Branch and Bound Algorithms

Satisfiability problem

Given a logic formula $f(x_1, x_2, \ldots, x_n)$ of n binary variables $x_1, x_2, \ldots, x_n$, is there a truth assignment for which the formula is **T**rue.

$f(x_1, x_2, x_3, x_4)$
$= (x_1 \vee \sim x_2 \vee \sim x_4) \wedge (x_2 \vee \sim x_3) \wedge (x_1 \vee x_3 \vee x_4)$

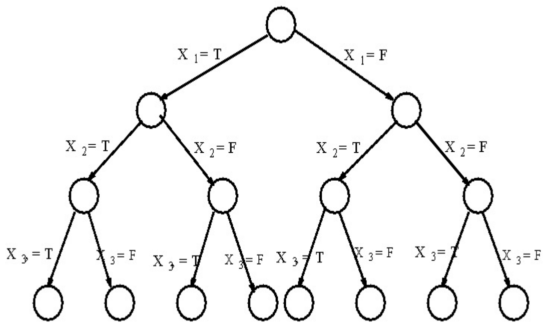
$f(\mathbf{F}, \mathbf{F}, \mathbf{F}, \mathbf{F}) = (\mathbf{T}) \wedge (\mathbf{T}) \wedge (\mathbf{F}) = \mathbf{F}$

$f(\mathbf{T}, \mathbf{T}, \mathbf{T}, \mathbf{T}) = (\mathbf{T}) \wedge (\mathbf{T}) \wedge (\mathbf{T}) = \mathbf{T}$

If there are n variables $x_1, x_2, \ldots, x_n$, then there are 2^n possible assignments.

Satisfiability problem

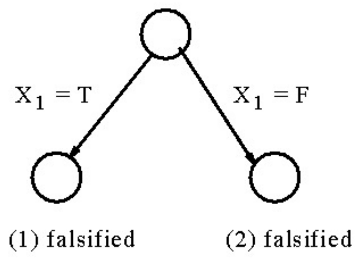
x_1	x_2	x_3
F	F	F
F	F	T
F	T	F
F	T	T
T	F	F
T	F	T
T	T	F
T	T	T



IIM, NCTU

3

- An instance:
  - $-x_1$..................(1)
  - x_1..................(2)
  - $x_2 \vee x_5$..........(3)
  - x_3..................(4)
  - $-x_2$..................(5)



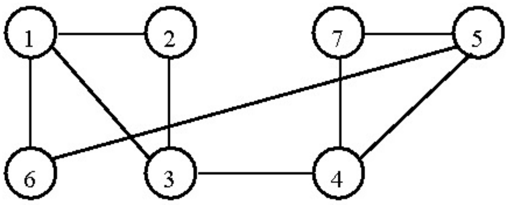
A partial tree for the satisfiability problem.

- We may not need to examine all possible assignments.

IIM, NCTU

4

Hamiltonian circuit problem

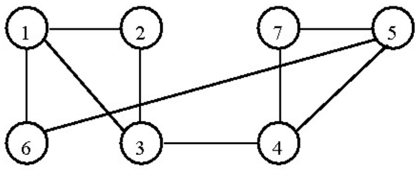


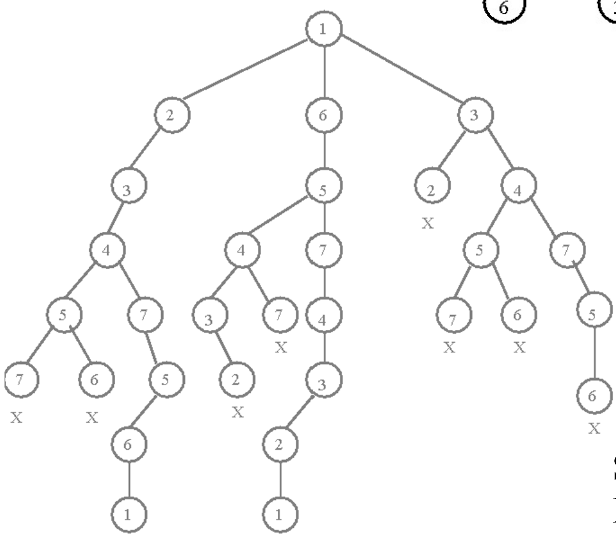
HC: 1-2-3-4-7-5-6-1

IIM, NCTU

5

Tree representation of the Hamiltonian circuit problem.





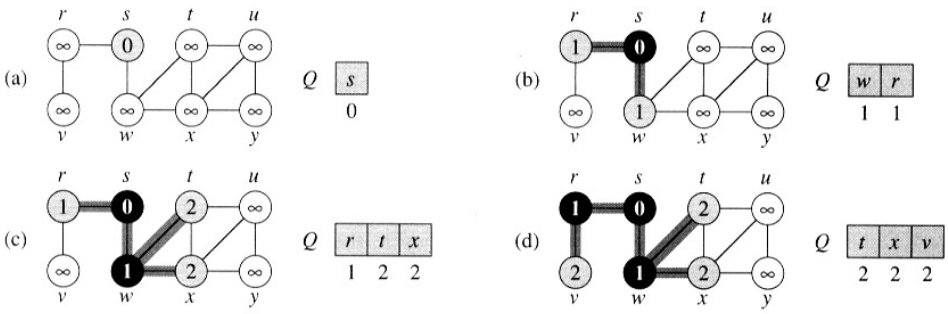
Search tree/
Enumeration tree

IIM, NCTU

6

P.7

Operations of Breadth First Search in a graph



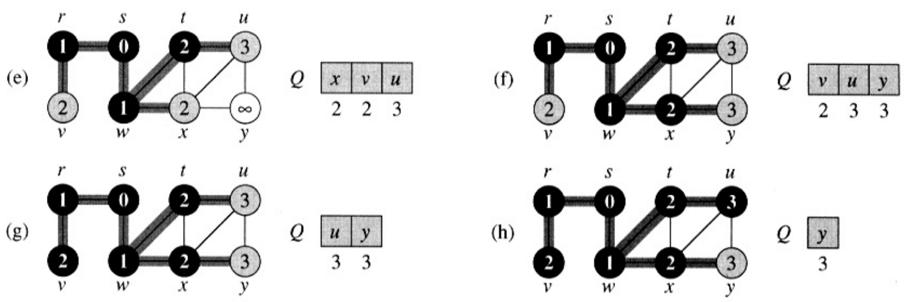
Chapter 22

IIM, NCTU

7

P.8

Operations of BFS

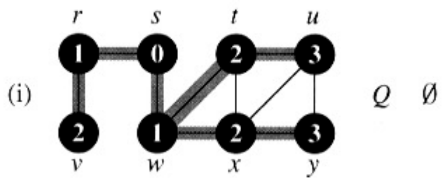


Chapter 22

IIM, NCTU

8

Operations of BFS



Breadth-first search

u.d: the distance from *s* to node *u*
u.π: the predecessor of node *u*

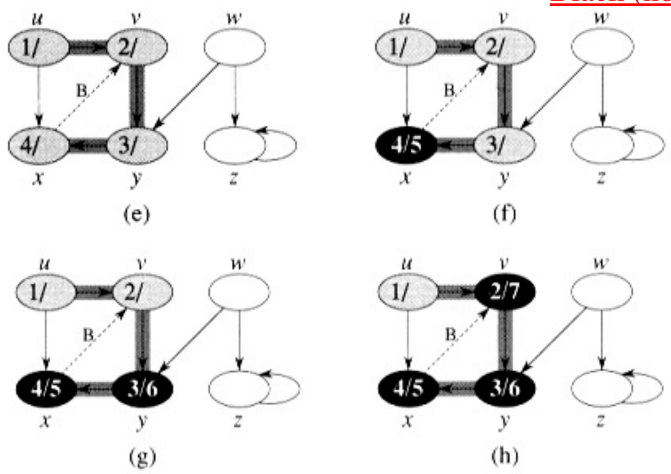
```
BFS(G, s)
for each vertex u ∈ G.V − {s}
    u.color = WHITE
    u.d = ∞
    u.π = NIL
s.color = GRAY
s.d = 0
s.π = NIL
Q = ∅
ENQUEUE(Q, s)
while Q ≠ ∅
    u = DEQUEUE(Q)
    for each v ∈ G.Adj[u]
        if v.color == WHITE
            v.color = GRAY
            v.d = u.d + 1
            v.π = u
            ENQUEUE(Q, v)
    u.color = BLACK
```

Time analysis: $O(V+E)$

P.11

Operations of Depth First Search

White (untouched)
Gray (explored)
Black (no more expansion)



Chapter 22

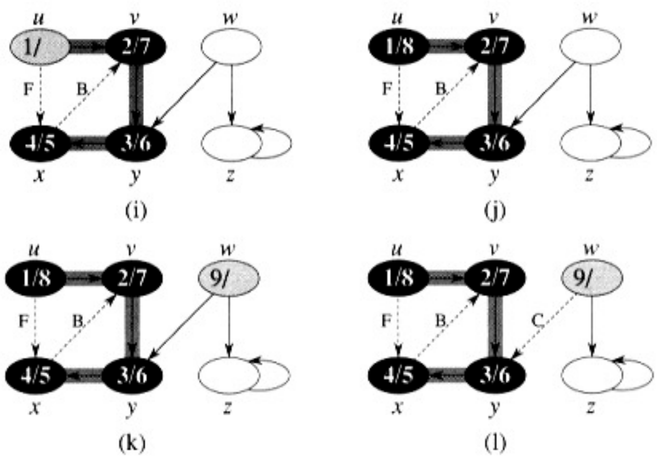
IIM, NCTU

11

P.12

DFS

White (untouched)
Gray (explored)
Black (no more expansion)



Chapter 22

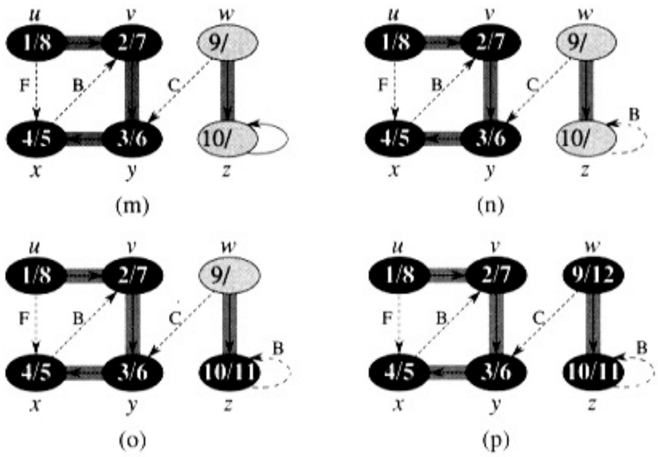
IIM, NCTU

12

P.13

DFS

White (untouched)
Gray (explored)
Black (no more expansion)



Chapter 22 IIM, NCTU

13

P.14

Depth-First Search

DFS(G)
for each vertex $u \in V[G]$
    $color[u] \leftarrow white$
    $\pi[u] \leftarrow NIL$
$time \leftarrow 0$
for each vertex $u \in V[G]$
    if $color[u] = white$
    then DFS-VISIT(u)

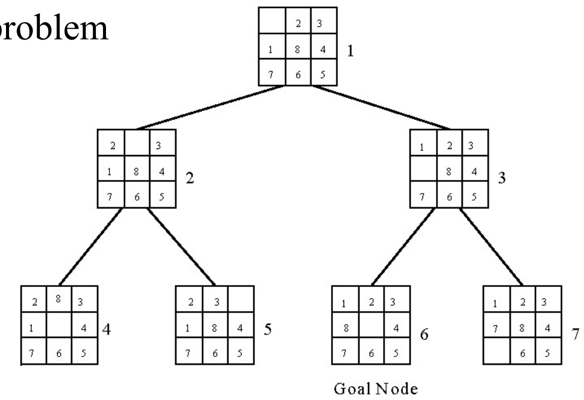
DFS-VISIT(u)
$color[u] = gray$
$d[u] \leftarrow time \leftarrow time + 1$
for each $v \in adj[u]$
    if $color[u] = white$
    then $\pi[v] \leftarrow u$
        DFS-VISIT(v)
$color[u] = black$
$f[u] \leftarrow time \leftarrow time + 1$

Chapter 22 IIM, NCTU

14

Breadth-First Search (BFS)

- 8-puzzle problem



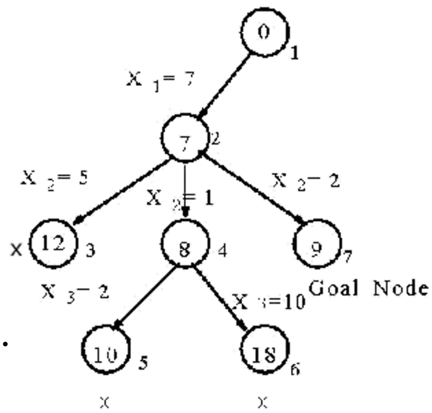
- Breadth-first search uses a queue to hold all expanded nodes.

IIM, NCTU

15

Depth-First Search (DFS)

- Sum of subset problem
  Given S={7, 5, 1, 2, 10},
  $\exists S' \subseteq S$ such that the sum of elements in S' = 9?



- A stack can be used to guide the depth-first search.

IIM, NCTU

16

Minimizing the Total Weighted Late Work

- There are n jobs to be processed on a machine
- Each job j is associated three parameters:
  - Processing length p_j
  - Due date d_j
  - Weight w_j
- In a schedule S, the completion time of job j is denoted by $C_j\,(S)$
- The late work of job j is given by $Y_j = \max\{p_j, C_j - d_j\}$
- We want to find a permutation of the jobs such that the total weighted late work $\Sigma w_j Y_j$ is minimized

IIM, NCTU

17

Hill Climbing, Steepest Descent

- A variant of depth-first search

  The method selects the locally optimal node to expand.
- e.g. 8-puzzle problem

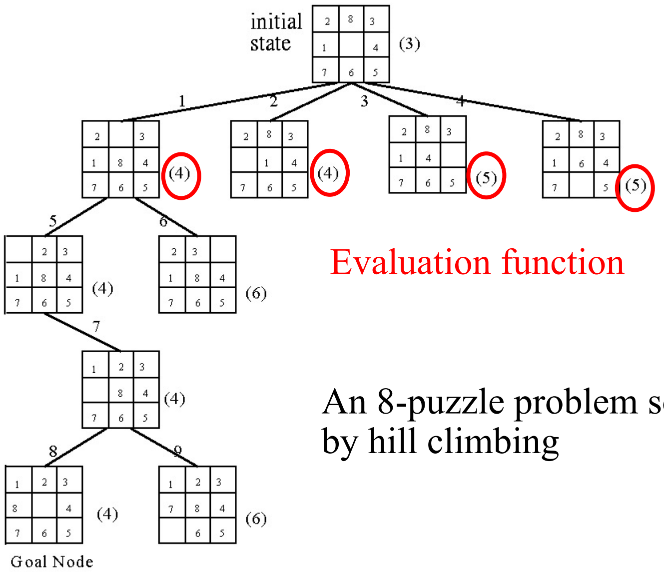
  evaluation function $f(n) = d(n) + w(n)$

  where $d(n)$ is the depth of node n

  $w(n)$ is the number of misplaced tiles in node n.

IIM, NCTU

18

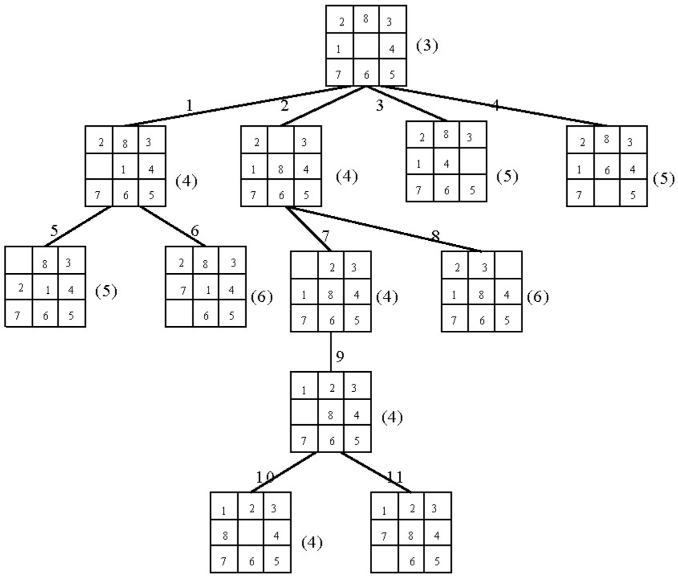


Evaluation function

An 8-puzzle problem solved by hill climbing

Best-First Search (Beam Search)

- Combing depth-first search and breadth-first search.
- Selecting the node with the best estimated cost among all active nodes.
- This method has a global view.



An 8-puzzle problem solved by best-first search

IIM, NCTU

Best-First Search Scheme

Step1: Form a one-element list consisting of the root node.

Step2: Remove the first element from the list. Expand the selected element. If one of the descendants of the first element is a goal node, then stop; otherwise, add the descendants into the list.

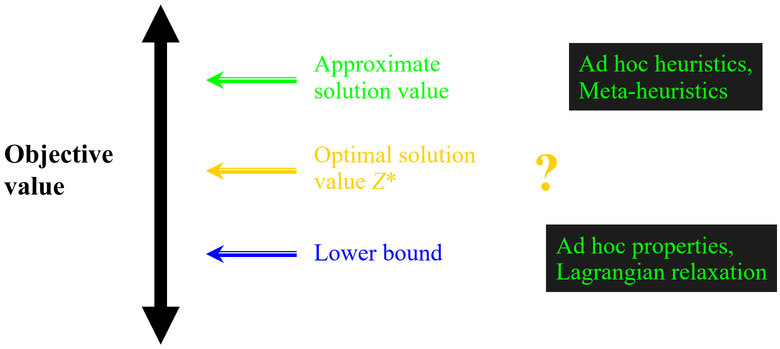
Step3: Sort the entire list by the values of some estimation function.

Step4: If the list is empty, then failure. Otherwise, go to Step 2.

IIM, NCTU

Why Lower Bounds

- Consider some *minimization* problem that is computationally challenging

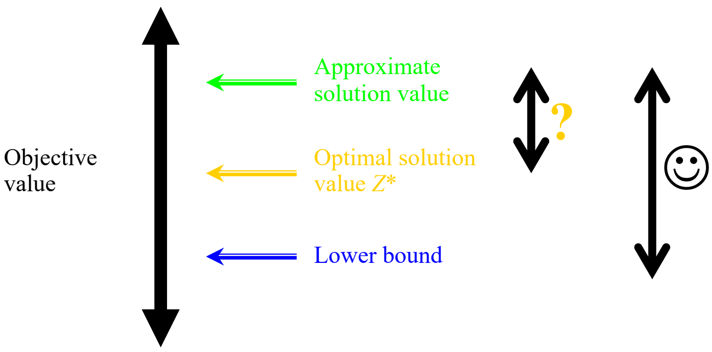


IIM, NCTU

23

Why Lower Bounds

- Consider some *minimization* problem that is computationally challenging

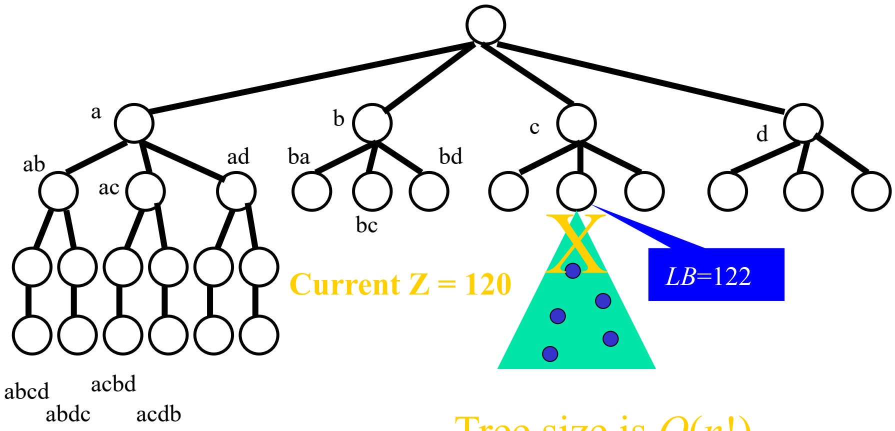


IIM, NCTU

24

Why Lower Bounds

4 (*a*, *b*, *c*, *d*) orders to be assigned to 4 positions



Tree size is $O(n!)$

IIM, NCTU

25

Lower bounds Vs. Upper bounds

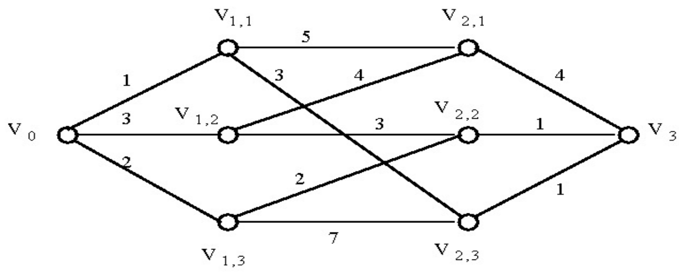
- Minimization problems
  - lower bound <= optimal value
  - larger lower bounds are preferred
  - approximate solutions are upper bounds
- Maximization problems
  - upper bound >= optimal value
  - smaller upper bounds are preferred
  - approximate solutions are lower bounds

IIM, NCTU

26

Branch-and-bound strategy

- This strategy can be used to efficiently solve optimization problems.
- e.g.



A multi-stage graph searching problem.

IIM, NCTU

27

Branch-and-Bound Algorithm

- An implicit enumeration approach to dealing with hard optimization problems
- Ingredients of a BaB algorithm
  - Enumeration Tree – The way in which the solution space is constructed and explored
  - Bounding Function – An estimation function of the objective value of a partial solution
  - Dominance/Elimination Rule – Conditions under which some part of the solution space can be skipped without sacrificing the optimality

IIM, NCTU

28

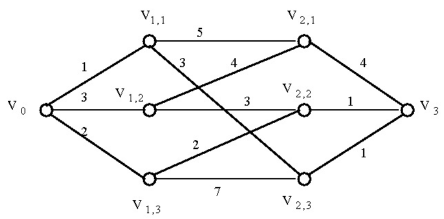
Efficiency of an Example BaB Algorithm

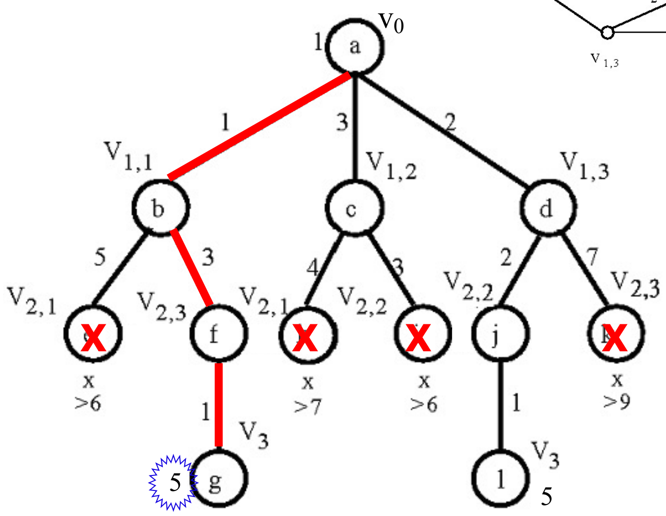
n	Crude enumeration CE		Dominance Rule DR		Lower Bound LB		DR & LB	
	Avg_Node	Avg_Time	llAvg_Node	Avg_Time	llAvg_Node	Avg_Time	llAvg_Node	Avg_Time
6	1956	0.00	624	0.00	52	0.00	33	0.00
8	109,600	0.02	15,477	0.01	154	0.00	67	0.00
10	9,864,100	1.49	711,574	0.24	428	0.00	117	0.01
12	1,302,061,344	205.73	47,312,610	15.98	834	0.00	151	0.00
14	2.370E+11	—	511,829,995	907.02	2472	0.01	265	0.00
15	3.555E+12	—	—	—	2423	0.01	278	0.00
20	6.613E+18	—	—	—	43,074	0.09	690	0.00
25	4.216E+25	—	—	—	584,526	1.27	1595	0.01
30	7.210E+32	—	—	—	760,590	1.94	2361	0.02
35	2.809E+40	—	—	—	9,096,072	23.78	3857	0.03
40	2.218E+48	—	—	—	—	—	5786	0.05
50	8.267E+64	—	—	—	—	—	42,507	0.24
60	2.619E+82	—	—	—	—	—	74,606	0.64
70	3.256E+100	—	—	—	—	—	281,947	1.64
80	1.945E+119	—	—	—	—	—	540,570	3.68
90	4.039E+138	—	—	—	—	—	3,247,508	18.22
100	2.537E+158	—	—	—	—	—	2,572,461	28.86

IIM, NCTU

29

- Solved by branch-and-bound





Cost = 5 is an upper bound

IIM, NCTU

30

Personnel assignment problem

- A linearly ordered set of persons $P=\{P_1, P_2, \ldots, P_n\}$ where $P_1 < P_2 < \ldots < P_n$
- A partially ordered set of jobs $J=\{J_1, J_2, \ldots, J_n\}$. Suppose that P_i and P_j are assigned to jobs $f(P_i)$ and $f(P_j)$ respectively. If $f(P_i) \le f(P_j)$, then $P_i \le P_j$. Cost C_{ij} is the cost of assigning P_i to J_j. We want to find a feasible assignment with the minimum cost. i.e.

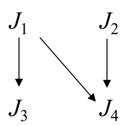
  $X_{ij} = 1$ if P_i is assigned to J_j

  $X_{ij} = 0$ otherwise.
- Minimize $\sum_{i,j} C_{ij}X_{ij}$

IIM, NCTU

31

- e.g. A partial ordering of jobs



- After topological sorting, one of the following 5 feasible sequences will be generated:

  J_1, J_2, J_3, J_4

  J_1, J_2, J_4, J_3

  J_1, J_3, J_2, J_4

  J_2, J_1, J_3, J_4

  J_2, J_1, J_4, J_3

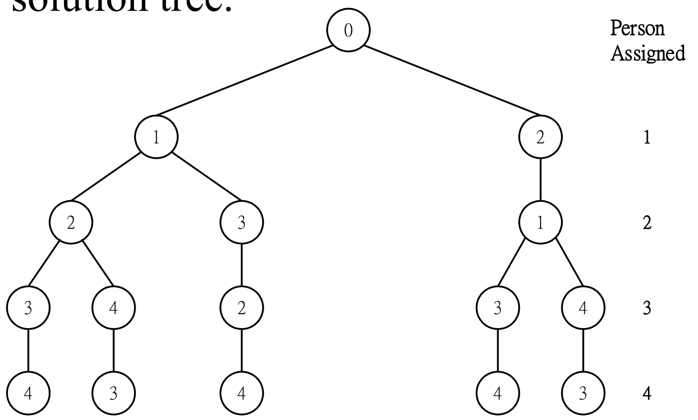
- One of the feasible assignments:
  $(P_1, J_1), (P_2, J_2), (P_3, J_3), (P_4, J_4)$

IIM, NCTU

32

Solution Tree

- All possible solutions can be represented by a solution tree.



IIM, NCTU

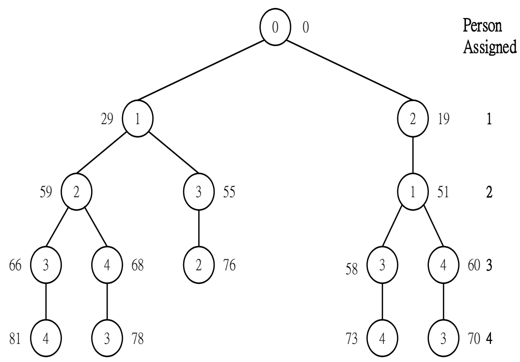
33

Cost matrix

- Cost matrix

Jobs / Persons	1	2	3	4
1	29	19	17	12
2	32	30	26	28
3	3	21	7	9
4	18	13	10	15

- Best-first search scheme:



Only one node is pruned away.

IIM, NCTU

34

Reduced cost matrix

- Cost matrix

Persons \ Jobs	1	2	3	4
1	29	19	17	12
2	32	30	26	28
3	3	21	7	9
4	18	13	10	15

- Reduced cost matrix

Persons \ Jobs	1	2	3	4	
1	17	4	5	0	(-12)
2	6	1	0	2	(-26)
3	0	15	4	6	(-3)
4	8	0	0	5	(-10)
		(-3)			

IIM, NCTU

35

- A reduced cost matrix can be obtained:

  subtract a constant from each row and each column respectively such that each row and each column contains at least one zero.

- Total cost subtracted: 12+26+3+10+3 = 54
- This is a *lower bound* of our solution.

IIM, NCTU

36

Traveling salesperson problem

- Cost matrix

i \ j	1	2	3	4	5	6	7
1	∞	3	93	13	33	9	57
2	4	∞	77	42	21	16	34
3	45	17	∞	36	16	28	25
4	39	90	80	∞	56	7	91
5	28	46	88	33	∞	25	57
6	3	88	18	46	92	∞	7
7	44	26	33	27	84	39	∞

IIM, NCTU

37

- Reduced cost matrix

i \ j	1	2	3	4	5	6	7	
1	∞	0	90	10	30	6	54	(-3)
2	0	∞	73	38	17	12	30	(-4)
3	29	1	∞	20	0	12	9	(-16)
4	32	83	73	∞	49	0	84	(-7)
5	3	21	63	8	∞	0	32	(-25)
6	0	85	15	43	89	∞	4	(-3)
7	18	0	7	1	58	13	∞	(-26)

Reduced: 84

IIM, NCTU

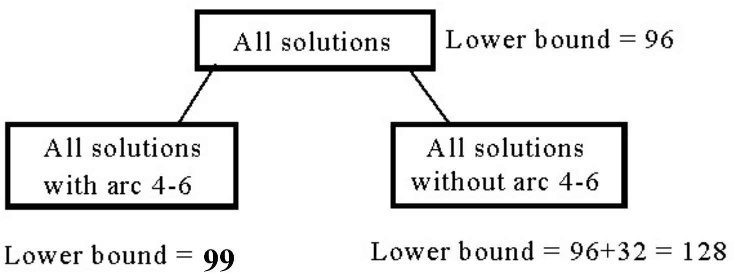
38

- Further reduced

i \ j	1	2	3	4	5	6	7
1	∞	0	83	9	30	6	50
2	0	∞	66	37	17	12	26
3	29	1	∞	19	0	12	5
4	32	83	66	∞	49	0	80
5	3	21	56	7	∞	0	28
6	0	85	8	42	89	∞	0
7	18	0	0	0	58	13	∞
			(-7)	(-1)			(-4)

Total cost reduced: 84+7+1+4 = 96 (lower bound)

- The highest level of a decision tree:



- A reduced cost matrix if arc (4,6) is included in the solution.

j / i	1	2	3	4	5	7
1	∞	0	83	9	30	50
2	0	∞	66	37	17	26
3	29	1	∞	19	0	5
5	3	21	56	7	∞	28
6	0	85	8	∞	89	0
7	18	0	0	0	58	∞

Arc (6,4) is changed to be infinity since it can not be included in the solution.

IIM, NCTU

41

- The reduced cost matrix for all solutions with arc 4-6

j / *i*	1	2	3	4	5	7	
1	∞	0	83	9	30	50	
2	0	∞	66	37	17	26	
3	29	1	∞	19	0	5	
5	0	18	53	4	∞	25	(-3)
6	0	85	8	∞	89	0	
7	18	0	0	0	58	∞	

- Total cost reduced: 96+3 = 99 (new lower bound)

IIM, NCTU

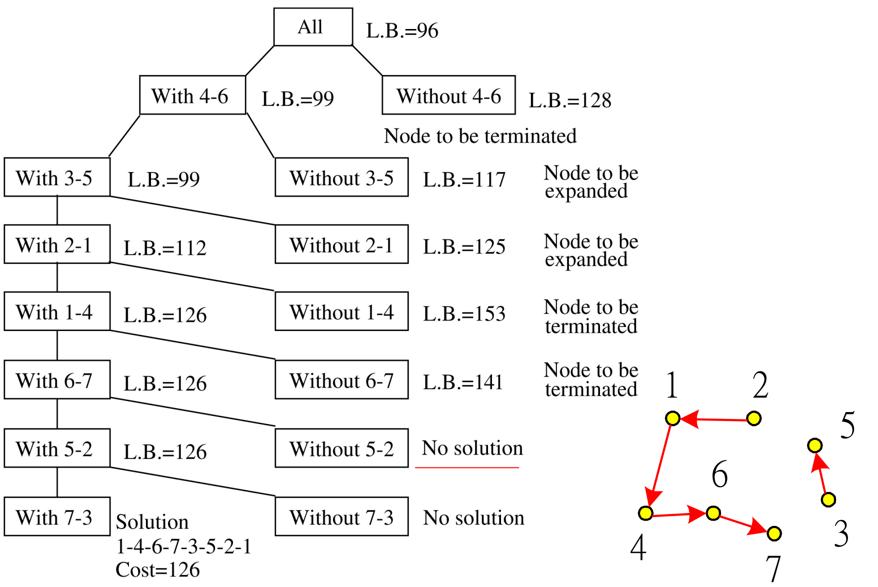
42

Arc (4, 6) is excluded

i \ j	1	2	3	4	5	6	7
1	∞	0	83	9	30	6	50
2	0	∞	66	37	17	12	26
3	29	1	∞	19	0	12	5
4	32	83	66	∞	49	∞	80
5	3	21	56	7	∞	0	28
6	0	85	8	42	89	∞	0
7	18	0	0	0	58	13	∞

IIM, NCTU

43



A branch-and-bound solution of a traveling salesperson problem.

IIM, NCTU

44

When will the expansion of a node be terminated?

- Current node is infeasible
- The lower bound value of the current node is greater than or equal to the best solution value ever encountered
- Upper bound and lower bound values of the current node are the same

IIM, NCTU

45

The 0/1 knapsack problem

- Positive integer $P_1, P_2, \ldots, P_n$ (profit)
  $W_1, W_2, \ldots, W_n$ (weight)
  M (capacity)

maximize $\sum_{i=1}^{n} P_i X_i$

subject to $\sum_{i=1}^{n} W_i X_i \leq M \quad X_i = 0 \text{ or } 1, i = 1, \ldots, n.$

The problem is modified:

minimize $-\sum_{i=1}^{n} P_i X_i$

IIM, NCTU

46

- e.g. $n = 6$, $M = 34$

i	1	2	3	4	5	6
P_i	6	10	4	5	6	4
W_i	10	19	8	10	12	8

$(P_i/W_i \geq P_{i+1}/W_{i+1})$

- A feasible solution: $X_1 = 1, X_2 = 1, X_3 = 0, X_4 = 0, X_5 = 0, X_6 = 0$
  $-(P_1+P_2) = -16$ (upper bound)
  Any solution higher than -16 cannot be optimal.

IIM, NCTU

47

Relax the restriction

- Relax our restriction from $X_i = 0$ or 1 to $0 \leq X_i \leq 1$ (knapsack problem)

Let $-\sum_{i=1}^{n} P_i X_i$ be an optimal solution for 0/1 knapsack problem and $-\sum_{i=1}^{n} P_i X_i'$ be an optimal solution for knapsack problem. Let $Y = -\sum_{i=1}^{n} P_i X_i$,

$Y' = -\sum_{i=1}^{n} P_i X_i'$.

$\Rightarrow Y' \leq Y$

IIM, NCTU

48

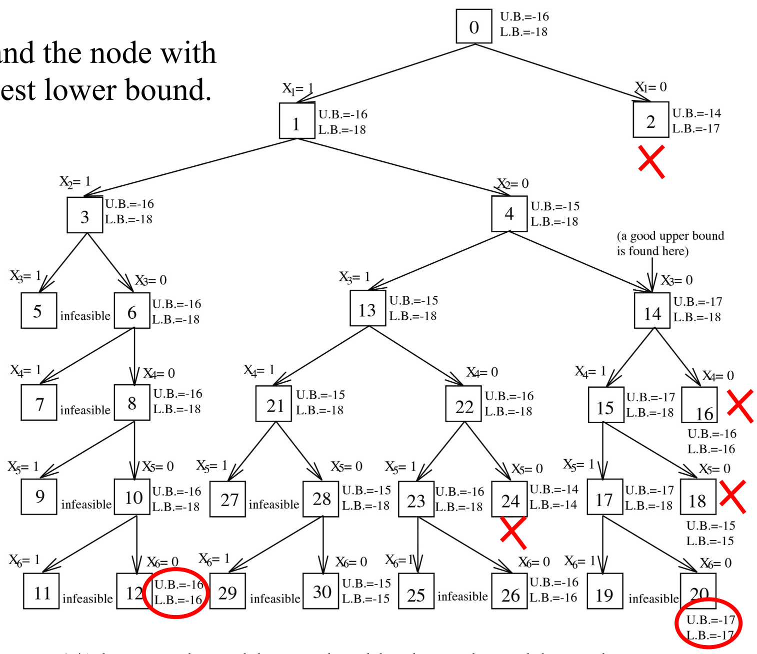
Upper bound and lower bound

- We can use the greedy method to find an optimal solution for knapsack problem:

  $X_1 = 1, X_2 =1, X_3 = 5/8, X_4 = 0, X_5 = 0, X_6 =0$
  $-(P_1+P_2+5/8P_3) = -18.5$ (lower bound)
  -18 is our lower bound. (only consider integers)

$\Rightarrow$ $-18 \leq$ optimal solution ≤ -16
optimal solution: $X_1 = 1, X_2 = 0, X_3 = 0, X_4 = 1, X_5 = 1, X_6 = 0$
$-(P_1+P_4+P_5) = -17$

Expand the node with the best lower bound.



0/1 knapsack problem solved by branch-and-bound strategy.

Dominance Rule

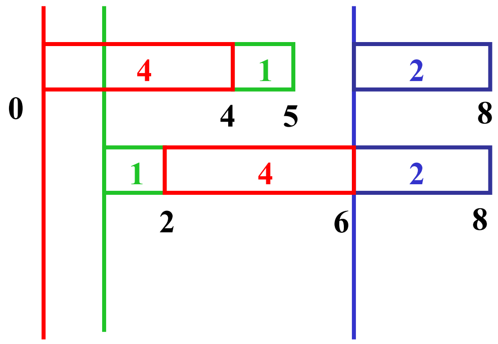
- If there exist items i and j with $p_i \geq p_j$ and $w_i \leq w_j$, then there is an optimal solution in which
  - (1) if item j is included, then item i is also included;
  - (2) if item i is excluded, then item j is also excluded;

IIM, NCTU

51

Minimizing the Total Completion Time on a Single Machine with Release Dates

- Each job has a processing time p_i and a release date r_i.
- We want to find a feasible schedule such that the Total completion time ΣC_i is minimized.



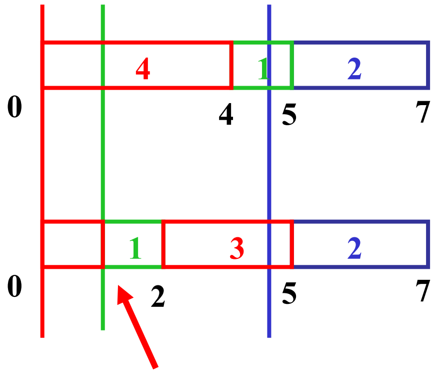
NP-hard

IIM, NCTU

52

Find a Lower Bound

- Consider the situation where *preemption* is allowed



Consider the objective $\Sigma w_i C_i$

IIM, NCTU

53

Total completion time minimization

- The total completion time minimization problem is strongly NP-hard
- The preemptive case is polynomially solvable
- Dominance Rule: If there exist jobs i and j with $p_i \le p_j$ and $r_i \le r_j$, then job i precedes job j in some optimal solution.
- Consider the objective $\Sigma w_i C_i$:
  - NP-hard even if preemption is allowed

IIM, NCTU

54

ELSEVIER

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers & Operations Research 31 (2004) 347–358

computers & operations research

www.elsevier.com/locate/dsw

Heuristics and lower bounds for the bin packing problem with conflicts

Michel Gendreau[a,b], Gilbert Laporte[c,d,*], Frédéric Semet[a,c]

[a] Centre de Recherche sur les Transports, Université de Montréal, C.P. 6128, Succursale Centre-ville, Montréal, Québec, Canada H3C 3J7

[b] Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, C.P. 6128, Succursale Centre-ville, Montréal, Québec, Canada H3C 3J7

[c] École des Hautes Études Commerciales, 3000, Chemin de la Côte-Sainte-Catherine, Montréal, Québec, Canada H3T 2A7

[d] LAMIH, Université de Valenciennes et du Hainaut Cambrésis, le Mont Houy, 59313 Valenciennes Cedex 9, France

Received 29 July 2002

IIM, NCTU

55

Bin Packing with Conflicts

- A set V of n items with weights $w_1, w_2, \ldots, w_n$
- Each bin has a capacity c
- Conflict graph representing the possible conflicts amongst the items
- Goal: Finding the minimum number of bins that can accommodate all items without violating conflict constraints

IIM, NCTU

56

Related Problems

- Examination scheduling
- Assignment of processes to processors
- Bounded graph coloring
- Delivery problem

Integer Programming Model

$$\text{minimize} \quad z = \sum_{k=1}^{n} y_k$$

Bin k is used or not

subject to

Item i is dispatched to bin k.

$$\sum_{i=1}^{n} w_i x_{ik} \leqslant c y_k \quad (k = 1, \ldots, n)$$

$$\sum_{k=1}^{n} x_{ik} = 1 \quad (i = 1, \ldots, n)$$

$$x_{ik} + x_{jk} \leqslant 1 \quad ((i, j) \in E, \ k = 1, \ldots, n)$$

$$y_k = 0 \text{ or } 1 \quad (k = 1, \ldots, n)$$

$$x_{ik} = 0 \text{ or } 1 \quad (i, k = 1, \ldots, n).$$

Problem-Solving Approaches

- Heuristics
  - FFD
  - Variants based upon coloring heuristics
  - Variants based upon clique heuristics
- Lower bounds
  - Relaxation of conflict constraints
  - Transportation problem

IIM, NCTU

59

Step 1: Determine a large clique D on $G=(V,E)$ using Johnson's (1974) heuristic, and assign each item of D to a different bin. Let r_k be the residual capacity of bin k and let $h = \text{argmin}_{i \in D}\{w_i\}$.

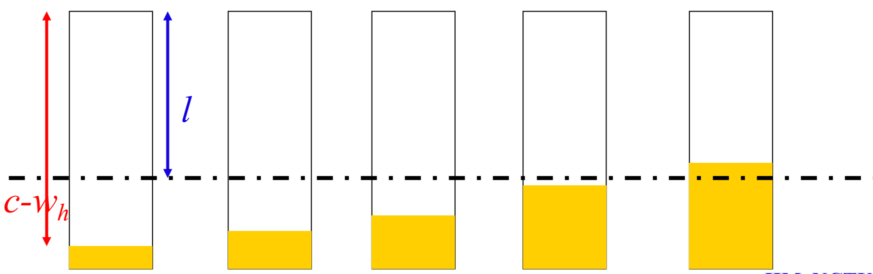
Step 2: For a given weight level $\ell \leqslant c - w_h$, define the three sets

$$I'_\ell = \{i \in V:\ w_i < \ell\},$$

$$I_\ell = \{i \in V:\ \ell \leqslant w_i \leqslant c - w_h\},$$

$$I''_\ell = \{i \in V: w_i > c - w_h\}.$$

In addition to the already created $|D|$ bins, create a dummy bin 0 of infinite residual capacity. Let K_ℓ be the set of bins k having a residual capacity $r_k \geqslant \ell$.



IIM, NCTU

60

Transportation Problem

$$\text{minimize} \quad \sum_{i \in I_\ell} \sum_{k \in K_\ell} c_{ik} x_{ik}$$

subject to

$$\sum_{k \in K_\ell} x_{ik} = w_i \quad (i \in I_\ell)$$

$$\sum_{i \in I_\ell} x_{ik} \leqslant r_k \quad (k \in K_\ell)$$

$$x_{ik} \text{ integer} \quad (i \in I_\ell, k \in K_\ell).$$

IIM, NCTU

61

BPC Lower Bound

$$b_\ell = |D| + \left\lceil \left(\sum_{i \in I_\ell} x_{i0} + \sum_{i \in I''_\ell} w_i + z_\ell \right) \Big/ c \right\rceil .$$

$$z_\ell = \max \left\{ 0, \sum_{i \in I'_\ell} w_i - \left(\sum_{k \in D} r_k - \sum_{i \in I_\ell} \sum_{k \in K_\ell \setminus \{0\}} x_{ik} \right) \right\} .$$

IIM, NCTU

62

Computational results for the Falkenauer uniform instances ($n = 120, c = 150$)

d (%)	MAX L_1	L_2	$\{L_1, L_2\}$	H1 UB/LB	sec.	H2 UB/LB	sec.	H3 UB/LB	sec.	H4 UB/LB	sec.	H5 UB/LB	sec.	H6 UB/LB	nb. opt.	sec.
0	48.3	48.3	48.3	**1.017**	0.1	1.279	0.5	1.049	0.5	1.261	1.9	1.302	1.3	**1.017**	2	0.2
10	48.3	48.3	48.3	1.025	0.1	1.281	0.6	1.080	0.4	1.242	2.1	1.263	1.2	**1.017**	2	0.2
20	48.3	48.3	48.3	**1.056**	0.1	1.291	0.6	1.109	0.6	1.180	2.8	1.213	1.2	**1.056**	0	0.3
30	48.3	48.3	48.3	**1.152**	0.2	1.295	0.7	1.192	0.7	1.188	4.6	1.237	1.0	1.192	0	0.3
40	48.3	49.7	49.7	1.248	0.2	1.351	1.0	1.289	1.0	1.289	13	1.349	0.8	**1.182**	0	0.4
50	48.3	59.8	59.8	1.197	0.2	1.265	1.3	1.260	1.3	1.242	15	1.284	0.6	**1.108**	0	0.5
60	48.3	72.1	72.1	1.130	0.2	1.178	1.7	1.178	1.8	1.165	13.5	1.192	0.4	**1.091**	0	0.5
70	48.3	85.8	85.8	1.081	0.3	1.100	2.0	1.100	2.0	1.091	16.4	1.112	0.3	**1.056**	0	0.7
80	48.3	96.3	96.3	1.046	0.4	1.059	2.2	1.059	2.3	1.051	15.7	1.063	0.4	**1.033**	0	0.6
90	48.3	108.3	108.3	1.017	0.4	1.029	2.4	1.029	2.4	1.017	16.1	1.023	0.3	**1.015**	4	0.6

IIM, NCTU

63

Table 4
Computational results for the Falkenauer uniform instances ($n = 1000, c = 150$)

d (%)	MAX L_1	L_2	$\{L_1, L_2\}$	H1 UB/LB	sec.	H2 UB/LB	sec.	H3 UB/LB	sec.	H4 UB/LB	sec.	H5 UB/LB	sec.	H6 UB/LB	nb. opt.	sec.
0	401.8	401.8	401.8	**1.012**	0.9	1.372	134.3	1.141	134.6	—	—	1.332	374.2	**1.012**	0	40.3
10	401.8	401.8	401.8	1.014	1.1	1.333	142.5	1.146	142.8	—	—	1.277	346.1	**1.012**	0	83.1
20	401.8	401.8	401.8	**1.037**	1.8	1.255	167.2	1.137	167.6	—	—	1.226	303.3	1.058	0	336.3
30	401.8	401.8	401.8	**1.095**	3.2	1.233	211.2	1.139	211.1	—	—	1.240	251.2	1.216	0	670.4
40	401.8	407.5	407.5	1.217	7.9	1.363	275.1	1.318	274.8	—	—	1.375	193.6	**1.149**	0	809
50	401.8	509.5	509.5	1.138	16.2	1.238	364.2	1.237	363.6	—	—	1.251	140.2	**1.063**	0	810.9
60	401.8	602.3	602.3	1.098	25.6	1.161	498.3	1.161	459.1	—	—	1.173	80.1	**1.047**	0	575.4
70	401.8	707.4	707.4	1.063	37.0	1.103	539.2	1.103	538.5	—	—	1.108	44.4	**1.027**	0	515.3
80	401.8	799.2	799.2	1.042	49.4	1.065	598.9	1.065	597.2	—	—	1.066	24.7	**1.024**	0	512.2
90	401.8	899.2	899.2	1.020	63.0	1.028	649.9	1.028	642.2	—	—	1.031	15.4	**1.011**	0	501.0

IIM, NCTU

64

Film Production Problem

Film Production Problem

- Holding Days :
  - The days when an actor is not required, but still has some other filming left to complete.

- The objective is to minimize the total holding cost of the actors/actress.

Film Production Problem

actor i	Shooting day j 1 2 3 4 5 6 7 8	Holding Cost c_i
1	1 1 1 0 0 1 0 0	810
2	1 1 0 0 0 1 1 0	700
3	0 1 0 1 0 0 0 1	910
4	0 1 0 1 1 1 1 0	880
5	0 1 0 0 0 0 0 1	400
6	0 0 1 1 0 0 0 1	610

Total cost = 2*810+3*700+4*910
+880+5*400+3*610

IIM, NCTU

67

Film Production Problem

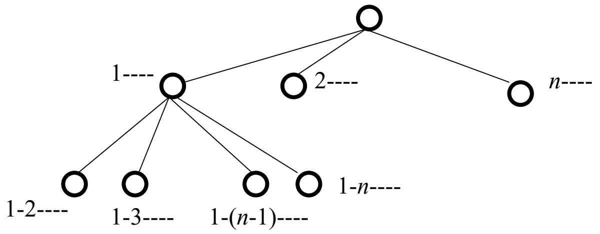
actor i	Shooting day j 8 2 3 4 5 6 7 1	Holding Cost c_i
1	0 1 1 0 0 1 0 1	810
2	0 1 0 0 0 1 1 1	700
3	1 1 0 1 0 0 0 0	910
4	0 1 0 1 1 1 1 0	880
5	1 1 0 0 0 0 0 0	400
6	1 0 1 1 0 0 0 0	610

Total cost = 3*810+3*700+1*910
+880+610

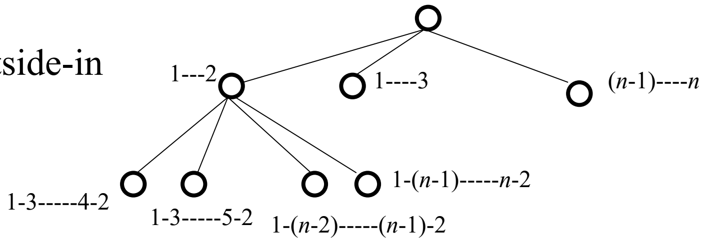
IIM, NCTU

68

Enumeration Tree



Outside-in



IIM, NCTU

69

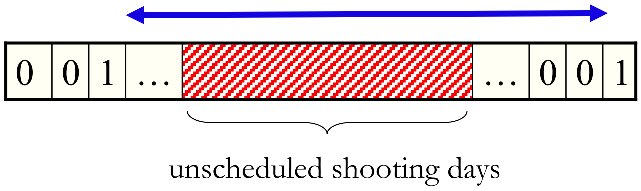
Lower Bound

Actor *i*	*T(s)* 4	2	3	1	5	8	7	6	**Holding Cost** c_i
1	0	1					0	0	$1,000
2	0	1					1	1	$200
3	1	0					0	0	$700
4	1	1					1	0	$350
5	0	1					0	0	$500
6	1	1					0	1	$840
7	0	0					1	0	$1,020
8	1	0					1	0	$980
9	0	0					1	1	$1,300
10	0	1					0	0	$450

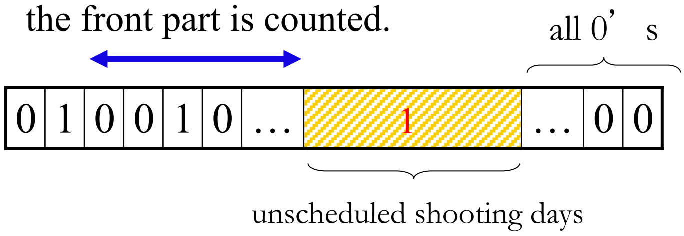
IIM, NCTU

70

Any 0 enclosed by two 1's is counted.

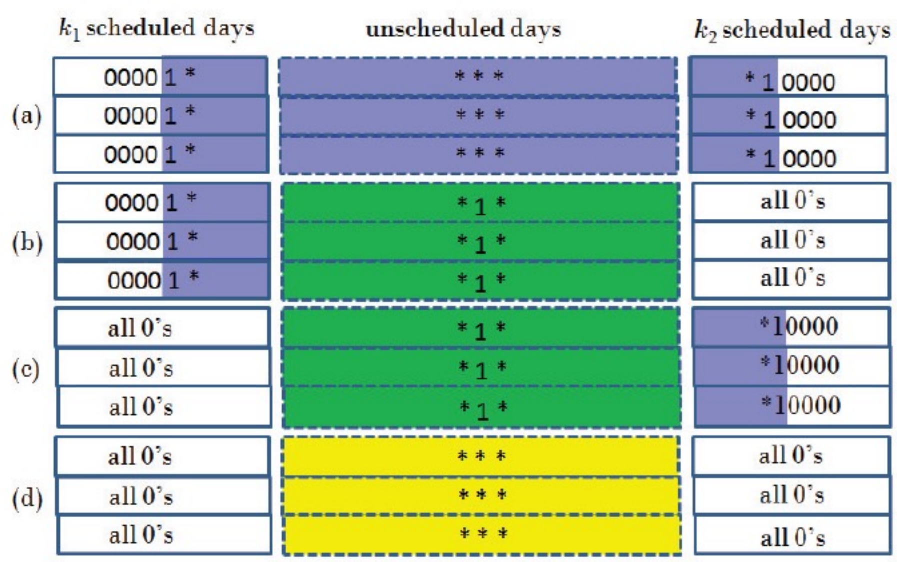


Any 0 already scheduled in
the front part is counted.



IIM, NCTU

71



IIM, NCTU

72

Lower Bound

Actor i	T(s) 4 2 3 1 5 8 7 6	Holding Cost c_i
√ 1	0 1 0 0	\$1,000
√ 3	1 0 0 0	\$700
√ 5	0 1 0 0	\$500
7	0 0 1 0	\$1,020
9	0 0 1 1	\$1,300
√ 10	0 1 0 0	\$450

IIM, NCTU

73

- Define two subsets:

$$E_P = \{i|\varepsilon_i(k) = 1 \text{ and } \lambda_i(k) = 0\}; \text{ and}$$

$$L_P = \{i|\varepsilon_i(k) = 0 \text{ and } \lambda_i(k) = 1\}.$$

- $\alpha_p(x,y)$ denotes the number of unscheduled shooting days which actor x is required but y is not.

IIM, NCTU

74

- P = 38----75, E_P = {1, 3, 5, 6} and L_P = {2, 4}.

actress	Shooting day j								Holding Cost
i	3	8	4	2	1	6	7	5	c_i
1	1	0	0	1	1	1	0	0	810
2	0	0	0	1	1	1	1	0	700
3	0	1	1	1	0	0	0	0	910
4	0	0	1	1	0	1	1	1	880
5	0	1	0	1	0	0	0	0	400
6	1	1	1	0	0	0	0	0	610

$\pi_P(1,3) = 2$; $\alpha_P(3,1) = 1$; $\alpha_P(5,1) = 0$; $\alpha_P(6;1) = 1$;

$\alpha_P(1,5) = 2$; $\alpha_P(3,5) = 1$; $\alpha_P(5,3) = 0$; $\alpha_P(6,3) = 0$;

$\alpha_P(1,6) = 3$; $\alpha_P(3,6) = 1$; $\alpha_P(5,6) = 1$; $\alpha_P(6,5) = 1$;

IIM, NCTU

75

- Extra holding days incurred by actors x and y is is no less than $min\{\alpha_P(x,y), \alpha_P(y,x)\}$.

x 1 | 000111111100 | 0

y 1 | 111110000000 | 0

Scheduled days

x 1 | 001111111000 | 0

y 1 | 111100000100 | 0

Scheduled days

IIM, NCTU

76

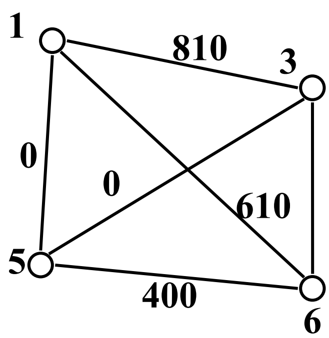
- The extra holding cost incurred by actors x and y is is no less than $min\{c_y\alpha_P(x,y), c_x\alpha_P(y,x)\}$.

- Formulate the problem of finding the extra holding cost for the subset E_P into a maximum weighted matching problem in graph theory by letting the weight of the edge connecting nodes x and y be defined as $min\{c_y\alpha_P(x,y), c_x\alpha_P(y,x)\}$.

IIM, NCTU

77

- Maximum weighted matching for E_P.
- With this graph, the maximum weighted matching of E_P is 810+400 = 1,210.



IIM, NCTU

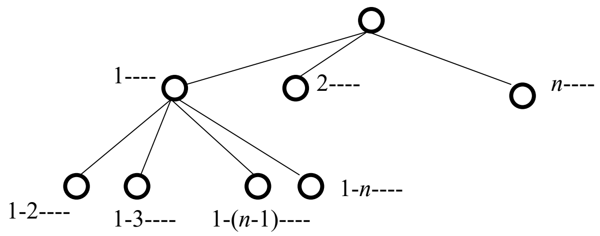
78

- The situation of L_P can be similarly estimated with a value of 700.
- Therefore, we conclude the example with a lower bound of P is 810+1,210+700 = 2,720.

IIM, NCTU

79

- Assume the conventional branching scheme is adopted.
- Can we deploy LB_1 or LB_2?



IIM, NCTU

80

$\beta_{i_1,\tilde{i}_2,i_3} \cup \beta_{i_1,i_2,\tilde{i}_3}$

a_{i_1}	**0**	1111 1111111	**0**
a_{i_2}	**1**	0000 1111111	**1**
a_{i_3}	**1**	1111 0000000	**1**

(a) Case x-1

$\beta_{i_1,i_2,\tilde{i}_3}$

a_{i_1}	**0**	1111111111	**1**
a_{i_2}	**1**	1111111111	**0**
a_{i_3}	**1**	0000000000	**1**

(b) Case x-4

$\beta_{i_1,\tilde{i}_2,\tilde{i}_3} \cup \beta_{i_1,i_2,i_3} \cup \beta_{i_1,\tilde{i}_2,\tilde{i}_3}$

a_{i_1}	**0**	000000111111111	**0**
a_{i_2}	**0**	111111110000000	**0**
a_{i_3}	**1**	000000110000000	**1**

(a) Case y-2

$\beta_{i_1,i_2,i_3} \cup \beta_{i_1,\tilde{i}_2,\tilde{i}_3}$

a_{i_1}	**0**	111111111111	**0**
a_{i_2}	**0**	111100000000	**1**
a_{i_3}	**1**	111100000000	**0**

or

$\beta_{i_1,i_2,i_3} \cup \beta_{i_1,\tilde{i}_2,\tilde{i}_3}$

a_{i_1}	**0**	111111111111	**0**
a_{i_2}	**0**	000000001111	**1**
a_{i_3}	**1**	000000001111	**0**

(b) Case y-5

Theorem 4.2. *Given partial schedule P,*

$$LB_3(P) = LB_2(P) + \textsc{Max-3-Grouping}(P)$$

is a lower bound on the optimal solution derived from P.

`IP-3-Grouping:`

$$\text{Maximize} \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} z_{i,j,k} Z_{i,j,k}$$

subject to

$$\sum_{j=1}^{m}\sum_{k=1}^{m} Z_{i,j,k} + \sum_{j=1}^{m}\sum_{k=1}^{m} Z_{j,i,k} + \sum_{j=1}^{m}\sum_{k=1}^{m} Z_{j,k,i} \leq 1 \qquad \text{for all } i;$$

$$Z_{i,j,k} \in \{0,1\} \qquad \text{for all } i,j,k.$$

TABLE 3. Lower bounds subject to outside-in branching scheme.

Density	lower bound	k=1 value	k=1 ratio	$k=3$ value	$k=3$ ratio	$k=5$ value	$k=5$ ratio	$k=10$ value	$k=10$ ratio
0.1	LB_1	11,413	1.00	106,365	1.00	241,830	1.00	515,424	1.00
	LB_2	22,297	1.95	129,686	1.22	265,002	1.10	530,076	1.03
	LB_3	50,023	4.38	143,619	1.35	272,854	1.13	531,030	1.03
0.2	LB_1	43,299	1.00	282,387	1.00	462,986	1.00	723,402	1.00
	LB_2	81,428	1.88	324,608	1.15	496,798	1.07	727,837	1.01
	LB_3	112,749	2.60	333,710	1.18	498,486	1.08	727,837	1.01
0.3	LB_1	102,531	1.00	393,930	1.00	564,109	1.00	693,117	1.00
	LB_2	160,744	1.57	439,086	1.11	585,516	1.04	693,716	1.00
	LB_3	188,908	1.84	442,229	1.12	585,516	1.04	693,716	1.00

IIM, NCTU

83

TABLE 4. Lower bounds subject to sequential branching.

Density	lower bound	k=1 value	k=1 ratio	$k=3$ value	$k=3$ ratio	$k=5$ value	$k=5$ ratio	$k=10$ value	$k=10$ ratio
0.1	LB_1	0	N/A	7,764	1.00	22,918	1.00	87,358	1.00
	LB_2	6,541	N/A	26,995	3.48	51,991	2.27	127,306	1.46
	LB_3	37,745	N/A	49,103	6.32	67,435	2.94	132,759	1.52
0.2	LB_1	0	N/A	11,737	1.00	35,605	1.00	11,7147	1.00
	LB_2	24,446	N/A	74,227	6.32	113,462	3.19	206,271	1.76
	LB_3	71,501	N/A	96,454	8.22	125,353	3.52	212,210	1.81
0.3	LB_1	0	N/A	14,769	1.00	39,563	1.00	113,626	1.00
	LB_2	44,348	N/A	119,872	8.12	275,254	6.96	236,553	2.08
	LB_3	95,967	N/A	135,891	9.20	281,540	7.12	236,553	2.08

IIM, NCTU

84

Dominance Rule

- Assume that k shooting days are already scheduled in P. We consider the (k-1)th and the kth days.
- If actress i is required on at least one of the first k-2 days of P, then $\varepsilon_i(k\text{-}2) = 1$; otherwise, $\varepsilon_i(k\text{-}2) = 0$.
- If $\sum_{j=1}^{n} t_{ij} - \sum_{j \in P} t_{ij} > 0$, $\lambda_i(n\text{-}k) = 1$; otherwise, let $\lambda_i(n\text{-}k) = 0$.

IIM, NCTU

85

Lagrangian Relaxation

- M.L. Fisher, An applications oriented guide to Lagrangian relaxation, *Interfaces*, Vol. 15, No. 2, 1985, pp. 10-21.
- M.L. Fisher, The Lagrangian relaxation method for solving integer programming problems, *Management Science*, Vol. 27, No. 1, 1981, pp. 1-18.

IIM, NCTU

86

u is a vector of multipliers

Max $Z = cX$
Subject to
$AX \le b$
$DZ \le e$
$X \ge 0$ and integral

Add a non-negative term into Z

Max Z
$= cX + u(b - AX)$
Subject to
$AX \le b$
$DZ \le e$
$X \ge 0$ and integral

Provides an upper bound on Z

Remove some constraints

1. Removing the constraints cannot decrease the optimal value
2. The relaxed problem is relatively easy to solve

Max $Z = cX + u(b - AX)$
Subject to
$DZ \le e$
$X \ge 0$ and integral

IIM, NCTU

87

$$
\begin{aligned}
&\text{Max } Z = 16x_1 + 10x_2 + 4x_4 \\
&\text{subject to} \\
&\quad \boxed{8x_1 + 2x_2 + x_3 + 4x_4 \le 10} \\
&\quad x_1 + x_2 \le 1 \\
&\quad x_3 + x_4 \le 1 \\
&\quad x_j = 0 \text{ or } 1,\ j = 1,2,3,4.
\end{aligned}
$$

$$
\begin{aligned}
&\text{Max } Z_D(u) = (16-8u)x_1 + (10-2u)x_2 + (0-u)x_3 + (4-4u)x_4 + 10u \\
&\text{subject to} \\
&\quad x_1 + x_2 \le 1 \\
&\quad x_3 + x_4 \le 1 \\
&\quad x_j = 0 \text{ or } 1,\ j = 1,2,3,4.
\end{aligned}
$$

For a fixed u, this problem can be easily solved.

Goal: Find some u>0 for which $Z_D(u)$ is minimized

IIM, NCTU

88

$$\text{Max } Z_D(u) = (16-8u)x_1 + (10-2u)x_2 + (0-u)x_3 + (4-4u)x_4 + 10u$$

subject to

$$x_1 + x_2 \le 1$$

$$x_3 + x_4 \le 1$$

$$x_j = 0 \text{ or } 1,\ j = 1,2,3,4.$$

$$\text{Max } Z = 16x_1 + 10x_2 + 4x_4$$

subject to

$$8x_1 + 2x_2 + x_3 + 4x_4 \le 10$$

$$x_1 + x_2 \le 1$$

$$x_3 + x_4 \le 1$$

$$x_j = 0 \text{ or } 1,\ j = 1,2,3,4.$$

u	x_1	x_2	x_3	x_4	$Z_D(u)$	Z
0	1	0	0	1	20	-
6	0	0	0	0	60	0
3	0	1	0	0	34	10
2	0	1	0	0	26	10
1	1	0	0	0	18	16
1	1	0	0	1	18	-
1	0	1	0	0	18	10
1	0	1	0	1	18	14
1/2	1	0	0	1	19	-
3/4	1	0	0	1	18.5	-

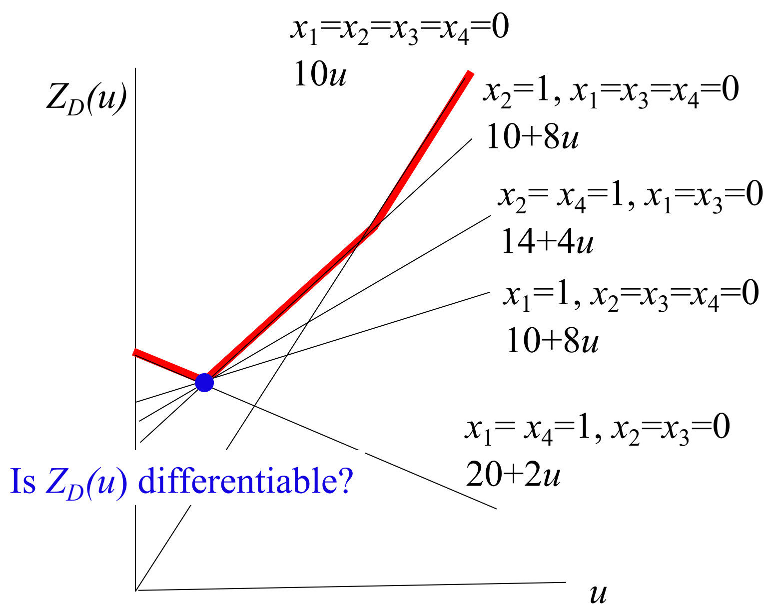
IIM, NCTU

89

- Is $Z_D(u) = 18$ optimal?
- If we substitute any x into the objective function for the Lagrangian problem, we obtain a linear function of u.
- The function is given by the upper envelop of a family of linear equations corresponding to the possible x's.

IIM, NCTU

90



IIM, NCTU

91

Non-differentiable Case

$$u^{k+1} = \max\{0,\ u^k - t_k(b - Ax^k)\}. \quad (6)$$

In this formula, t_k is a scalar stepsize and x^k is an optimal solution to $(LR_u{}^k)$, the Lagrangian problem with dual variables set to u^k.

IIM, NCTU

92

$u^0 = 0$
$u^1 = 0 - (-2) = 2$
$u^2 = \max\{0, 2\text{-}8\} = 0$
$u^3 = 0 - (-2) = 2$
$u^4 = \max\{0, 2\text{-}8\} = 0$

Table 2: Subgradient method with $t_k = 1$ for all k.

$u^0 = 0$
$u^1 = 0 - (-2) = 2$
$u^2 = \max\{0, 2 - ½(8)\} = 0$
$u^3 = 0 - ¼(-2) = ½$
$u^4 = ½ - ⅛(-2) = ¾$
$u^5 = ¾ - 1/16(-2) = ⅞$
$u^6 = ⅞ - 1/32(-2) = 15/16$

Table 3: Subgradient method with $t_k = 1, ½, ¼, ⅛, \ldots$

$u^0 = 0$
$u^1 = 2$
$u^2 = \max\{0, 2 - 1/3(8)\} = 0$
$u^3 = 0 - 1/9(-2) = 2/9$
$u^4 = 2/9 - 1/27(-2) = .296$
$u^5 = .296 - 1/81(-2) = .321$
$u^6 = .321 - 1/243(-2) = .329$
$u^7 = .329 - 1/729(-2) = .332$

Table 4: Subgradient method with $t_k = 1, 1/3, 1/9, 1/27, 1/81, \ldots$

If $t_k \to 0$ and $\sum_{i=1}^{k} t_i \to \infty$ as $k \to \infty$

then $Z_D(u_k)$ converges to its optimal

solution value Z_D.

IIM, NCTU

93

An effective procedure

$$t_k = \frac{\lambda_k\left(Z_D(u^k) - Z^*\right)}{\sum_{i=1}^{m}\left(b_i - \sum_{j=1}^{n} a_{ij} x_j^k\right)^2}$$

IIM, NCTU

94

Relationship with LP Solutions

the LP dual of example (1) – (5) is

$$\min\ 10_u + v_1 + v_2 + w_1 + w_2 + w_3 + w_4$$

$$8u + v_1 + w_1 \geq 16$$
$$2u + v_1 + w_2 \geq 10$$
$$u + v_2 + w_3 \geq 0$$
$$4u + v_2 + w_4 \geq 4$$
$$u,\ v_1,\ v_2,\ w_1, \ldots, w_4 \geq 0.$$

The optimal solution to the primal LP is $x_1 = 1$, $x_2 = 0$, $x_3 = 0$, $x_4 = ½$ and the optimal solution to the dual LP is $u = 1$, $v_1 = 8$, $v_2 = w_1 = \ldots w_4 = 0$.

$$Z_D = \min_{u \geq 0} \{ \max_{\substack{x \\ Dx \leq e \\ x \geq 0 \text{ and integral}}} (cx + u(b - Ax)) \}$$

$$\leq \min_{u \geq 0} \{ \max_{\substack{x \\ Dx \leq e \\ x \geq 0}} (cx + u(b - Ax)) \}$$

$$\text{(by } LP \text{ duality)} = \min_{u \geq 0} \{ \min_{\substack{v \geq 0 \\ vD \geq c - uA}} ub + ve \}$$

$$= \min_{\substack{u, v \geq 0 \\ uA + vD \geq c}} ub + ve$$

$$\text{(by } LP \text{ duality)} = \max_{\substack{Ax \leq b \\ Dx \leq e}} cx$$

$$= Z_{LP}$$

$Z_D < Z_{LP}$

$Z_D = Z_{LP}$

LP relaxation can improve the Lagrangian bound.

Max $Z = 16x_1 + 10x_2 + 4x_4$

subject to

$$8x_1 + 2x_2 + x_3 + 4x_4 \le 10$$
$$x_1 + x_2 \le 1$$
$$x_3 + x_4 \le 1$$
$$x_j = 0 \text{ or } 1,\ j = 1,2,3,4.$$

$$Z_D(v_1,v_2) = \max\ (16-v_1)x_1 + (10-v_1)x_2 + (0-v_2)x_3 + (4-v_2)x_4 + v_1 + v_2$$

subject to $8x_1 + 2x_2 + x_3 + 4x_4 \le 10$ (2)

$0 \le x_j \le 1,\ j=1,\ldots,4$ (5)

x_j integral, $j=1,\ldots,4$ (6)

IIM, NCTU

97

Lagrangian Solution

v_1	v_2	λ_k	x_1	x_2	x_3	x_4	$Z_D(v_1,v_2)$	Z^*
0	0	1	1	1	0	0	26	0
13	0	1	0	0	0	1	17	4
(feasible with $Z = 4$)								
0	0	1	1	1	0	0	26	4
11	0	1	1	0	0	0	16	16
(feasible with $Z = 16$)								

Table 5: The subgradient method applied to the improved relaxation.

IIM, NCTU

98

Key Issues for Lagrangian Relaxation

- Which constraint set to relax
- How to compute good multipliers *u*
- How to deduce a good, feasible solution to the original problem

IIM, NCTU

99