Automatic Scheduling of Hypermedia Documents with Elastic Times

Celso C. Ribeiro

Joint work with M.T. Medina and L.F.G. Soares

Computer Science Department
Catholic University of Rio de Janeiro, Brazil

Workshop on
New Trends in Scheduling Parallel and Distributed Systems
Luminy, October 1-5, 2001

Contents:

Hypermedia systems should have some desirable features, such as:

- supporting a rich set of capabilities for each media segment (video, audio, text, pictures, animation, ...):
  - alternatives for different presentations
  - presentation duration flexibility
  - behavior changes during presentation
- allowing the explicit definition of temporal relationships among media segments with non-deterministic times
- supporting a temporal formatting algorithm that
  - considers non-determinism
  - allows on-the-fly corrections whenever unpredictable events (network delays, user interactions) occur

A presentation event is a media segment presentation

- d^{min}: minimum allowed presentation time
- d^{max}: maximum allowed presentation time
- d^{ideal}: duration with best presentation quality
- c: cost of shrinking or stretching the presentation duration

Scheduling algorithm: determination of starting times and optimal durations d^{opt} (which will be tried by the formatter) so as to ensure spatial and temporal consistency

- at compile time: build an execution plan to guide the scheduling of the object presentation
- at run time: online adjustments to handle unpredictable events (network delays, user interactions)

Difficult of performing fast rescheduling in run time led the systems that make time adjustments to make them only at compile time.

Other systems: Firefly, Isis, Madeus (linear programming, PERT, constraints, ...)

Problem: find (at compile time) the otimal starting times and durations of objects to be presented, to ensure spatial and temporal consistency of the presentation while respecting limits on shrinking and stretching the ideal duration of each object.

Objectives:

(1) minimize the total cost of shrinking and stretching objects;

(2) minimize the number of presentation events that will have their durations different than the ideal ones;

(3) spread the changes in object duration proportionally to their duration; and

(4) find the shortest and longest presentation durations.

Objectives (1), (3), and (4) can be dealt with at compile time by linear programming models.

Although conflicting, objectives

(1) minimize the total cost of shrinking and stretching objects and

(2) minimize the number of presentation events that will have their durations different than the ideal ones

are the most natural.

Possible strategy to handle both of them: select the best solution with respect to (2) among all those which optimize (1). The combinatorial nature of the minimization of the number of objects whose ideal duration is modified makes objective (2) the most difficult to be tackled by combinatorial algorithms.

- Extendable mixed 0-1 LIP formulation using objective (2)
- Fast and effective primal heuristic based on variable fixations
- Lagrangean relaxation and dual heuristic

Hypermedia document represented by a directed graph $G = [N, A]$:

- set $N_s = \{1, \ldots, n\}$ of synchronization points where presentation events either start or stop
- set $N = \{0\} \cup \{1, \ldots, n\} \cup \{n+1\}$ of nodes, where 0 and $n+1$ denote respectively the initial and the final nodes
- set A of arcs:
  - presentation events are arcs (i, j) with both extremities $i, j \in \{1, \ldots, n\}$ and durations d_{ij}^{min}, d_{ij}^{ideal}, and d_{ij}^{max}
  - dummy initial arcs $(0, j)$ with $j \in \{1, \ldots, n\}$ and durations $d_{0j}^{min} = 0$, $d_{0j}^{ideal} = 0$, and $d_{0j}^{max} = \infty$
  - dummy final arcs $(i, n+1)$ with $i \in \{1, \ldots, n\}$ and durations $d_{i,n+1}^{min} = 0$, $d_{i,n+1}^{ideal} = 0$, and $d_{i,n+1}^{max} = \infty$

Variables:

- continuous variable $t_i \geq 0$, for every $i \in N$
  - time in which synchronization takes place at $i \in N_s$
  - presentation starting time: $t_0 = 0$
  - end of presentation: t_{n+1}
- continuous variable $x_{ij} \in [d_{ij}^{min}, d_{ij}^{max}]$, for every $(i,j) \in A$
  - duration of each presentation (or dummy) event $(i,j) \in A$
- binary variable $y_{ij} \in \{0,1\}$, for every event $(i,j) \in A$
  - $y_{ij} = 1$, if the ideal duration of event (i,j) is modified
  - $y_{ij} = 0$, otherwise
- artificial costs c_{ij} associated to every arc $(i,j) \in A$
  - $c_{ij} = 1$, if (i,j) is a presentation event
  - $c_{ij} = 0$, otherwise (i.e., (i,j) is a dummy event)

<u>Formulation</u>: mixed 0-1 integer programming problem ASHD

$$
\begin{cases}
z^* = \min \displaystyle\sum_{(i,j)\in A} c_{ij} \cdot y_{ij} \\
\text{with:} \\
\begin{array}{rrlll}
t_i - t_j + x_{ij} & & = 0 & \forall (i,j) \in A & (1) \\
x_{ij} & -b_{ij} \cdot y_{ij} & \leq d_{ij}^{ideal} & \forall (i,j) \in A & (2) \\
-x_{ij} & +a_{ij} \cdot y_{ij} & \leq -d_{ij}^{ideal} & \forall (i,j) \in A & (3) \\
y_{ij} & & \in \{0,1\} & \forall (i,j) \in A & (4) \\
t_i & & \geq 0 & \forall i \in N \setminus \{0\} & \\
t_0 & & = 0 & &
\end{array}
\end{cases}
$$

$$
\begin{cases}
a_{ij} = d_{ij}^{min} - d_{ij}^{ideal} \\
b_{ij} = d_{ij}^{max} - d_{ij}^{ideal}
\end{cases}
$$

<u>Formulation</u>: mixed 0-1 integer programming problem ASHD

$$
\begin{cases}
z^* = \min \displaystyle\sum_{(i,j)\in A} c_{ij} \cdot y_{ij} \\
\text{with:} \\
\begin{array}{rrlll}
t_i - t_j + x_{ij} & & = 0 & \forall (i,j) \in A & (1) \\
x_{ij} & -b_{ij} \cdot y_{ij} & \leq d_{ij}^{ideal} & \forall (i,j) \in A & (2) \\
-x_{ij} & +a_{ij} \cdot y_{ij} & \leq -d_{ij}^{ideal} & \forall (i,j) \in A & (3) \\
y_{ij} & & \in \{0,1\} & \forall (i,j) \in A & (4) \\
t_i & & \geq 0 & \forall i \in N \setminus \{0\} & \\
t_0 & & = 0 &
\end{array}
\end{cases}
$$

$$
\begin{cases}
a_{ij} = d_{ij}^{min} - d_{ij}^{ideal} \\
b_{ij} = d_{ij}^{max} - d_{ij}^{ideal}
\end{cases}
$$

<u>Open question</u>: complexity of ASHD?

$\overline{\text{ASHD}}$ is the LP relaxation of ASHD: $y_{ij} \in \{0,1\} \Rightarrow 0 \leq y_{ij} \leq 1$

<u>Primal heuristic</u> `PH` for problem ASHD:

```
procedure PH
    Solve ASHD̄ and let (ȳ, x̄, t̄) be its optimal solution;
    while ∃(i,j) ∈ A : 0 < ȳ_ij < 1 do
        Fix to one in ASHD all variables y_ij such that ȳ_ij = 1;
        Fix to zero in ASHD all variables y_ij such that ȳ_ij = 0;
        Variable_selection: select and fix at either zero or
            one in ASHD a variable y_st with a fractional
            value 0 < ȳ_st < 1 in ASHD̄;
        Solve ASHD̄ and let (ȳ, x̄, t̄) be its optimal solution;
    end_while;
    return ȳ;
end PH;
```

```
procedure Variable_selection
    if ∃(i,j) ∈ A, i ≠ 0, j ≠ n+1 : 0 < ȳ_ij ≤ α then do
        (s,t) ← argmin_{(i,j)∈A,i≠0,j≠n+1} {ȳ_ij : 0 < ȳ_ij ≤ α};
        Fix y_st to zero in ASHD;
        if ASHD̄ is not feasible then fix y_st to one in ASHD;
    else if ∃(i,j) ∈ A, i ≠ 0, j ≠ n+1 : 1 − α ≤ ȳ_ij < 1 then do
            (s,t) ← argmax_{(i,j)∈A,i≠0,j≠n+1} {ȳ_ij : 1 − α ≤ ȳ_ij < 1};
            Fix y_st to one in ASHD;
        else do
            (s,t) ← argmin_{(i,j)∈A,i≠0,j≠n+1} min{ȳ_ij, 1 − ȳ_ij};
            if y_st ≤ 0.5 then do
                Fix y_st to zero in ASHD;
                if ASHD̄ is not feasible then fix y_st to one in ASHD;
            else fix y_st to one in ASHD;
    return (s,t);
end Variable_selection;
```

Primal heuristic

UltraSPARC station, CPLEX 5.0, 35 instances, $n = 20 \rightarrow 1000$

			Best value		Time (s)		Duality
Instance	Nodes	Arcs	PH	Exact	PH	Exact	gap (%)
a1	20	40	22	20	0.09	2.98	59.6
a2	20	80	62	62	0.29	25.07	47.8
a3	20	120	103	102	0.76	2328.75	56.4
c1	40	80	45	≤ 42	0.28	* 11535.65	57.3
c2	40	160	125	≤ 124	1.05	> 4 hs	52.3
d1	50	100	53	53	0.33	4246.48	51.5
d2	50	200	156	≤ 154	1.54	* 9190.92	52.3
e1	100	200	103	≤ 102	1.29	* 10087.15	53.1
e2	100	400	303	≤ 304	6.74	* 9715.30	54.5

* maximum size of the enumeration tree exceeded

Conclusions:

- PH found good solutions within small computation times for real-size instances.
- "Reasonable" computation times, except for very large instances (500 or more nodes).
- PH found the optimal solutions for two of seven instances for which the exact optimal is known and was off by exactly one unit for two others.
- Average and maximum gaps are respectively 54.5% and 59.6%.
- Very large duality gaps even for small instances for which PH found the optimal solution:

Conclusions:

- PH found good solutions within small computation times for real-size instances.
- "Reasonable" computation times, except for very large instances (500 or more nodes).
- PH found the optimal solutions for two of seven instances for which the exact optimal is known and was off by exactly one unit for two others.
- Average and maximum gaps are 54.5% and 59.6%.
- Very large duality gaps even for small instances for which PH found the optimal solution:
  - conjecture: solutions are quite good, but bounds are poor
  - search for polyhedral cuts or valid inequalities should be pursued if one wants to develop efficient exact methods

<u>Lagrangean dual</u> of problem ASHD:

- Associate non-negative dual multipliers λ_{ij}^{+} and λ_{ij}^{-} to constraints (2) and (3) for each arc $(i,j) \in A$:
- Lagrangean function: $L(x, y, t, \lambda^{+}, \lambda^{-}) =$
$$\sum_{(i,j)\in A} c_{ij} \cdot y_{ij} + \lambda_{ij}^{+}(x_{ij} - b_{ij} \cdot y_{ij} - d_{ij}^{ideal}) + \lambda_{ij}^{-}(-x_{ij} + a_{ij} \cdot y_{ij} + d_{ij}^{ideal})$$
- Dual function:

$$\begin{cases} w(\lambda^{+}, \lambda^{-}) = \min \quad L(x, y, t, \lambda^{+}, \lambda^{-}) \\ \text{with:} \\ t_i - t_j + x_{ij} = 0 \quad \forall (i,j) \in A \qquad (1) \\ y_{i,j} \in \{0,1\} \quad \forall (i,j) \in A \qquad (4) \\ d_{ij}^{min} \leq x_{ij} \leq d_{ij}^{max} \quad \forall (i,j) \in A \qquad (5) \\ t_i \geq 0 \quad \forall i \in N \setminus \{0\} \\ t_0 = 0. \end{cases}$$

Dual heuristic `DH`:

- Combines the primal heuristic and the solution of the dual problem by subgradient optimization.
- Uses the multipliers obtained along the solution of the dual problem to periodically generate reduced costs c'_{ij} associated with the 0-1 variables y_{ij}.
- Periodically applies the primal heuristic `PH`.
- Reduced costs c'_{ij} replacing the original costs c_{ij} in the primal heuristic are computed according with two different schemes:
  - $c'_{ij} = c_{ij} + a_{ij} \cdot \lambda^-_{ij} - b_{ij} \cdot \lambda^+_{ij}$ (Lagrangean multiplipliers)
  - $c'_{ij} = (1 - \hat{y}_{ij}) \cdot c_{ij}$ (primal aproximation)
- Stopping criteria: total number of iterations of the subgradient algorithm equals 1200 or the relative difference between two consecutive dual function values becomes less than 10^{-4}.

<u>Dual heuristic</u> `DH` for problem ASHD:

```
procedure DH
    Apply PH to compute a feasible solution ȳ;
    Set the upper bound z_best ← Σ_{(i,j)∈A} c_ij · ȳ_ij and y_best ← ȳ;
    while stopping criteria not attained do
        Perform a certain number of subgradient iterations;
        Recompute reduced costs c' using the current dual
            multipliers (λ⁺, λ⁻);
        Apply PH using c' to compute a new feasible solution ȳ;
        if Σ_{(i,j)∈A} c_ij · ȳ_ij < z_best
        then set z_best ← Σ_{(i,j)∈A} c_ij · ȳ_ij and y_best ← ȳ;
    end;
    return ȳ;
end DH;
```

Dual heuristic (1/2)

			DH		PH		Exact
Instance	Nodes	Arcs	z_{best}	s	z_{best}	s	z^*
a4	20	160	142	18.23	144	1.05	141
a5	20	200	178	24.88	180	1.72	176
b1	30	60	31	8.35	32	0.16	31
b3	30	180	154	21.87	155	1.46	$\leq$ 151
b4	30	240	209	31.88	210	2.64	$\leq$ 208
c3	40	240	202	32.83	203	2.64	$\leq$ 205
c4	40	320	280	49.20	283	4.71	$\leq$ 282
c5	40	400	362	67.19	363	7.54	$\leq$ 367
d2	50	200	154	28.63	156	1.54	$\leq$ 154
d4	50	400	352	69.73	354	7.07	$\leq$ 358
d5	50	500	451	95.40	453	11.62	$\leq$ 452

Dual heuristic (2/2)

			DH		PH		Exact
Instance	Nodes	Arcs	z_{best}	s	z_{best}	s	z^*
e3	100	600	505	142.63	506	16.69	$\leq$ 520
e4	100	800	699	219.34	702	30.47	$\leq$ 709
f1	500	1000	510	395.25	515	42.01	$\leq$ 520
f3	500	3000	2539	3602.95	2541	782.47	$\leq$ 2551
f5	500	5000	4514	5678.42	4519	1659.88	$\leq$ 4553
g1	1000	2000	1043	786.30	1045	208.05	$\leq$ 1177

Conclusions:

- The dual heuristic DH consistently improves upon the solutions found by the primal heuristic PH, finding better solutions for 17 out of the 35 test problems.
- However, the computational times observed for the dual heuristic DH are much larger.
- Need for better/faster construction algorithms and local search procedures (work-in-progress) based on variable complementation.

Concluding remarks and extensions:

- Primal heuristic: effective in finding good solutions in small processing times (real-size instances); quite promising not only in algorithmic terms, but also concerning its integration within existing document formatting systems.

Concluding remarks and extensions:

- Primal heuristic: effective in finding good solutions in small processing times (real-size instances); quite promising not only in algorithmic terms, but also concerning its integration within existing document formatting systems.
- Dual heuristic: better solutions, but at the cost of much larger computation times.

Concluding remarks and extensions:

- Primal heuristic: effective in finding good solutions in small processing times (real-size instances); quite promising not only in algorithmic terms, but also concerning its integration within existing document formatting systems.
- Dual heuristic: better solutions, but at the cost of much larger computation times.
- Establish the computational complexity of problem ASHD.

Concluding remarks and extensions:

- Primal heuristic: effective in finding good solutions in small processing times (real-size instances); quite promising not only in algorithmic terms, but also concerning its integration within existing document formatting systems.
- Dual heuristic: better solutions, but at the cost of much larger computation times.
- Establish the computational complexity of problem ASHD.
- Replace CPLEX by a customized LP algorithm to speedup the solution of the linear relaxation $\overline{\text{ASHD}}$.

Concluding remarks and extensions:

- Primal heuristic: effective in finding good solutions in small processing times (real-size instances); quite promising not only in algorithmic terms, but also concerning its integration within existing document formatting systems.
- Dual heuristic: better solutions, but at the cost of much larger computation times.
- Establish the computational complexity of problem ASHD.
- Replace CPLEX by a customized LP algorithm to speedup the solution of the linear relaxation $\overline{\text{ASHD}}$.
- Improved construction procedures and local search strategies.

Concluding remarks and extensions:

- Primal heuristic: effective in finding good solutions in small processing times (real-size instances); quite promising not only in algorithmic terms, but also concerning its integration within existing document formatting systems.
- Dual heuristic: better solutions, but at the cost of much larger computation times.
- Establish the computational complexity of problem ASHD.
- Replace CPLEX by a customized LP algorithm to speedup the solution of the linear relaxation $\overline{\text{ASHD}}$.
- Improved construction procedures and local search strategies.
- Integration of such heuristics and the bicriteria optimization approach within an existing document formatter to perform compile time scheduling and even run time adjustments.

Automatic Scheduling of Hypermedia Documents with Elastic Times

Maira T. Medina
Celso C. Ribeiro
Luiz Fernando G. Soares

Computer Science Department
Catholic University of Rio de Janeiro

Paper: //www-di.inf.puc-rio.br/∼celso/artigos/ashd.pdf

Slides: //www-di.inf.puc-rio.br/∼celso/talks/luminy.pdf