Optimal Scheduling of Precedence-constrained Task Graphs on Heterogeneous Distributed Systems with Shared Buses

Sanjit Kumar Roy\(^1\), Sayani Sinha\(^2\), Kankana Maji\(^2\), Rajesh Devaraj\(^1\), and Arnab Sarkar\(^1\)

\textit{IEEE ISORC 2019}

May 9, 2019

\(^1\)Indian Institute of Technology Guwahati
\(^2\)Jadavpur University
# Table of Contents

1. Introduction
2. The Models
3. ASAP/ALAP
4. ILP Formulation
   - ILP1
   - ILP2
5. Experimental Evaluation
   - Experiment-1
   - Experiment-2
   - Experiment-3
6. Case Study
7. Conclusion
8. Bibliography
Applications in many time-critical cyber-physical systems are often represented as *Precedence-constrained Task Graphs* (PTGs).

There is an increasing trend towards their implementation on distributed heterogeneous platforms
- consisting of heterogeneous processing elements
- shared buses (CAN, LIN, FlexRay etc.) [1]

On a distributed platform consisting of heterogeneous processing and communication resources,
- execution of a task may require different amounts of time on different processing elements.
- transmission of a message may require different amounts of time on different communication resources.
Given a PTG representing a real-time application and a heterogeneous platform, successful execution/transmission of the task/message nodes while satisfying all timing, precedence and resource related specifications, is ultimately a *scheduling* problem.

- Scheduler design schemes for PTGs can be broadly classified as *static* (offline) and *dynamic* (online) [2].
- In safety-critical systems such as automotive/avionic systems [3], it is often advisable that all timing requirements be guaranteed off-line, before putting the system in operation [4].
- Hence, static off-line scheduling schemes are preferred in such systems to provide a high degree of timing predictability [5].
Most existing real-time static scheduling approaches for PTGs are list scheduling based heuristic schemes [2, 6, 7].

A majority of them attempt to minimize the overall schedule length (makespan minimization).

Such an objective allows maximization of the spare computation bandwidth in the system, which may be used to perform other useful activities.

Many of them assume that the underlying execution platform consists of a fully connected system of processing elements.

There exists a significant class of cyber-physical systems with bus based shared communication links among processors.
Introduction: Heuristic vs Optimal

- **Heuristic schedules**
  - typically based on the satisfaction of a set of sufficiency conditions
  - cannot take into consideration all necessary schedulability requirements
  - schedules are sub-optimal in nature

- **Optimal solutions**
  - can make a fundamental difference in resource-constrained time-critical systems with respect to performance, reliability and other non-functional metrics like cost, power, space etc
  - Optimal schedules can act as benchmarks allowing accurate comparison and evaluation of heuristic solutions [8]
We design an *Integer Linear Programming (ILP)* based static optimal real-time scheduling strategy for PTGs executing on a distributed platform consisting of heterogeneous processing nodes and inter-connected through a set of heterogeneous shared buses.
A set of resources \( \{R_1, R_2, \ldots, R_{p+b}\} \) among which,

- \( \{R_1, R_2, \ldots, R_p\} \) denote a set \( P = \{P_1, P_2, \ldots, P_p\} \) of \( p \) heterogeneous processing elements
- \( \{R_{p+1}, R_{p+2}, \ldots, R_{p+b}\} \) denote a set \( B = \{B_1, B_2, \ldots, B_b\} \) of \( b \) heterogeneous shared buses
- Each processing node \( P_i \) is connected to all \( b \) buses
**Computation Model**

\[ V_1 = T_1 \]
\[ V_2 = T_2 \]
\[ V_3 = T_3 \]
\[ V_4 = T_4 \]
\[ V_5 = T_5 \]
\[ V_6 = T_6 \]
\[ V_7 = M_1 \]
\[ V_8 = M_2 \]
\[ V_9 = M_3 \]
\[ V_{10} = M_4 \]
\[ V_{11} = M_5 \]
\[ V_{12} = M_6 \]

**Figure:** (a) PTG \( G \), (b) Platform Model \( \rho \), (c) Computation-time Matrix (\( CT \)) and (d) Communication-time Matrix (\( CM \)).
A *Precedence-constrained Task Graph* (PTG) $G$ is described by a quadruple $G = (V, E, CT, CM)$ where,

- $V = \{V_1, V_2, \ldots, V_{n+m}\}$ represents a set of nodes
- $\{V_1, V_2, \ldots, V_n\}$ represent a set $T = \{T_1, T_2, \ldots, T_n\}$ of $n$ task nodes
- $\{V_{n+1}, V_{n+2}, \ldots, V_{n+m}\}$ denote a set $M = \{M_1, M_2, \ldots, M_m\}$ of $m$ message nodes
- $E \subseteq V \times V$ is a set of edges that describe the *precedence-constraints* among nodes in $V$.
- $CT$ is a $n \times p$ *computation-time matrix*
- $CM$ is a $m \times b$ *communication-time matrix*
Assumptions

- Single source node $T_1$
- Single sink node $T_n$
- Both source ($T_1$) and sink ($T_n$) nodes are tasks.
- Each task node $T_i$ is preceded/succeeded by one or more message nodes.
- Each message node $M_k$ is preceded/succeeded by a single task node.
- The communication time for $M_k$ is negligible if both preceding and succeeding task nodes are mapped to same processing element.
Problem Formulation

Given a PTG \( G = (V, E, CT, CM) \) with end-to-end deadline \( D \), \( p \) processing elements and \( b \) buses, find:

- A task node assignment \( V_i \mapsto R_j; 1 \leq i \leq n \) and \( 1 \leq j \leq p \)
- A message node assignment \( V_i \mapsto R_j; n + 1 \leq i \leq n + m \) and \( p + 1 \leq j \leq p + b \)
  - If both the preceding and succeeding task nodes of message node \( M_i \) are mapped to the same processing element then, \( V_i \mapsto \emptyset \)
- A start time for each task node and message node, such that
  - length of the total schedule is minimized and
  - meets the deadline \( D \)
Earliest/Latest Start Times for PTG Nodes

Let, $t^{s}_i$ and $t^{l}_i$ be the ASAP and ALAP time of node $V_i$, respectively

- **ASAP time computation of task nodes:**
  - Ignore message nodes in the PTG
  - Set ASAP time of the source task node, $t^{s}_1 = 1$
  - Compute ASAP times of the remaining task nodes recursively (downward) as follows:

$$
t^{s}_i = \max_{T_j \in \text{pred}(T_i)} (t^{s}_j + \min_{r \in [1,p]} CT_{jr})
$$

where, $\text{pred}(T_i)$ is the set of immediate predecessors of task node $T_i$
Earliest/Latest Start Times for PTG Nodes

- **ALAP time computation of task nodes:**
  - Ignore message nodes in the PTG
  - Set ALAP time for the sink task node as,
    \[ t_l^n = D - \min_{r \in [1,p]} CT_{nr} \]
  - Compute ALAP times of the remaining task nodes recursively (upward) as follows:
    \[ t_l^i = \min_{T_j \in \text{succ}(T_i)} \left( t_l^j - \min_{r \in [1,p]} CT_{ir} \right) \]

where, \( \text{succ}(T_i) \) is the set of immediate successors of task node \( T_i \)
**ASAP/ALAP computation procedure for message nodes:**

- **ASAP time of a message node** \( M_k \) is,
  \[
  t_{n+k}^s = t_i^s + \min_{r \in [1,p]} CT_{ir}
  \]
  where, \( T_i \) is the predecessor task node of \( M_k \)

- **ALAP time of a message node** \( M_k \) is,
  \[
  t_{n+k}^l = t_j^l - \min_{r \in [1,b]} CM_{kr}
  \]
  where, \( T_j \) is the successor task node of \( M_k \)
ILP Formulation: ILP1

We define binary decision variable,

\[ X_{irt} = \begin{cases} 
1 & \text{if node } i \text{ starts its execution/transmission} \\
& \text{on } r^{th} \text{ resource at time step } t \\
0 & \text{Otherwise} 
\end{cases} \]

where, \( i = 1, 2, \ldots, n + m; \ r = 1, 2, \ldots, p + b; \ t = 1, 2, \ldots, D \)
**ILP1**

**Unique Start Time Constraints:**
Start time of each task node should be unique,

\[
\forall i \in [1, n] \quad \sum_{r=1}^{p} \sum_{t=t_i^s}^{t_i^l} X_{irt} = 1 \quad (1)
\]

Start time of each message node should be unique,

\[
\forall M_k \mid T_i = \text{pred}(M_k) \text{ and } T_j = \text{succ}(M_k), \quad \sum_{r=p+1}^{p+b} \sum_{t=t_{k'}^s}^{t_{k'}^l} X_{k'r}t = 1 - Y_k \quad (2)
\]

where,

\[
k' = n + k \text{ and } Y_k = \sum_{r=1}^{p} \sum_{t_1=t_i^s}^{t_i^l} \sum_{t_2=t_j^s}^{t_j^l} X_{irt_1} * X_{jrt_2}
\]

**Figure:** PTG
We introduce another binary decision variable $U_{krt_1 t_2}$ ($= X_{irt_1} \times X_{jrt_2}$) to linearize the non-linear term, 

$$Y_k = \sum_{r=1}^{p} \sum_{t_1=t_i^l}^{t_i^s} \sum_{t_2=t_j^l}^{t_j^s} U_{krt_1 t_2}$$

(3)

Now, the non-linear variables $U_{krt_1 t_2}$ can be linearized using the following three inequalities,

$$X_{irt_1} \geq U_{krt_1 t_2}$$

(4)

$$X_{jrt_2} \geq U_{krt_1 t_2}$$

(5)

$$U_{krt_1 t_2} \geq X_{irt_1} + X_{jrt_2} - 1$$

(6)
Resource Constraints:
A resource can execute at most one task/message node at a given time.
For processing element:

\[ \forall t \in [1, D] \text{ and } \forall r \in [1, p] \quad \sum_{i=1}^{n} \sum_{t'=\psi}^{t} X_{irt'} \leq 1 \quad (7) \]

where, \( \psi = t - CT_{ir} + 1 \).

For bus element:

\[ \forall t \in [1, D] \text{ and } \forall r \in [1, b] \quad \sum_{i=1}^{m} \sum_{t'=\psi}^{t} X_{i'r't'} \leq 1 \quad (8) \]

where, \( i' = i + n, r' = r + p \) and \( \psi = t - CM_{ir} + 1 \).
Dependency Constraints:
Dependencies between nodes must be satisfied,

$$\forall M_k \mid T_i = \text{pred}(M_k) \text{ and } T_j = \text{succ}(M_k),$$

$$\sum_{r=1}^{p} \sum_{t=t_i^s}^{t_i^l} (t + CT_{ir}) \cdot X_{irt} \leq \sum_{r=p+1}^{p+b} \sum_{t=t_i^s}^{t_i^l} t \cdot X_{k'rt} \leq$$

$$+ \sum_{r=1}^{p} \sum_{t=t_j^s}^{t_j^l} t \cdot X_{jrt} \cdot Y_k$$

(9)

where, $k' = n + k$. 

Figure: PTG
Dependency Constraints Contd.

We replace the non-linear term $Y_k \times X_{jrt}$ by $Z_{krt}$ and linearize by,

\[
Z_{krt} \leq X_{jrt} \quad (10)
\]

\[
Z_{krt} \leq Y_k \quad (11)
\]

\[
Z_{krt} \geq Y_k + X_{jrt} - 1 \quad (12)
\]

\[\forall M_k | T_j = succ(M_k),\]

\[
\sum_{p+1}^{p+b} \sum_{t=t_k'}^{t_{k'}^l} (t + CM_{kr}) \times X_{k'rt} \leq \sum_{r=1}^{p} \sum_{t=t_j^s}^{t_j^l} t \times X_{jrt} \quad (13)
\]

where, $k' = n + k$. 

Figure: PTG
Objective function: Minimize schedule length of the PTG.

\[
\text{Minimize} \sum_{r=1}^{p} \sum_{t=t_n^s}^{t_n^l} X_{nrt}(t + CT_{nr})
\]

subject to constraints presented in equations 1 - 13.
Linearization in equations 10 to 12 may be avoided by replacing equation 9 with the following two equations.

\[
\forall M_k | T_i = \text{pred}(M_k) \text{ and } T_j = \text{succ}(M_k),
\]

\[
\sum_{r=1}^{p} \sum_{t=t_i^l}^{t_i^l} (t + CT_{ir}) \times X_{irt} \leq \sum_{r=p+1}^{p+b} \sum_{t=t_{k'}^l}^{t_{k'}^l} t \times X_{k'rt}
\]

\[
+ \sum_{r=1}^{p} \sum_{t=t_j^l}^{t_j^l} t \times X_{jrt} \times Y_k
\]

where, \( k' = n + k \).
∀\( M_k \) | \( T_i = \text{pred}(M_k) \) and \( T_j = \text{succ}(M_k) \),

\[
\sum_{r=1}^{p} \sum_{t=t_i^s}^{t_i^l} (t + CT_{ir}) \ast X_{irt} \leq \sum_{r=1}^{p} \sum_{t=t_j^s}^{t_j^l} t \ast X_{jrt} \quad (15)
\]

\[
\sum_{r=1}^{p} \sum_{t=t_i^s}^{t_i^l} (t + CT_{ir}) \ast X_{irt} \leq \sum_{r=p+1}^{p+b} \sum_{t=t_k^s}^{t_k^l} t \ast X_{k'rt} + C \ast Y_k \quad (16)
\]

where, \( k' = n + k \) and \( C \) is a sufficiently large constant.
Experimental Setup

- We evaluate and compare the performance of ILP1 and ILP2.
- Performance metrics
  - #Constraints generated
  - Time required to generate a solution
- Experiments have been conducted using six standard PTGs.
- The scenarios considered differ in terms of,
  - Number of processing elements ($p$)
  - Number of buses ($b$)
  - Communication to Computation Ratio ($CCR$)
  - Deadline ($D$)
- All experiments are carried out using the CPLEX optimizer [9] version 12.6.2.0, executing on a system having Intel(R) Xeon(R) CPU running Linux Kernel 2.6.32-042stab123.1.
Experimental Setup

(a) PTG1 [10]  
(b) PTG2 [11]  
(c) PTG3 [11]  
(d) PTG4 [6]  
(e) PTG5 [6]  
(f) PTG6 [11]

Figure: Benchmark PTGs from [6, 10, 11]
Experiment-1

Compared ILP1 and ILP2

- #processing elements \((p) = 4\)
- #buses \((b) = 2\)
- *Communication to Computation Ratio (CCR) = 0.5*
- Execution/transmission times generated from a uniform random distribution within the range 5 ms to 15 ms and scaled properly

<table>
<thead>
<tr>
<th>PTG</th>
<th>(n)</th>
<th>(m)</th>
<th>(D)</th>
<th>(SL)</th>
<th>Running Time</th>
<th>#Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(ILP1)</td>
<td>(ILP2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(ILP1)</td>
<td>(ILP2)</td>
</tr>
<tr>
<td>PTG1</td>
<td>6</td>
<td>7</td>
<td>32</td>
<td>32</td>
<td>0.19</td>
<td>0.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4681</td>
<td>4112</td>
</tr>
<tr>
<td>PTG2</td>
<td>6</td>
<td>7</td>
<td>37</td>
<td>37</td>
<td>0.34</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8734</td>
<td>7925</td>
</tr>
<tr>
<td>PTG3</td>
<td>8</td>
<td>9</td>
<td>46</td>
<td>42</td>
<td>7.68</td>
<td>3.72</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>28601</td>
<td>26918</td>
</tr>
<tr>
<td>PTG4</td>
<td>10</td>
<td>15</td>
<td>42</td>
<td>38</td>
<td>35.57</td>
<td>5.73</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>48905</td>
<td>46064</td>
</tr>
<tr>
<td>PTG5</td>
<td>14</td>
<td>19</td>
<td>80</td>
<td>72</td>
<td>111.44</td>
<td>24.73</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>147313</td>
<td>141512</td>
</tr>
<tr>
<td>PTG6</td>
<td>16</td>
<td>24</td>
<td>72</td>
<td>67</td>
<td>1577.40</td>
<td>171.98</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>226443</td>
<td>218535</td>
</tr>
</tbody>
</table>

**Table:** Running time (seconds) and #constraints for PTGs
Experiment-2

Compared ILP1 & ILP2 (varying number of task and message nodes)

- PTG6a: Eliminate message nodes $M_{11}, M_{16}, M_{17}$ and task node $T_9$ from PTG6
- PTG6b: Eliminate message nodes $M_9, M_{14}, M_{15}$ and task node $T_8$ from PTG6a

<table>
<thead>
<tr>
<th>PTG</th>
<th>$n$</th>
<th>$m$</th>
<th>$D$</th>
<th>$SL$</th>
<th>Running Time</th>
<th>#Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ILP1</td>
<td>ILP2</td>
</tr>
<tr>
<td>PTG6</td>
<td>16</td>
<td>24</td>
<td>72</td>
<td>67</td>
<td>1577.40</td>
<td>171.98</td>
</tr>
<tr>
<td>PTG6a</td>
<td>15</td>
<td>21</td>
<td>69</td>
<td>64</td>
<td>208.07</td>
<td>35.81</td>
</tr>
<tr>
<td>PTG6b</td>
<td>14</td>
<td>18</td>
<td>63</td>
<td>58</td>
<td>53.56</td>
<td>10.17</td>
</tr>
</tbody>
</table>

Table: Performance comparison w.r.t PTGs 6, 6a and 6b (second)
Experiment-3

This experiment compares run time overheads incurred by ILP2

- Parameters are,
  - $p \in \{2, 4\}$
  - $b \in \{1, 2\}$
  - $CCR \in \{0.25, 0.5, 0.75\}$
  - $DR \in \{1.0, 1.1, 1.2\}$
  - $DR$ refers to the ratio $(D : SL)$

<table>
<thead>
<tr>
<th></th>
<th>$CCR = 0.25$</th>
<th></th>
<th>$CCR = 0.5$</th>
<th></th>
<th>$CCR = 0.75$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$SL$</td>
<td>$DR$</td>
<td>$SL$</td>
<td>$DR$</td>
<td>$SL$</td>
</tr>
<tr>
<td></td>
<td>(b = 1)</td>
<td></td>
<td>(b = 2)</td>
<td></td>
<td>(b = 1)</td>
</tr>
<tr>
<td>$p = 2$</td>
<td>57</td>
<td>10.93</td>
<td>17.46</td>
<td>131.25</td>
<td>55</td>
</tr>
<tr>
<td>(b = 1)</td>
<td>57</td>
<td>9.61</td>
<td>21.61</td>
<td>79.76</td>
<td>54</td>
</tr>
<tr>
<td>$p = 4$</td>
<td>42</td>
<td>37.30</td>
<td>109.53</td>
<td>168.96</td>
<td>45</td>
</tr>
<tr>
<td>(b = 1)</td>
<td>37</td>
<td>1.53</td>
<td>14.76</td>
<td>23.71</td>
<td>38</td>
</tr>
</tbody>
</table>

Table: Running time of ILP2 (in seconds) w.r.t PTG4 for different #resources, $DR$ and $CCR$
Case Study: Adaptive Cruise Controller

Figure: ACC Block Diagram [12]
Case Study

<table>
<thead>
<tr>
<th></th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
<th>T7</th>
<th>T8</th>
<th>T9</th>
<th>T10</th>
<th>T11</th>
<th>T12</th>
<th>T13</th>
<th>T14</th>
<th>T15</th>
<th>T16</th>
<th>T17</th>
<th>T18</th>
<th>T19</th>
<th>T20</th>
<th>T21</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>29</td>
<td>29</td>
<td>26</td>
<td>29</td>
<td>21</td>
<td>43</td>
<td>36</td>
<td>21</td>
<td>37</td>
<td>25</td>
<td>21</td>
<td>20</td>
<td>36</td>
<td>29</td>
<td>43</td>
<td>36</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>P2</td>
<td>25</td>
<td>27</td>
<td>29</td>
<td>35</td>
<td>23</td>
<td>45</td>
<td>43</td>
<td>25</td>
<td>43</td>
<td>28</td>
<td>25</td>
<td>30</td>
<td>30</td>
<td>27</td>
<td>40</td>
<td>40</td>
<td>18</td>
<td>17</td>
<td>25</td>
<td>22</td>
</tr>
<tr>
<td>P3</td>
<td>32</td>
<td>21</td>
<td>27</td>
<td>27</td>
<td>20</td>
<td>37</td>
<td>45</td>
<td>24</td>
<td>45</td>
<td>26</td>
<td>19</td>
<td>25</td>
<td>40</td>
<td>31</td>
<td>45</td>
<td>30</td>
<td>23</td>
<td>24</td>
<td>20</td>
<td>24</td>
</tr>
<tr>
<td>P4</td>
<td>30</td>
<td>35</td>
<td>34</td>
<td>26</td>
<td>17</td>
<td>40</td>
<td>40</td>
<td>29</td>
<td>40</td>
<td>20</td>
<td>18</td>
<td>26</td>
<td>32</td>
<td>28</td>
<td>42</td>
<td>34</td>
<td>20</td>
<td>18</td>
<td>19</td>
<td>25</td>
</tr>
</tbody>
</table>

Table: Computation time (in ms) of task nodes

|       | M_{10} | M_{11} | M_{12} | M_{13} | M_{14} | M_{15} | M_{16} | M_{17} | M_{18} | M_{19} | M_{20} | M_{21} | M_{22} | M_{23} | M_{24} | M_{25} | M_{26} | M_{27} | M_{28} | M_{29} | M_{30} | M_{31} | M_{32} | M_{33} |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| B_1   | 1      | 1      | 1      | 1      | 2      | 2      | 2      | 3      | 1      | 1      | 2      | 1      | 1      | 1      | 3      | 1      | 1      | 1      | 2      | 1      | 1      | 2      | 1      |
| B_2   | 2      | 2      | 1      | 1      | 1      | 1      | 1      | 2      | 2      | 3      | 3      | 2      | 2      | 2      | 3      | 3      | 3      | 3      | 1      | 2      | 2      | 2      | 1      |

Table: Transmission time (in ms) of message nodes

Figure: Gantt chart representation of the schedule
Observations:

- ILP2 takes approximately 21872 secs (≈6 hours)
- Makespan is 146 ms
- Message nodes $M_{14}, M_{17}, M_{18}, M_{21}, M_{23}, M_{26}, M_{29}$ and $M_{32}$ are absent in the schedule
- All scheduling constraints are satisfied
This work considers the problem of computing optimal schedules for PTGs executing on distributed systems consisting of heterogeneous processing nodes and inter-connected via a limited number of shared buses.

The first version of the proposed ILP formulation requires two sets of computationally expensive linearizations.

Proposed an improved version of the ILP which reduces computational overheads by elegantly avoiding a sub-set of linearizations that are required to handle dependency constraints.

Experimental analysis using standard benchmark PTGs reveal the practical efficacy of our scheme.

Finally, a case study on a cruise control application has been presented.


Thank You