

**SCHEDULING WITH TWO TYPES OF PENALTIES
UNDER UNCERTAINTY AND ITS APPLICATION TO
MAINTENANCE PLANNING**

Hue Chi (Trudy) Lam

Supervisors: Hanyu Gu, Yakov Zinder

A thesis submitted in fulfillment of the requirements for the degree of
Doctor of Philosophy

School of Mathematical and Physical Sciences
Faculty of Science
University of Technology Sydney

November 2022

Certificate of Original Authorship

I, Hue Chi (Trudy) Lam declare that the thesis titled “*Scheduling with two types of penalties under uncertainty and its application to maintenance planning*”, is submitted in fulfillment 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.

Name: Hue Chi (Trudy) Lam

Signed: Production Note:
Signature removed prior to publication.

Date: 12 April 2022

Abstract

This thesis studies a number of combinatorial optimization problems with uncertainty, which have applications in the fields of maintenance planning. In particular, the research in Chapter 3 was conducted as part of the project with a major maintenance center, and Chapter 5 considers a maintenance planning problem faced by a French electricity transmission system operator. The problem was subject of the recent ROADEF/EURO challenge 2020 competition.

This thesis is comprised of several chapters. Chapters 1 and 2 introduce the thesis topics and provide the literature survey, respectively. Chapter 3 focuses on developing maintenance plans for a fleet of passenger trains. The key feature of our model is that it considers the uncertain duration of maintenance, the center's engineering restrictions, and the rail operator's operational requirements. We propose a Genetic Algorithm, an Iterated Local Search, and a hybrid two-stage optimization procedure that combines Jensen's Inequality based relaxation with Iterated Local Search to solve the problem.

Chapter 4 focuses on improving the utilization of resources when scheduling jobs that share multiple resources. The motivation for this research comes from scheduling problems that arise in healthcare and rolling stock maintenance services. The problem is distinct from the problem in Chapter 3: the processing of each job requires several types of resources; and the penalty for capacity violation is calculated based on not only the capacity of resource but also an upper bound on the resource expansion. The solution approaches presented in Chapter 3 cannot be directly applied to this problem because their performances rely on having a good-quality initial solution produced by an exact method. We instead propose a Genetic Algorithm enhanced by local search and present a method for assessing solution quality based on Sample Average Approximation.

Chapter 5 is concerned with a large-scale planning problem arising in the maintenance of power distribution grid. Due to the extreme hazards involved when performing maintenance operations on the high-voltage lines, individual transmission lines have to be shut down for the duration of maintenance. The goal is to build intervention schedules that minimize the impact of planned outages on the reliability of power networks. A unique feature of this problem is that the duration, resource consumption, and risk value of an intervention depend on when it starts. Using the problem instances provided by the ROADEF competition, we demonstrated the effectiveness of the proposed Iterated Local Search with self-adaptive perturbation and obtained the 2nd prize.

Acknowledgements

I would like to start by expressing my sincere gratitude to Dr Hanyu Gu for giving me the opportunity to conduct the research under his supervision. This research would not have been possible without his guidance and advice, as well as the scholarly discussions that have sharpened my thinking and enhanced my knowledge in the area of Operations Research. I would like to thank Professor Yakov Zinder for his advice on my writings and his time and patience in answering my questions. Conversations with Professor Zinder have always encouraged me to challenge my ideas and claims and allowed me to think further than what meets the eyes.

My special thank to the Faculty of Science for providing the HDR fund, which supported me financially to attend two international conferences. The 9th Manufacturing Modelling, Management and Control Conference (MIM), which was held in Berlin, Germany, in August 2019. The 6th International Conference on Computer Science, Applied Mathematics, and Applications (ICCSAMA) was held in Hanoi, Vietnam, in December 2019. During my time in Hanoi, I have also had the chance to visit the Operations Research Laboratory (ORLab) at the University of Engineering and Technology, Vietnam National University. I would like to thank Dr Minh-Hoang Ha for giving me this opportunity and to the members of ORLab for taking the time to work with me and show me around Hanoi.

I gratefully acknowledge the Computational facilities provided by the UTS eResearch High Performance Computer Cluster.

I wish to thank my fellow doctoral students for their encouragement and support at various stages of my graduate study. The shared jokes and occasional drinks after work before lockdown have made this challenging but rewarding PhD journey less stressful and more enjoyable. I am indebted to my family, who has supported me through the most challenging

time, especially my parents in Vietnam, who constantly assured me that the PhD study is completely worth it despite not understanding what my research was. It is to my family that I dedicate this thesis.

Contents

Certificate of Original Authorship	i
Abstract	ii
Acknowledgements	iv
List of Figures	ix
List of Tables	xii
1 Introduction	1
1.1 Planning of Rolling Stock Maintenance	2
1.2 Scheduling of Jobs Sharing multiple Resources	4
1.3 Planning of Grid Operation-based Outage Maintenance	5
1.4 Contributions	7
2 Literature Review	11
2.1 Stochastic Programming	11
2.1.1 Stochastic programs with recourse	12
2.1.2 Solution methods	14
2.1.3 Sample Average Approximation	15
2.2 Metaheuristic	16
2.2.1 Genetic Algorithm	17

2.2.2	Iterated Local Search	19
2.3	Planning of Rolling Stock Maintenance	20
2.4	Scheduling Jobs Sharing Multiple Resources	23
2.5	Planning of Grid Operation-based Outage Maintenance	26
3	Planning of Rolling Stock Maintenance	29
3.1	Introduction	29
3.2	Mathematical Programming Formulation	33
3.2.1	Nonlinear Integer Programming Formulation	37
3.2.2	Evaluation of the Objective Function	37
3.2.3	Integer Linear Programming Relaxation based on Jensen's Inequality	40
3.3	Genetic Algorithm Approach	43
3.3.1	The decoding procedure	45
3.3.2	Evolutionary process	46
3.4	Genetic Algorithm based Matheuristic	50
3.5	Hybrid Two-stage optimization Procedure	53
3.5.1	Mixed Integer Linear Program MILP	55
3.5.2	Local Search Subroutines	56
3.5.3	Iterated Local Search	62
3.6	Computational Results	64
3.6.1	Comparison of GA and GA-based matheuristic	66
3.6.2	Comparison of the Performance of MIPM and MILP	69
3.6.3	Comparison of Hybrid ILS and Multi-start ILS	71
3.6.4	Visualization of Quality of Arrival Plan	76
3.7	Summary	78
4	Scheduling of Jobs Sharing multiple Resources	79
4.1	Introduction	80

4.2	Mixed integer linear programming formulation	82
4.2.1	Mixed integer linear program	82
4.2.2	Evaluation of the objective function	85
4.3	Sample Average Approximation	87
4.4	Hybrid Genetic Algorithm	89
4.4.1	Representation of chromosome and definition of fitness function . . .	91
4.4.2	Parent selection and crossover	92
4.4.3	Mutation	93
4.4.4	Local search method	93
4.5	Computational Results	95
4.5.1	Generation of test instances	96
4.5.2	HGA parameter setting	97
4.5.3	Comparisons between the proposed HGA and CPLEX on small problem instances	97
4.5.4	Performance evaluation of the proposed HGA on large problem instances	99
4.5.5	Sensitivity analysis	101
4.6	Summary	105
5	Planning of Grid Operation-based Outage Maintenance	106
5.1	Introduction	107
5.2	Mathematical programming formulation	109
5.2.1	Notations	109
5.2.2	Objective function	110
5.2.3	Mixed integer linear programming formulation	113
5.3	Approximation of quantile term in objective function and iterative updating algorithm	115
5.4	Confidence method approaches	119
5.5	Iterated local search	122
5.5.1	Subroutine INITIAL	125

5.5.2	Subroutine SEARCH	126
5.5.3	Subroutines PERTURB_SHIFT and PERTURB_SWAP	131
5.6	Computational results	133
5.6.1	Model MILP vs. Model A-MILP	135
5.6.2	Evaluation of the IterUpdate algorithm	136
5.6.3	CM-heuristic vs. cs-CM heuristic	138
5.6.4	Comparison of the ILS with benchmark results	140
5.7	Summary	144
6	Conclusion	146
6.1	Summary of Work	146
6.2	Future Work	150
	Bibliography	154

List of Figures

3.1 Random key encoding example. Each train-set is assigned a random number generated according to the uniform distribution $U(0, 1)$, which determine its priority in the decoding procedure. Here, train-set 2 has the highest priority, followed by train-set 1, 5, 3, and train-set 4 will be the last. 45

3.2 Resource-based crossover example. The crossover operator takes the gene values 0.41 and 0.05 from the father chromosome and places them in the same position in the child chromosome. The crossover operator fills in the missing gene values in the remaining places in the order they are defined in the mother chromosome. A big enough number (i.e., 5000) is then added to the genes indexed 2 and 3 to give the values 5000.41 and 5000.05. 48

3.3 Two-point crossover example. Two crossover points $t_1 = 1$ and $t_2 = 4$ are selected. The values of genes 2, 3, and 4 are obtained from the father chromosomes, whereas the values of genes 1 and 5 are obtained from the mother chromosomes. 49

3.4 Mutation example. For each gene, one random number is generated from $U(0, 1)$. Since the random number of gene 1 is smaller than *mutation_prob* (i.e., $0.04 < 0.05$), the corresponding gene value is replaced by a new random key, i.e. 0.65. 49

3.5 Example of the sequence decoding by train types. 51

3.6 Heat maps displaying the probability of having more than 5 train-sets residing in the maintenance center for each day across the planning horizon of one year for cases (a) $\alpha = 1, \beta = 1000$; and (b) $\alpha = 1, \beta = 1$ 77

(a)	77
(b)	77
4.1	Example of the half-uniform crossover.	92
4.2	Change in the objective function value with number of generations of the hybrid GA on the pilot set of (a) small problems and (b) large problems. . .	97
(a)	97
(b)	97
5.1	An example of the interventions having (left) positive and (right) negative slopes. Given a time t and $\tau = 0.8$, in the left figure, we have $Q_\tau^t = 128$ and $\widehat{Q}_\tau^t = Q_{1,\tau}^t + Q_{2,\tau}^t = 69 + 59 = 128$. In the right figure, we have $Q_\tau^t = 128$ and $\widehat{Q}_\tau^t = Q_{1,\tau}^t + Q_{2,\tau}^t = 69 + 59 = 128$	116
5.2	An example of the interventions having both positive and negative slopes. Given a time t and $\tau = 0.8$, we have $Q_\tau^t = 107$ but $\widehat{Q}_\tau^t = Q_{1,\tau}^t + Q_{2,\tau}^t = 59 + 73 = 132$	117
5.3	<i>one-shift</i> evaluation.	129
5.4	<i>two-swap</i> evaluation.	130
5.5	Convergence of the approximate Z_2 to the true Z_2	137

List of Tables

3.1	Parameters for the train types.	65
3.2	Parameters of probability distribution for cycle time by train types.	65
3.3	Assignment of α and β for all the cases.	66
3.4	Comparison of the performance of GA and GA-based matheuristic	67
3.5	Analysis of the performance of GA-based matheuristic with $\Gamma \in \{1, 5, 10\}$, when the algorithm is run in 1800 seconds.	68
3.6	The number of generations performed to produce the solutions in Table 3.5.	68
3.7	Performance of GA-based matheuristic with $\Gamma \in \{1, 5, 10\}$ for Case 1, when the algorithm is run for 40 generations.	68
3.8	Comparison of the performance of MIPM and MILP	69
3.9	Improvements in solution quality of MIPM and MILP by the local search LS1.	70
3.10	Improvements in solution quality of MIPM and MILP by the sequential local search SLS.	70
3.11	Summary of the eight algorithms used in Section 3.6.3.	72
3.12	Performance of hybrid ILS with LS1 and LS1'	73
3.13	Performance of hybrid ILS with SLS and SLS'	73
3.14	Performance of multi-start ILS with LS1 and LS1'	74

3.15	Performance of multi-start ILS with SLS and SLS'	74
3.16	Summary of the effects of the different neighborhoods.	75
3.17	Comparison between the performance of all solution approaches.	75
4.1	Comparison of the exact method and the proposed hybrid GA on small instances.	98
4.2	Summary of results for hybrid GA on large instances.	100
4.3	The 95% confidence interval (CI) for the optimality gap at $\hat{\mathbf{x}}$, where $\hat{\mathbf{x}}$ is the best solution obtained from the hybrid GA.	101
4.4	The 95% confidence interval (CI) for the optimality gap at $\hat{\mathbf{x}}$, where $\hat{\mathbf{x}}$ is the best solution obtained from the hybrid GA (continue).	102
4.5	Sensitivity analysis on the performance of SAA and HGA with H for instance 80-5-2 ⁸⁰ -S2.	103
4.6	Sensitivity analysis on the performance of SAA and HGA with ψ for instance 80-5-2 ⁸⁰ -S2.	104
4.7	Sensitivity analysis on the performance of HGA with λ_m and λ_c for large instances, in terms of solution quality and time.	104
5.1	Instances characteristics in ROADEF/EURO challenge 2020.	134
5.2	Comparison of models MILP and A-MILP on dataset A instances.	135
5.3	Summary of $\beta_t^0, t \in T$ for the IterUpdate algorithm on four datasets.	137
5.4	Results obtained by CM-heuristic and cs-CM heuristics for dataset A.	139
5.5	Results obtained by CM-heuristic and cs-CM heuristics for dataset B.	140
5.6	Results obtained by CM-heuristic and cs-CM heuristics for dataset C.	140
5.7	Results obtained by CM-heuristic and cs-CM heuristics for dataset X.	140
5.8	Performance of ILS on dataset A instances.	141

5.9	Performance of ILS on dataset B instances.	142
5.10	Performance of ILS on dataset C instances.	142
5.11	Performance of ILS on dataset X instances.	143