

University of Technology Sydney

Doctor of Philosophy

Algorithms for scheduling without preemptions

by
Julia Memar

Thesis supervisors: A/Prof Dr Yakov Zinder, Dr Hanyu Gu

February 2022

School of Mathematical and Physical Sciences

Declaration

I, Julia Memar, declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Mathematical and Physical Sciences, Faculty of Science at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signed: Production Note:
 Signature removed prior to publication.

Date: 10 February 2022

Algorithms for scheduling without preemptions

by

Julia Memar

Submitted to the School of Mathematical and Physical Sciences
on February 10, 2022, in fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis is concerned with algorithms for scheduling without preemptions and it contributes to research as follows.

The new area of research, which has gained attention only in the last 15 years, is concerned with flow shop models where the storage requirement varies from job to job and a job occupies the storage continuously from the start of its first operation till the completion of its last operation. This thesis contributes to research by developing a new approach of constructing feasible solutions for such flow shop problems with job-dependent storage. This approach utilises Lagrangian relaxation and decomposition - the techniques that have never been used before for such flow shop problems. In this thesis, several Lagrangian relaxation and decomposition-based heuristics are developed for NP -hard flow-shop problems with job-dependent storage and the effectiveness of these heuristics is demonstrated by the results of computational experiments.

In this thesis, a new discrete optimisation procedure is introduced. This optimisation procedure can be viewed as an alternative exact method to a branch and bound algorithm for a class of discrete optimisation problems with certain properties. This class includes several NP -hard scheduling problems. This discrete optimisation procedure is an iterative algorithm, that searches for a feasible solution with the objective value of the current lower bound or determines that such a solution does not exist. Various methods of how this search can be carried out are investigated, and these methods are compared computationally in application to a scheduling problem.

The worst-case analysis of a polynomial-time approximation algorithm for a maximum lateness scheduling problem with parallel identical machines, arbitrary processing times and arbitrary precedence constraints is provided. The algorithm is a modification of the Brucker-Garey-Johnson algorithm originally developed as an exact algorithm for the case of the problem with unit execution time tasks and precedence constraints represented by an in-tree. For the case when the largest processing time does not exceed the number of machines, a worst-case performance guarantee which is tight for arbitrary large instances of the considered maximum lateness problem has been obtained. It is shown that, if the largest processing time is greater than the num-

ber of machines, then the worst-case performance guarantee for the list algorithm, obtained by Hall and Shmoys, is tight.

Thesis supervisors: Associate Professor Dr Yakov Zinder, Dr Hanyu Gu

Acknowledgments

- I would like to express my gratitude to my supervisors, Associate Professor Dr. Yakov Zinder and Dr. Hanyu Gu, for their guidance, for countless enlightening talks and hundreds of hours spent to share with me their wisdom.
- I would like to thank my esteemed co-authors, Professor Dr. Alexander Kononov, Sobolev Institute of Mathematics, Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Russia, and Dr. Gaurav Singh, Research and Innovation Manager, BHP, Perth, for sharing their knowledge and expertise with me.
- A huge thank you to my family - for their eternal patience and bearing with me and this thesis for all these years.
- Finally, I would like to dedicate this thesis to my mother, Lubov Kuzminichna Bazanova, who has always believed in me.

This doctoral thesis has been examined by a Committee of the School of Mathematical and Physical Sciences, Faculty of Sciences, UTS as follows:

Contents

1	Introduction and motivation for research	19
1.1	Thesis Organization	22
2	Background and literature overview	25
2.1	Scheduling theory: classification of problems	25
2.2	Flow shop problems and algorithms	27
2.3	Lagrangian Relaxation	33
2.4	Scheduling problems on parallel machines	34
3	Lagrangian relaxation and decomposition-based algorithms for flow shops with job-dependent buffer requirements	47
3.1	Introduction	48
3.2	Two-stage flow shop with storage and objective to minimise total weighted completion time	51
3.2.1	Problem Description	51
3.2.2	Lagrangian relaxation	52
3.2.3	Lagrangian heuristics	54
3.2.4	Computational experiments	63
3.2.5	Conclusion	72
3.3	Two-stage flow shop with storage and objective to minimise maximum completion time	76
3.3.1	Problem Description	76
3.3.2	Lagrangian relaxation-based heuristic	77

3.3.3	Bin-packing heuristic	81
3.3.4	Barrier heuristic	83
3.3.5	Lower Bound	86
3.3.6	Computational Experiments	87
3.3.7	Conclusion	93
3.4	Two-stage hybrid flow shop with a storage and objective to minimise total weighted tardiness	96
3.4.1	Problem Description	96
3.4.2	Choice of the Planning Horizon	97
3.4.3	Integer Programming Formulation	99
3.4.4	Lagrangian relaxation and decomposition	102
3.4.5	Lagrangian relaxation-based optimisation procedure	104
3.4.6	Scaling	105
3.4.7	Permutation heuristic	107
3.4.8	Computational experiments	108
3.4.9	Conclusion	121

4 Discrete optimisation with polynomially detectable boundaries and restricted level sets 123

4.1	Introduction	125
4.2	Level sets	127
4.3	Examples of the problems with the considered properties	128
4.3.1	<i>Property 1</i> : scheduling on dedicated machines - the boundary of the feasible region	128
4.3.2	<i>Property 2</i> and <i>Property 3</i> : level sets of $\max_{1 \leq j \leq n} \varphi_j(x_j)$	132
4.3.3	<i>Property 2</i> and <i>Property 3</i> in multi-objective optimisation	133
4.4	Description of the Discrete Optimisation Procedure	136
4.4.1	Introduction	136
4.4.2	Descending method	137
4.4.3	Ascending method	141

4.4.4	Descending-ascending method	145
4.5	Application	146
4.5.1	Description of the problem and preliminaries	146
4.5.2	Descending method	150
4.5.3	Ascending method	154
4.5.4	Descending-ascending method	160
4.6	Computational Experiments	162
4.7	Conclusion	165
5	The worst-case analysis for an approximation algorithm for a maximum lateness problem	167
5.1	Introduction and description of the problem	167
5.2	BGJ-algorithm	169
5.3	The structure of a schedule	171
5.4	Lower Bounds on the Optimal Value of $G(\sigma)$	174
5.5	Worst-Case Performance Guarantee	176
5.6	The case of the problem when $p_{max} > m$	182
5.7	Conclusion	192
6	Conclusion and further research	193

List of Figures

3-1	Upper and Lower bounds change with iterations: 25 jobs	72
3-2	Upper and Lower bounds change with iterations: 50 jobs	73
3-3	Relative error for 25 job instances, in %	73
3-4	Relative error for for 50 job instances, in %	74
3-5	Average CPU time for instance with different number of jobs	74
3-6	Relative Error for 25 jobs instances	91
3-7	Relative Error for 50 jobs instances	92
3-8	Relative Error for 100 jobs instances	93
3-9	Average time per instance	94
3-10	LB^{buffer} vs $LB^{Johnson}$	94
3-11	Relative Error	112
3-12	Step τ : comparison of scaling and no scaling options	115
3-13	Convergence of the algorithms: upper and lower bounds	115
3-14	Relative Error - larger instances, Ω_5 buffer size	119
3-15	Average time per instance	120
4-1	Time per group of tasks	165
5-1	Set of tasks: $p_{max} \leq m$	183
5-2	Schedules: $p_{max} \leq m$	184
5-3	Set of tasks: $p_{max} > m$	190
5-4	Schedules: $p_{max} > m$	191
5-5	$G(\sigma) \rightarrow 2G(\sigma^*)$	192

List of Tables

3.1	5 jobs instances, buffer size $\Omega_{1.0}$	66
3.2	5 jobs instances, buffer size $\Omega_{1.5}$	66
3.3	5 jobs instances, buffer size $\Omega_{2.0}$	67
3.4	10 jobs instances, buffer size $\Omega_{1.0}$	67
3.5	10 jobs instances, buffer size $\Omega_{1.5}$	67
3.6	10 jobs instances, buffer size $\Omega_{2.0}$	68
3.7	25 jobs instances, buffer size $\Omega_{1.0}$	68
3.8	25 jobs instances, buffer size $\Omega_{1.5}$	69
3.9	25 jobs instances, buffer size $\Omega_{2.0}$	69
3.10	50 jobs instances, buffer size $\Omega_{1.0}$	70
3.11	50 jobs instances, buffer size $\Omega_{1.5}$	70
3.12	50 jobs instances, buffer size $\Omega_{2.0}$	71
3.13	Quality of solution for instances with buffer size $\Omega_{1.0}$, proportion of instances, in %	89
3.14	Quality of solution for instances with buffer size $\Omega_{1.5}$, proportion of instances, in %	90
3.15	Quality of solution for instances with buffer size $\Omega_{2.5}$, proportion of instances, in %	90
3.16	Quality of solution for instances with buffer size $\Omega_{4.5}$, proportion of instances, in %	90
3.17	5 – 50 – 5 – 2, Ω_2 instances relative error from the best, in %	110
3.18	5 – 50 – 5 – 2, Ω_3 instances relative error from the best, in %	110
3.19	5 – 50 – 5 – 2, Ω_5 instances relative error from the best, in %	110

3.20	5 – 100 – 5 – 2, Ω_2 instances relative error from the best, in %	110
3.21	5 – 100 – 5 – 2, Ω_3 instances relative error from the best, in %	111
3.22	5 – 100 – 5 – 2, Ω_5 instances relative error from the best, in %	111
3.23	10 – 100 – 5 – 2, Ω_2 instances relative error from the best, in %	111
3.24	10 – 100 – 5 – 2, Ω_3 instances relative error from the best, in %	111
3.25	10 – 100 – 5 – 2, Ω_5 instances relative error from the best, in %	112
3.26	Order on the 1st stage vs. order on the 2nd stage: 5 – 50 – 5 – 2 instances, in %	113
3.27	Order on the 1st stage vs. order on the 2nd stage: 5 – 100 – 5 – 2 instances, in %	113
3.28	Order on the 1st stage vs. order on the 2nd stage: 10 – 100 – 5 – 2 instances, in %	114
3.29	Smaller planning horizon T: objective value and time, 5 – 50 – 2, Ω_2	116
3.30	Smaller planning horizon T: objective value and time, 10 – 100 – 5 – 2, Ω_2	117
3.31	Lagrangian heuristic vs. permutation heuristic: instances with 10 batches	118
3.32	Lagrangian heuristic vs. permutation heuristic: instances with 12 batches	118
4.1	Proportion of instances, in %, solved to optimality	164
4.2	Comparison of ascending and descending algorithms: number of iterations	164