CSIT113
Problem Solving

UNIT 5
GREEDY APPROACH

UNIVERSITY OF WOLLONGONG AUSTRALIA

SIM GLOBAL EDUCATION

Overview

- Introduction of Greedy Approach through illustrative example
- Summary on Using Greedy Approach to Solve Knapsack Problems
- The Strength and limitation of Greedy Approach
- Using Greedy Approach to Accept and Schedule Jobs

First Problem – Knapsack Problem

- Alabama Smith, the adventurous archaeologist, is exploring a hidden temple when he comes across a pile of treasure.
- He is able to carry at most 50Kg of treasure out of the temple.
- Sadly, because of the nearby volcano, which is on the verge of a violent eruption, the temple will soon be engulfed in lava.
- The treasure consists of 5 objects each with a different weight and value.

object	1	2	3	4	5
weight	5	10	15	20	25
value	2000	3000	6600	4000	6000

First Problem – Knapsack Problem

- Al is lucky to have a hacksaw with him so he can cut an object into two pieces if he has to.
- What is the most valuable load he can take from the temple?
- A brute force approach to this problem does not seem attractive.
- There are simply too many possible options to consider.
- By the time Al looks at all the options the lava will have covered the temple, the treasure and him!

Greedy Approach

- To solve this problem before the lava arrives he clearly needs a better strategy.
- The strategy he chooses, and which we are going to investigate, is called the greedy approach. Greedy approach can solve quite a lot of combinatorial problems efficiently.
- **Greedy approach** constructs a solution through a sequence of steps, each expanding a partially constructed solution so far until a complete solution to the problem is reached. On each step, the choice made must be:
  - ❑ feasible
  - ❑ locally optimal (best)
  - ❑ irrevocable

Greedy Approach

- Feasible means the choice must satisfy the problem's constraints.
- Locally optimal (best) means it has to be the **best** local choice among all feasible choices available on that step.
- Irrevocable means once it is made, it cannot be reversed back in the subsequent steps.
- Hold on though, what do we mean by "best"?
  - Most valuable?
  - Lightest?
  - Some other measure...?
- We will look at our options in turn.

Strategy 1: pick the most valuable object.

- Object 3 has the highest value so we put all of it in the pack.

object	1	2	3	4	5
weight	5	10	15	20	25
value	2000	3000	6600	4000	6000
pack			1.0		

- The pack now has:
  - Weight = 15Kg
  - Value = $6,600

Strategy 1: pick the most valuable object.

- Object 5 has the next highest value so we put all of it in the pack.

object	1	2	3	4	5
weight	5	10	15	20	25
value	2000	3000	6600	4000	6000
pack			1.0		1.0

- The pack now has:
  - Weight = 40Kg
  - Value = $12,600

Strategy 1: pick the most valuable object.

- Object 4 has the next highest value so we put half of it in the pack.

object	1	2	3	4	5
weight	5	10	15	20	25
value	2000	3000	6600	4000	6000
pack			1.0	0.5	1.0

- The pack now has:
  - Weight = 50Kg
  - Value = $14,600
- The pack is now full with a value of $14,600
- Can we do better?

Strategy 2: pick the lightest object.

- Object 1 is the lightest so we put all of it in the pack.

object	1	2	3	4	5
weight	5	10	15	20	25
value	2000	3000	6600	4000	6000
pack	1.0				

- The pack now has:
  - Weight = 5Kg
  - Value = $2,000

Strategy 2: pick the lightest object.

- Object 2 is the next lightest so we put all of it in the pack.

object	1	2	3	4	5
weight	5	10	15	20	25
value	2000	3000	6600	4000	6000
pack	1.0	1.0			

- The pack now has:
  - Weight = 15Kg
  - Value = $5,000

Strategy 2: pick the lightest object.

- Object 3 is the next lightest so we put all of it in the pack.

object	1	2	3	4	5
weight	5	10	15	20	25
value	2000	3000	6600	4000	6000
pack	1.0	1.0	1.0		

- The pack now has:
  - Weight = 30Kg
  - Value = $11,600

Strategy 2: pick the lightest object.

- Object 4 is the next lightest so we put all of it in the pack.

object	1	2	3	4	5
weight	5	10	15	20	25
value	2000	3000	6600	4000	6000
pack	1.0	1.0	1.0	1.0	

- The pack now has:
  - Weight = 50Kg
  - Value = $15,600
- Once again, the pack is full; this time with a value of $15,600
- This is better but is it the best we can do?

Strategy 3: pick the object with the greatest value per Kg.

- We add another row to our table value/Kg

object	1	2	3	4	5
weight	5	10	15	20	25
value	2000	3000	6600	4000	6000
value/Kg	400	300	440	200	240

Strategy 3: pick the object with the greatest value per Kg.

- Object 3 has the highest ratio so we put all of it in the pack.

object	1	2	3	4	5
weight	5	10	15	20	25
value	2000	3000	6600	4000	6000
value/Kg	400	300	440	200	240
pack			1.0		

- The pack now has:
  - Weight = 15Kg
  - Value = $6,600

Strategy 3: pick the object with the greatest value per Kg.

- Object 1 has the next highest ratio so we put all of it in the pack.

object	1	2	3	4	5
weight	5	10	15	20	25
value	2000	3000	6600	4000	6000
value/Kg	400	300	440	200	240
pack	1.0		1.0		

- The pack now has:
  - Weight = 20Kg
  - Value = $8,600

Strategy 3: pick the object with the greatest value per Kg.

- Object 2 has the next highest ratio so we put all of it in the pack.

object	1	2	3	4	5
weight	5	10	15	20	25
value	2000	3000	6600	4000	6000
value/Kg	400	300	440	200	240
pack	1.0	1.0	1.0		

- The pack now has:
  - Weight = 30Kg
  - Value = $11,600

Strategy 3: pick the object with the greatest value per Kg.

- Object 5 has the next highest ratio so we put 80% of it in the pack.

object	1	2	3	4	5
weight	5	10	15	20	25
value	2000	3000	6600	4000	6000
value/Kg	400	300	440	200	240
pack	1.0	1.0	1.0		0.8

- The pack now has:
  - Weight = 50Kg
  - Value = $16,400
- The pack is, once again, full; with an even higher value of $16,400
- This is the best possible outcome.

But really!

- In real life sawing up a gem-encrusted gold statue will result in a huge loss of value. So, Strategy 3 may not be appropriate. Strategy 3 can only be considered for finding:
  - the most valuable set of objects that will fit into a fixed capacity using either whole or partial objects -- this kind of problem is called, **continuous knapsack problem**.
- For the previous problem, we have to find:
  - the most valuable set of objects that will fit in the capacity using only whole objects -- this kind of problem is called, **discrete knapsack problem**.
- **Those options we have tried for the previous problem are for illustration purpose, the methods for finding optimal solutions for the two types of knapsacks are summarized in the next slide.**

Knapsack Problems: Important Summary

- Continuous knapsack: **Optimal solutions** can be found by applying Greedy Approach with highest value per weight best strategy. This can be proved (the proof is not within the scope of this course).
- Discrete knapsack: **Approximate optimal solutions** can be found by taking the best solution found from the following two strategies:
  - ➢highest value best
  - ➢highest value per weight best
- Therefore, discrete knapsack is harder problem which we will look at it later again using other method.

Second Problem: Paying the Bills

- We want to pay a bill with cash.
- We have banknotes with the following denominations:
  - \$100, \$50, \$20, \$10, \$5
- And coins:
  - \$2, \$1, 50c, 20c, 10c, 5c
- We want to use as few items of currency as possible.
- Can you devise a greedy strategy to achieve this?

Greedy payment

- Simply stated our greedy strategy is to use as many of each item as we can in descending order.
- Thus to pay a bill of \$379.45 we would use:

Three \$100 notes	Two \$2 coins
One \$50 note	No \$1 coins
One \$20 note	No 50c coins
No \$10 notes	Two 20c coins
One \$5 note	No 10c coins
	One 5c coin

- This is the best possible solution
- Is this always the case?

Zlotovian Currency

- The tiny country of Zlotavia bases its currency on the Zlyg.
- It has banknotes in the following denominations.
- Ż100, Ż30, Ż10, Ż7 and Ż1.
- Does the greedy strategy work in Zlotavia?
- Consider paying a bill of Ż15
  - The greedy strategy says use one Ż10 and five Ż1 notes, a total of six notes.
  - The optimal solution uses only three notes: two Ż7 notes and a single Ż1 note.
- Clearly we need to be careful in using greedy strategies, they do not always give the best answer.
- But, at least, they do give us an answer.

Third Problem: Egyptian Fractions

- Any rational number can be expressed as the sum of a series of fractions, each with a numerator of one.
- These are the so-called Egyptian fractions.
- For example: $3/4 = 1/2 + 1/4$
- Another example: $2\frac{3}{4} = 1/1 + 1/1 + 1/2 + 1/4$
- Can you apply Greedy Approach for finding Egyptian fractions with the least number of fractions in the sum?
- Can you find a greedy strategy for finding Egyptian fractions?
- How about we find the largest fraction of the form $1/n$ which is no bigger than the value we seek and repeat this process with whatever is left over?

Egyptian Fractions

- Let’s try with 7/15

Fraction to find	Best fraction	Amount still to find
7/15	1/3	7/15 – 1/3 = 2/15
2/15	1/8	2/15 – 1/8 = 1/120
1/120	1/120	1/120 – 1/120 = 0

- And we have our answer:
  - 7/15 = 1/3 + 1/8 + 1/120
- Once again, is this optimal?

Egyptian Fractions

- Let’s try with 5/121
- The greedy strategy gives us the answer:
  - 5/121 = 1/25 + 1/757 + 1/763309 + 1/873960180913+ 1/1527612795642093418864225
- The best answer, however, is:
  - 5/121 = 1/33 + 1/121 + 1/363
- There are even worse cases.
- Once again, the take home message is be careful when you decide to use a greedy strategy.

Don’t Panic!

- Despite the last two examples, there are lots of problems for which the greedy strategy works perfectly.
- Just not every problem.
- Let’s look at a more complex problem where a greedy approach does give the best solution.

Forth Problem: The Widget Factory

- A factory produces custom widgets; each one is different.
- Each widget takes one day to make.
- We have *n* customers.
- Each customer wants a specific widget.
- Each customer has a specific deadline.
  - If we don’t have the widget ready we lose the sale
- Each customer widget has a different profit.
- How can we maximize the total profit?

We are too good for our own good!

- Here's the problem
- At the start of the week we know each customer's order
  - The deadline
  - The profit
- The difficulty is that we have too many orders to make every widget on time.
- For each order we have to decide:
  - Do we make the widget?
  - If so, when do we make it?

A simple example

- We have only four customers for a two day period.
- Their properties are as follows:

Customer	1	2	3	4
Profit	50	10	15	30
Deadline	2	1	2	1

- We can construct a *schedule*, a sequence of jobs in daily order to tell the factory what to produce.
- Let's start with a brute force approach:

Lots of schedules!

Schedule	Profit	Schedule	Profit
1	50	2, 3	25
2	10	2, 4	BAD
3	15	3, 1	65
4	30	3, 2	BAD
1, 2	BAD	3, 4	BAD
1, 3	65	4, 1	80
1, 4	BAD	4, 2	BAD
2, 1	60	4, 3	45

- Some schedules are impossible; these are marked as BAD.
- The best schedule is 4, 1 with a total profit of 80

Greedy for Profit

- A possible greedy strategy works as follows:
  - Find the most profitable job.
  - Add it to the schedule as late as possible.
  - Repeatedly try to add the next most profitable job to the schedule, again as late as possible, until no more jobs are left.

A simple example

- Let us try this strategy with our simple example:

Customer	1	2	3	4
Profit	50	10	15	30
Deadline	2	1	2	1

- Job 1 is the most profitable. Schedule it for day 2.
- Job 4 is next and we can fit it in day 1. Schedule it.
- Job 3 is next but we have no room in the schedule. Skip it.
- Job 2 remains but we still have no room. Skip it
- This gives us the schedule 4, 1 which is optimal.

A bigger example

- This time we have ten orders over four days:

customer	1	2	3	4	5	6	7	8	9	10
profit	10	8	5	4	10	7	7	9	6	5
deadline	1	2	4	4	2	4	1	1	1	3

- The first step is to sort them by their profit:

customer	1	5	8	2	6	7	9	3	10	4
profit	10	10	9	8	7	7	6	5	5	4
deadline	1	2	1	2	4	1	1	4	3	4

- We can now try to schedule each job in order.

A bigger example

customer	1	5	8	2	6	7	9	3	10	4
profit	10	10	9	8	7	7	6	5	5	4
deadline	1	2	1	2	4	1	1	4	3	4

- Choose job 1
- We can fit it in our schedule

Day	1	2	3	4
Job	1			
Profit	10			

A bigger example

customer	1	5	8	2	6	7	9	3	10	4
profit	10	10	9	8	7	7	6	5	5	4
deadline	1	2	1	2	4	1	1	4	3	4

- Next choose job 5
- We can fit it in our schedule

Day	1	2	3	4
Job	1	5		
Profit	10	10		

A bigger example

customer	1	5	8	2	6	7	9	3	10	4
profit	10	10	9	8	7	7	6	5	5	4
deadline	1	2	1	2	4	1	1	4	3	4

- Next consider job 8
- We can't fit it in our schedule

Day	1	2	3	4
Job	1	5		
Profit	10	10		

A bigger example

customer	1	5	8	2	6	7	9	3	10	4
profit	10	10	9	8	7	7	6	5	5	4
deadline	1	2	1	2	4	1	1	4	3	4

- Next consider job 2
- We can't fit it in our schedule

Day	1	2	3	4
Job	1	5		
Profit	10	10		

A bigger example

customer	1	5	8	2	6	7	9	3	10	4
profit	10	10	9	8	7	7	6	5	5	4
deadline	1	2	1	2	4	1	1	4	3	4

- Next choose job 6
- We can fit it in our schedule

Day	1	2	3	4
Job	1	5		6
Profit	10	10		7

A bigger example

customer	1	5	8	2	6	7	9	3	10	4
profit	10	10	9	8	7	7	6	5	5	4
deadline	1	2	1	2	4	1	1	4	3	4

- Next consider job 7
- We can't fit it in our schedule

Day	1	2	3	4
Job	1	5		6
Profit	10	10		7

A bigger example

customer	1	5	8	2	6	7	9	3	10	4
profit	10	10	9	8	7	7	6	5	5	4
deadline	1	2	1	2	4	1	1	4	3	4

- Next consider job 9
- We can't fit it in our schedule

Day	1	2	3	4
Job	1	5		6
Profit	10	10		7

A bigger example

customer	1	5	8	2	6	7	9	3	10	4
profit	10	10	9	8	7	7	6	5	5	4
deadline	1	2	1	2	4	1	1	4	3	4

- Next choose job 3
- We can fit it in our schedule

Day	1	2	3	4
Job	1	5	3	6
Profit	10	10	5	7

A bigger example

- The schedule is now full so we do not need to go any further.

Day	1	2	3	4
Job	1	5	3	6
Profit	10	10	5	7

- The final schedule is 1, 5, 3, 6 for a total profit of 32.
- Can we show that applying Greedy Approach with this strategy will always find the best solution (maximum profit)?
- Yes! But we aren't going to ☺
- Some of you will see the proof of this in CSCI203 next year.

Wait!

- Did anyone see the magic in what we just did?
  - How did I get from:

customer	1	2	3	4	5	6	7	8	9	10
profit	10	8	5	4	10	7	7	9	6	5
deadline	1	2	4	4	2	4	1	1	1	3

to:

customer	1	5	8	2	6	7	9	3	10	4
profit	10	10	9	8	7	7	6	5	5	4
deadline	1	2	1	2	4	1	1	4	3	4

- Sorting looks pretty magical!
- We will look at sorting later in this subject.