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

Maria Alejandra Ayala[1], Christian Artigues[1], Bernat Gacias[1]

[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.

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

Industrial context : VLIW instruction schedling

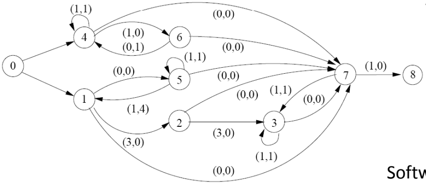
- The instruction scheduling problem is defined by a set of operations to schedule, a set of dependences between these operations, and a target processor micro-architecture. An operation is considered as an instance of an instruction in a program text.
- Modulo scheduling is a software pipelining framework, based only on 1-periodic cyclic schedules with integral period λ.
- 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.

Introduction and problem presentation

```
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, G128
BRF_7 b135, L?__0_8
```

loops



schedule instructions to end the program in minimum time

↓

Optimize loop performance

↓

Software pipeline
- Cyclic scheduling problem
- **Modulo scheduling**

Introduction and problem presentation

Resource constrained modulo scheduling problem : data

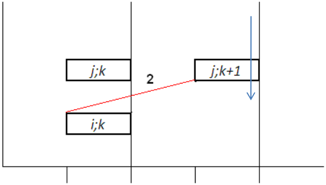
- A set of n generic operations with unit durations and a set of m resources.
- Each resource has a limited availability B_s.
- Each generic operation requires b_i^s units of each resource s.
- A set E of dependences is defined where for each $(i, j) \in E$, a latency θ_i^j and a distance ω_i^j are given.

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 $\sigma_i \equiv \sigma_i^0$

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 $\sigma_i \bmod \lambda$

Introduction and problem presentation

Resource constrained modulo scheduling problem

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

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

Remark 1 : strongly NP-hard

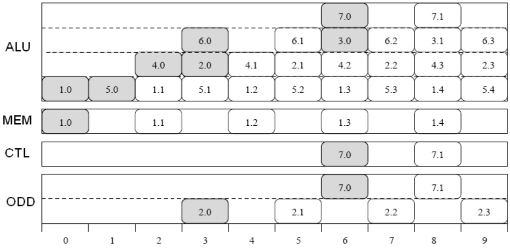
Remark 2 : For a fixed λ, find a feasible solution for a resource-constrained project scheduling problem with minimum and maximum time lags.

Introduction and problem presentation

Exemple.

Resource	Capacity
ALU	4
MEM	1
CTL	1
ODD	2

RESERVATION	ALU	MEM	CTL	ODD
ALU	1	0	0	0
ALUX	2	0	0	1
MUL	1	0	0	1
MULX	2	0	0	1
MEM	1	1	0	0
MEMX	2	1	0	1
CTL	1	0	1	1



$\lambda = 2$

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 λ. 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).

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

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

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

$$x_i^t \in \{0,1\}, \forall i = 1, ..., n, \forall t \in \{0, \ldots, T\} \tag{4}$$

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

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

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

A method to obtain a lower bound λ_{LP} is to solve repetitively linear programs via linear or binary search to find the smallest λ for which $\tilde{\mathcal{X}}(\lambda)$ is non-empty.

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.

Lagrangian relaxation.

Let now the linear expression

$$L_{\delta,\lambda}(x) = \sum_{s=1}^{m}\sum_{\tau=0}^{\lambda-1}\delta_{\tau,s}(\sum_{i=1}^{n-1}\sum_{t \bmod \lambda=\tau} b_i^s x_i^t) - \sum_{s=1}^{m}\sum_{\tau=0}^{\lambda-1}\delta_{\tau,s}B_s$$

and the LP

$$\min\{L_{\delta,\lambda}(x) : x \in \hat{\mathcal{X}}(\lambda)\} \tag{6}$$

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

Lagrangian relaxation.

Proposition 1.
$\tilde{\mathcal{X}}(\lambda)$ is empty if for some $\delta \geq 0$, $\hat{\mathcal{X}}(\lambda)$ is empty or $L^*_{\delta,\lambda} > 0$.
A lower bound λ_{LR} for the minimal period can be obtained by finding the smallest λ value (through linear of binary search) for which the so-called lagrangian dual problem (LR) defined by :

$$\max_{\delta \geq 0} \min\{L_{\delta,\lambda}(x) : x \in \hat{\mathcal{X}}(\lambda)\} \quad (7)$$

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

Lagrangian subproblem.

- Lagrangian subproblem is defined by

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

if we define weights

$$f_{i,t} = \sum_{s=1}^{m} b_i^s \delta_{t \bmod \lambda, s}. \tag{9}$$

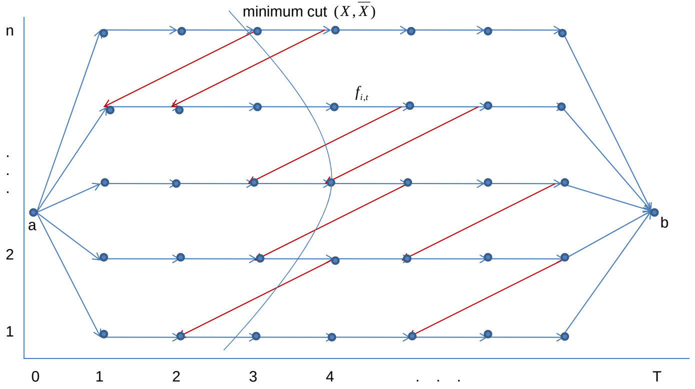
The Lagrangian subproblem can be written as

$$\min\{\sum_{i=1}^{n-1} \sum_{t=0}^{T-1} f_{i,t} x_i^t : x \in \hat{\mathcal{X}}(\lambda)\} \tag{10}$$

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

Lagrangian subproblem

- vertex $v_{i,t}$ $\forall i \in \{1 \ldots n\},\ t \in \{0 \ldots T\}$.
- Assignment arcs $(v_{i,t}, v_{i,t+1})$ $\forall i \in \{1 \ldots n\},\ t \in \{0 \ldots T\}$.
- Temporals arcs $(v_{i,t}, v_{i,t+(\theta_i^j - \lambda \omega_i^j)})$ $\forall (i,j) \in E,\ t \in \{0 \ldots T\}$.
- $f_{i,t}$ Arc capacities.



Lagrangian subproblem

Theorem 1 (Möhring et al 2003) describes the transformation of (8) to a minimum cut problem.

Theorem

There is a one to one correspondance between minimum $a - b$ cuts $(X, \bar{X})$ of G wth 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 $(X, \bar{X})ofG$

Solving the lagrangian dual poblem.

Subgradient algorithm for min $\sum_{i=1}^{n-1} \sum_{t=0}^{Tmax-1} f_{i,t} x_i^t : x \in \hat{\mathcal{X}}(\lambda)$

1: $\delta = (\delta_{\tau,s})_{\tau \in \{0,\dots,\lambda-1\}}^{s \in \{1,\dots,m\}} = 0.$
2: Solve max flow and get min cut $(X, \bar{X}) \rightarrow L_{\delta,\lambda}(x)$.
3: **if** $L_{\delta,\lambda}(x) > 0$ **then**
   STOP.
4: **end if**
5: Compute subgradient $g_{s,\tau}^k = \sum_{i=i}^{n-1} b_i^s (\sum_{t \bmod \lambda = \tau} x_{i,t}^k) - B_s$.
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 λ for instruction scheduling instances of the production compiler for the STMicroelectronics ST200 VLIW processor family.

Experimental results.

TABLE: Lower bound for λ

Instances	λ_0	$\lambda_{LP,LR}$	CPU LP (s)	CPU LR (s)	CPU LR^* (s)
adpcm-st231.1	11	21	14.1	20.22	9.002
gsm-st231.1	24	24	35.85	22.03	18.37
gsm-st231.2	12	26	29.31	24.03	17.589
gsm-st231.5	4	11	13.27	11.25	4.02
gsm-st231.6	3	7	4.12	2.21	0.010
gsm-st231.7	4	11	5.09	4.03	1.006
gsm-st231.10	1	4	1.03	0.98	0.052
gsm-st231.20	3	6	1.29	1.12	0.044
gsm-st231.29	4	11	4.32	3.27	1.005
gsm-st231.40	2	10	13.25	12.27	4.10

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 λ