cs681fig15-18.cdr
Wednesday, April 24, 2024 10:09:52 AM
Color profile: Disabled

Composite Default screen

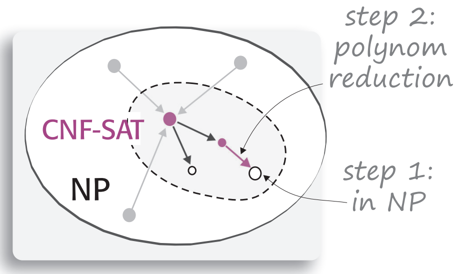
NP-Complete Problems
Practical Implications

➪ Karp reducibility

Recall, theory relies on decision versions of such problems.

Exercise
Describe a polynomial function to reduce *all-pair counts of k-edge paths* in graph to problem of matrix multiplications, determine efficiency.

- ➪ 2-Step proof (Karp 21)
- ➪ Solving one is enough
- ➪ Important to recognize



Choices grow combinatorially, must search through an exponentially increasing space.

➪ Combinatorial explosion

➪ Interesting instances?

Branch-and-bound can cut search drastically for <u>some</u> instances (unpredictable)

Dealing with Difficulty

➪ **Search/state space**

Build on promising ones, cut short otherwise.

Evaluate partially constructed solutions

When any solution is **feasible** (e.g., HC in TSP), a path may turn unpromising late, requiring a backtrack to look for the optimal one.

- ✎ Backtracking
- ✎ Branch-and-bound (optimization)

Track components of a partial solution (in exhaustive search the search space is made up of fully formed solutions).

➪ State-space trees

➪ Compare methods (tentative, more later)

cs681fig15-18.cdr
Wednesday, April 24, 2024 10:09:54 AM
Color profile: Disabled

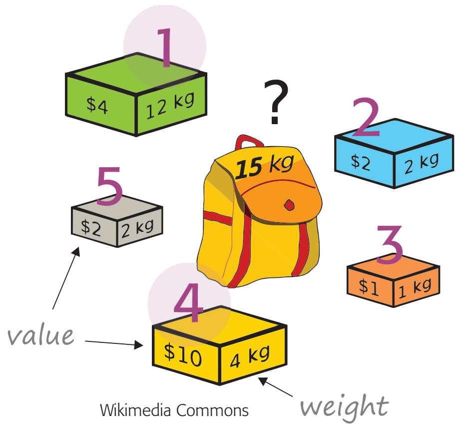
Composite Default screen

Knapsack Problem

➪ Exhaustive search

	{1}	12	4
	{2}	2	2
	{3}	1	1
	{4}	4	10
	{5}	2	2
	{1,2}	14	6
	{1,3}	13	5
	{1,4}	16	14
	. . .		
×	{**1**,2,**4**}	18	16
	. . .		
×	{**1**,3,**4**}	17	–
	. . .		
×	{**1**,**4**,5}	18	–
	. . .		
×	{**1**,2,3,**4**}	19	–
	. . .		
×	{**1**,2,**4**,5}	20	–
×	{**1**,3,**4**,5}	19	–
	. . .		
×	{**1**,2,3,**4**,5}	21	–

Clearly, once {1,4} is encountered, any subset containing 1,4 is not worth looking into.



Wikimedia Commons

Classic formulation (statement)

Fit the most valuable set of items without exceeding knapsack capacity

cs681fig15-18.cdr
Wednesday, April 24, 2024 10:09:54 AM
Color profile: Disabled

Composite Default screen

Knapsack Problem
An Upper Bound

➪ **Bounding function**

4	\$40	10
7	\$42	6
5	\$25	5
3	\$12	4

➪ Bounds, rationale *value per-unit weight*

$$10 \bullet \{\overset{2}{7}, \overset{4}{3}, \overset{1}{4}, \overset{3}{5}\}$$

$$\{\$42, \$12, \$40, \$25\}$$

➪ Example

$$v + (\underset{10}{W} - w)\frac{v_{i+1}}{\underset{0}{w_{i+1}}}$$

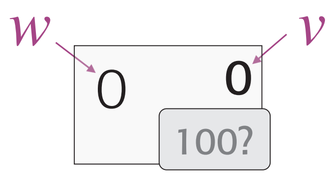
Quiz
Describe (in words) the multiplicative ratio v/w in the bounding function. Interpret the term W–w.
Ans. this slide.

Exercise
Write the expression for the initial upper bound.

➪ State node

✎ A partial solution

✎ Initially, sack empty



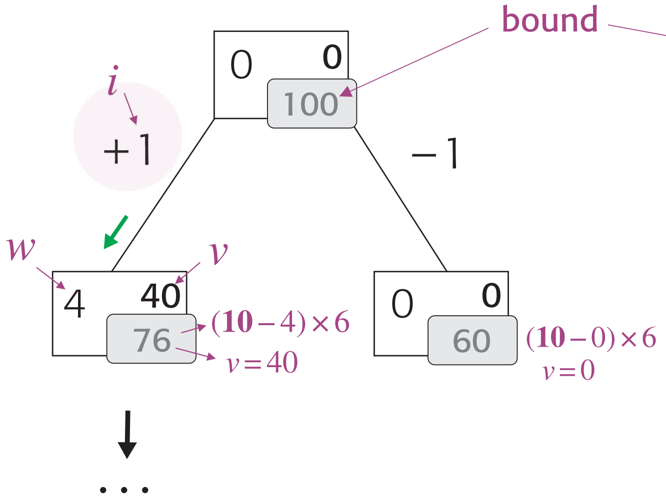
Simply value per weight (SAR/kg or \$/lb), i.e, best choice to add (in value) in remaining capacity, if possible.

Try next highest per-unit value

A naturally binary decision to add or not item i=1.

Quiz
Why pick the bigger number?

In remaining capacity, after weight 4 is fit (whose value was 40), pick next item.



$$v + (\underbrace{W}_{10} - w)\frac{v_{i+1}}{\underbrace{w_{i+1}}_{0}}$$

i	w_i	v_i	v_i/w_i
1	4	40	10
2	7	42	6
3	5	25	5
4	3	12	4

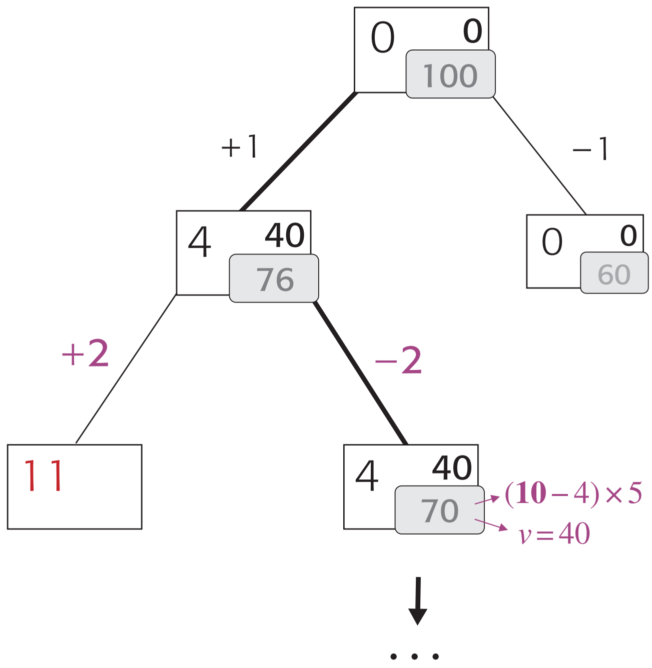
+	−
76 ✓	60

Knapsack Problem
A Search Tree

➪ Feasible solution

Nodes reflect components of a partial solution, transitions choices leading to another one.

Not **feasible**, i.e., exceeded capacity (10).



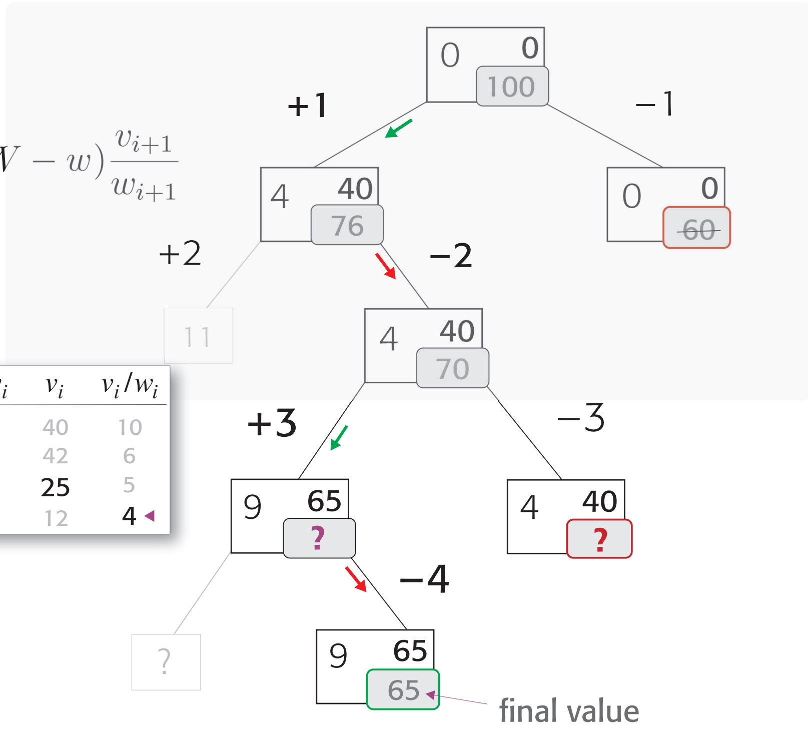
$$v + (\underset{10}{W} - w)\frac{v_{i+1}}{w_{i+1}}$$

i	w_i	v_i	v_i/w_i
1	4	40	10
2	7	42	6
3	5	25	5
4	3	12	4

Knapsack Problem
Branch-and-Bound

Red cards mark feasible solutions (criteria for a candidate solution are satisfied), green marks the optimal one according to search.

$$v + (W - w)\frac{v_{i+1}}{w_{i+1}}$$



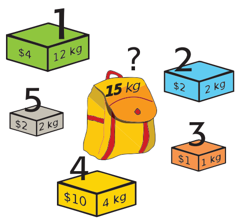
i	w_i	v_i	v_i/w_i
1	4	40	10
2	7	42	6
3	5	25	5
4	3	12	4

3rd item can fit, adding 5, 25 to partial solution.

Last item (i=4) will not fit, hence may not be added.

Branch-bound Knapsack
Check Result

Quiz
Determine the counts: nodes max in tree, nodes bound checked, nodes bound calculation avoided.

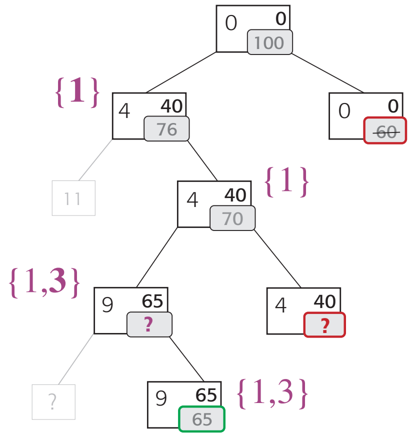


Wikimedia Commons

Exercise
Construct a tree for the Knapsack instance from the review slide.

Exercise
Compare to the exhaustive search solution. Hints: use Excel to generate table, lookup websites that generate subsets.

i	w_i	v_i	v_i / w_i
1	4	40	10
2	7	42	6
3	5	25	5
4	3	12	4



2 4 **1 3**
1 2 **3 4**
$10 \bullet \{7, 3, 4, 5\}$
$\{\$42, \$12, \$40, \$25\}$
▲ ▲

ϕ	0	0
{1}	7	42
{2}	3	12
{3}	4	40
{4}	5	25
{1,2}	10	54
{1,3}	11	–
{1,4}	12	–
{2,3}	7	52
{2,4}	8	37
▶ {3,4}	9	65 ◀
{1,2,3}	14	–
{1,2,4}	15	–
{1,3,4}	16	–
{2,3,4}	12	–
{1,2,3,4}	19	–

Efficiency? ***compare***

cs681fig15-18.cdr
Wednesday, April 24, 2024 10:09:59 AM
Color profile: Disabled

Composite Default screen

Branch-bound Knapsack
Compare Methods

Convert search of combinatorially increasing items to a tree search (DFS, backtracking etc., guided by a bounding function in branch-bound).

	From	*To*
Search	Sequential	Tree
State-space	Full solution	Partial solution
Exclusion	Cost function	Boundary function
Optimal solution		
Guarantee	*Always*	
At cost	$\Theta(2^n)$	
Solution path		

cs681fig15-18.cdr
Wednesday, April 24, 2024 10:09:59 AM
Color profile: Disabled

Composite Default screen

Branch-bound Knapsack Efficiency

cs681fig15-18.cdr
Wednesday, April 24, 2024 10:09:59 AM
Color profile: Disabled
Composite Default screen

Traveling Salesman

	a	b	c	d	e
a	–	3	1	5	8
b	3	–	6	7	9
c	1	6	–	4	2
d	5	7	4	–	3
e	8	9	2	3	–

HC observations:
Tour pattern, combinatorial object, arbitrary role of tour start/end verts, opposite tours redundant (pick: 2-before-3 rule).

Quiz
How many unique HC are checked by exhaustive search? (Ans. next slide).

Classic formulation (statement)

Shortest tour through a set of cities visiting each exactly once before returning to the start

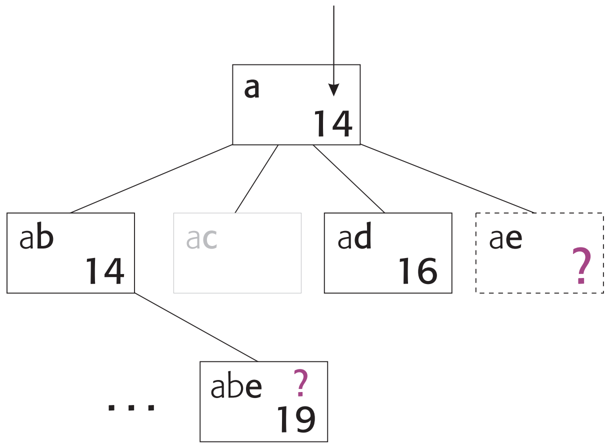
Traveling Salesman
A Lower Bound

➪ Nearest pair average

An upper bound previously signaled a more promising branch to find an optimally <u>maximized</u> value-sum.

	a	b	c	d	e
a	–	**3**	**1**	5	8
b	**3**	–	**6**	7	9
c	**1**	6	–	4	**2**
d	5	7	**4**	–	**3**
e	8	9	**2**	**3**	–

$$\left\lceil \left[(1+3)+(3+6)+(1+2)+(3+4)+(2+3)\right]/2 \right\rceil$$



	a	b	c	d	e
a	–	**3**	**1**	5	8
b	**3**	–	**6**	7	9
c	**1**	6	–	4	**2**
d	5	7	**4**	–	**3**
e	8	9	**2**	**3**	–

	a	b	c	d	e
a	–	3	**1**	**5**	8
b	**3**	–	**6**	7	9
c	**1**	6	–	4	**2**
d	**5**	7	4	–	**3**
e	8	9	**2**	**3**	–

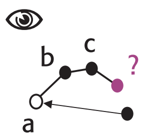
Exercise
Write an expression for indicated bounds.

Branch-and-Bound Optimal Tour

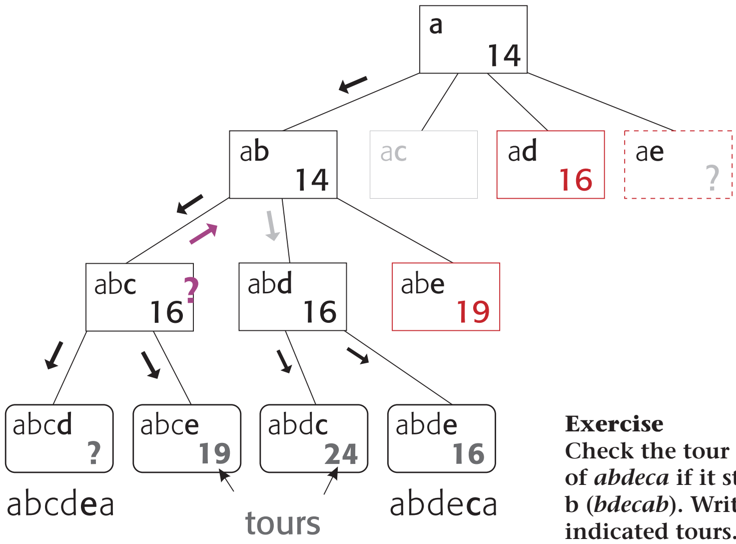
- Live node
- Best-first rule

		a	b	c	d	e
1	a	–	3	1	5	8
2	b	3	–	6	7	9
3	c	1	6	–	4	2
4	d	5	7	4	–	3
5	e	8	9	2	3	–

1,2,3,4,5,**1**	24	1,5,2,4,3,1	29
1,3,2,4,5,1	25	1,2,5,4,3,1	20
1,4,2,3,5,1	28	1,4,5,2,3,1	24
1,2,4,3,5,1	24	1,5,4,2,3,1	25
1,3,4,2,5,1	29	1,2,4,5,3,1	16
1,4,3,2,5,1	32	1,4,2,5,3,1	24
1,4,3,5,2,1	23	1,3,2,5,4,1	24
1,3,4,5,2,1	20	1,2,3,5,4,1	19
1,5,4,3,2,1	24	1,5,3,2,4,1	28
1,4,5,3,2,1	19	1,3,5,2,4,1	24
1,3,5,4,2,1	16	1,2,5,3,4,1	23
1,5,3,4,2,1	24	1,5,2,3,4,1	32



Choices end once the before-last internal city is determined.



Check paths starting with 14, 15.

Exercise
Write an expression for indicated bound.

Exercise
Check the tour length of *abdeca* if it starts at b (*bdecab*). Write indicated tours.

	a	b	c	d	e
a	–	**3**	1	5	8
b	3	–	6	**7**	9
c	**1**	6	–	4	2
d	5	7	4	–	**3**
e	8	9	**2**	3	–

cs681fig15-18.cdr
Wednesday, April 24, 2024 10:09:59 AM
Color profile: Disabled

Composite Default screen

Branch-and-Bound
Performance Summary

⇨ **Heuristic rule**

Not all instances, good ones (solvable in reasonable time) unpredictable, bounding functions vary (some better tailored for application or a subset of instances).

Problem may suggest **heuristics** to speed up solution (cut branches earlier, for example).

⇨ Limitations vs opportunities

⇨ Compare to brute-force

⇨ Assignment vs. the other two

Tractable based on known efficiency (how come?) Is there a lower bound?

Exercise
Compare efficiency of solution of the assignment problem by the B&B in textbook to the best versions of the **Hungarian method** and to a trivial lower bound.

LB node abe, starting from ab(14)
(1+3)+(3+9)+(1+2)+(3+4)+(2+9) = 37, 37/2 = 18.5

	a	b	c	d	e
a	–	**3**	**1**	5	8
b	**3**	–	6	7	**9**
c	**1**	6	–	4	**2**
d	5	7	**4**	–	**3**
e	8	**9**	**2**	3	–

LB abc, starting from ab(14)
(1+3)+(3+6)+(1+6)+(3+4)+(2+3) = 32, 32/2 = 16

	a	b	c	d	e
a	–	**3**	**1**	5	8
b	**3**	–	**6**	7	9
c	**1**	**6**	–	4	2
d	5	7	**4**	–	**3**
e	8	9	**2**	**3**	–