A Robust Approach for Project Scheduling Problem

Xin Shen, Jubiao Yang
advised by: John E. Mitchell

Rensselaer Polytechnic Institute
Troy, NY 12180

7th AIMMS-MOPTA Optimization Modeling Competition
Lehigh University, Bethlehem, PA, 2015

Outline

Introduction

Objective: to maximize the net present value of the project portfolio (sum of benefits and costs of portfolio projects discounted appropriately with hurdle rate).

Projects may have *dependencies*:

- nonsimultaneity (e.g. resource constraints on teams/equipments)
- single precedence (e.g. a project is decomposed into phases)
- alternative precedence (e.g. parallel-approach effort to overcome technical hurdles)

Introduction

Projects are subject to risks of bad luck (delay/failure/delay and failure).

Deterministic Approach: prepare for a certain bad-luck scenario beforehand (incl. scenario with no bad luck) and schedule a portfolio.

disproportionate depreciation of portfolio value can be caused by "chain reaction" of bad lucks, thanks to project dependencies.

Robust Approach: to have the largest portfolio value under the worst possible outcome scenario (resilience to bad luck).

Outline

Project dependencies

nonsimultaneity: if $i \nsim j$, then $\Delta_{ij} = \Delta_{ji} = -1$

alternative precedence: if $\{i_1, \cdots, i_N\} \vdash j$, then $\Delta_{i_1 j} = \cdots = \Delta_{i_N j} =$ a unique positive integer

single precedence: $i \succ j \quad \Leftrightarrow \quad \{i\} \vdash j$, thus a special case of $I \vdash j$ and can be treated the same way

e.g. for a project pool of $\{p_1, p_2, p_3, p_4, p_5\}$ with $p_1 \nsim p_2$, $p_2 \nsim p_3$, $p_3 \succ p_1$, $p_1 \succ p_4$, $\{p_2, p_5\} \vdash p_4$, $p_3 \succ p_4$, $\{p_1, p_3\} \vdash p_5$:

$$\mathbf{\Delta} = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{array} \begin{array}{c} \begin{array}{ccccc} p_1 & p_2 & p_3 & p_4 & p_5 \end{array} \\ \begin{pmatrix} & -1 & & 1 & 1 \\ -1 & & -1 & 2 & \\ 1 & -1 & & 3 & 1 \\ & & & & \\ & & & 2 & \end{pmatrix} \end{array}$$

Model

Binary variable X_{jt}:

$$X_{jt} = \begin{cases} 1, & \text{if Project } j \text{ starts at the beginning of the } i^{th} \text{ month} \\ 0, & \text{otherwise} \end{cases}$$

User-controlled parameters q_j^δ and q_j^f:

$$q_j^\delta = \begin{cases} 1, & \text{if knew beforehand that Project } j \text{ would be delayed} \\ 0, & \text{otherwise} \end{cases}$$
$$q_j^f = \begin{cases} 1, & \text{if knew beforehand that Project } j \text{ would fail} \\ 0, & \text{otherwise} \end{cases}$$

Thus the adjusted durations and costs are:

$$\tilde{d}_j = d_j + q_j^\delta d_j^+,\ \tilde{c}_j = c_j + q_j^\delta c_j^+,\ \forall j \in J$$

Formulation

- a project can start at most once:

$$\sum_{t=1}^{T} X_{jt} \leq 1 - q_j^f, \ \forall j \in J$$

- a project cannot start if it cannot complete by the deadline:

$$\sum_{t \geq T+1-\tilde{d}_j} X_{jt} = 0, \ \forall j \in J$$

- for $i \nsim j$, i cannot be started within d_j months after j started, vice versa:

$$\sum_{t-\tilde{d}_j+1 \leq t' \leq t+\tilde{d}_i-1} X_{jt'} + X_{it} \leq 1, \ \forall i \nsim j, \ \forall t \in \{1, \cdots, T\}$$

Formulation

- a project can start at most once:

$$\sum_{t=1}^{T} X_{jt} \leq 1 - q_i^f, \ \forall j \in J$$

- a project cannot start if it cannot complete by the deadline:

$$\sum_{t \geq T+1-\tilde{d}_j} X_{jt} = 0, \ \forall j \in J$$

- for $i \nsim j$, i cannot be started within d_j months after j started, vice versa:

$$\sum_{t-\tilde{d}_j+1 \leq t' \leq t+\tilde{d}_i-1} X_{jt'} + X_{it} \leq 1, \ \forall i \nsim j, \ \forall t \in \{1, \cdots, T\}$$

Formulation

- a project can start at most once:

$$\sum_{t=1}^{T} X_{jt} \leq 1 - q_i^f, \ \forall j \in J$$

- a project cannot start if it cannot complete by the deadline:

$$\sum_{t \geq T+1-\tilde{d}_j} X_{jt} = 0, \ \forall j \in J$$

- for $i \nsim j$, i cannot be started within d_j months after j started, vice versa:

$$\sum_{t-\tilde{d}_j+1 \leq t' \leq t+\tilde{d}_i-1} X_{jt'} + X_{it} \leq 1, \ \forall i \nsim j, \ \forall t \in \{1, \cdots, T\}$$

Formulation

- for $I \vdash j$, j cannot be started until at least one of the projects in I has been finished:

$$\sum_{i \in I} \sum_{t' \leq t - \tilde{d}_i} X_{it'} \geq X_{jt}, \ \forall I \vdash j, \ \forall t \in \{1, \cdots, T\}$$

- the objective function can be evaluated:

$$NPV_{\gamma}(S, T) = -\sum_{j,t} \gamma^{t} \cdot \tilde{c}_j \cdot X_{jt} + \sum_{j,t} \gamma^{t+\tilde{d}_j} \cdot b_j \cdot X_{jt}$$

Formulation

- for $I \vdash j$, j cannot be started until at least one of the projects in I has been finished:

$$\sum_{i \in I} \sum_{t' \leq t - \tilde{d}_i} X_{it'} \geq X_{jt}, \ \forall I \vdash j, \ \forall t \in \{1, \cdots, T\}$$

- the objective function can be evaluated:

$$NPV_{\gamma}(S, T) = -\sum_{j,t} \gamma^{t} \cdot \tilde{c}_j \cdot X_{jt} + \sum_{j,t} \gamma^{t+\tilde{d}_j} \cdot b_j \cdot X_{jt}$$

Outline

Problem Description

Define:

$$
\begin{aligned}
I_f &= \{\text{Project that fails}\} \\
I_\delta &= \{\text{Project that delays but succeeds}\} \\
I_{f|\delta} &= \{\text{Project that delays and then fails}\}
\end{aligned}
$$

Each project has probabilities to delay, to fail, or to delay and then fail. The cost of each occurence of bad luck can be calculated from these probabilities. The total cost is:

$$w(I_f, I_\delta, I_{f|\delta}) := \sum_{j \in I_f} [-log_2(p_{f,j})] + \sum_{j \in I_\delta} [-log_2((1-p_{f,j})p_{\delta,j})] + \sum_{j \in I_{f|\delta}} [-log_2((1$$

Given a bad luck budget W, want to find a schedule (S, T) that yields the largest NPV in the worst case.

Adaptive Scheduling

The schedule will be updated once any bad luck happens. The purpose is to reduce the loss.

Assumptions:

- Bad Luck only happens to a project at its scheduled ending time.
- Only projects that are scheduled to start at and after the time of bad luck can be updated.
- When updating the schedule, we don't take future bad luck into account.

Adaptive Scheduling

The schedule will be updated once any bad luck happens. The purpose is to reduce the loss.

Assumptions:

- Bad Luck only happens to a project at its scheduled ending time.
- Only projects that are scheduled to start at and after the time of bad luck can be updated.
- When updating the schedule, we don't take future bad luck into account.

Adaptive Scheduling

The schedule will be updated once any bad luck happens. The purpose is to reduce the loss.

Assumptions:

- Bad Luck only happens to a project at its scheduled ending time.
- Only projects that are scheduled to start at and after the time of bad luck can be updated.
- When updating the schedule, we don't take future bad luck into account.

Adaptive Scheduling

Example:

$$1 \succ 2 \succ 3 \text{ and } c_1 = c_2 = c_3 = 1$$
$$b_1 = b_2 = 0, b_3 = 10, d_1 = d_2 = d_3 = 12$$

The deadline is Month 36

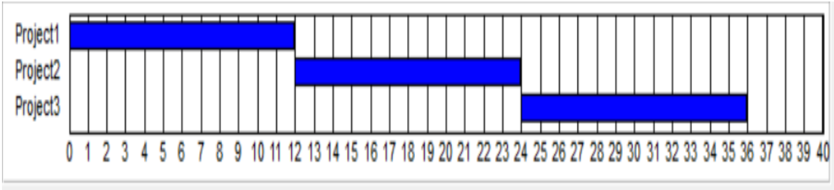


Figure: Initial Schedule

Adaptive Scheduling

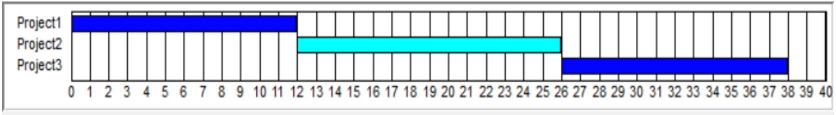


Figure: Schedule affected by bad luck

Project 3 cannot be completed by the deadline.

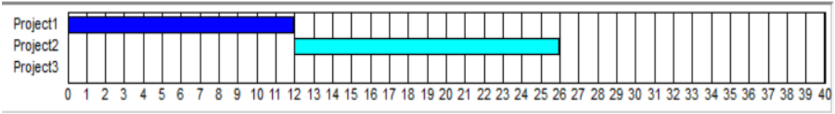


Figure: Updated Schedule after Delay

Delete Project 3 in the updated schedule.

Adaptive Scheduling

Given an initial schedule (S, T) and a triple of bad luck $(I_f, I_\delta, I_{f|\delta})$, can get a final realization (S', T') by solving a series of optimization problems.

Estimation of Worst Case Scenario

Given the budget W, want to find a triple of bad luck $(I_f, I_\delta, I_{f|\delta})$ that maximizes the loss.

Cannot incorporate adaptive scheduling into a single optimization problem.

Brute-force approach: Try every combination of $(I_f, I_\delta, I_{f|\delta})$ and pick the one that yields the maximum loss. Computationally infeasible.

General Framework

Data: Initial Schedule (S, T), budget of bad luck W

while *Stopping Criteria not satisfied* **do**

Find $(I_f, I_\delta, I_{f|\delta})$ in the worst case scenario under Adaptive Scheduling;

Robustify (S, T) based upon $(I_f, I_\delta, I_{f|\delta})$;

end

Output (S, T)

The initial schedule can be the solution in the simple case.

Estimation of Worst Case Scenario

- Our approach: Greedy Heuristics.
- Pros: Computationally efficient. Can get a bad scenario and identify critical projects in a reasonable amount of time.
- Cons: No proof for the optimality of $(I_f, I_\delta, I_{f|\delta})$.

Estimation of Worst Case Scenario

- Our approach: Greedy Heuristics.
- Pros: Computationally efficient. Can get a bad scenario and identify critical projects in a reasonable amount of time.
- Cons: No proof for the optimality of $(I_f, I_\delta, I_{f|\delta})$.

Estimation of Worst Case Scenario

- Our approach: Greedy Heuristics.
- Pros: Computationally efficient. Can get a bad scenario and identify critical projects in a reasonable amount of time.
- Cons: No proof for the optimality of $(I_f, I_\delta, I_{f|\delta})$.

Estimation of Worst Case Scenario

Data: Initial Schedule (S,T), budget of bad luck W,$(I_f, I_\delta, I_{f|\delta}) = \emptyset$
Result: Triple of bad luck in the worst case $(I_f, I_\delta, I_{f|\delta})$
while *1* **do**
    for *i in S, p in* $\{Failure, delay, both\}$ *with* $W_{ip} \leq W$ **do**
    end
end

Robustification of Schedule

After getting the triple $(I_f, I_\delta, I_{f|\delta})$ in the worst case scenario, can take the following measures to robustify the schedule:

- If Project i fails in the worst case scenario:
  - i is in set I with $I \vdash j$. Add another project $k \in I$ to the schedule.
  - Delete Project i.
- If Project i is delayed in the worst case scenario:
  - Move the starting time of i forward if possible.
  - Delete Project i.

Robustification of Schedule

After getting the triple $(I_f, I_\delta, I_{f|\delta})$ in the worst case scenario, can take the following measures to robustify the schedule:

- If Project i fails in the worst case scenario:
  - i is in set I with $I \vdash j$. Add another project $k \in I$ to the schedule.
  - Delete Project i.
- If Project i is delayed in the worst case scenario:
  - Move the starting time of i forward if possible.
  - Delete Project i.

Robustification of Schedule

After getting the triple $(I_f, I_\delta, I_{f|\delta})$ in the worst case scenario, can take the following measures to robustify the schedule:

- If Project i fails in the worst case scenario:
  - i is in set I with $I \vdash j$. Add another project $k \in I$ to the schedule.
  - Delete Project i.
- If Project i is delayed in the worst case scenario:
  - Move the starting time of i forward if possible.
  - Delete Project i.

Robustification of Schedule

After getting the triple $(I_f, I_\delta, I_{f|\delta})$ in the worst case scenario, can take the following measures to robustify the schedule:

- If Project i fails in the worst case scenario:
  - i is in set I with $I \vdash j$. Add another project $k \in I$ to the schedule.
  - Delete Project i.
- If Project i is delayed in the worst case scenario:
  - Move the starting time of i forward if possible.
  - Delete Project i.

Robustification of Schedule

After getting the triple $(I_f, I_\delta, I_{f|\delta})$ in the worst case scenario, can take the following measures to robustify the schedule:

- If Project i fails in the worst case scenario:
  - i is in set I with $I \vdash j$. Add another project $k \in I$ to the schedule.
  - Delete Project i.
- If Project i is delayed in the worst case scenario:
  - Move the starting time of i forward if possible.
  - Delete Project i.

Robustification of Schedule

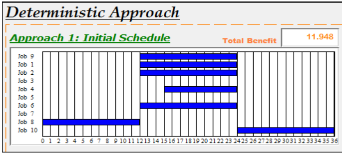
After getting the triple $(I_f, I_\delta, I_{f|\delta})$ in the worst case scenario, can take the following measures to robustify the schedule:

- If Project i fails in the worst case scenario:
  - i is in set I with $I \vdash j$. Add another project $k \in I$ to the schedule.
  - Delete Project i.
- If Project i is delayed in the worst case scenario:
  - Move the starting time of i forward if possible.
  - Delete Project i.

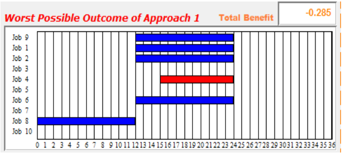
Outline

Small-size Example

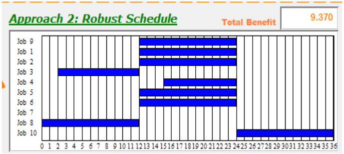
The small-size Example contains 13 projects. We test variuous choices of bad luck budget W and hurdle rate γ.



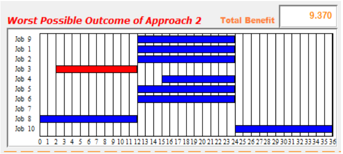
(a) Initial Schedule



(b) Intial Worst Case



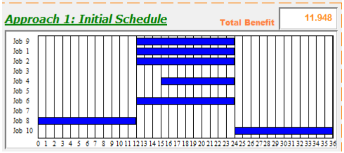
(c) Robust Schedule



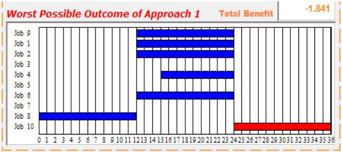
(d) Robust Worst Case

Figure: $W = 1.4$, $\gamma = 0.99$

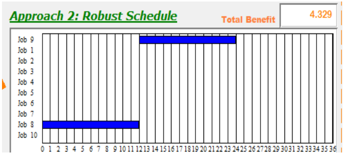
Small-size Example



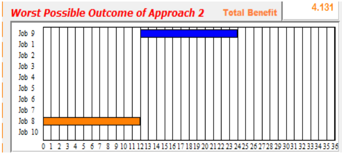
(a) Initial Schedule



(b) Intial Worst Case



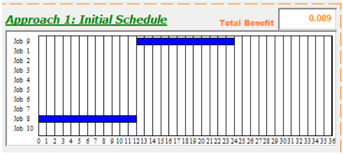
(c) Robust Schedule



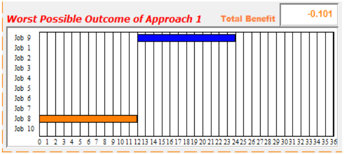
(d) Robust Worst Case

Figure: $W = 2.5$, $\gamma = 0.99$

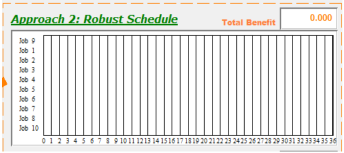
Small-size Example



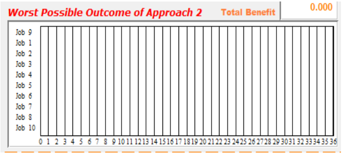
(a) Initial Schedule



(b) Initial Worst Case



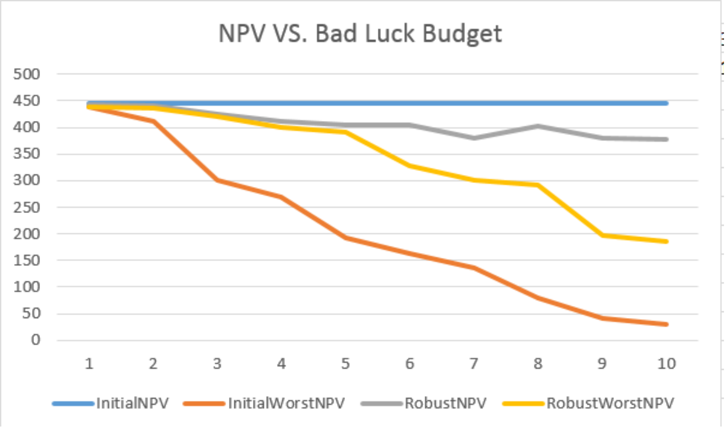
(c) Robust Schedule



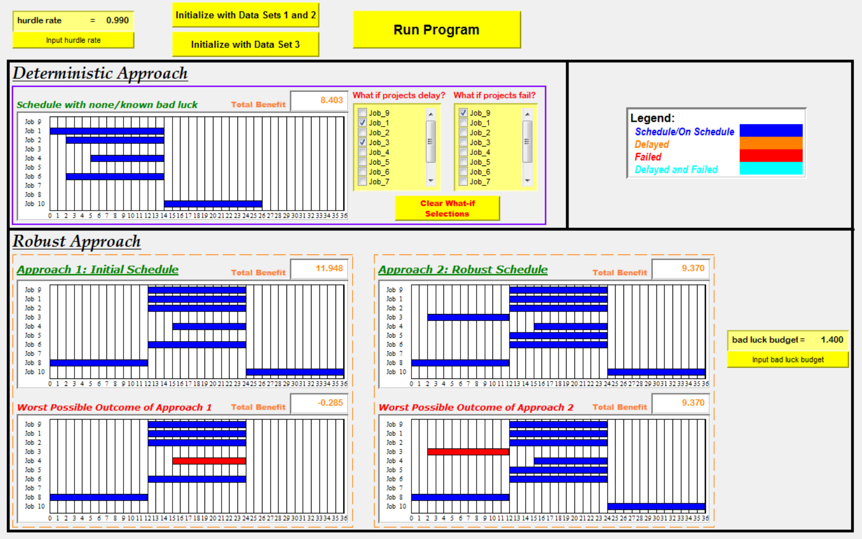
(d) Robust Worst Case

Figure: $W = 2.5$, $\gamma = 0.95$

Large-size Example



Graphical User Interface



Outline