**CS 341: ALGORITHMS**

Lecture 6: greedy algorithms II
Readings: see website
Trevor Brown
https://student.cs.uwaterloo.ca/~cs341
 trevor.brown@uwaterloo.ca

---

**OPTIMALITY PROOF**

for greedy interval selection

---

**Goal:** choose as many disjoint intervals as possible, (i.e., without any overlap)

**Algorithm:**

Sort the intervals in increasing order of finishing times. At any stage, choose the earliest finishing interval that is disjoint from all previously chosen intervals (i.e., the local evaluation criterion is $f_i$).

---

**PROVING OPTIMALITY**

Consider an input $A[1..n]$
- Let $G$ be the greedy solution
- Let $O$ be an optimal solution
- "Greedy stays ahead" argument
  - Intuition: out of the given set of intervals, greedy picks as many as optimal

---

**VISUAL EXAMPLE**

Input

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$G$

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_1$</td>
<td>$G_2$</td>
<td>$G_3$</td>
<td>$G_4$</td>
<td>$G_5$</td>
</tr>
</tbody>
</table>

$O$

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_1$</td>
<td>$O_2$</td>
<td>$O_3$</td>
<td>$O_4$</td>
<td>$O_5$</td>
</tr>
</tbody>
</table>

How to compare $G$ and $O$? Imagine reordering $O$ to match $G!$

---

**CRUCIAL:** We are NOT assuming the optimal algorithm uses the same sort order.

We are merely imagining reordering the intervals chosen by the optimal algorithm so we can easily compare their finish times to intervals in $G$.
REORDERING O BY INCREASING FINISH TIME

Now O' and G are both ordered by increasing finish time.
This ordering helps us leverage what we know about G in our comparison with O'.
Argue for a prefix of the intervals sorted this way, G chooses as many as O'.

COMPARING O' WITH G

Looks like \( f(G_i) \leq f(O'_i) \) and \( f(G_2) \leq f(O'_2) \)...
If this trend holds in general, then out of the intervals with finish time \( \leq f(O'_i) \),
G chooses as many intervals as O'!

PROVING LEMMA: \( f(G_i) \leq f(O'_i) \) FOR ALL i

Base case: \( f(G_1) \leq f(O'_1) \) since G chooses the interval with the earliest finish time first.

PROVING LEMMA: \( f(G_i) \leq f(O'_i) \) FOR ALL i

Inductive step: assume \( f(G_{i-1}) \leq f(O'_{i-1}) \). Show \( f(G_i) \leq f(O'_i) \).
- Since O' is feasible, \( f(O'_{i-1}) \leq s(O'_i) \)
- So \( f(G_{i-1}) \leq s(O'_i) \)
- So G can choose \( O'_i \) if it has the smallest finish time
- So \( f(G_i) \leq f(O'_i) \)

USING THIS LEMMA

- Suppose \( |O'| > |G| \) to obtain a contradiction
  - So if G chooses k intervals, O' chooses at least \( k + 1 \)
By the lemma, \( f(G_k) \leq f(O_k) \)
Since \( O' \) is feasible, \( f(O'_k) \leq s(O'_{k+1}) \)
But then G can, and would, pick \( O'_{k+1} \).
- Contradiction!

A DIFFERENT PROOF

"Slick" ad-hoc approaches are sometimes possible...
Let \( F = \{ f_1, \ldots, f_n \} \) be the finishing times of the intervals in \( X \).

No interval finishes strictly to the left

No interval starts strictly to the right

Would be chosen by greedy (contradiction)

So, in addition to the intervals in \( X \), only the following types of intervals are possible:

- Contains \( f_i \)
- Contains \( f_j \)
- Contains \( f_i \) and \( f_j \)

Thus, every interval contains some finishing time in \( F \).

And, two intervals cannot contain the same element of \( F \).

So, there must be as many finishing times in \( F \) as there are intervals in \( X \). QED

**POSSIBLE GREEDY STRATEGIES FOR KNAPSACK PROBLEMS**

**Strategy 1:** consider items in decreasing order of profit (i.e., we maximize the local evaluation criterion \( p_i \))

Let's try an example input:

- Profits \( P = [20, 50, 100] \)
- Weights \( W = [10, 20, 10] \)
- Weight limit \( M = 10 \)

Algorithm selects last item for 100 profit

Looks optimal in this example

**POSSIBLE GREEDY STRATEGIES FOR KNAPSACK PROBLEMS**

**Strategy 1:** consider items in decreasing order of profit (i.e., we maximize the local evaluation criterion \( p_i \))

How about a second example input:

- Profits \( P = [20, 50, 100] \)
- Weights \( W = [10, 20, 100] \)
- Weight limit \( M = 10 \)

Algorithm selects last item for 10 profit

Not optimal!
POSSIBLE GREEDY STRATEGIES FOR KNAPSACK PROBLEMS

**Strategy 2:** consider items in increasing order of weight
(i.e., we minimize the local evaluation criterion $w_i$)

Counterexample
- Profits $P = [20, 50, 100]$
- Weights $W = [10, 20, 100]$
- Weight limit $M = 10$
Algorithm selects first item for 20 profit
- It could select half of second item, for 25 profit!

**Strategy 3:** consider items in decreasing order of profit divided by weight
(i.e., we maximize local evaluation criterion $p_i/w_i$)

Let’s try our first example input
- Profits $P = 20, 50, 100$
- Weights $W = [10, 20, 100]$
- Weight limit $M = 10$
Profit divided by weight
- $P/W = [2, 2.5, 1]$
Algorithm selects last item for 100 profit (optimal)

POSSIBLE GREEDY STRATEGIES FOR KNAPSACK PROBLEMS

**Strategy 3:** consider items in decreasing order of profit divided by weight
(i.e., we maximize local evaluation criterion $p_i/w_i$)

Let’s try our second example input
- Profits $P = 20, 50, 100$
- Weights $W = [10, 20, 100]$
- Weight limit $M = 10$
Profit divided by weight
- $P/W = [2, 2.5, 1]$
Algorithm selects second item for 25 profit (optimal)

Algorithm selects second item for 25 profit (optimal)

POSSIBLE GREEDY STRATEGIES FOR KNAPSACK PROBLEMS

**Strategy 3:** consider items in decreasing order of profit divided by weight
(i.e., we maximize local evaluation criterion $p_i/w_i$)

Let’s try our second example input
- Profits $P = 20, 50, 100$
- Weights $W = [10, 20, 100]$
- Weight limit $M = 10$
Profit divided by weight
- $P/W = [2, 2.5, 1]$
Algorithm selects last item for 100 profit (optimal)

INFORMAL FEASIBILITY ARGUMENT
(SHOULD BE GOOD ENOUGH TO SHOW FEASIBILITY ON ASSESSMENTS)

Feasibility: all $x_i$ are in $[0, 1]$ and total weight is $\leq M$

Either everything fits in the knapsack, or:
- When we exit the loop, weight is exactly $M$
- Every time we write to $x_i$ it’s either 0, 1 or $(M - \text{weight})/w_i$ where weight + $w_i > M$
  - Rearranging the latter we get $(M - \text{weight})/w_i < 1$
  - And weight $\leq M$, so $(M - \text{weight})/w_i \geq 0$
  - So, we have $x_i \in [0, 1]$
MINOR MODIFICATION TO FACILITATE FORMAL PROOF

**Loop invariant:**

\[ \sigma - M = 0 \]

Rearrange to get

\[ w \cdot x + M = \sigma \cdot \text{weight} \]

Rearrange to get

\[ w \cdot x + M = 0 \]

Prove invariant holds at end of iteration

**Case 1:**

\[ \sigma = w \cdot x - M \]

Hence \( M = -w \cdot x \)

So \( w \cdot x \leq 0 \)

Base case. Initially weight = 0 and \( \sigma = x = 0 \).

- So \( w = 0 = \sum_{i=1}^{n} w_i \cdot 0 = \sum_{i=1}^{n} w_i \leq M \)

Inductive step.

Suppose invariant holds at start of iteration \( i \).

- Let weight', \( x'_i \) denote values of weight, \( x_i \) at end of iteration \( i \).
- Prove invariant holds at end of iteration \( i \).
- i.e., \( \sigma, x'_i \) \in [0,1] and weight' = \( \sum_{i=1}^{n} w'_i \cdot x'_i \leq M \)

**MINOR MODIFICATION TO FACILITATE FORMAL PROOF**

**GreedyRationalKnapsack**\( (p_1, \ldots, p_n), (w_1, \ldots, w_n), M) \)

1. \( X = [0, \ldots, 0] \)
2. \( w = 0 \)
3. for \( i = 1 \) to \( n \)
   - if weight + \( w_i \) > \( M \) then
     - weight = \( M \)
   - else
     - \( x[i] = 1 \)
     - weight = weight + \( w_i \)
4. return \( X \)

Optional slide, just for your notes

**FORMAL FEASIBILITY ARG**

**WTP:** \( v_i : x'_i \in [0,1] \)

and \( \text{weight} = \sum_{i=1}^{n} w'_i \cdot x'_i \leq M \)

**Case 1:** weight + \( w_i \leq M \)

- \( x'_i = 1 \) which is in \([0,1]\) (by line 11)
- weight' = weight + \( w_i \) (by line 12)
- and this is \( \leq M \) by the case
- weight' = \( \sum_{i=1}^{n} x'_i w_k + w_i \) (by invariant)
- weight' = \( \sum_{i=1}^{n} x'_i w_k + x'_i w_i \) (since \( x'_i = 1 \))
- and \( x'_i = x_k \) for all \( k \neq i \) and \( x_i = 0 \) so \( \sum_{k=1}^{n} x'_i w_k = x'_i w_i + \sum_{k=1}^{n} x'_i w_k \)
- Rearrange to get \( \sum_{k=1}^{n} x'_i w_k = \sum_{i=1}^{n} x'_i w_k - x'_i w_i \)
- So weight' = \( \sum_{i=1}^{n} x'_i w_k - x'_i w_i \) + \( x'_i w_i = \sum_{i=1}^{n} x'_i w_k \)

Optional slide, just for your notes

**FORMAL FEASIBILITY ARG**

**WTP:** \( v_i : x'_i \in [0,1] \)

and \( \text{weight} = \sum_{i=1}^{n} w'_i \cdot x'_i \leq M \)

**Case 2:** weight + \( w_i > M \)

- We have weight = \( M - w_i \) (by case)
- and \( w_i < M \) (by invariant)
- So \( w_i = \sum_{k=1}^{n} x'_i w_k \) (by line 8)
- weight' = \( \sum_{i=1}^{n} x'_i w_k \) (by invariant)
- But \( x'_i = x_k \) for all \( k \neq i \) and \( x_i = 0 \) so \( \sum_{k=1}^{n} x'_i w_k = x'_i w_i + \sum_{k=1}^{n} x'_i w_k \)
- Rearrange to get \( \sum_{i=1}^{n} x'_i w_k = \sum_{i=1}^{n} x'_i w_k - x'_i w_i \)
- So weight' = \( \sum_{i=1}^{n} x'_i w_k - x'_i w_i \) + \( M - \text{weight} \)
- And \( M - \text{weight} \) so weight' = \( \sum_{i=1}^{n} x'_i w_k \)

Optional slide, just for your notes

**OPTIONAL – AN EXCHANGE ARGUMENT**

For simplicity, assume that the profit / weight ratios are all distinct, i.e.,

\[
\frac{p_1}{w_1} > \frac{p_2}{w_2} > \ldots > \frac{p_n}{w_n}
\]

Suppose the greedy solution is \( X = \{x_1, \ldots, x_n\} \) and the optimal solution is \( Y = \{y_1, \ldots, y_n\} \).

We will prove that \( X = Y \), i.e., \( x_j = y_j \) for \( j = 1, \ldots, n \). Therefore there is a unique optimal solution and it is equal to the greedy solution.

Suppose \( X \neq Y \). To obtain a contradiction

Pick the smallest integer \( j \) such that \( x_j \neq y_j \). If \( X \) and \( Y \) are identical up to \( x_j \) and \( y_j \), respectively

EXCHANGE ARGUMENT

for proving optimality

Optional slide, just for your notes

Optional slide, just for your notes

Optional slide, just for your notes

Optional slide, just for your notes
What's the relationship between $x_j$ and $y_j$?

Can we have $y_j > x_j$?

No! Greedy would take more of item $j$ if it could.

Must have $y_j < x_j$.

Since item $j$ is worth more per unit weight, replacing even a tiny amount of item $j$ with item $j$ will improve the solution.

So, we remove an infinitesimal $\delta > 0$ of weight of item $k$, and add it weight of item $j$. 
FEASIBILITY OF $Y'$

To show $Y'$ is feasible, we show $y'_j \geq 0, y'_j \leq 1$ and $\text{weight}(Y') \leq M$

Let's show $y'_j \geq 0$
- By definition, $y'_j = y_j - \frac{\delta}{w_j}$
- So, $y'_j \geq 0$ iff $y_j - \frac{\delta}{w_j} \geq 0$ iff $\delta \leq y_j w_j$
- And we know $y_j$ and $w_j$ are both positive
- So, this constrains $\delta$ to be smaller than this positive number
- Therefore, it is possible to choose positive $\delta$ s.t. $y'_j \geq 0$

Existence proof: but not constructive

FEASIBILITY OF $Y''$

To show $Y''$ is feasible, we show $y''_j \geq 0, y''_j \leq 1$ and $\text{weight}(Y'') \leq M$

Now let's show $y''_j \leq 1$
- By definition, $y''_j = y_j + \frac{\delta}{w_j}$
- So, $y''_j \leq 1$ iff $y_j + \frac{\delta}{w_j} \leq 1$ iff $\delta \leq (1 - y_j)w_j$
- Recall $y_j < 1$, so $y_j < 1$, which means $(1 - y_j) > 0$
- So, this constrains $\delta$ to be smaller than some positive number

SUPERIORITY OF $Y''$

Finally, we show $\text{weight}(Y'') \leq M$

- Recall changes to get $Y''$ from $Y$
  - We move $\delta$ weight from item $k$ to item $j$
  - This does not change the total weight!
  - So $\text{weight}(Y'') = \text{weight}(Y') \leq M$
  - Therefore, $Y''$ is feasible!

Contradicts optimality of $Y'$

To assume $\delta = 0$ is bad.

So, if $\delta > 0$ then $\delta \left( \frac{w_k}{p_k} - \frac{w_j}{p_j} \right) > 0$
- Since $\delta > 0$, we have $\text{profit}(Y') > \text{profit}(Y'$).
WHAT IF ELEMENTS DON’T HAVE DISTINCT PROFIT/WEIGHT RATIOS?

OPTIMALITY PROOF WITHOUT DISTINCTNESS

- There may be many optimal solutions
  
  **Key idea:** Let \( y \) be an optimal solution that matches \( X \) on a maximal number of indices

- Observe: if \( X \) is really optimal, then \( y = X \)

- Suppose not for contra
  
  We will modify \( y \), preserving its optimality, but making it match \( X \) on **one more index** (a contradiction!)

Covering the next 9 slides is homework!
To show $Y'$ is feasible, we show $\text{weight}(Y') \leq M$ and $y'_k \geq 0, y'_j \leq 1$.

**Feasibility of $Y'$**

- Showing $y'_k \geq 0$
  
  By definition, $y'_k = y_k - \frac{\delta}{w_k} \geq 0$ iff $\delta \leq y_k w_k$

- But $\delta$ is the minimum of $w_j (x_j - y_j)$ and $w_k (y_k - x_k) \leq w_k y_k$
  
  And $w_k (y_k - x_k) \leq w_k y_k$ so $\delta \leq y_k w_k$

- Showing $y'_j \leq 1$
  
  $y'_j = y_j + \frac{\delta}{w_j} \leq 1$ iff $\frac{\delta}{w_j} \leq 1 - y_j$ iff $\delta \leq w_j (1 - y_j)$ (rearranging)

- $\delta \leq w_j (x_j - y_j)$ (definition of $\delta$)

- And $w_j (x_j - y_j) \leq w_j (1 - y_j)$ (by feasibility of $X$, i.e., $x_j \leq 1$)

**Profit of $Y'$**

$\text{profit}(Y') = \text{profit}(Y) + \frac{\delta}{w_j} p_j - \frac{\delta}{w_k} p_k = \text{profit}(Y) + \delta \left( \frac{p_j}{w_j} - \frac{p_k}{w_k} \right)$

Since $j$ is before $k$, and we consider items with more profit per unit weight first, we have $\frac{p_j}{w_j} \geq \frac{p_k}{w_k}$.

Since $\delta > 0$ and $\frac{p_j}{w_j} \geq \frac{p_k}{w_k}$, we have $\delta \left( \frac{p_j}{w_j} - \frac{p_k}{w_k} \right) \geq 0$

Since $Y'$ is optimal, this cannot be positive

So $Y'$ is a new optimal solution that matches $X$ on one more index than $Y$.

Contradiction: $Y$ matched $X$ on a maximal number of indices.

**Summarizing Exchange Arguments**

- If inputs are distinct
  
  So there is a unique optimal solution

- Let $O \neq G$ be an optimal solution that beats greedy
  
  Show how to change $O$ to obtain a better solution

- If not
  
  There may be many optimal solutions

- Let $O \neq G$ be an optimal solution that matches greedy on as many choices as possible
  
  Show how to change $O$ to obtain an optimal solution $O'$ that matches greedy for even more choices.