Optimization Problem Set

John has many activities to do during the next 5 hours (300 minutes).

The following are a list of his activities and what time do they start and finish. Time 0 corresponds to now and time 300 correspond to 5 hours from now (300 minutes).

Activity Number	Start Time	End Time
1	0	100
2	50	150
3	180	220
4	210	260
5	120	190
6	190	270
7	140	190
8	150	180
9	200	210
10	80	170

John can only do at most one activity at a time. He wants to do as many activities as possible.

(i) Give an example of a set of activities that are a **feasible solution** to John's problem.

(ii) What is the corresponding **objective value** of that solution (i.e. how many activities did you pick)?

(iii) What makes a set of activities **feasible** or not?

(iv) What is the largest set of activities that you can come up with that are feasible?

(v) What is the logic you used?

Jessica is trying to serve her many customers. Each customer requires a certain amount of minutes of her time. Once she starts serving a customer, she must not interrupt it.
The following are a list of the times required for serving each customer

Customer Number	Time Required
1	30
2	10
3	20
4	15
5	30
6	100
7	50
8	90
9	10
10	60

Jessica must serve all customers. Her goal is to minimize the average waiting time of her customers. Customers wait from time zero (now) until Jessica is finished serving them.

(i) What would describe a **feasible solution** to Jessica's problem? Give an example of a feasible solution.

(ii) What is the corresponding **objective value** of that solution?

(iii) What is the best solution that you can come up with?

(iv) What is the logic you used?

Exercises

(i) Find the optimal solution to John's problem with the following activities.

Activity Number	Start Time	End Time
1	20	120
2	10	50
3	80	290
4	210	250
5	150	190
6	190	280
7	240	300
8	0	80
9	30	130
10	90	170

Solution:

We pick the activity with smallest end time: **Activity 2**.

The remaining activities that do not conflict with Activity 2 are:

Activity Number	Start Time	End Time
3	80	290
4	210	250
5	150	190
6	190	280
7	240	300
10	90	170

Of those remaining, the activity with smallest end time is: **Activity 10**

The remaining activities that do not conflict with 10 are:

Activity Number	Start Time	End Time
4	210	250
6	190	280
7	240	300

Of those remaining, the activity with smallest end time is: **Activity 4**

There are no remaining activities that do not conflict with Activity 4.

Thus, our optimal solution to John's problem is doing activities 2, 10 and 4.

The optimal value is 3.

(ii) Find the optimal solution to Jessica's problem with the following data.

Customer Number	Time Required
1	10
2	50
3	5
4	65
5	90
6	10
7	45
8	75
9	35
10	80

Solution

Jessica must serve the customers in order of lowest to highest time required. So the optimal solution is to serve customers as follows:

Customer Number	Start Time	Finish Time
3	0	5
1	5	15
6	15	25
9	25	60
7	60	105
2	105	155
4	155	220
8	220	295
10	295	375
5	375	465

This will get the average wait time (optimal value) to be 172.

(iii) Consider the following other rules for John's problem:

- Choose as the last activity the one with latest finishing time, remove all incompatible activities and repeat.
- Choose as the last activity the one with latest starting time, remove all incompatible activities and repeat.

Will any of these rules always produce an optimal solution? Why or why not?

Solution

The first rule will not.

Consider the following activities:

Activity Number	Start Time	End Time
1	0	50
2	0	25
3	25	49

The first rule will choose only activity 1. The optimal is clearly picking activities 2 and 3.

Now consider the second rule. This is essentially equivalent to the rule of picking as the first activity the one with the earliest finishing time (Why? Try to think about it.)

Thus, it also produces an optimal solution always.

(iv) **Challenge** Consider a variation of Jessica's problem where customers have a due time by which they want their activity completed

Customer Number	Time Required	Due Time
1	30	50
2	10	30
3	20	100
4	15	80
5	30	200
6	100	300
7	50	90
8	90	400
9	10	300
10	60	500

The *lateness* of a customer is the difference between the time their activity finished and their due date. For instance, if customer 1 finished at time 40, then its lateness is -10. If customer 1 finished at time 70, its lateness is 20.

Develop an algorithm to solve the problem of:

- Minimizing the sum of the lateness of all customers

  Some thoughts

  - Come up with several different solutions and compare the objective value of Jessica's original problem with this new one
  - Try out with a small (say 4 or 5) customers first to get a better feel
  - If you think you have a good algorithm, can you prove it always gives the optimal solution?

Solution

Let us look at what happens with only 4 customers.

Say, for instance Jessica served customers in order 1, 2, 3, 4. This is how they would be served:

Customer Number	Start Time	Finish Time	Lateness
1	0	30	-20
2	30	40	10
3	40	60	-40
4	60	75	-5

Value for original problem:

$$\frac{30+40+60+75}{4} = \frac{205}{4} = 51.25$$

Value for lateness problem:

$$-20+10-40-5 = -55$$

Say, for instance Jessica served customers in order 2, 1, 3, 4. This is how they would be served:

Customer Number	Start Time	Finish Time	Lateness
2	0	10	-20
1	10	40	-10
3	40	60	-40
4	60	75	-5

Value for original problem:

$$\frac{10+40+60+75}{4} = \frac{185}{4} = 46.25$$

Value for lateness problem:

$$-20-10-40-5=-75$$

Say, for instance Jessica served customers in order 3, 2, 4, 1. This is how they would be served:

Customer Number	Start Time	Finish Time	Lateness
3	0	20	-80
2	20	30	0
4	30	45	-35
1	45	75	25

Value for original problem:

$$\frac{20+30+45+75}{4} = \frac{170}{4} = 42.5$$

Value for lateness problem:

$$-80+0-35+25=-90$$

Have a close look at the numerators of the fractions for the original problem's value and the values for the lateness problem. Notice that the difference is always constant equal to 260.

Note, in addition, that the sum of the due times of customers 1 through for is **EXACTLY** 260!

This is not a coincidence. Note that lateness of a customer equals completion time minus due time.

So the sums of the lateness equals the sum of the completion times minus the sum of the due times. For instance, for serving customers in order 3,2,4,1, we get the sum of the lateness by calculating lateness of each customer and adding it up as:

$$(20-100)+(30-30)+(45-80)+(75-50)$$

which is equivalent to doing

$$(20+30+45+75)-(100+30+80+50)$$

Since the same logic applies to any sequence, we note that the second part will always be equal to -260, so all we need to do is minimize the first part, which is minimizing the sum of the finish times, which is in fact the same as minimizing the average wait time.

- Minimizing the largest lateness out of all customers

  Some thoughts

  - Come up with several different solutions and compute their objective values
  - Try out with a small (say 4 or 5) customers first to get a better feel
  - If you have a solution (for instance customers served in order 1,2,...), how would you try to improve its objective value?
  - If you think you have a good algorithm, can you prove it always gives the optimal solution?

Solution

An algorithm that guarantees to give an optimal solution is to serve customers with lowest due date first.

The proof that it always works is a little tricky (and was probably much to expect to be proved), but if you could come up with a reasonable algorithm and think about the problem, that was the main goal.

In any case, I will give a sketch of an argument by an example.

Suppose that the customers are ordered so their due dates are from smallest to largest.

Pick a solution that executes jobs $1, \ldots, j-1$ in order, for some j, then executes job $k > j$ instead of j.

For instance, if a solution executes jobs in order $(1, 2, 3, 4, 10, 7, 6, 8, 5, 9)$ then $j = 5$, $k = 10$.

For instance, if a solution executes jobs in order $(5, 2, 3, 4, 10, 7, 6, 8, 1, 9)$ then $j = 1$, $k = 5$.

Consider the solution we would get by altering the current one by executing job k after job j.

For instance, if a solution executes jobs in order $(1, 2, 3, 4, 10, 7, 6, 8, 5, 9)$ then the new solution would be $(1, 2, 3, 4, 7, 6, 8, 5, 10, 9)$

For instance, if a solution executes jobs in order $(5, 2, 3, 4, 10, 7, 6, 8, 1, 9)$ then the new solution would be $(2, 3, 4, 10, 7, 6, 8, 1, 5, 9)$

Note that the completion times of all jobs between k and j decrease (thus their lateness decreases). Also, the completion times (and thus lateness) of any jobs before k and after j does not change.

Thus, the only lateness that may increase in the new solution is the lateness of job k.

Now note that the completion time of job k in the new solution is exactly the same as the completion time of job j in the old solution (Think about why).

However, the due date of j is $\leq$ than the due date of job k. So the lateness of job k in the new solution cannot be more than the lateness of job j in the old solution.

Thus, the maximum lateness will not increase if we do this.