

# Forced-Directed List-Scheduling

Vytaras Juraska  
Electronics Engineering (6<sup>th</sup> Semester)  
Hamm-Lippstadt Hochschule  
Lippstadt, Germany  
vytaras.juraska@stud.hshl.de

**Abstract**—an explanation and a deep dive on the working concept, the history of development and the usage of a very specific extension of an already existent algorithm. Understanding and analyzing the relevancy and connection when applying specified algorithm to the current time and age technology standards.

## I. INTRODUCTION

This specific scheduling algorithm is essentially an extension of an already existent scheduling algorithm - Forced-Directed scheduling. It adds an extension of List-scheduling to the mentioned algorithm. It was created to have the most optimal solution for a scheduling problem, where both resources and the latency of the operations is constrained.

**NOTE** Throughout the explanation of Force-Direct and Force-Direct List-Scheduling algorithms, the references ([1] and [2]) are using this specific sequencing diagram (Figure 1).

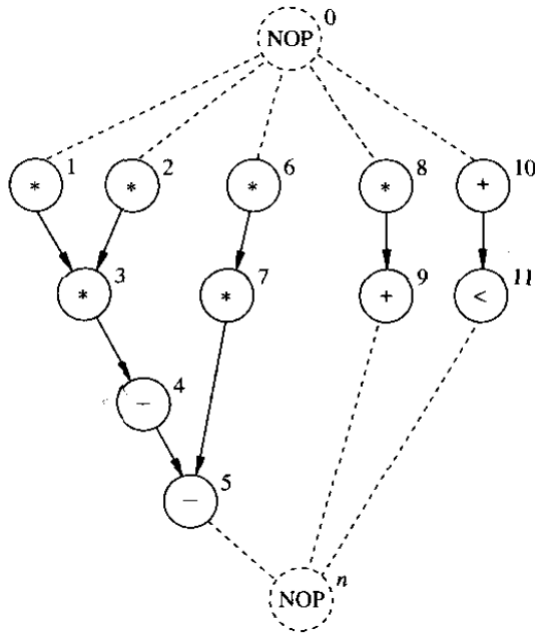


Fig. 1. Initial Sequencing Graph

### A. Force-Direct Relevant Part

Force-Directed scheduling algorithm was created to find the most optimal and sufficient approach on how to solve resource

and latency constrained (limited) scheduling problems. Visually it is, in a sense, where each of it's components have a force, which is pushing each component away from each other, creating a natural distance between each other, with a corresponding length (Figure 2).

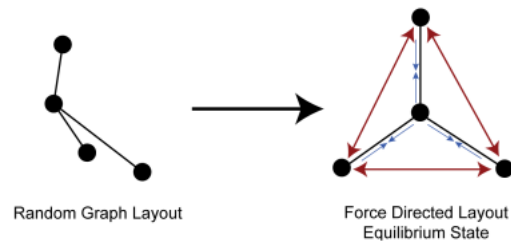


Fig. 2. Force-Directed visual layout example

Considering how the Forced-Directed List-Scheduling works, we have to understand the basics of Forced-Directed algorithm. The time constrain has two variables, the earliest time variable and the latest time variable when the operation is expected to be executed, both of the variable form a, so called, time frame. This concept has been taken from and can be calculated with ASAP (As Soon As Possible) and ALAP (As Late As Possible) algorithms.

Probability of operations is defined as follows: outside the time frame it is zero, inside the time frame, it is the reciprocal of the frame width. Hence the larger the width of the time frame, the lower the probability is for scheduled operation to be successfully executed in the corresponding time step. So in this specific algorithm, it is required to not only have a limited time frame, but also the most optimal solution is to keep the time frame as narrow as possible.

**NOTE** Further explanation would lead to the first example of Force-Direct Scheduling calculations, the example refers to Initial Sequencing Graph. It's time distribution graph for the components in the exercise (Figure 3).

### B. List-Scheduling Relevant Part

Relatively simple explanation of List-Scheduling is giving some sort of property to each of the component and defining a position of the component in the list corresponding to it properties unit.

### C. Force-Direct List-Scheduling

As mentioned in the Force-Direct algorithm introduction, this specific solution is an addition of List-Scheduling to pre-

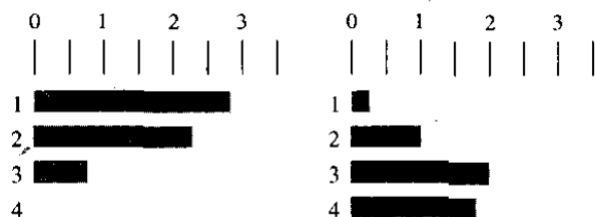


Fig. 3. Distribution graph for multiplexer and ALU

viously mentioned algorithm. The conditions of the problem are similar to Force-Direct algorithm's, a limited and minimal time frame of the operation, caused by the fixed hardware constrains.

Force-Direct used to figure out the priority of each process

## II. HISTORY OF DEVELOPMENT

A detailed timeline of how this method came to be developed

## III. USAGE AND APPLIANCE

Where is it applied in real situations, where is it used in

**NOTE** In the [2] reference, and in the other various resources this specific graph was displayed as a common comparison figure for FDLS. Referenced description - Fifth-Order Digital Elliptical Wave Filter (Figure 4) (sounds like a tongue twister).

Algor <sup>m</sup>	17	18	19	21																				
FDS	<table border="1"> <tr><td>+</td><td>+</td><td>+</td></tr> <tr><td>x</td><td>x</td><td>x</td></tr> </table>	+	+	+	x	x	x	<table border="1"> <tr><td>+</td><td>+</td><td>+</td></tr> <tr><td>x</td><td>x</td><td></td></tr> </table>	+	+	+	x	x		<table border="1"> <tr><td>+</td><td>+</td></tr> <tr><td>x</td><td>x</td></tr> </table>	+	+	x	x	<table border="1"> <tr><td>+</td><td>+</td></tr> <tr><td>x</td><td></td></tr> </table>	+	+	x	
+	+	+																						
x	x	x																						
+	+	+																						
x	x																							
+	+																							
x	x																							
+	+																							
x																								
FDLS	<table border="1"> <tr><td>+</td><td>+</td><td>+</td></tr> <tr><td>x</td><td>x</td><td>x</td></tr> </table>	+	+	+	x	x	x	<table border="1"> <tr><td>+</td><td>+</td></tr> <tr><td>x</td><td>x</td></tr> </table>	+	+	x	x		<table border="1"> <tr><td>+</td><td>+</td></tr> <tr><td>x</td><td></td></tr> </table>	+	+	x							
+	+	+																						
x	x	x																						
+	+																							
x	x																							
+	+																							
x																								
ASAP	<table border="1"> <tr><td>+</td><td>+</td><td>+</td><td>+</td></tr> <tr><td>x</td><td>x</td><td>x</td><td>x</td></tr> </table>	+	+	+	+	x	x	x	x															
+	+	+	+																					
x	x	x	x																					
LS			<table border="1"> <tr><td>+</td><td>+</td></tr> <tr><td>x</td><td>x</td></tr> </table>	+	+	x	x																	
+	+																							
x	x																							
FDS, FDLS	<table border="1"> <tr><td>+</td><td>+</td><td>+</td></tr> <tr><td>⊗</td><td>⊗</td><td></td></tr> </table>	+	+	+	⊗	⊗		<table border="1"> <tr><td>+</td><td>+</td><td>+</td></tr> <tr><td>⊗</td><td></td><td></td></tr> </table>	+	+	+	⊗			<table border="1"> <tr><td>+</td><td>+</td></tr> <tr><td>⊗</td><td></td></tr> </table>	+	+	⊗						
+	+	+																						
⊗	⊗																							
+	+	+																						
⊗																								
+	+																							
⊗																								

+ Adder      x Multiplier      ⊗ Pipelined multiplier

Fig. 4.

## IV. ADVANTAGES

Any positive advantages related to this method, why is it useful to use this method

## V. DISADVANTAGES

Any disadvantages, which might lead to certain issues, where this method can not be applied

## VI. EVALUATION

Personal opinion, is it useful, is there future for this method?

## VII. CONCLUSION

Final thoughts, concluding the topic

## REFERENCES

- [1] Giovanni De Micheli "Synthesis and Optimization of Digital Circuits"
- [2] P. G. Paulin and J. P. Knight, "Force-directed scheduling for the behavioral synthesis of ASICs,"