Solving Stochastic Parallel Machine Scheduling using a Metaheuristic Approach with Efficient Robustness Estimation.

Utrecht University

M. S. Hessey
3496724

Supervisor: J. M. Van Den Akker
J. A. Hoogeveen

Computing Science
University Utrecht

Proposal for MSc Thesis

2018

Contents

Chapter 1

Robustness Measure Evaluation

Notation	Definition	Description
FS_j	$\min_{ß \in \sigma_j}\{s_i - (s_j + p_j)\}$	Free slack of job j.
BFS_j^γ	1 if $FS_j \geq \gamma \cdot p_j$, 0 otherwise.	Binary value indicating if the free slack exceeds a percentage of the processing time.
UFS_j^γ	$\min\{\gamma \cdot p_j, FS_j\}$	Free Slack with upperbound.
TS_j	$s_j(ESS) - s_j(LSS)$	The Total slack of job j is the difference in starting times of j in the earliest / latest start schedules.
BFS_j^γ	1 if $TS_j \geq \gamma \cdot p_j$, 0 otherwise.	Binary value indicating if the total slack exceeds a percentage of the processing time.
UFS_j^γ	$\min\{\gamma \cdot p_j, TS_j\}$	Total Slack with upperbound.
SDR_j	TS_1/p_1	Slack Duration Ratio

Table 1.1: Robustness measure list. For symbols, refer to A.1

1.1 Methodology

We compare the Robustness Measures in table 1.1 to the following performance measures:

- The makespan (determined by 1000 simulation runs)
- The sum of all linear start delays: $\sum_i \mathbf{s_i} - s_i$. Note that jobs are never started before their planned start time.
- The start time punctuality: the percentage of jobs started before $1.05 \cdot s_j$.

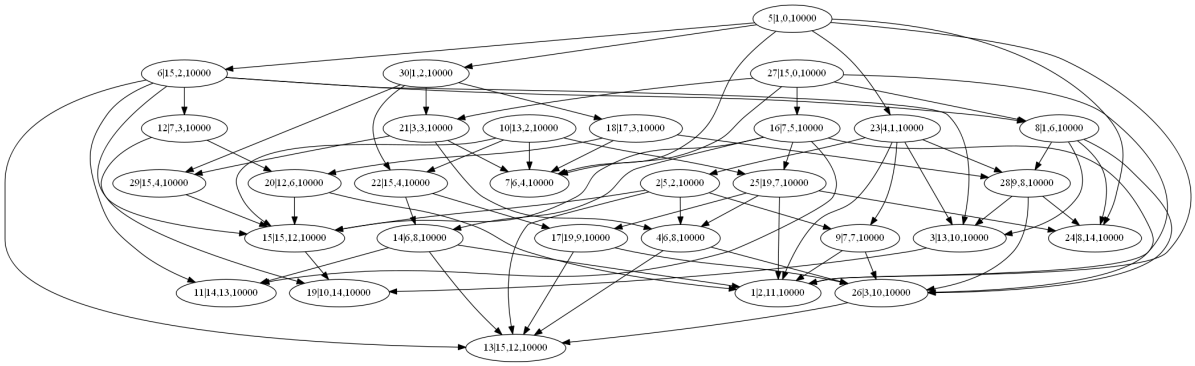

Figure 1.1: The precedence graph in the 30J-75R-8M data file. Each node represents a job j and gives its processing time, release date and due date as: $j|p_j, r_j, tododefj$

1.2 Data

We consider RMs (all RMs). and compare their accuracy in predicting performance measures PM (all PMs). We do this for several input problems (all input problems). We use the same problem instances used in [PAH]. These are titled todo format $nJ-rR-mM$ where n is the number of jobs, r the number of relations and m the number of machines. For each job, the processing time is a natural number between 1 and 20 and the release date is a natural number between 0 and $\lfloor n/2 \rfloor$. All released jobs are assumed to be able to start as soon as all predecessors. That is, we have 0-lag finish start precedence relations. Precedence relations are randomly selected, such that no cycles occur. Furthermore, release dates form a partial order in the graph. That is, if job j is a descendent of job i, then $r_j > r_i$. As an example, a graphical representation of $30J - 75R - 8M$ is shown in figure 1.1.

It is worth noting that the maximum depth is 5, which is more than n/m. As the maximum depth increases, the maximum depth path is more likely to determine the makespan. Therefore, solutions become more likely to be interchangeable. This makes local search difficult in such cases. However, in this case, processing times along the maximum depth path are short.

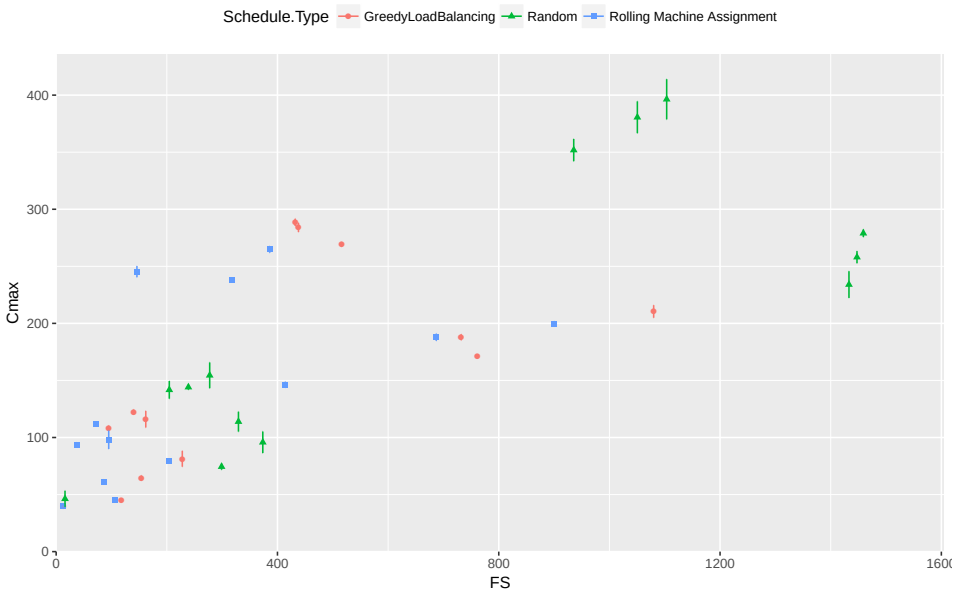

Figure 1.2: Free slack vs makespan.

1.3 Results

1.3.1 Free Slack based measures

Figures 1.2, 1.3 and 1.4 show that the is some correlation between free slack measures and the makespan. All three measures exhibit similar behaviour.

In general, less free slack results in a smaller makespan, as is to be expected. Schedules with a lot of free slack (such as those built by random assignment) are in general inefficient. Interestingly free slack does not seem to reduce variance in makespan.

Linear start delay (figures 1.5, 1.6 and 1.7) seems almost uncorrelated with the free slack measures.

Start Punctuality (figures 1.8, 1.9 and 1.10) does increase as the measures first increase. However at some point there is so much slack that adding extra does not improve punctuality (as it is already almost 100%).

Perhaps these schedules are too poor for Free Slack to be a good measure. Indeed, slack is only a good thing if introduced in an otherwise tight schedule. Adding slack to a bad schedule just makes it worse. As a next step, I will investigate the effect on better schedules.

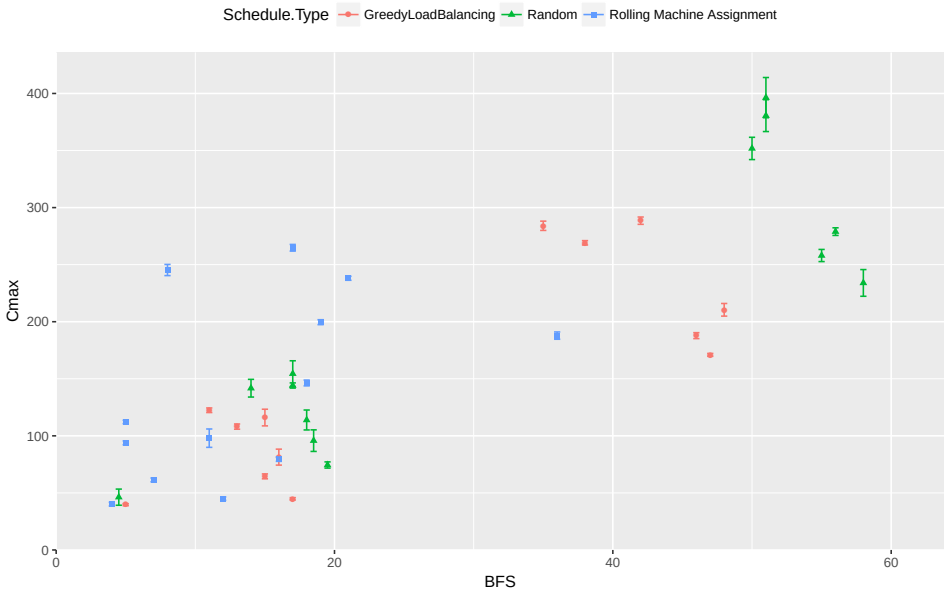

Figure 1.3: Binary (0.25) free slack vs makespan.

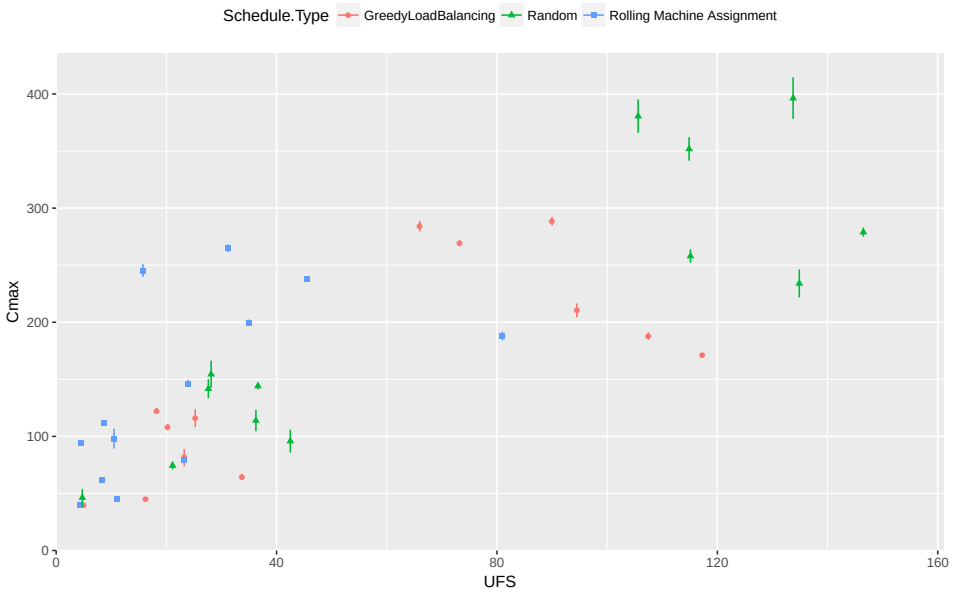

Figure 1.4: Upper bound (0.25) free slack vs makespan.

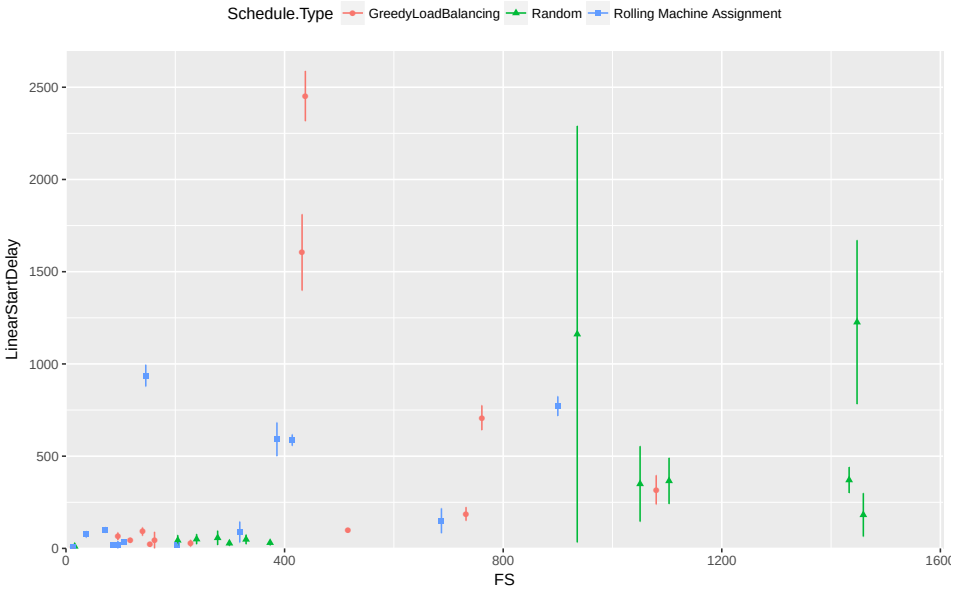

Figure 1.5: Free slack vs linear start delay

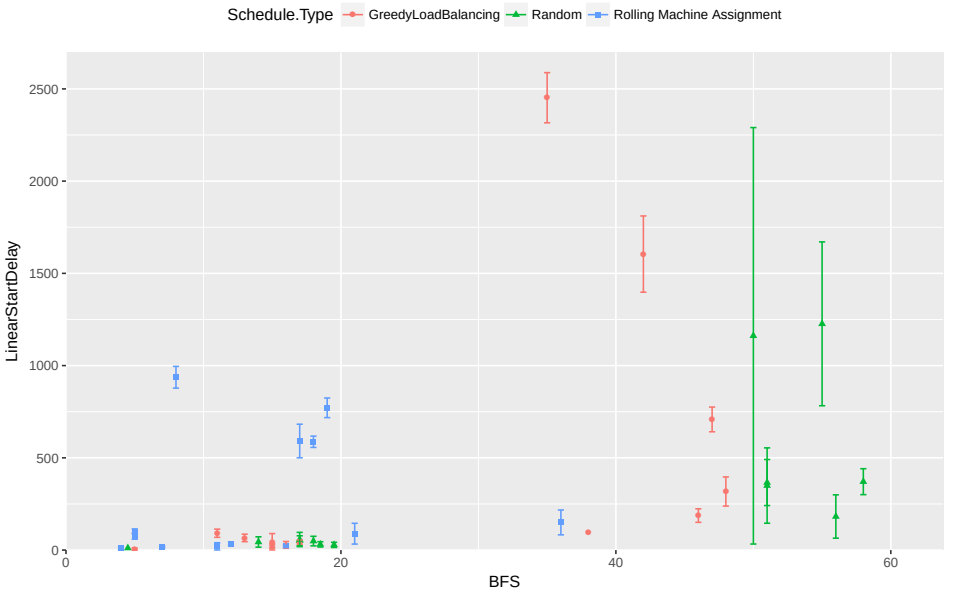

Figure 1.6: Binary (0.25) free slack vs linear start delay

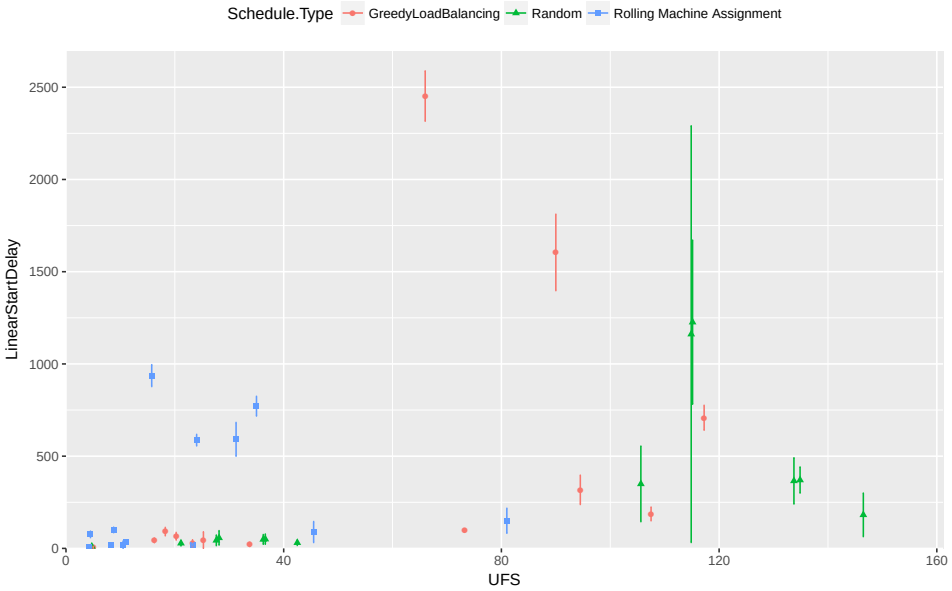

Figure 1.7: Upper bound (0.25) Free slack vs linear start delay

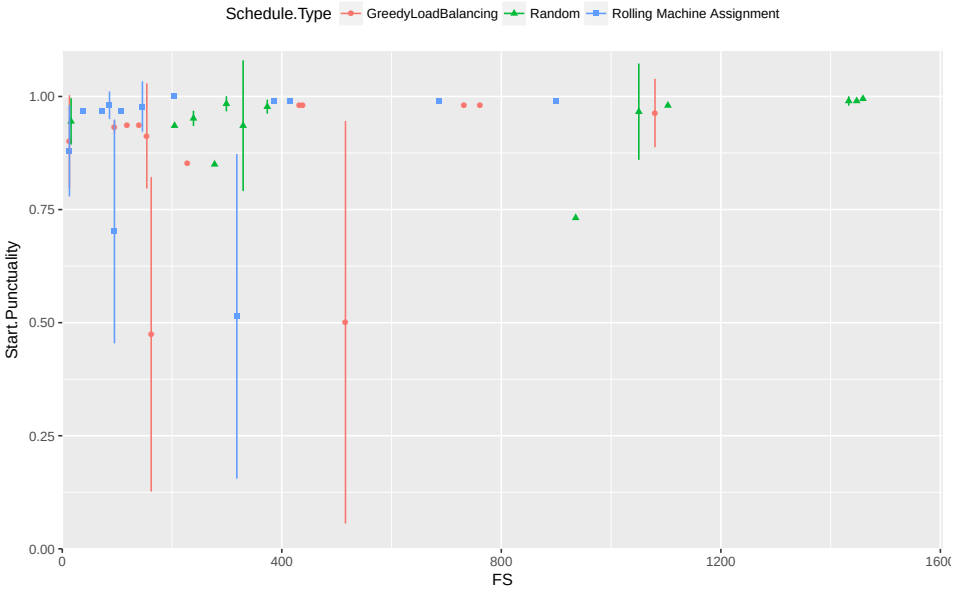

Figure 1.8: Free slack vs start punctuality

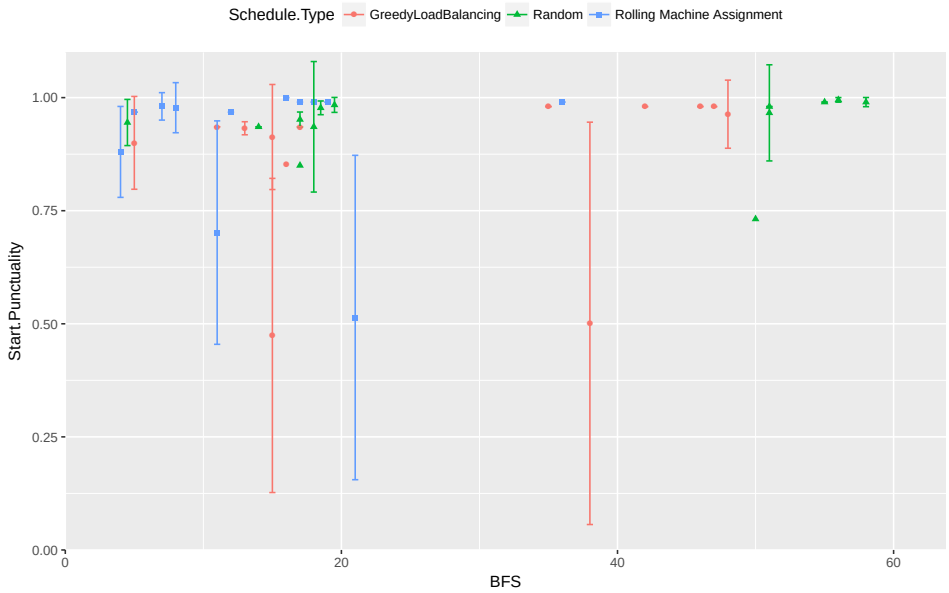

Figure 1.9: Binary (0.25) free slack vs start punctuality

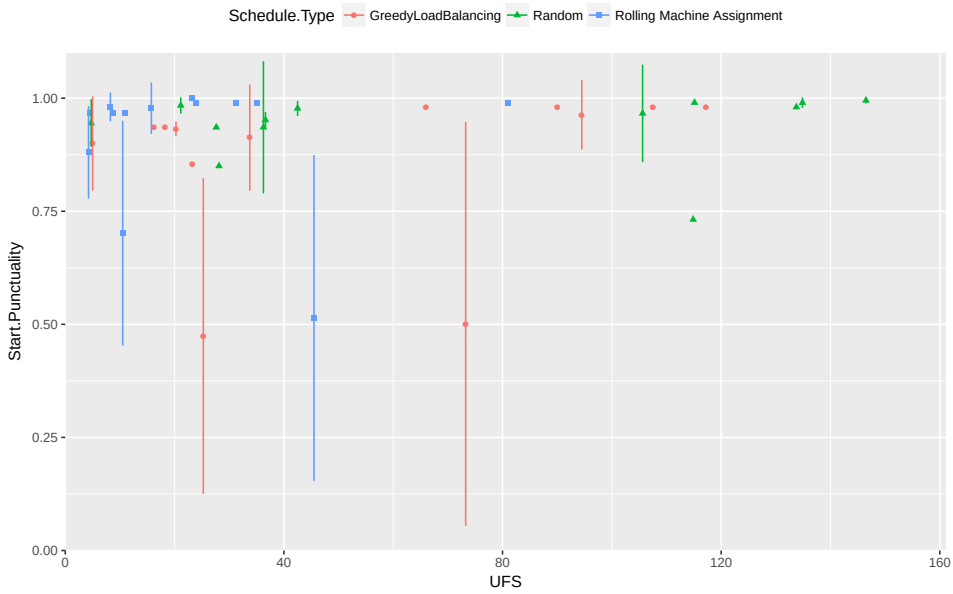

Figure 1.10: Upper bound (0.25) Free slack vs start punctuality

Bibliography

[PAH] Guido Passage, Marjan van den Akker, and Han Hoogeveen. “Improving the performance of local search for stochastic parallel machine scheduling by estimating the makespan”. MA thesis.

Appendix A

Notation

An overview of the symbols used in the 3 field notation:

- P_m Parallel machine scheduling on m machines
- $C(\sigma)_j$ The completion time of job j in schedule σ
- r_j The release date of job j: the time at which it becomes available for processing
- d_j The due date of job j: the time by which we would prefer to have job j finished (a soft constraint).
- $\bar{d_j}$ The deadline of job j: the time by which job j must be finished (a hard constraint).
- $C_{max}(\sigma)$ The maximum completion time in schedule σ: The smallest time by which all jobs are finished. Also known as the *makespan* $C_{max}(\sigma) = \max_j\{C(\sigma)_j\}$
- $L_{max}(\sigma)$ The maximum lateness in schedule σ: The largest difference between due date and completion time of a job. $L_{max}(\sigma) = \max_j\{C(\sigma)_j d_j\}$
- prec Denotes that precedence relations between jobs exist. This work discusses only precedence constraints where a job may start as soon as all it's predecessors are completed, known as *0-lag finish-start* precedence contraints.
- $\mathbf{p}_j$ Denotes that the processing times are stochastic.

An overview of abbreviations used:

- RM: Robustness measure
- PMS: Parallel machine scheduling

Symbol	Meaning
Job properties	
p_j	Mean processing time of job
$\mathbf{p_j}$	Realised processing time of job j
r_j	Release date of job
Schedule Properties	Sometimes followed by (S) to indicate the schedule.
σ_j	Direct successors of job j
π_j	Direct predecessors of job j
${\sigma_j}^*$	Transitive successors of job j
${\pi_j}^*$	Transitive predecessors of job j
$s_j(ESS)$	Start time of job j in the earliest start schedule
$s_j(LSS)$	Start time of job j in the latest start schedule
s_j	Planned start time of job j in a schedule.
$\mathbf{s_j}$	Realised start time of job j in a schedule.

Table A.1: List of symbols frequently used in formulae.

- SPMS: Stochastic parallel machine scheduling
- RCPSP: Resource constrained project scheduling problem
- SRCPSP: Stochastic resource constrained project scheduling problem
- LS: Local Search