Design and Analysis of Algorithms

L32: Reliaility Design
Dynamic Programming

Dr. Ram P Rustagi
Sem IV (2020-Even)
Dept of CSE, KSIT
rprustagi@ksit.edu.in

Resources

- Text book 2: Horowitz
  - Sec <u>**5 . 8**</u>

Example Problem

- Example: consider you need to complete 4 number of assignments successfully to pass the course.
- Each assignment can be attempted any number of times.
- Probability of successful attempt at each assignment
  $P(a_1)=0.8, P(a_2)=0.9, P(a_3)=0.85, P(a_4)=0.75$
- Time (hrs) taken per attempt for each assignment
  $T(a_1)=3h, T(a_2)=5h, T(a_3)=4h, P(a_4)=2h$
- Total time (hrs) available to you for all 4 assignments
  - 20 hours
- Problem: Define the number of attempts for each assignment so as to increase the pass probability
- Pass probability if each assignment is done only once
  $P(a_1)*P(a_2)*P(a_3)*P(a_4)=0.8*0.9*0.85*0.75=0.459$
- Max possible attempts for each assignment
  $u_1=3,\ u_2=2,\ u_3=2,\ u_4=6$

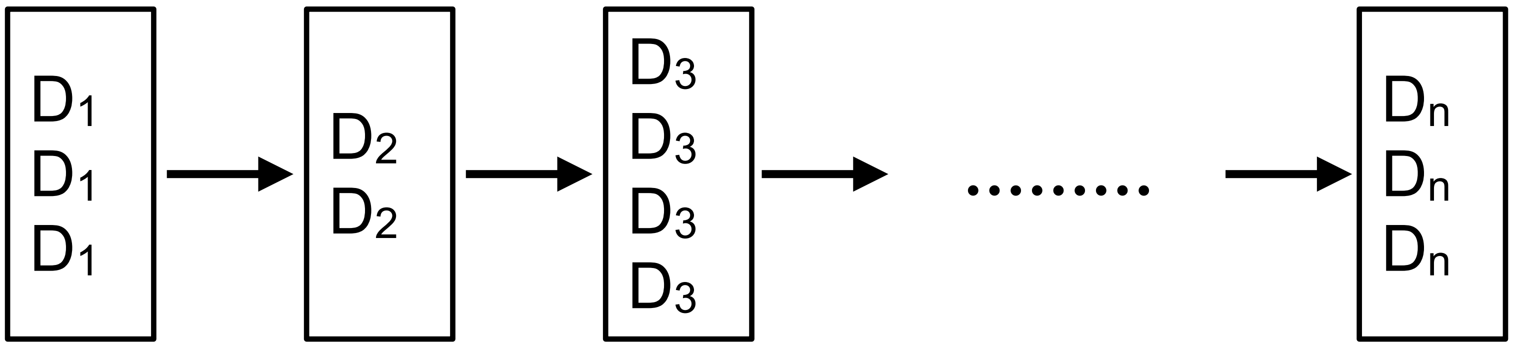
Example Problem

- Consider the number of attempts for each assignment are represented by n_1, n_2, n_3, n_4.
- The probability of success of i^{th} assignment is
  $1-(1-p_i)^{n_i}$
- Ex: Prob of success with 2 attempts for assignment 1
  =1- P(failure at both attempts)
  $=1-(1-P(a_1))*(1-P(a_1))=1-0.2*0.2=0.96$
- The probability of successfullly completing all assignments
  $\Pi_{1\le i\le 4}(1-(1-p_i)^{n_i})$
- Goal:
  – Maximize $\Pi_{1\le i\le 4}(1-(1-p_i)^{n_i})$
  – Subject to $\Sigma_{1\le i\le 4}\ t_i*n_i \le c$,
    – where c (e.g. =20) is max time available, and
    – t_i time taken per attempt for i^{th} assignment

Reliability Design

- Application: Problem with multiplicative optimization function.
- Problem: Design a system that is composed of n devices connected in series
  - Let r_i be the reliability of device D_i.
    - r_i is probability(reliability) that D_i will function properly.
  - The reliability of entire system is Πr_i
  - When n is large (e.g. 20),
    - Even if r_i is high e.g. 0.95,
    - The reliability of system is $(0.95)^{20}=0.358$
  - Thus, it is desirable to duplicate the devices
    - Multiple copies of same device parallelly connected
    - So as to increase overall reliability of the system.

Multiple Devices in Parallel



- If device D_i with a reliability probability of r_i,
  - Has m_i copies connected in parallel, then
  - Probability that all of m_i devices will malfunction
    $(1-r_i)^{m_i}$
- Thus, reliability of machines at stage i is $1-(1-r_i)^{m_i}$
- Example: $r_i=0.95$, $m_i=2$, then reliability is 0.9975
- Assume that reliability at stage i is given by $\emptyset_i(m_i)$
  - It may also depend upon switching circuit as well

Reliability Design Problem

- Problem:
  - Use device duplication to maximize reliability
  - Under the constraint of total cost.
- Let $c_i > 0$ be the cost of i^{th} device.
- Let c be the max cost allowed for the system.
- Thus, the problem is mathematically defined as

  maximize $\Pi_{1 \leq i \leq n} \varnothing_i (m_i)$

  subject to $\Sigma_{1 \leq i \leq n} c_i m_i \leq c$ .... (1)

  and $1 \leq m_i \leq u_i$, m_i is intger, and $1 \leq i \leq n$
- Thus, similar to knapsack problem, we can apply dynamic programming technique to solve reliability design problem

Reliability Design Problem: DP Approach

- Since each $c_i > 0$, and $m_i > 0$, then
  - Let u_i denotes the max number of i^{th} device
  - Each device has to be used once.

    $\Sigma_{1 \le j \le n} c_j$ is cost of each device using once

    $c - \Sigma_{1 \le j \le n} c_j$ is remaining cost after using each device once
  - The max value u_i for i^{th} device would be

  $u_i = \lfloor (c - \Sigma_{1 \le j \le n} c_j + c_i) / c_i \rfloor = (c - \Sigma_{1 \le j \ne i \le n} c_j) / c_i$
- An optimal solution $m_1, m_2, \ldots, m_n$ is the result of sequence of decisions.
- Let $f_i(x)$ represents the max value of $\Pi_{1 \le j \le i} \varnothing_j(m_j)$ subject to the contraints

  $\Sigma_{1 \le j \le i} c_j m_j \le x$, and $1 \le m_j \le u_j,\ 1 \le j \le i$.
- The optimal solution then is $f_n(c)$

Reliability Design Problem: DP Approach

- The last decision for n^{th} device requires m_n to be chosen from $\{1, 2, 3, \ldots, u_n\}$.
- After the value m_n is chosen,
  - Remaining decisions must be made w.r.t. $c - c_n m_n$.
  - The principle of optimality should be used.
- The recurrence relation becomes

$$f_n(c) = \max_{1 \leq m_n \leq u_n} \left\{ \phi_n(m_n) f_{n-1}(c - c_n m_n) \right\} \ldots\ldots(2)$$

- For any $f_i(x), i \geq 1$, the generalized equation is

$$f_i(x) = \max_{1 \leq m_i \leq u_i} \left\{ \phi_i(m_i) f_{i-1}(x - c_i m_i) \right\} \ldots\ldots(3)$$

Example: Reliability Design

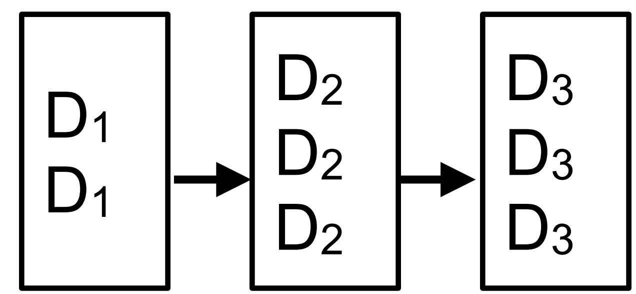
- Consider 3 devices with their costs and reliabilities as
  - $c_1=30, c_2=15, c_3=20, \ r_1=0.9, r_2=0.8, r_3=0.5$
- The max system cost is $c=105$
- Computation for designing the system:

  $\Sigma c_i=30+15+20=65$

  $u_1=(105-65+30)/30=70/30=2$

  $u_2=(105-65+15)/15=55/15=3$

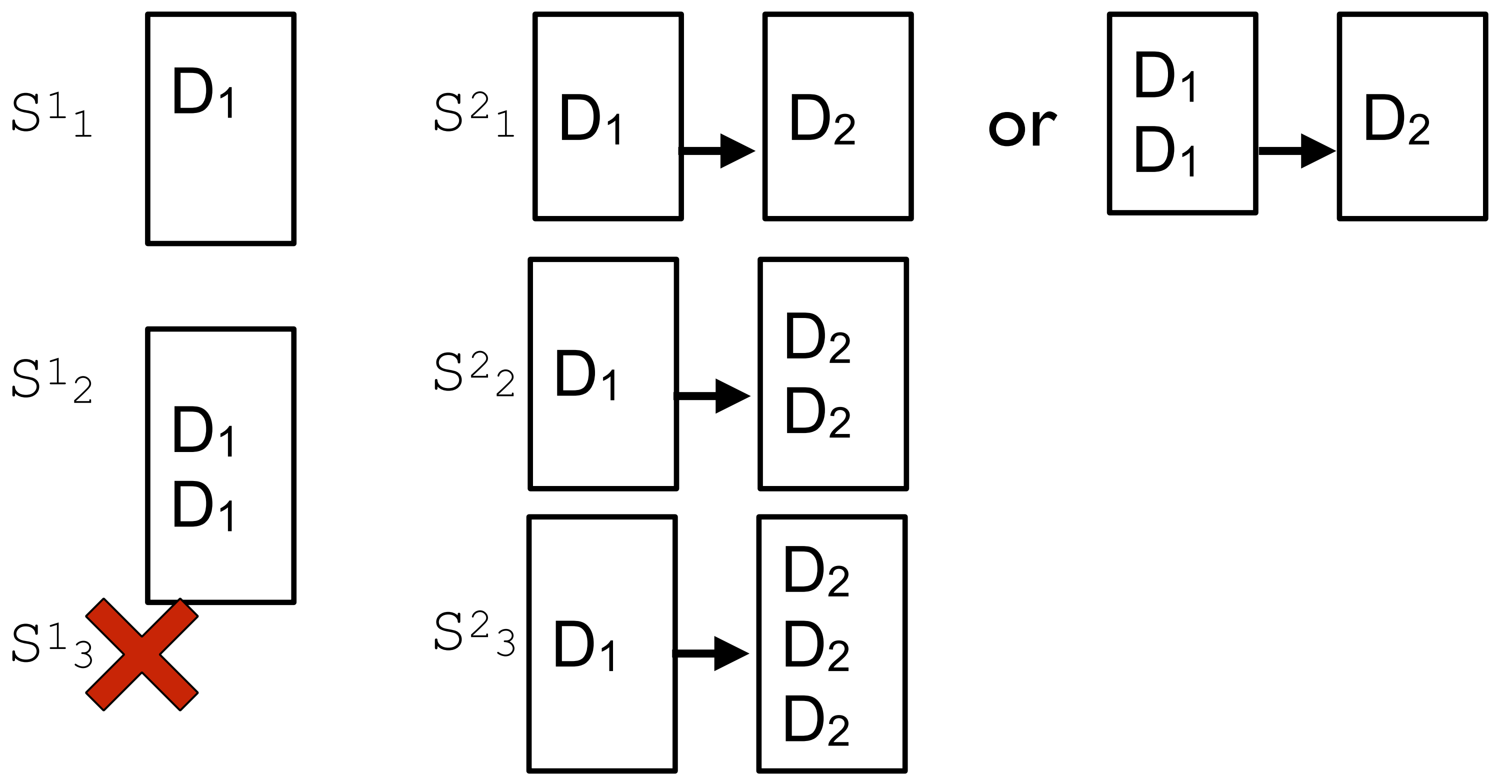
  $u_3=(105-65+20)/20=60/20=3$



- Consider the decision sequence m_1, m_2 and m_3.
- Starting from tuple $S^0=\{(1,0)\}$,
  - Compute S^i from S^{i-1} by trying out all possible values for m_i and combining the results.

Example: Reliability Design

- Let $S^i{}_j$ represent all tuples obtainable from S^{i-1} by choosing $m_i=j$.
  - $S^i{}_1 \Rightarrow D_i$ is used once, $S^i{}_2 \Rightarrow D_i$ is used 2 times, …
    - Devices $D_1, D_2, \ldots, D_{i-1}$ are to be used as applicable
- Example, $C=105$; $c_1=30, c_2=15, c_3=20$



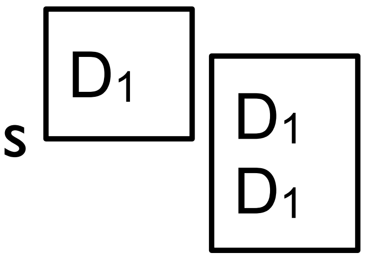
Example: Reliability Design

- Example, $C=105$; $c_1=30, c_2=15, c_3=20$
  - Devices $D_1, D_2, \ldots, D_{i-1}$ are to be used as applicable
- For device D_1, $u_1=2$, possible values for m_1 are $1, 2$
- For device D_2, $u_2=3$, possible values for m_2 are $1, 2, 3$
- For device D_3, $u_3=3$, possible values for m_3 are $1, 2, 3$

$S^1_1=\{(0.9, 30)\}$ #D_i is used once

$S^1_2=\{(1-(1-0.1)^2, 30*2)\}$ #D_i is used 2 times

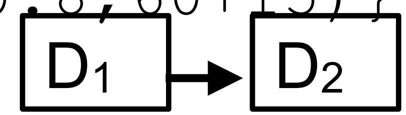
$=\{(0.99, 60)\}$

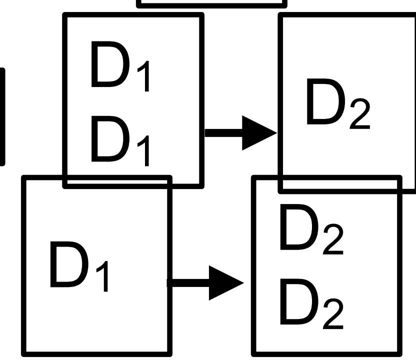


$m_1=1, m_2=1$ $m_1=2, m_2=1$

$S^2_1=\{(0.9*0.8, 30+15), (0.99*0.8, 60+15)\}$

$=\{(0.72, 45), (0.792, 75)\}$





$S^2_2=\{(0.9*(1-(1-0.2)^2, 30+15*2)\}$ $m_1=1, m_2=2$

$=\{(0.9*0.96, 30+30)\} = \{(0.864, 60)\}$

$m_1=2, m_2=2$ infeasible

Reliability Design Problem: DP Approach

- Initial value (when no device is used, reliability is 1)

  $f_0(x)=1 \ \forall x, \ 0 \le x \le c.$
- Let S^i consists of tuples of the form (f, x), where

  $f=f_i(x)$

  $S^i= \{S^i{}_j : 1 \le m_j \le u_j\}$
- There is at most one tuple for each different x,
  - Results from a sequence of decisions $m_1, \ldots, m_n$.
- The dominance rule is
  - (f_1, x_1) dominates (f_2, x_2) iff $f_1 \ge f_2$ and $x_1 \le x_2$.
  - Keep the dominant tuple (f_1, x_1) and
  - Discard the dominated tuple (f_2, x_2) from S^i.
  - Because dominant tuple provides higher reliability at lower cost

Example: Reliability Design

- Continuing

  $m_1=1, m_2=3$

  $S^2_3=\{(0.9*(1-(1-0.2)^3, 30+15*3)$

  $=\{(0.9*0.992, 30+45)\}$

  $=\{(0.8928, 75)\}$

  $m_1=2, m_2=3$ infeasible

  The tuple value $(0.99*0.992, 60+45)=(0.98208, 105)$ is eliminated as left with cost of 0, which is not enough for D_3

- Combining S^2_1, S^2_2, and S^2_3, we get

  $S^2_1=\{(0.72, 45), \cancel{(0.792, 75)\}}$

  $S^2_2=\{(0.864, 60)\}$

  $S^2_3=\{(0.8928, 75)\}$

  $S^2=\{(0.72, 45), (0.864, 60), (0.8928, 75)\}$

  The tuple value $(0.792, 75)$ is eliminated as it is dominated by $(0.864, 60)$ using dominance rule

  $0.864 \geq 0.792$, and $60 \leq 75$

Example: Reliability Design

- Continuing S^3_1={(0.9*0.8*0.5,30+15+20), **m_1=1,m_2=1,m_3=1**
  (0.99*0.8*0.5,30*2+15+20) **m_1=2,m_2=1,m_3=1**
  (0.9*0.96*0.5,30+15*2+20), **m_1=1,m_2=2,m_3=1**
  (0.9*0.992*0.5,30+15*3+20)} **m_1=1,m_2=3,m_3=1**
  ={(0.36,65),~~(0.396,95)~~,{0.432,80),~~(0.4464,95)~~}
  S^3_2={(0.9*0.8*0.75,30+15+20*2), **m_1=1,m_2=1,m_3=2**
  (0.9*0.96*0.75,30+15*2+20*2)} **m_1=1,m_2=2,m_3=2**
  ={(0.54,85),(0.648,100)}
  S^3_3={(0.9*0.8*0.875,30+15+20*3)} **m_1=1,m_2=1,m_3=3**
  ={~~(0.63,105)~~}
- Combining S^3_1, S^3_2, and S^3_3, we get
  S^3={(0.36,65),(0.432,80),(0.54,85),(0.648,100)}
  Note: Other values are dominated.
- The best design is (0.648,100) i.e. m_1=1,m_2=2,m_3=2

Summary

- Understanding reliability
- Reliability in stages
- Overall summary of DP
  – Principle of optimality
  – Multi-stage graphs
  – Transitive closure: Warshall's algorithm
  – All pair shortest path: Floyd's algorithm
  – Optimal binary search trees
  – Knapsack problem
  – Bellman-Ford algorithm
  – Traveling Sales Person problem
  – Reliability design