Lagrangian relaxation-based lower bound for resource-constrained modulo scheduling.

Maria Alejandra Ayala\textsuperscript{1}, Christian Artigues\textsuperscript{1}, Bernat Gacias\textsuperscript{1}

\textsuperscript{1}LAAS-CNRS; Université de Toulouse, France

\{mayala,artigues,bgacias\}@laas.fr

ISCO 2010
Outline

1. Introduction and problem presentation.
2. Time indexed linear programming formulation for the resource constrained modulo scheduling.
3. Lagrangian relaxation
4. Experimental results.
5. Conclusion and future work.
Introduction and problem presentation.

Time indexed linear programming formulation for the resource constrained modulo scheduling.

Lagrangian relaxation

Experimental results.

Conclusion and future work.
Industrial context: VLIW instruction scheduling

- The instruction scheduling problem is defined by a set of operations to schedule, a set of dependences between these operations, and a target processor micro-architecture. An operation is considered as an instance of an instruction in a program text.
- Modulo scheduling is a software pipelining framework, based only on 1-periodic cyclic schedules with integral period $\lambda$.
- The modulo scheduling algorithm must take into account the constraints of the target processor, this is, latencies of operations, resources, and size of the register files.
int prod(int n, short a[], short b) {
    int s = 0, i;
    for (i = 0; i < n; i++) {
        s += a[i] * b;
    }
    return s;
}

L?__0_8:
LDH_1 g131 = 0, G127
MULL_2 g132 = G126, g131
ADD_3 G129 = G129, g132
ADD_4 G128 = G128, 1
ADD_5 G127 = G127, 2
CMPNE_6 b135 = G118, G118
BRF_7 b135, L?__0_8

schedule instructions to end the program in minimum time

Optimize loop performance

Software pipeline

Cyclic scheduling problem

Modulo scheduling
Resource constrained modulo scheduling problem : data

- A set of $n$ generic operations with unit durations and a set of $m$ resources.
- Each resource has a limited availability $B_s$.
- Each generic operation requires $b^s_i$ units of each resource $s$.
- A set $E$ of dependences is defined where for each $(i, j) \in E$, a latency $\theta^j_i$ and a distance $\omega^j_i$ are given.
Introduction and problem presentation

Dependence constraints

Exemple: Let \((i, j)\) such that
\[ \theta_i^j = 2, \omega_i^j = 1 \]

Start time of instance \((i,k)\)
\[ \sigma_i^k = \sigma_i^0 + k\lambda \]
Let \(\sigma_i = \sigma_i^0\)
Dependence constraint
\[ \sigma_j^{k+\omega_i^j} \geq \sigma_i^k + \theta_i^j \]

\[ \sigma_i + \theta_i^j - \omega_i^j \lambda \leq \sigma_j \quad \forall (i, j) \in E \]

Resource constraints

Each task uses each resource such that \(\exists s, b_i^s > 0\) at time \(\sigma_i \mod \lambda\)
Introduction and problem presentation

Resource constrained modulo scheduling problem

- Decision variables: start time $\sigma_i \in \{0, \ldots, T\}$, where $T$ is the time horizon, for each generic operation $i = 1, \ldots, n$

- Minimize $\lambda$, subject to
  - modulo resource constraints
    \[
    \sum_{\sigma_i \mod \lambda = \tau} b_i^s \leq B_s, \quad \forall \tau \in \{0, \ldots, \lambda - 1\} \text{ and } \forall s \in \{1, \ldots, m\}
    \]
  - dependence constraints
    \[
    \sigma_j \geq \sigma_i + \theta_i^j - \lambda \omega_i^j, \quad \text{for all } (i, j) \in E
    \]

Remark 1: strongly NP-hard

Remark 2: For a fixed $\lambda$, find a feasible solution for a resource-constrained project scheduling problem with minimum and maximum time lags.
## Introduction and problem presentation

### Example.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>4</td>
</tr>
<tr>
<td>MEM</td>
<td>1</td>
</tr>
<tr>
<td>CTL</td>
<td>1</td>
</tr>
<tr>
<td>ODD</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RESERVATION</th>
<th>ALU</th>
<th>MEM</th>
<th>CTL</th>
<th>ODD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALU</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ALUX</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>MUL</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>MULX</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>MEM</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MEMX</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>CTL</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Diagram

\[ \lambda = 2 \]
Outline

1. Introduction and problem presentation.
2. Time indexed linear programming formulation for the resource constrained modulo scheduling.
3. Lagrangian relaxation
4. Experimental results.
5. Conclusion and future work.
Time indexed linear programming formulation for the resource constrained modulo scheduling.

Proposed by Dupont de Dinechin (2004). A fixed $\lambda$. Binary variables $x_i^t$ such that $x_i^t = 1$ if only if $\sigma_i = t$, $\sigma_i = \sum_{t=0}^{T} t x_i^t$. The feasible set $\mathcal{X}(\lambda)$ is described by (1-4).

\[
\sum_{t=0}^{T} x_i^t = 1, \forall i = 1, \ldots, n. \tag{1}
\]

\[
\sum_{i=1}^{n} \sum_{k=0}^{\lfloor \frac{T}{\lambda} \rfloor} x_i^{T+k\lambda} b_i^s \leq B_s, \forall \tau \in \{0, \ldots, \lambda - 1\}, s \in \{1, \ldots, m\} \tag{2}
\]

\[
\sum_{h=t}^{T} x_i^h + \sum_{h=0}^{t+\theta_i^j-\lambda \omega_i^j-1} x_j^h \leq 1, \forall t \in \{0, \ldots, T\}, (i,j) \in E \tag{3}
\]

\[
x_i^t \in \{0, 1\}, \forall i = 1, \ldots, n, \forall t \in \{0, \ldots, T\} \tag{4}
\]
Time indexed linear programming formulation for the resource constrained modulo scheduling.

Let $\tilde{X}(\lambda)$, the polyhedron defined by the linear relaxation of $X(\lambda)$, i.e. constraints (1-3) and (5).

$$0 \leq x_i^t \leq 1, \forall i = 1, ..., n, \forall t \in \{0, \ldots, T\}$$  \hspace{1cm} (5)

A method to obtain a lower bound $\lambda_{LP}$ is to solve repetitively linear programs via linear or binary search to find the smallest $\lambda$ for which $\tilde{X}(\lambda)$ is non-empty.
1 Introduction and problem presentation.

2 Time indexed linear programming formulation for the resource constrained modulo scheduling.

3 Lagrangian relaxation

4 Experimental results.

5 Conclusion and future work.
Lagrangian relaxation.

Let now the linear expression

\[
L_{\delta,\lambda}(x) = \sum_{s=1}^{m} \sum_{\tau=0}^{\lambda-1} \delta_{\tau,s} \left( \sum_{i=1}^{n} \sum_{t \mod \lambda = \tau} b_i^s x_i^t \right) - \sum_{s=1}^{m} \sum_{\tau=0}^{\lambda-1} \delta_{\tau,s} B_s
\]

and the LP

\[
\min \{ L_{\delta,\lambda}(x) : x \in \hat{X}(\lambda) \}
\] (6)

- \( \delta_{\tau,s} \geq 0 \), for \( \tau \in \{0, \ldots, \lambda - 1\} \), \( s \in \{1, \ldots, m\} \)
- \( \hat{X}(\lambda) \) is the polyhedron defined by constraints (1),(3) and (5).
- \( L_{\delta,\lambda}^* \) denote its optimal solution value.
Proposition 1.

\( \hat{X}(\lambda) \) is empty if for some \( \delta \geq 0 \), \( \hat{X}(\lambda) \) is empty or \( L^*_{\delta,\lambda} > 0 \).

A lower bound \( \lambda_{LR} \) for the minimal period can be obtained by finding the smallest \( \lambda \) value (through linear of binary search) for which the so-called lagrangian dual problem (LR) defined by:

\[
\max_{\delta \geq 0} \min \{ L_{\delta,\lambda}(x) : x \in \hat{X}(\lambda) \} \tag{7}
\]

has a non positive optimal solution (\( L^*_\lambda \leq 0 \)). Furthermore, (LR) has the integrity property since \( Co\{x \in \{0,1\} \cap \hat{X}(\lambda)\} = \hat{X}(\lambda) \), where \( Co(X) \) denotes the convex hull of set \( X \). It follows that \( \lambda_{LR} = \lambda_{LP} \).
Lagrangian subproblem.

Lagrangian subproblem is defined by

\[
\min \{ L_{\delta,\lambda}(x) : x \in \hat{X}(\lambda) \} \tag{8}
\]

if we define weights

\[
f_{i,t} = \sum_{s=1}^{m} b_{i}^{s} \delta_{t} \mod \lambda, s. \tag{9}
\]

The Lagrangian subproblem can be written as

\[
\min \left\{ \sum_{t=0}^{T-1} \sum_{i=1}^{n-1} f_{i,t}x_{i}^{t} : x \in \hat{X}(\lambda) \right\} \tag{10}
\]

A special case of the scheduling problem with start-time-dependent costs described in Möhring et al (2003).
Lagrangian subproblem

- **vertex**: $v_{i,t} \forall i \in \{1...n\}, t \in \{0...T\}.$
- **Assignment arcs**: $(v_{i,t}, v_{i,t+1}) \forall i \in \{1...n\}, t \in \{0...T\}.$
- **Temporals arcs**: $(v_{i,t}, v_{i,\theta/\lambda-\mu}) \forall (i, j) \in E, t \in \{0...T\}.$

Arc capacities $f_{i,t}$.

Minimum cut $(X, \overline{X})$.
Theorem 1 (Möhring et al 2003) describes the transformation of (8) to a minimum cut problem.

**Theorem**

*There is a one to one correspondence between minimum $a - b$ cuts $(X, \bar{X})$ of $G$ with exactly $n$ forward arcs and the optimal solutions $x$ of (8) by virtue of*

$x_{i,t} = 1$ if $(v_{i,t}, v_{i,t+1})$ is a forward arc of the cut $(X, \bar{X})$ and $x_{i,t} = 0$ otherwise. *The value $w(x)$ of an optimal solution of (8) equals the capacity $c(X, \bar{X})$ of any minimum cut $(X, \bar{X})$ of $G$.*
Solving the lagrangian dual problem.

Subgradient algorithm for \( \min \sum_{i=1}^{n-1} \sum_{t=0}^{T_{\text{max}}-1} f_{i,t} x_{i}^{t} : x \in \tilde{X}(\lambda) \)

1: \( \delta = (\delta_{\tau,s})^{s\in\{1,\ldots,m\}}_{\tau\in\{0,\ldots,\lambda-1\}} = 0. \)
2: Solve max flow and get min cut \((X, \bar{X}) \rightarrow L_{\delta,\lambda}(x)\).
3: \textbf{if} \( L_{\delta,\lambda}(x) > 0 \) \textbf{then}
   \hspace{1em} \text{STOP.}
4: \textbf{end if}
5: Compute subgradient \( g_{s,\tau}^{k} = \sum_{i=i}^{n-1} b_{i}^{s}(\sum_{t \text{ mod } \lambda=\tau} x_{i,t}^{k}) - B_{s} \).
6: Compute scalar step size \( \gamma^{k} = \gamma(L^{*} - L_{\delta}(x^{k}))/\|g^{k}\|^{2} \).
7: Update the vector of Lagrangian multipliers of the \( k+1 \)th iteration
   \( \rightarrow \delta^{k+1} = [\delta^{k} + \gamma^{k} g^{k}]^{+} \).
8: If no substantial improvement could be achieved within 15 iterations, STOP.
9: Jump to step 2.
Outline

1. Introduction and problem presentation.
2. Time indexed linear programming formulation for the resource constrained modulo scheduling.
3. Lagrangian relaxation
4. Experimental results.
5. Conclusion and future work.
Experimental results.

We implemented the Lagrangian relaxation for the RCMSP to compute a lower bound for the period $\lambda$ for instruction scheduling instances of the production compiler for the STMicroelectronics ST200 VLIW processor family.
### Table: Lower bound for $\lambda$

<table>
<thead>
<tr>
<th>Instances</th>
<th>$\lambda_0$</th>
<th>$\lambda_{LP,LR}$</th>
<th>CPU $LP$ (s)</th>
<th>CPU $LR$ (s)</th>
<th>CPU $LR^*$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>adpcm-st231.1</td>
<td>11</td>
<td>21</td>
<td>14.1</td>
<td>20.22</td>
<td>9.002</td>
</tr>
<tr>
<td>gsm-st231.1</td>
<td>24</td>
<td>24</td>
<td>35.85</td>
<td>22.03</td>
<td>18.37</td>
</tr>
<tr>
<td>gsm-st231.2</td>
<td>12</td>
<td>26</td>
<td>29.31</td>
<td>24.03</td>
<td>17.589</td>
</tr>
<tr>
<td>gsm-st231.5</td>
<td>4</td>
<td>11</td>
<td>13.27</td>
<td>11.25</td>
<td>4.02</td>
</tr>
<tr>
<td>gsm-st231.6</td>
<td>3</td>
<td>7</td>
<td>4.12</td>
<td>2.21</td>
<td>0.010</td>
</tr>
<tr>
<td>gsm-st231.7</td>
<td>4</td>
<td>11</td>
<td>5.09</td>
<td>4.03</td>
<td>1.006</td>
</tr>
<tr>
<td>gsm-st231.10</td>
<td>1</td>
<td>4</td>
<td>1.03</td>
<td>0.98</td>
<td>0.052</td>
</tr>
<tr>
<td>gsm-st231.20</td>
<td>3</td>
<td>6</td>
<td>1.29</td>
<td>1.12</td>
<td>0.044</td>
</tr>
<tr>
<td>gsm-st231.29</td>
<td>4</td>
<td>11</td>
<td>4.32</td>
<td>3.27</td>
<td>1.005</td>
</tr>
<tr>
<td>gsm-st231.40</td>
<td>2</td>
<td>10</td>
<td>13.25</td>
<td>12.27</td>
<td>4.10</td>
</tr>
</tbody>
</table>
Outline

1. Introduction and problem presentation.
2. Time indexed linear programming formulation for the resource constrained modulo scheduling.
3. Lagrangian relaxation
4. Experimental results.
5. Conclusion and future work.
Conclusion and future work

The results show that the proposed lagrangian relaxation method allows to obtain the same lower bound as the direct solving of the LP formulation with substantially lower computational requirements.

- Heuristic based on lagrangian relaxation
- Direct minimization of $\lambda$