chang@umbc.edu

Greedy Algorithms

- intuitive
- usually fast
- requires proof

Conventional Wisdom

- don't be greedy
- first is the worst,
  second is the best
  third is the one with the treasure chest
- largest handful not the most

but, I have a mathematical proof of optimality

Activity Selection

n activities (TV shows)

activity i has start time s_i & finish time f_i

Problem:

find a subset of activities that do not overlap:

activity i ends before activity j starts

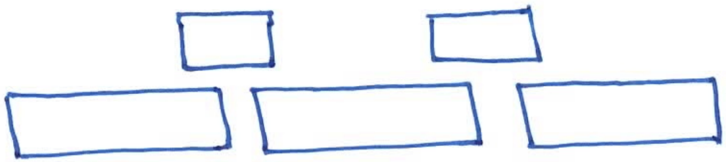
either $f_i \leq s_j$ or $f_j \leq s_i$

j ends before i starts

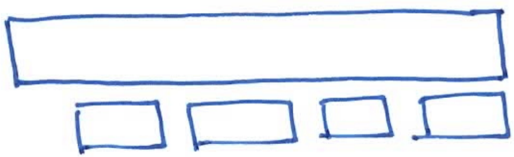
Maximize number of activities in the subset.

Greedy selection criteria that don't work:

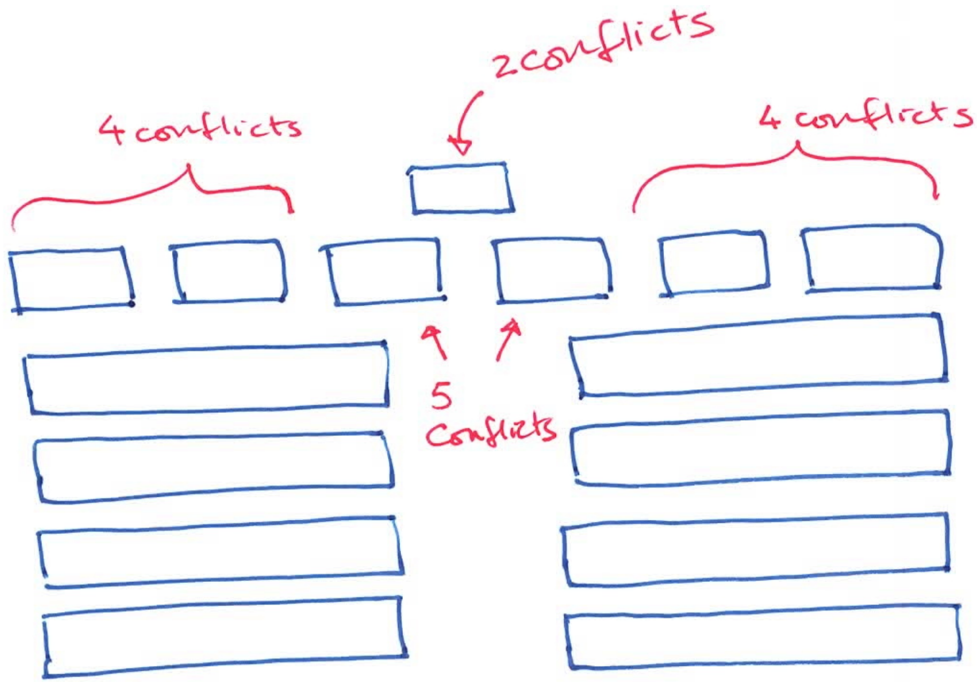
1. Shortest first



2. Earliest start time



③ Fewest conflict first



Earliest Finishing Time:

1. Pick activity i s.t. f_i is smallest of all available activities.
2. Remove all activities that conflict with i.
3. Add i to the schedule.
4. If any activities remain, repeat.

Prove that earliest finish time is optimum

- A proof is necessary!

- Q: what is the last line of the proof before you can claim optimality?

and sufficient too

Necessary condition for optimality:

For every "legal" schedule X, the schedule produced by the earliest finish time greedy algorithm has at least as many activities as in X.

Very common totally fallacious argument:

Earliest finishing time is optimum because it greedily selects the activity that has the smallest finishing time. This gives a schedule that has more activities than any other schedule.

not a proof, just restates what the algorithm does

This is what must be proven. Just saying that it is true does not constitute a proof.

Variation: It is always better to pick the activity with the earliest finishing time.

But WHY?

Need an argument that uses earliest finish time

Proof by induction on number of activities n.

Induction Hypothesis $P(n)$:

The greedy earliest finishing time algorithm produces an optimum schedule for every Activity Selection Problem that has n activities.

Base Case:

proof by "inspection"

$P(1)$ holds since greedy will pick the only available activity and that is optimum.

Induction Case:

"Strong" induction.

Assume $P(k)$ holds for all $k < n$.

Means we already know Greedy is optimum for smaller problems

Need to show that $P(n)$ holds.

Yes, we can do that!

Proof by contradiction:

WAKE UP! This will be on a Quiz and in the Final Exam!!

Suppose some schedule X has more activities than schedule A, produced by greedy.

Let activities in A & X be (in sorted order):

$$A = (a_1, a_2, a_3, \ldots\ldots, a_t)$$

$$X = (x_1, x_2, x_3, \ldots\ldots, x_{t+1})$$

if X has even more activities just ignore them.

What do we know?

$f(a) =$ finish time of acti ity a.

$s(a) =$ start time of activity a.

$f(a_1) \leq f(x_1)$ ← why?

$f(a_1) \leq s(x_2)$ ← why?

So we can replace x_1 with a_1 and still have a legal schedule

$$A = (a_1, a_2, a_3, \ldots, a_t)$$
$$X = (\underset{a_1}{\cancel{x_1}}, x_2\ x_3, \ldots, x_t, x_{t+1})$$

$f(a_1) \leq f(x_1)$

because x_1 was available to the greedy algorithm when a_1 was picked. If $f(x_1) < f(a_1)$ then x_1 had an earlier finish time than a_1 and would have been chosen instead.

means x_1 finished before x_2 started

$f(x_1) \leq s(x_2)$

since x_1 & x_2 do not conflict.

So, we also know that $f(a_1) \leq s(x_2)$.

Let S be the original set of activities.

includes a_1

Let $S_1 = S - \{\text{activities that conflict with } a_1\}$

What solution does greedy produce for problem S_1?

FLASH $\boxed{\text{Important:}}$ FLASH

BLINK! BLINK! BLINK! BLINK!

Whatever is the solution produced by greedy for S_1 must be the optimum solution for S_1 because $|S_1| < |S|$, BY THE INDUCTION HYPOTHESIS.

This is where we use the induction hypothesis.

Solution produced by greedy for problem S_1:

$A_1 = (a_2, a_3, a_4, \ldots, a_t)$

This is A with a_1 removed.
Relies on optimal substructure.

BUT

$X_1 = (x_2, x_3, x_4, \ldots, x_t, x_{t+1})$

is also a solution for problem S_1

How can this be?

We argued earlier that $f(a_1) \le s(x_2)$.
We also know that $s(x_2) < s(x_3) < \cdots < s(x_{t+1})$
So, none of $x_2, \ldots, x_{t+1}$ conflict with a_1.
Then, greedy failed to produce an opt. solution!

IT'S a CONTRADICTION!

END OF PROOF BY CONTRADICTION

So, our original assumption that some schedule X has more activities than A must be wrong.

Continue with outer proof by induction.

Therefore, A is optimum.

because it has at least as many activities as any other schedule X.

Thus, greedy produces an optimum schedule for any Activity Selection Problem with n activities.

$\therefore$ P(n) holds.

Hiding the proof by induction.

Suppose greedy isn't always optimum.
Let n be the smallest s.t. for some problem S
with n activities, greedy fails to produce opt. solution.

Let $A = (a_1, \ldots, a_t)$ be the greedy solution.
Let $X = (x_1, \ldots, x_t, x_{t+1})$ be a better solution.

$\vdots \; \}$ same as before

$A_1 = (a_2, \ldots, a_t)$ is the greedy solution for problem S_1.
$X_1 = (x_2, \ldots, x_t, x_{t+1})$ is a better solution for problem S_1.

Since $|S_1| < |S|$, n is not smallest. $\Rightarrow\Leftarrow$

Technicality:

① "Let n be the smallest s.t. ..."

uses the well-foundedness of natural numbers:

Every non-empty set of natural numbers has a least element.

② Maybe $n=1$. Then, what?

$$\text{“I still} \left\{ \begin{array}{l} \text{am confused by} \\ \text{hate} \\ \text{loathe} \end{array} \right\} \text{induction.”}$$

... my whole life long...

Use the ... method

pronounced dot, dot, dot

The ... method

Still uses proof by contradiction.

Suppose some schedule X is better than the greedy solution A.

$$A = (a_1, a_2, a_3, \ldots, a_t)$$

$$X = (\underset{a_1}{\cancel{x_1}}, x_2, x_3, \ldots, x_t, x_{t+1})$$

We can swap x_1 for a_1 and still have a legal schedule because

Since a_1 has earliest finish time. $\rightarrow f(a_1) \le f(x_1) \le s(x_2)$

What about x_2?

$$A = (a_1, a_2, a_3, \ldots, a_t)$$

$$X = (\underset{a_1}{\cancel{x_1}}, \underset{a_2}{\cancel{x_2}}, x_3, \ldots, x_t, x_{t+1})$$

activity x_2 was available when a_2 was picked by greedy. So, $f(a_2) \le f(x_2) \le s(x_3)$

so a_2 & x_3 do not conflict

So we can swap x_2 for a_2.

What about x_3? x_4? $\underbrace{\ldots}_{\text{hence the name}}$ x_t?

$$A = (a_1, a_2, a_3, \ldots, a_t)$$

$$X = (\underset{a_1}{\cancel{x_1}}, \underset{a_2}{\cancel{x_2}}\ \underset{a_3}{\cancel{x_3}}, \ldots, \underset{a_t}{\cancel{x_t}}, x_{t+1})$$

So, $X' = (a_1, a_2, a_3, \ldots, a_t, x_{t+1})$ is a solution.

What's the contradiction?

Proof by contradiction ending #1:

X is a legal schedule, so

$$f(a_t) \le f(x_t) \le s(x_{t+1})$$

means activity x_{t+1} was available after a_t was picked. The "real" greedy algorithm would have picked x_{t+1}.

Therefore, $A = (a_1, a_2, \ldots, a_t)$ cannot be the solution produced by greedy. $\Rightarrow\Leftarrow$

not a "mushy" ending. We can point to p and $\neg p$.

Proof by contradiction ending #2: (alternate ending)

Since x_{t+1} was not picked by greedy, it must have conflicted with one of $a_1, \ldots, a_t$ and was removed.

Thus, $s(x_{t+1}) < f(a_i)$ for some i.

Since $f(a_1) < f(a_2) < \cdots < f(a_t)$, $s(x_{t+1}) < f(a_t)$.

But $f(a_t) \leq f(x_t)$, so $s(x_{t+1}) < f(x_t)$.

assuming $f(x_t) < f(x_{t+1})$

Therefore, activities x_t and x_{t+1} conflict.

Thus, X is not a legal schedule. $\Rightarrow\Leftarrow$

Fractional Knapsack

like Knapsack except you can divide items.

Knapsack with capacity K.

n items.

$V(x)$ = value of item x

$w(x)$ = weight of item x

If fraction α of item x taken,

then value $= \alpha V(x)$

weight $= \alpha w(x)$

Take items such that total value is maximized & total weight $\leq K$.

Greedy:

1. Sort items by v/w ratio

2. Starting with highest v/w ratio, take as much of next item as knapsack can hold

Suppose some packing B has higher value than greedy solution A.

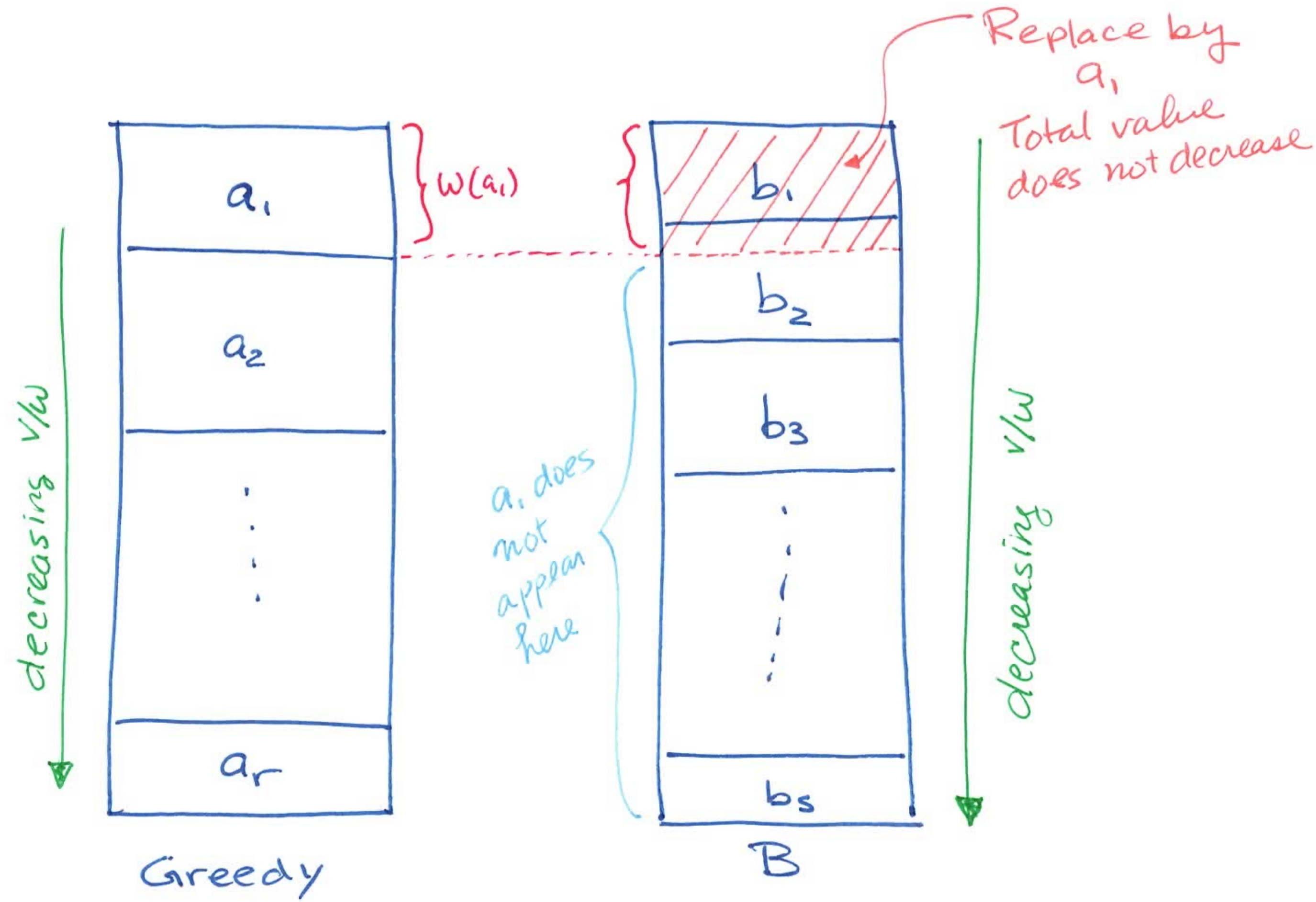
$$A = (a_1, a_2, \dots, a_r)$$

$$B = (b_1, b_2, \dots, b_s)$$

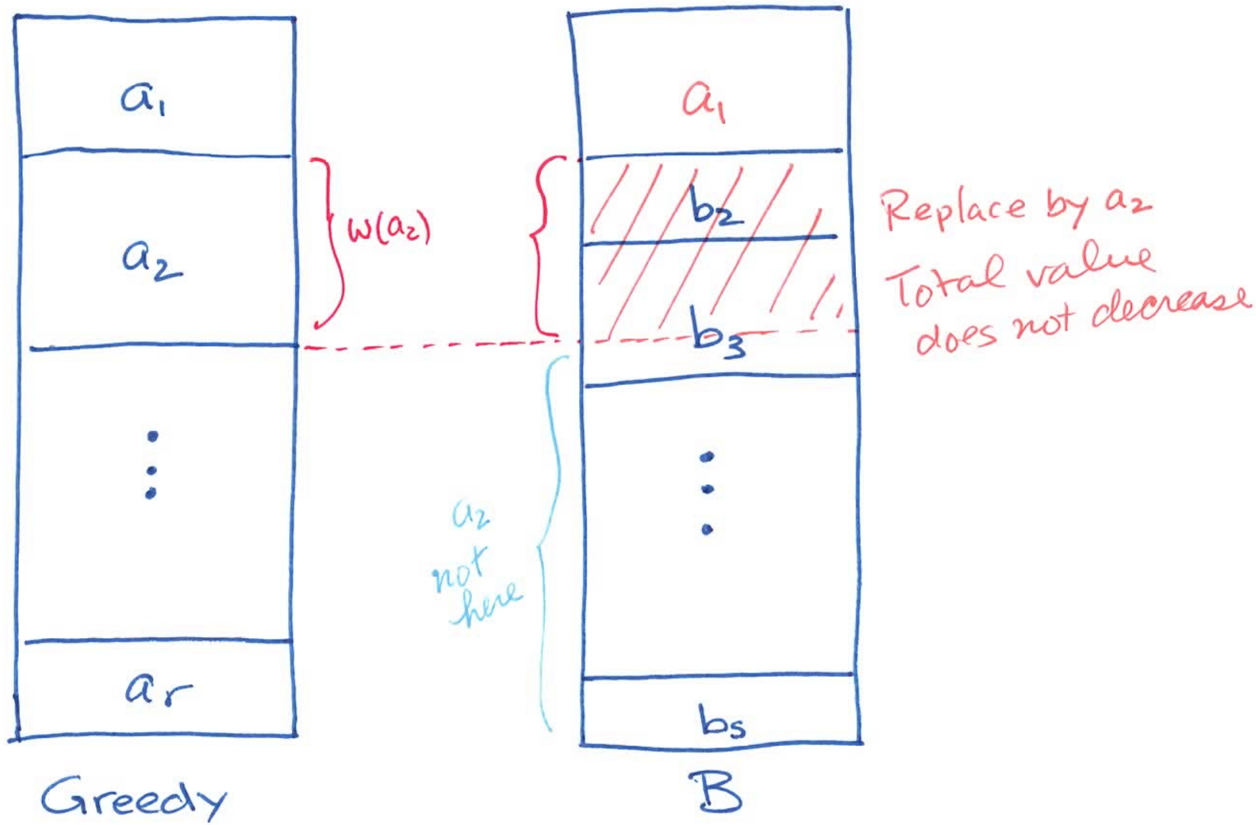
no relationship assumed about r & s

Assume items are ordered by v/w ratio.

Note: B does not use v/w ratio, but we can still rearrange the items after they are picked.



Replace by a_1
Total value does not decrease
a_1
$w(a_1)$
a_2
a_r
decreasing v/w
Greedy
b_1
b_2
b_3
b_s
a_1 does not appear here
decreasing v/w
B


a_1
a_2
$w(a_2)$
a_r
Greedy
a_1
b_2
b_3
b_s
B
Replace by a_2
Total value does not decrease
a_2 not here

... Replace all of B with A.

Each replacement guaranteed not to decrease total value of B.

Let $B^{(i)} =$ B after i^{th} replacement.

first replacement

r = # of items in A

$$v(B) \leq v(B^{(1)}) \leq v(B^{(2)}) \leq \cdots \leq v(B^{(r)}) = v(A)$$

$A = B^{(r)}$

Thus, $v(B) \leq v(A)$.

suppose B is <u>better</u> than A...

But, we assumed $v(B) > v(A)$ $\Rightarrow\Leftarrow$