Force-Directed Scheduling (FDS) (Timed-Constrained Scheduling)

Basic idea — Place each operation into the appropriate control step, so as to balance operations of the same type across all possible control steps

If the number of operations of a particular type are balanced across all possible control steps,

the number of functional units of a particular type are also balanced across all possible control steps,

thus requiring the minimum number of functional units

---

compute the mobility range of each operation

while there are operators left to schedule {

for every unscheduled operation

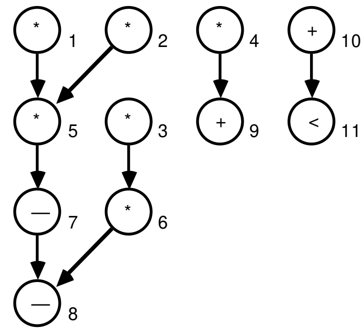
for every feasible control step
compute the cost of scheduling this operation into this control step

schedule the operation with the lowest cost into the appropriate control step

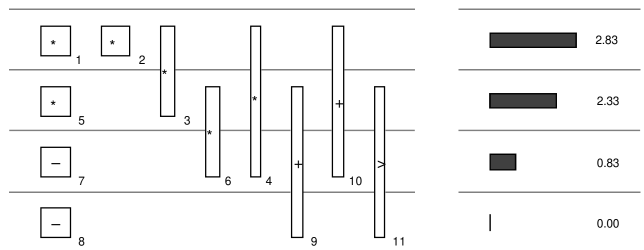
}

Diffeq — Force-Directed Scheduling Example

The variation on the differential equation example, in data flow graph form:



The probability of scheduling operations into control steps, and the operator cost for multiplications:



Force-Directed Scheduling (FDS) (cont.)

Construct a set of probability distribution graphs (i.e., bar graphs), constructing a separate distribution graph for each operation type

A bar graph for a particular operation type (e.g., multiplication) represents the expected operator cost (EOC) in each stae

The expected operator cost (EOC) for control step s_j for a particular type of operation k is given by:

$$EOC_{j,k} = c_k \sum_{i, s_j \in mrange(o_i)} p_j(o_i)$$

where o_i is an operation of type k, and c_k is the cost of the functional unit performing the operation of type k

In the example, the bar graph for the multiplication operations is constructed as follows:

$EOC_{1,multiplication}$ =

$c_{multiplication}\ (\ p_1(o_1) + p_1(o_2) + p_1(o_3) + p_1(o_4)\ ) =$

$c_{multiplication}\ (1.0 + 1.0 + 0.5 + 0.33) = 2.83\ c_{multiplication}$

$EOC_{2,multiplication}$ =

$c_{multiplication}\ (1.0 + 0.5 + 0.5 + 0.33) = 2.33\ c_{multiplication}$

$EOC_{3,multiplication}$ =

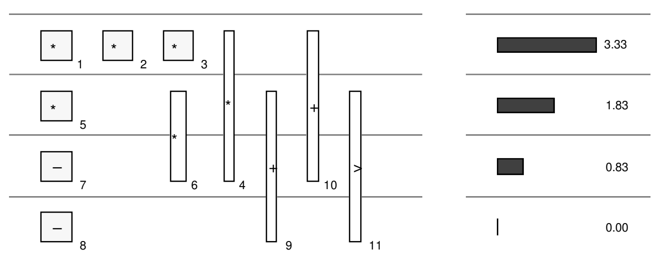
$c_{multiplication}\ (0.5 + 0.33) = 0.83\ c_{multiplication}$

$EOC_{4,multiplication}$ =

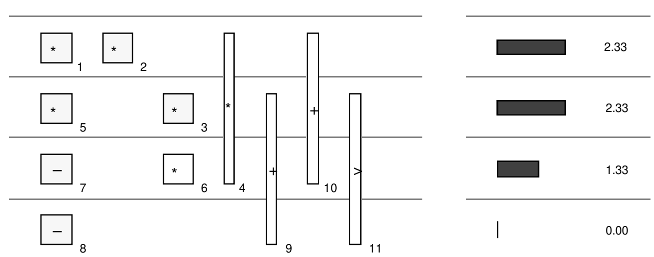
$c_{multiplication}\ (0.0) = 0.0$

Diffeq — Force-Directed Scheduling Example (cont.)

If operation 3 is scheduled into control step 1:



If operation 3 is scheduled into control step 2:



The assignment of operation 3 to control step 2 does a better job of balancing the expected functional unit costs for the multiplications

A Force-Directed Scheduling Algorithm

call ASAP(V);
call ALAP(V);

/* while there are operators left to schedule */
while there exists an o_i such that $E_i \neq L_i$ **do**

maxgain = – ∞;

/* for each unscheduled operation */
for o_i such that $E_i \neq L_i$ **do**

/* for control step in the operation's mobility range */
for each j, $E_i \leq j \leq L_i$ **do**

/* find the schedule that produces the maximum reduction in cost*/

S_{work} = schedule_op($S_{current}$, o_i , s_j);

adjust_distributions(S_{work} , o_i , s_j);

if cost($S_{current}$) – cost(S_{work}) > *maxgain* **then**

maxgain = cost($S_{current}$) – cost(S_{work});

BestOp = o_i ;

BestStep = s_j ;

endif

endfor

endfor

/* assign the appropriate operation to the appropriate control step so as to produce the maximum reduction in cost */

$S_{current}$ = schedule_op($S_{current}$, *BestOp*, *BestStep*);

adjust_distributions($S_{current}$, *BestOp*, *BestStep*);

endwhile

Force-Directed Scheduling — The "Force" Analogy

Consider each time step as a set of discrete locations on a line, and each operation as an ball that can rest at any of these locations (multiple balls can rest at the same location)

- Each ball can't really rest at *any* location — only those in its mobility range
- Imagine a spring from each ball to each location where it can potentially rest
- Imaging a spring from each ball to the balls that are its predecessor and successor
- When all these springs reach equilibrium, the best schedule is achieved
- In each iteration of the algorithm, for each operation and for each control step in its mobility range, we will calculate a set of *forces* based on the distribution graphs
- Then we will select the operation / control step assignment with the lowest (or most negative) force, which represents the best balancing of the operations among the control steps

Force-Directed Scheduling — Computing the Forces

Recall that the expected operator cost (EOC) for control step s_j for a particular type of operation k is given by:

$$EOC_{j,k} = c_k \sum_{i, s_j \in mrange(o_i)} p_j(o_i)$$

where o_i is an operation of type k, and c_k is the cost of the functional unit performing the operation of type k

First, calculate the force associated with every feasible control step assigment of each operation

For an operation o_i with a mobility range $E_i \leq j \leq L_i$, the force associated with its assignment to control step j is given by:

$$Force_{i,j,k} = EOC_{j,k} - \sum_{m=E_i}^{L_i} \frac{EOC_{m,k}}{L_i - E_i + 1}$$

In English, the force associated with the tentative assignment of operation o_i to control step s_j is equal to the difference between:

- the expected operator cost (EOC) for that control step
- minus the average EOC over all control steps in the operation's mobility range

Thus the force is directly proportional to the additional hardware required to schedule an operation into a particular control step

Diffeq — Force-Directed Scheduling Example (cont.)

Continuing the diffeq example,

$Force_{3,1,multiplication}$ = 2.83 – (2.83 + 2.33) / 2 = +0.25

$Force_{3,2,multiplication}$ = 2.33 – (2.83 + 2.33) / 2 = –0.25

Continuing the example, considering the assignment of operation 3 to either control step 1 or control step 2:

$TotalForce_{3,1,multiplication}$ =

$Force_{3,1,multiplication}$ = (*operation 6 not affected*)

2.83 – (2.83 + 2.33) / 2 = +0.25

$TotalForce_{3,2,multiplication}$ =

$Force_{3,2,multiplication}$ + $Force_{6,3,multiplication}$ =

2.33 – (2.83 + 2.33) / 2 + 0.83 – (2.33 + 0.83) / 2 =

–0.25 + (–0.75) = –1.0