Design and Analysis of Algorithms

L38: Reliaility Design
Dynamic Programming

Dr. Ram P Rustagi
Sem IV (2019-H1)
Dept of CSE, KSIT/KSSEM
rprustagi@ksit.edu.in

Resources

- Text book 2: Horowitz
  - Sec `5.1,5.2,5.4,`**5.8**`,5.9`
- Text book 1: Levitin
  - Sec `8.2-8.4`
- R1: Introduction to Algorithms
    - Cormen et al.

Example Problem

- Example: consider you need to complete 4 number of assignments successfully to pass the course.
- Each assignment can be attempted multiple 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

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

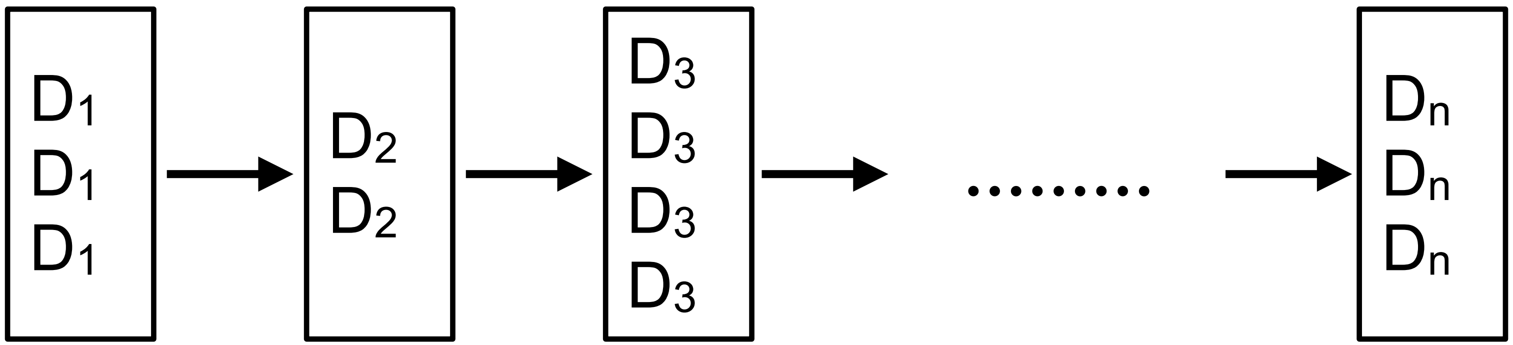
  $1-(1-p_i)^{n_i}$
- The probability of successfullly completing assignment

  $\Pi_{1\leq i\leq 4}(1-(1-p_i)^{n_i})$
- Goal:
  - Maximize $\Pi_{1\leq i\leq 4}(1-(1-p_i)^{n_i})$
  - Subject to $\Sigma_{1\leq i\leq 4}\ t_i n_i \leq c$,
    - where c (e.g. =20) is max time available, and
    - t_i time taken per attempt by 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 that D_i will function properly.
  - The reliability of entire system is Πr_i
  - When n is large (e.g. 10),
    - even though r_i is high e.g. 0.9,
    - the reliability of system is $(0.9)^{10}=0.348$
  - 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: $r_i=0.99$, $m_i=2$, then reliability is 0.9999
- 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 $\Pi_{1 \le i \le n} \emptyset_i (m_i)$

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

  and to $m_i \ge 1$ and integer, $1 \le i \le n$
- 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.
  - The max value u_i for i^{th} device would be

$$u_i = (c - \Sigma_{1 \le j \ne i \le n} c_j) / c_i = \lfloor (c + c_i - \Sigma_{1 \le j \le n} c_j) / c_i \rfloor$$

- 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 $\Pi_{1 \le i \le n} \emptyset_i(m_i)$ subject to the contraints

  $\Sigma_{1 \le j \le n} c_j m_j \le x$, 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

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

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

  $f=f_i(x)$
- There is at most one tuple for each different x,
  - that 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 $x_1 \le x_2$.
- The dominated tuples can be discarded from S^i.

Example: Reliability Design

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

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

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

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

  $u_3=(105+20-65)/20=60/20=3$
- 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 $m_i=j$. Thus
- For device D_1, $u_1=2$, possible values for m_1 are $1,2$

$S^1_1=\{(0.9,30)\}$

$S^1_2=\{(0.9,30),(1-(1-0.1)^2,30*2)\}=$

$=\{(0.9,30), (0.99,60)\}$

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

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

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

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

$=\{(0.864,60)\}$

The tuple value $(0.99*0.96,60+30)=(0.9504.90)$ is eliminated as left with cost of 15, which is not enough for D_3

Example: Reliability Design

- Continuing

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

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

  ={(0.8928,75)}

  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), (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),
(0.9*0.96*0.5,30+15*2+20),
(0.9*0.992*0.5,30+15*3+20)}
={(0.36,65),{0.432,80),(0.4464,95)}

S^3_2={(0.9*0.8*0.75,30+15+20*2),
(0.9*0.96*0.75,30+15*2+20*2)}
={(0.54,85),(0.648,100)}

S^3_3={(0.9*0.8*0.875,30+15+20*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