

**A GENETIC ALGORITHM APPROACH FOR SOLVING
DYNAMIC JOB SHOP SCHEDULING PROBLEM**

Pitigalage Buddhika Chandradeepa Kurera

(138116D)

Degree of Master of Engineering

Department of Mechanical Engineering

University of Moratuwa

Sri Lanka

February 2019

A GENETIC ALGORITHM APPROACH FOR SOLVING DYNAMIC JOB SHOP SCHEDULING PROBLEM

Pitigalage Buddhika Chandradeepa Kurera

(138116D)

Thesis/Dissertation submitted in partial fulfillment of the requirements for the degree
of Master of Engineering

Department of Mechanical Engineering

University of Moratuwa

Sri Lanka

February 2019

Declaration

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where due reference is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature :
Buddhika Kurera (138116D)

Date :

The above candidate has carried out research for the Master's thesis under my supervision.

Signature of the Supervisor :
Dr.V.P.C. Dassanayake

Date :

Acknowledgements

I would like to thank my project supervisor Dr. V.P.C. Dissanayake, Senior Lecturer Department of Mechanical Engineering, University of Moratuwa, for being the sole guidance throughout the project.

I would like to extend my sincere gratitude to the Python and LaTeX communities for their efforts on maintaining these Free and Open Source Software projects which are used by the whole scientific community.

I would also use this opportunity to express my gratitude to everyone who supported me from the beginning to the end of this research project, in numerous ways, especially my family and friends.

Abstract

Job Shop Scheduling Problem (JSSP) is a non-deterministic, polynomial-time (NP) hard combinatorial optimization problem. It is one of the most common problems in manufacturing due to its widespread application and the usability across the manufacturing industry. Due to the vast solution space the JSSP problem deals with, it is impossible to apply brute force search techniques to obtain an optimal solution. Indeed, it is not possible to obtain an optimal solution when the number of jobs and the machines increase. Numerous researches have been carried out studying many approaches to solve this problem. In this research, Genetic Algorithm (GA) which is another widely used nonlinear optimization technique has been used to propose an algorithm. A novel chromosome representation (indirect) with an encoding based on time is introduced in this research. The proposed solution is capable of handling multiple disruptions which are new job arrivals, sudden machine breakdown and unplanned machine maintenance. The proposed algorithm is tested against benchmark problems in Static JSSP and some developed scenarios to simulate Dynamic JSSP conditions. The results show that the proposed algorithm generates near optimal schedules for Static JSSP. This algorithm can be used as a planning tool by the planners. It is possible to simulate almost all the real-life scenarios using this algorithms and schedules can be generated satisfying the required conditions. The algorithm can be developed further by employing a local search algorithm which produced more precious, optimal schedules.

Keywords: Dynamic Job Shop Scheduling Problem, Genetic Algorithm, Disruptions in Job Shops

List of Tables

4.1	Type codes	44
5.1	Parent Selection Method – TS and Next Generation Selection Method – TS for problems of 10 x 10 (Jobs x Machines)	50
5.2	Parent Selection Method – Random and Next Generation Selection Method – TS for problems of 10 x 10 (Jobs x Machines)	50
5.3	Parent Selection Method – RWS and Next Generation Selection Method – RWS for problems of 10 x 10 (Jobs x Machines)	50
5.4	Fully Completed Operations per Job, <i>LA19^A</i>	52
5.5	Partially Completed Operations per Job, <i>LA19^A</i>	52

List of Figures

3.1	Generated Feasible Solutions	30
3.2	Assigning Operations to Machines	31
3.3	Continuity of STU on the timeline	32
3.4	Crossover and Mutation process	35
3.5	Crossover and Mutation	36
4.1	Format of the Events	43
4.2	Examples for defined Events	43
5.1	Movement of Mutation Rate	48
5.2	Convergence against the number of generations	49
5.3	Generated Schedule for modified LA19 problem (<i>LA19^A</i>)	51
5.4	After including the new job in <i>LA19^A</i>	53

List of Abbreviations

AI - Artificial Intelligence
ANN - Artificial Neural Network
BKS - Best-Known Solutions
BLSA - Bottleneck Local Search Agent
CR - Crossover Rate
CSV - Comma-Separated Values
DSPN - Distributed Problem Solving Networks
EDD - Earliest Due Date
FCFS - First Come, First Served
GA - Genetic Algorithm
GBML - Genetic Algorithm-based Machine Learning
JSSP - Job Shop Scheduling Problem
LSOA - Local Search Optimization Algorithm
MR - Mutation Rate
MUSA - Matchup Schedule Algorithm
PBX - Position Based Crossover
PSO - Particle Swarm Optimization
RP - Rescheduling Point
RS - Random Selection
RWS - Roulette Wheel Selection
SE - Scheduling Engine
SI - Swarm Intelligence
SPT - Shortest Processing Time
STU - Starting Time Unit
TS - Tournament Selection
VNS - Variable Neighbor Search

Table of Contents

Declaration	ii
Acknowledgements	iii
Abstract	iv
List of Tables	v
List of Figures	vi
List of Abbreviation	vii
1 Introduction	1
2 Literature Review	3
2.1 Job Shop Scheduling Problem (JSSP)	3
2.1.1 Varieties of JSSP	4
2.2 Methods for Solving Static JSSP	5
2.2.1 Heuristic Approach	5
2.2.2 Neighborhood Search Method	6
2.2.3 Shifting Bottleneck	6
2.2.4 Simulated Annealing	7
2.2.5 Tabu Search	7
2.2.6 Artificial Intelligence	7
2.2.7 Expert Systems	8
2.2.8 Distributed AI – Agents	8
2.2.9 Artificial Neural Network (ANN)	9
2.2.10 Evolutionary Algorithms	10
2.3 Method for Solving Dynamic JSSP	12
2.3.1 Heuristics	15
2.3.2 Metaheuristics	16
2.3.3 Simulated annealing and Tabu search	17
2.3.4 Genetic Algorithms (GA)	17
2.3.5 Artificial Intelligence (AI)	18
2.3.6 Artificial Neural Network (ANN)	18
2.4 Chapter Summary	19

3	Methodology	20
3.1	Problem Definition	20
3.1.1	Disruption Demanding A Schedule Change	22
3.1.2	Conditions to Satisfy When Scheduling	24
3.1.3	Requirements to Satisfy	24
3.2	Methodology	25
3.2.1	GA related to JSSP	26
3.2.2	How GA works	26
3.3	Design of the Algorithm	28
3.3.1	Chromosome Representation - A novel approach	28
3.3.2	Generating Initial Population	29
3.3.3	Fitness Calculation	33
3.3.4	Crossover	34
3.3.5	Mutation	36
3.3.6	Natural Selection	36
3.4	Optimization	37
3.5	Dynamic JSSP	37
3.6	Disruptions	38
3.6.1	Arrival of new jobs	38
3.6.2	Unplanned machine breakdown	39
3.6.3	Unplanned machine maintenance	40
4	Implementation	41
4.1	Implementation Approach	41
4.2	Standards in the Implementation	42
4.3	Inputs to the Algorithm	42
4.3.1	Operation Sequence per Job	42
4.3.2	Duration per operations	43
4.3.3	Information on machine unavailability	43
4.3.4	Historical Information on the Current Schedule	44
4.3.5	Information on Constraints	44
4.3.6	Graphically Visualization the Schedule	45
5	Results and Analysis	46
5.1	Tests for Solving Static JSSP	46
5.1.1	Parent and Next Generation Selection	48
5.2	Dynamic JSSP	51
5.2.1	Scenario A - Arrival of new jobs	51
5.2.2	Scenario B - Machine break-down	53
5.2.3	Scenario C - Unplanned machine maintenance	54
5.2.4	Scenario D - Start time fixed operations	54
5.3	Analysis	54
6	Conclusion	57
6.1	Conclusion	57
6.2	Future Work	58

A	Program Code	65
A.1	Implementation of the Proposed Algorithm	65
A.2	Tournament Selection Algorithm	75
A.3	Roulette Wheel Selection Algorithm	78