A new PDE-based approach for construction scheduling and resource allocation

Paul Gabet, Julien Nachef
CE 291F – Project Presentation
Spring 2014
Problem Statement

• What is the schedule of a project?
  • A chronological list of tasks (Task 1, Task 2, etc.)
  • Constraints: Precedents and Following Tasks
  • Nodes/Events

Example of schedule
Problem Statement

- Cost of a task $i$ is a function of the desire duration
  \[ C_i = C_i(d_i) \]

- Payment depends on time of completion

\[ \text{Cost of resources (\$)} \]
\[ \text{Time (t)} \]

\[ \text{Task completed quickly with massive resources involved} \]
\[ \text{Task completed slowly with minimum price} \]

\[ \text{Earnings (\$)} \]
\[ \text{Time (t)} \]
\[ T_c \]
**Problem Statement**

- **Control**: Resources allocated to all the future tasks

  - Task duration $d_i = E_i - S_i$
  - Time of events $S_i$, $E_i$ and $E_{node}$
  - Time of project completion $E_{end}$

- **Objective**:

  $\text{Maximize: } P(E_{end}) - \sum_i C_i$

Subject to:

- $C_i = C_i(d_i)$ for all tasks $i$
- $d_i = E_i - S_i$ for all tasks $i$
- *Precedence and Following tasks constraints*
Industry State of the Art

• Widely-used of the Critical Path Method (CPM)

• Main limitations
  • No systematic optimization of resource allocation
  • Approximate iterative method
  • Not designed for real-time adjustment
    ➢ When deviation occurs between actual and scheduled progress
    ➢ Addresses poorly criticality of tasks
Our PDE approach

- Local resource allocation (sub-optimality)

\[ C(E_1) = C_1(E_1 - S) + C_2(E - E_1) \]

Minimum for \( C'(E_1) = 0 \)

\[ C_1'(E_1 - S) = C_2'(E - E_1) \]

Tasks behave like springs of tension \( c_i'(d_i) \)

**Dynamic D1**

\[ \frac{\partial E_1}{\partial t} = -\gamma \cdot [C_1'(E_1 - S) - C_2'(E - E_1)] \]
General Case

• Local resource allocation (sub-optimality)

Dynamic D1:

\[
\frac{\partial E_{\text{node}}}{\partial t} = -\gamma \cdot \left( \sum_i c'_i (E_{\text{node}} - S_i) - \sum_j c'_j (E_j - E_{\text{node}}) \right) \text{, for all nodes}
\]

Dynamic D2:

\[
\frac{\partial^2 E_{\text{node}}}{\partial^2 t} = -\gamma \cdot \left( \sum_i c'_i (E_{\text{node}} - S_i) - \sum_j c'_j (E_j - E_{\text{node}}) \right) + f \left( \frac{\partial E}{\partial t} \right) \text{, for all nodes}
\]

➤ Behaves as a real spring network dynamic
• Optimization of schedules without changing task duration
  • Use of Genetic algorithms (Tarek Hegazy, 1999)
  • Use of Back propagation models (Elazouni et al., 1997)

• Optimization of schedules and resource allocation simultaneously
  • Use of Lyapunov function and a global dynamic (Adeli, 2001)

• What we do is different:
  • We are using a local dynamic to optimize an entire schedule
Contribution

• Use PDE approach for resource allocation in project scheduling

• Proof of convergence of the model toward optimal schedule

• Proof of local exponential stability of the model

• Implementation of the model on Matlab (Oriented Object)
• Demonstration of good behavior on
  • An elementary project (9 tasks)
  • A real construction project (~250 tasks)

• Provide a new approach to task criticality

• Simulation in real-time
  • Benefits of continuous scheduling optimization along with task progress assimilation
**Dynamic D1:**

For all node

\[
\frac{\partial E_{node}}{\partial t} = -\gamma \cdot (\sum_i c'_i(E_{node} - S_i) - \sum_j c'_j(E_j - E_{node})
\]

**Assumption:**

Cost functions \((c_i)_i\) are strictly convex.

**Proposition:**

Dynamic D1 has a unique equilibrium which is the optimal schedule.

**Theorem:**

Dynamic D1 converges locally exponentially towards optimum.
Accuracy and exponential convergence

**Dynamic D1:**
For all node
\[
\frac{\partial E_{\text{node}}}{\partial t} = -\gamma \cdot (\sum_i c_i'(E_{\text{node}} - S_i) - \sum_j c_j'(E_j - E_{\text{node}})
\]

**Theorem:**
Dynamic D1 converges locally exponentially towards optimum.

**Scheme of proof:**
Linearization of D1 near optimum \((E^*)_{\text{node}}\):
\[
\frac{d\Delta E}{dt} = -A\Delta E \quad \text{where} \quad (\Delta E)_{\text{node}} = (E)_{\text{node}} - (E^*)_{\text{node}}
\]

\[
\begin{pmatrix}
e_1 \\
e_2
\end{pmatrix}
\]
\[
\begin{pmatrix}
\sum_{j \in V(e_1)} c_j'' & -c_i'' \\
-c_i'' & \sum_{j \in V(e_2)} c_j''
\end{pmatrix}
\]
\(a\)

and matrix \(A\) = 
\[
\begin{pmatrix}
c_j'' & -c_i'' \\
-c_i'' & c_j''
\end{pmatrix}
\]  
+ positive diagonal matrix

\(\triangleright\) A is Hurwitz if cost functions \((c_i)_i\) are strictly convex near equilibrium.
Elementary Project

- Task 1: Obtain Building permit
- Task 2: Install temporary networks
- Task 3: Idle task
- Task 4: Order cable trays, work space outlets
- Task 5: Order router, cabling, etc.
- Task 6: Install cable trays
- Task 7: Install work space outlets
- Task 8: Install, test new file servers
- Task 9: Test communication systems
<table>
<thead>
<tr>
<th>Time (days)</th>
<th>Profit (k$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Schedule</td>
<td>10</td>
</tr>
<tr>
<td>Optimized Schedule</td>
<td>6.9</td>
</tr>
<tr>
<td>% improvement</td>
<td><strong>31%</strong></td>
</tr>
</tbody>
</table>
Task Criticality

- How to characterize the criticality of task 1?
  - Spring analogy: tension $c'_1(d_1)$
  - $c'_1(d_1)$: cost to finish task 1 one day earlier
    - or
  - how difficult it is to catch up a one day delay
Criticality on an elementary example

- Evolution of the criticality of each task while optimizing the schedule
Criticality on real construction project

- Change of the desired time completion of the project
Assimilation of progress data

P=0% $P_{estimated}$ P=100%

Progress P

t
Assimilation of progress data

\[ P = 0\% \quad P_{\text{estimated}} \quad P_{\text{data}} \quad P = 100\% \]

Progress P

\( t \)
Assimilation of progress data

P = 0%  \quad P_{\text{estimated}}  \quad P = 100%

P_{\text{data}}

Progress P

t
Assimilation of progress data

$P_{estimated} = P_{data}$

End of task is loose:
- Future tasks can take more time and less resources.
- The project can be advanced.
  
  ➢ Apply dynamic D1

Only if continuous optimization of resource allocation.
Real-time simulation of project

SIMULATION OF REALITY (TASK PROGRESS)

Time t

Random progress $\Delta P$
on current tasks

Sparse progress data

SCHEDULING

➢ Assimilate progress data
➢ Update effort on current task

Time $t + \Delta t$

Random progress $\Delta P$
on current tasks

Iterate dynamic D1
➢ Adjust allocated resources

CE 291F - Project Presentation 22
Real-time simulation of project

- On-going work: quantify the gain of continuously optimizing the schedule
A new PDE-based approach for construction scheduling and resource allocation

Paul Gabet, Julien Nacheff
CE 291F – Project Presentation
Spring 2014

Thank you