Extending Scheduling Algorithms to Minimise the Impact of Bounded Uncertainty Using Interval Programming

by

Ahmad Hany Mahmoud Hossny
MSc and BSc(Hons) Computer Science

Submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

Deakin University

May, 2014

DEAKIN UNIVERSITY
ACCESS TO THESIS - A

I am the author of the thesis entitled

Extending Scheduling Algorithms to Minimise the Impact of Bounded Uncertainty Using Interval Programming

submitted for the degree of

This thesis may be made available for consultation, loan and limited copying in accordance with the Copyright Act 1968.

'I certify that I am the student named below and that the information provided in the form is correct'

Full Name: .....Ahmad Hany Mahmoud Hossny.....
(Please Print)

Signed: ..... Signature Redacted by Library **.....**

Date: ..... 29 October 2014.....

DEAKIN UNIVERSITY
CANDIDATE DECLARATION

I certify the following about the thesis entitled (10 word maximum)

Extending Scheduling Algorithms to Minimise the Impact of Bounded Uncertainty Using Interval Programming

submitted for the degree of Doctor Of Philosophy

a. I am the creator of all or part of the whole work(s) (including content and layout) and that where reference is made to the work of others, due acknowledgment is given.

b. The work(s) are not in any way a violation or infringement of any copyright, trademark, patent, or other rights whatsoever of any person.

c. That if the work(s) have been commissioned, sponsored or supported by any organisation, I have fulfilled all of the obligations required by such contract or agreement.

I also certify that any material in the thesis which has been accepted for a degree or diploma by any university or institution is identified in the text.

'I certify that I am the student named below and that the information provided in the form is correct'

Full Name: .....Ahmad Hany Mahmoud Hossny .....
(Please Print)

Signed: ..... Signature Redacted by Library **.....**

Date: .....29 October 2014 .....

Extending Scheduling Algorithms to Minimise the Impact of Bounded Uncertainty Using Interval Programming

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

Submitted by:

Ahmad Hany Hossny
MSc and BSc(Hons) Computer Science

Under supervision of:

– A.Prof. Doug Creighton

– Prof. Saeid Nahavandi

– Dr. Michael Johnstone

March 2014

Deakin University

Centre for Intelligent Systems Research

DEAKIN UNIVERSITY

Centre for Intelligent Systems Research

CANDIDATE DECLARATION

I certify that the thesis entitled: **"Extending Scheduling Algorithms to Minimise the Impact of Bounded Uncertainty Using Interval Programming"** that is submitted for the degree of: **"Doctor of Philosophy in Computer Science"** is the result of my own work and that is where reference is made to the work of others, due acknowledgement is given.

I also certify that any material in the thesis, which has been accepted for a degree by any other university or institution is identified in the text.

Full Name .....

Signed .....

Date .....

Abstract

SCHEDULING under bounded or interval-based uncertainty presents its own challenge due to the lack of information describing the data between the lower bound and the upper bound. This challenge arise when the distribution or membership function of the problem are not defined. Bounded uncertainty causes a trade-off between the optimality and the certainty of the estimated value for the objective function of the scheduling problem. Relaxing the uncertainty constraint of the scheduling problem will lead to an accurate estimate for the objective function with a high margin of uncertainty. On the other hand, optimising the objective function considering the uncertainty constrain, will lead to an inaccurate estimates with a low margin of uncertainty.

Researchers have tackled the problem of scheduling under bounded uncertainty, many of the proposed techniques use the mixed integer linear programming (MILP). MILP minimises the bounded uncertainty if the problem is formulated as a system of linear equations. As some scheduling problems cannot be formulated as a system of linear equations, they cannot be solved using MILP. Such problems are usually formulated and solved using the branch and bound heuristics or the meta-heuristic techniques. Here the need arises for an algorithmic extension that can minimise the impact of bounded uncertainty on the results.

The objective of the proposed research is to minimise the impact of bounded uncertainty of the input parameters on the estimated output cost function for the heuristic and meta-heuristic algorithms. The uncertainty of the estimated objective function is measured by

the distance between the upper bound and lower bound of the optimal interval value. The distance of the optimal interval is compared to the distance between the two estimates of the objective function using the upper and lower bound. Such an objective is considered a preventive scheduling technique as it reduces the uncertainty impact before it happens.

This research introduces the concept of performing the calculation on interval inputs, then approximating the final result instead of approximating the uncertain input before performing the calculations. The proposed methodology analyses the scheduling algorithm, extends it to do the arithmetic and logic operations using intervals rather numbers and estimate the final result. This methodology uses program slicing and interval programming techniques including interval arithmetic, interval algebra and interval logic.

By extending Bratley's algorithm, the uncertainty impact on the objective function is reduced by 7.2%. The extended interval-based Hodgson's algorithm minimised impact of bounded uncertainty by more than 12%. Extending McNaughton's algorithm minimised the uncertainty by 7.1%, but the result was not guaranteed to be optimal. The uncertainty reduction percentages vary according to the input data. The margin of uncertainty minimisation is directly correlated with the distance between the upper and the lower bounds of the input data as well as the number of overlaps between intervals.

According to the stated research, experiments and results, it is concluded that interval programming can reduce the impact of bounded uncertainty on objective functions according to the number of tasks, the uncertainty ranges, the number of interval overlaps. The proposed research minimised the impact of bounded uncertainty of the input parameters on the estimate of the objective function. The methodology has great potential in fundamental and applied research. Fundamental research extensions include the integration with sensitivity analysis, using affine arithmetic. Applied research extensions include integration with prediction intervals-based systems, adding functional observers to the algorithms to ensure that the uncertainty range does not diverge or inflate.

List of Publications

- Hossny, A.; Nahavandi, S.; Creighton, D.;, "Minimizing Bounded Uncertainty Impact on McNaughton's Scheduling Algorithm via Interval Programming", in IEEE international conference on Systems Man and Cybernetics, 2013. IEEE-SMC 2013. Manchester, UK, 13-16 Oct. 2013.

- Hossny, A.; Nahavandi, S.; Creighton, D.;, "Minimizing Bounded Uncertainty Impact on Scheduling with Earliest Start and Due-date Constraints via Interval Computation", in Emerging Technology and Factory Automation, 2012. ETFA 2012. International Conference on, September 2012.

Research Contributions

This research contributes to the algorithms science by minimising the impact of bounded or interval-based uncertainty on the scheduling algorithms in a preventive way by extending the algorithm in a way that copes with the uncertain nature of the input parameters. The contribution of this research can be summarised in the following three points:

1. Introduced the concept of executing the algorithm using the uncertain variables, then deciding how to resolve the uncertainty in the decision making stage using the final uncertain result. Such concept maintains the uncertainty of the variables at its level or less, as it minimises the impact of uncertainty changes happening due to the algorithm execution, such uncertainty changes include the uncertainty propagation, divergence, inflation and cyclic drifts.

2. Proposed a new methodology that uses interval programming and program slicing to extend any algorithm to be able to run using the interval-based input parameters. The new methodology can minimise the total uncertainty of the end result and maintain the optimality of the cost function. The interval-based extensions can be applied to any scheduling algorithm, including the heuristics and meta-heuristics.

3. Applying the new concept and methodology to three different scheduling algorithms. the results of the three algorithms with the interval-base extension outperformed the

results of the same three algorithms using numerical calculations or point estimates. The marginal enhancement of the results varies between 0% and 22% according to the problem and the data nature. The marginal enhancement was never less than zero, which implies that the worst case scenario returns the same performance as numerical or point-based estimates without any extra complexity in performance.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

Uncertainty is the lack of accuracy and lack of preciseness in any estimated or calculated value that may act as input or output of any system. It is usually results from the lack of information and sometimes from overwhelming with information that may confuse the decision maker. It has multiple definitions and perceptions in different perspectives including physics, weather, market, language, and many other domains, but in all cases it is a matter of decision making.

Uncertainty is considered one of the famous challenges meeting any decision maker as it leads to the risk of a wrong decision by assuming some wrong values as correct ones. The risk can be reduced by paying extra costs in some domains such as market uncertainty, industrial uncertainty. The risk can also be reduced by considering the worst case scenarios in the planning such as in supply chain, cargo or traffic. It is important to consider that some risks are not tolerable or acceptable, such as human related or medical risks, especially that the cost will not be just wasted money or time. Decision makers want to minimise the uncertainty to zero, if possible, or at least minimise its effect on the objective function.

Uncertainty in engineering domain is usually formulated using one of four paradigms

including:

1. Logical uncertainty (Logical Ambiguity)
2. Mathematical uncertainty (Arithmetic Intervals)
3. Probability uncertainty (Confidence Intervals, Credible Intervals)
4. Fuzzy logic

Logical uncertainty is the multiple interpretation of the same item leading to confusion, especially in human related domains such as language processing and information retrieval. Uncertainty is considered logical if the problem is represented in propositional logic or in rule-based format with multiple rules applying to the same problem leading to multiple solutions that might be not correct. This problem is usually named as ambiguity. To minimise the ambiguity issue, the rule induction process should induce exact rules matching to the desired scenarios to be covered. On the other hand, this leads to less coverage and less flexibility of the logical system (e.g. information retrieval). Some researchers integrate rule-based systems with probability to minimise the ambiguity.

Uncertainty is represented in mathematical calculations by the term precision, especially in a floating number related processing, considering how many decimal places are needed to provide accurate enough result. The term precision can easily be represented by intervals as 3.547 are in the interval [3.54, 3.55]. Usually 3.547 is approximated to 3.55 which accepts margin of error equals 0.003, but in sensitive systems when approximation is applied to many values it leads to effective deviation of the final results. Here arises the importance of interval processing.

Mathematical uncertainty can be minimised either by minimising the interval size (if possible) for every interval input to the system, or to implement the optimisation techniques using interval arithmetic and interval algebra. Intervals can be represented as number range

where x value is not known exactly but known within lower and upper limits x_l and x_h (x_l, x_h). Another presentation for the interval is the centre (mean) and error (radius) of the interval (m, r).

Although the interval contains the x value for sure, it is difficult to say where exactly x is located inside the interval, so it is not recommended to select specific value or even crop the range to smaller range assuming that it will make it less certain with specific percentage, on contrary it may exclude the exact value from the interval, which may lead to wrong answers. The main advantage of applying interval programming is to minimise the overall interval result.

Bounded form: is a formulation that uses only the upper and lower bounds of the uncertain parameters. It is used when there exist no enough information describing the uncertainty nature and when there is no historical data to build any kind of distribution or regression. The uncertainty factor is described by interval $\theta \in [\theta_{min}, \theta_{max}]$ *or* $|\hat{\theta} - \theta| \leq \varepsilon|\theta|$, where θ is real value, $\hat{\theta}$ is the estimated value, and $\varepsilon > 0$ is the error range or uncertainty measure. Dealing with bounded form should be managed by interval computation, interval arithmetic and interval algebra.

Probability description: is a formulation of the uncertainty as distribution built according to previous data representing the history of this problem, which is known as "frequentist probability" and represented as confidence intervals. Uncertainty can also be formulated based on the predicted data of the future, which is known as "Bayesian probability" and presented by credible intervals or prediction intervals. This problem can be solved by many techniques such as Monte Carlo, regression analysis or simulation techniques in order to minimise uncertainty source.

Fuzzy logic is a form of multi-valued logic dealing with reasoning, which is fixed or relative represents uncertainty on its membership function. It depends on the membership function that models who certain a specific value is totally true or partially true or totally

false.

Fuzzy uncertainty can be an extension to the logical uncertainty by increasing number of logical values. It can also be mapped to probabilistic uncertainty as the probability cares if the random sample inside or outside the distribution. Meanwhile, fuzzy distribution gives additional factor of how much the sample belongs to the distribution.

Fuzzy description: is similar to probability formulation with difference that it has membership function, so it is considered as adding one more dimension to probability curve for how sure does that value belongs to this set, so it is considered as abstraction of probability.

Fuzzy is different than probability that the area under the curve can be more than 1, and it can have part of the curve as uniform value and another parts of the curve as distribution, it all depends on the membership function. Handling fuzzy-based uncertainty is divided into two parts, first is similar to bounded form where the membership function is uniform, and second part will be handled similar to probability distribution in curve based areas of membership functions.

In the standard set theory, the value of the statement can be determined by the Boolean membership function A(x) represented in equations 1.1, 1.2 and 1.3.

$$\mu A(x) = \begin{cases} 1 & iff \ x \in A \\ 0 & iff \ x \notin A \end{cases} \tag{1.1}$$

Fuzzy theory accepts membership function A(x) between 0 and 1 for continuous values:

$$\mu A(x) = \begin{cases} 1 & iff \ x \in A \\ 0 & iff \ x \notin A \\ p; \ 0<p<1 & if \ x \ partially \ belongs \ to \ A \end{cases} \tag{1.2}$$

$$\mu A(x) = \begin{cases} 1 & iff \ x \in A \\ 0 & iff \ x \notin A \\ 1 - \frac{(ax-\theta)}{\Delta\theta} & if \ \theta < ax < \theta + \Delta\theta \end{cases} \tag{1.3}$$

1.2 Motivation

1.2.1 Scheduling importance

Scheduling is a famous decision making problem that targets allocating different resources to different tasks. The most frequent resources to be allocated are the time, money, human resources and machines. The importance of the scheduling algorithms arises in case the resources are limited. Such shortage of resources leads to a need for better utilisation of every resource in order to optimise the objective function [1–3].

Scheduling has important role in different domains including project planning, production lines, supply chains, bus schedules, traffic estimation, bank queues, chemical plants, electricity load balancing [4–6], CPU process and memory management and medical resource allocation including operating rooms, intensive care units and emergency rooms [7–9]. Airports is a another example of scheduling importance; as the number of runways is limited, and the number of aircrafts willing to land is high, which requires accurate allocation of time slots, in which the aircraft plane will use a specific runway and how long will it use such runway [10–12].

Every scheduling problem has a set of variables that is used for the formulation and the optimisation algorithm. Such variables represent the properties of each machine and the relations between them. These variables also represent the properties and dependencies of the tasks or the jobs processed on each machine. The properties of each task include the processing time, the release time, the setup cost, the due date, the actual completion time,

the deadline, the weights, and the dependencies [13, 14].

The objective can be a single function or multiple functions that should be satisfied together. The objective function can be either minimization or maximization problem. The objective functions of scheduling problems can be optimising the make-span, the maximum lateness, maximum tardiness, total setup cost, total setup time, total flow time, total completion time, total earliness, total tardiness, number of tardy (late) jobs, total weighted completion time, weighted number of tardy jobs, total weighted earliness or total weighted tardiness.

Scheduling problems are categorised according to the given properties for the tasks, the properties of the machines, the constraints and the objective function [13, 15]. The properties of the task can be static if they are all defined from the beginning of the scheduling process till the end. The task properties can also be dynamic, where the properties of the task can change at runtime or any new task can be assigned to a machine at the runtime [16, 17].

The properties of the machine include the number that can be single machine or multiple machines. Another property is the relation between the machines as sequential or parallel processing in case of multiple machines. Another factor for categorization is the certainty of the input when all conditions are assumed to be known, which is a deterministic problem. If the input is not certain or there exist uncertainty in the properties of the tasks or the properties of the machines, The model is considered stochastic [13–18].

1.2.2 Scheduling algorithms

There exist many techniques and algorithms used to achieve the scheduling objectives or to optimise the cost function for different scheduling problems according to the nature of the problem. Some of the famous algorithms used in solving many problems are:

1. Simulated Annealing: is an optimisation technique for NP-complete problems such as traveling salesman. It uses probability to find an approximated global optimum for the cost function in the search space. Such heuristic is mainly useful in discrete search space

2. Tabu Search: is mathematical technique that enhances the local search performance using memory structures describing visited solutions then mark the potential solutions as taboo.

3. Genetic Algorithms: is a heuristic search technique that imitates the evolution process. Such heuristic is usually used to create good enough solutions to optimisation problems using techniques inspired by evolution, such as inheritance, mutation, selection, and crossover.

4. Ant Colony Optimisation: is a probabilistic technique that constitutes some metaheuristic optimisations to solve complex computational problems that can be reduced to finding routes through graphs.

5. Beam Search: is an optimisation of best-first search that reduces its memory requirements, such heuristic algorithm tracing the graph by expanding the most promising node within a limited set.

6. Constraint Programming: is form of declarative programming that states the constraints as relations between variables.

1.2.3 Scheduling uncertainty

Scheduling uncertainty is a challenge faces every decision maker in different domains and leads to state assumptions leading to risks of failure or costs of risk avoidance and insurance. For example, airport flight lags cause cost as the limited availability of runways leads to tight schedule of landing waiting and take off actions, any unexpected changes in such schedule will lead to extra waste of cost and time. Another example is intensive care rooms in hospitals, which are usually limited and needed and its estimates are usually not trusted, as you can never state for sure when will the patient need intensive care and for how long.

Simulation, forecasting and prediction helps a lot in decision making with one concern that its data are not certain, they are just estimates with some margin of error represented by math, probability or fuzzy logic. By applying scheduling on the predicted values, which are not certain, it will not give fully trusted schedules or plans.

Early trials to formulate the scheduling problem focused mainly on the discrete-time scheduling formulation, as the time axis is divided into a sequence of ranges of equal duration. This is mentioned by Bassett et al. [19] and Kondili *et al.* [20]). Recently a many researchers have focused on finding efficient techniques and algorithms to deal with continuous-time representation. After the continuous time representation is formulated, many alternatives have appeared to reduce the computational complexity of the resulting model.

The scheduling process differs according to the given information for each task and each resource including the release date, due date and the processing time. Many researchers assumes the nature of the input data to be deterministic. This assumption is true within simple systems or digital systems, such as CPU process scheduling. On contrary, the certainty of given information in real vary according to situation and problem nature.

The lack of certainty (uncertainty) results from uncontrollable parameters or unknown

parameters that affect the problem itself. e.g. Flight control with weather factor, or traffic scheduling with runtime sudden crashes. The recent researchers studying the scheduling under uncertainty try to find a paradigm or methodology to reduce it in systematic way, in order to find an optimal solution with minimum uncertainty providing the optimal, reliable and feasible schedule.

1.3 Problem statement

The main challenge with bounded uncertainty of the input data is the lack of information between the lower bound and the upper bound for each interval, especially that the interval does not have any function describing the nature of the uncertainty. Such challenge lead to search for alternative solutions similar to the interval perturbation, multiple estimates and assumption of a specific value [21–24] .

Interval perturbation tries all possible values inside each interval. This technique guarantees to try all possible solutions until the optimum solution is found. Although this technique finds the optimum solution, extensive processing is required, as it executes the algorithm number of times equal to the multiplication of the distance between the boundaries for each interval.

In case the number of intervals is limited, interval perturbation can be applicable, but the processing time grows exponentially by every task added to the problem. Each interval is divided to a number of steps with a specific size, which is another factor affecting the interval perturbation. The step size affects the total number of steps, which affects the results optimality and the processing time. Reducing the step size increases the number of steps, which affect the performance and the optimality [25–28].

Another solution for the bounded uncertainty is the multiple estimate for the objective function. The first estimate uses the lower bound and the second estimate uses the upper

bound, then take the average. The problem with this technique is the inflation of the uncertainty, where the execution of the algorithm using the lower bound may lead the first estimate to go lower and the upper bound to go higher. This will result to increase the distance between the final lower estimate and final the upper estimate, which increases the uncertainty of the result and the difficulty of decision making.

The third technique used to solve the bounded uncertainty is to assume any specific value between the lower and the upper bound. This is the most simple way to execute any algorithm with uncertain input, but it implies the risk of the deviation between the estimated value and the optimum value.

1.4 Objective

The proposed research focuses on the mathematical formulation of uncertainty, which is based on intervals. interval-based uncertainty can be identified as tolerance intervals or prediction intervals or bounded representation. The only available information regarding the bounded uncertainty is the upper and the lower bounds of every interval. To reduce the interval-based uncertainty on heuristic and meta-heuristic algorithms, the algorithm should use the interval arithmetic and logic as replacement to the standard operations defined in the standard scheduling algorithms.

To update a scheduling algorithm to use interval input rather numerical input, each interval can be summarised to a selected number from it according to one of the following options:

1. Estimate on lower bounds
2. Estimate on upper bound
3. Random value from each interval

4. midpoint of each interval

Selecting a number at the beginning of the calculations means neglecting the potential uncertainty propagation or the effect of uncertainty in the calculation. By performing further calculations the deviation or the drift from the optimal solution increases. This can be verified by solving the problem with the lower bound selection and upper bound selection, then compare the solutions. The difference between the two solutions represents the total uncertainty.

The premises here is that if the algorithm is applied on the interval as one unit, it will give overall better performance than applying it on a selected number from the interval. To apply the algorithm on interval basis, each of interval operations and interval algebra should be identified in order to implement the algorithm using the newly defined operators.

The basic operations identified by interval arithmetic are:

- Addition: $[a, b] + [c, d] = [a + c, b + d]$
- Subtract: $[a, b] - [c, d] = [a - d, b - c]$
- Multiply: $[a, b] \times [c, d] = [min(ac, ad, bc, bd), max(ac, ad, bc, bd)]$
- Division: $[a, b] \div [c, d] = [min(a/c, a/d, b/c, b/d), max(a/c, a/d, b/c, b/d)]$
- Boolean comparisons

The proposed concept is applied by extending the Matlab toolbox *"Torsche"*, which is developed by Sucha et al. [29]. Extending the toolbox allowed it to support the interval operations including the interval arithmetic and interval logic as well as the functions using them such as sort and search. Bartley's algorithm, Hodgson's algorithm and McNaughton's algorithm have been implemented on interval basis using the toolbox *Torsche* and changed the code to accept the added functionality.

The three algorithms used the sort functionality, which is re-implemented to support interval comparisons according to interval algebra by James Allen [30–33]. A comparison between two intervals depends on four criteria, which are lower bound, upper bound, median and radius. one of the challenges is when the lower bound of interval I_1 is less than the lower bound of interval I_2 but upper bound is higher. Sometimes, scheduling algorithms need a special purpose sorting algorithm according to the context. For example, if the due-date is the variable to sorted, the sort can be applied on the upper bound, and if the release-date is the variable to be optimised, it should be sorted on the lower-bound.

To evaluate the algorithm performance; The scheduling algorithm is applied using number arithmetic on lower bound and upper bound, then calculate the cost function in the two cases, then measure the difference to know the level of uncertainty. The less the distance between the lower and upper bound of every interval leads to more certain and accurate result of the problem optimisation. Then apply the same algorithm using interval arithmetic and calculate the objective function then measure the distance again to identify the certainty/uncertainty level.

1.5 Scope and limitations

The proposed research focuses on how to minimise the impact of bounded uncertainty on scheduling algorithms. The methodology starts by analysing the scheduling algorithm using program slicing in order to find the possible trajectories of the algorithm. Second the uses interval programming is used to extend the algorithm, which imply using interval arithmetic and interval logic.

To emphasize the success of the methodology, three experiments have been performed using three algorithms, one of them used real life data and the other two used generated data. The results of each algorithm are analysed and correlated with the input in order to

find out the factors affecting the performance in terms of uncertainty nature and the nature of the problem.

Although there exist many uncertainty formulations, this research focus only on the bounded uncertainty, which is formulated mathematically using intervals. Other types of uncertainty including the logical ambiguity, probabilistic distributions and fuzzy representation are out of the scope of this research. The other formulations of uncertainty can be the future research.

Scheduling uncertainty is taken as case study for extending the algorithms to deal with uncertainty. In this research, only preventive scheduling is studied. Corrective scheduling is not studied as it needs to consider runtime factors similar to runtime type identification and runtime performance tracking, which are out of the scope of this research.

1.6 Thesis outline

The thesis outline is organised as follows: The second chapter is a literature and critical review of uncertainty formulation, scheduling techniques and scheduling under uncertainty solutions. The second chapter covers in the first section the uncertainty definitions, its taxonomy, sources and management strategies. The second section in the review chapter describes in brief the different formulations of scheduling problems, scheduling classification and scheduling challenges in terms of complexity and uncertainty. The third section describes the preventive scheduling under bounded uncertainty, as the preventive scheduling is sub-class of scheduling techniques and bounded uncertainty is sub-class of quantitative uncertainty. The third section also describes different techniques to minimise the impact of bounded uncertainty on the scheduling algorithms, such as interval perturbation, mixed integer programming and combinatorial with the critical review of each.

Chapter three explains the proposed methodology to minimise the impact of bounded

uncertainty on scheduling algorithms and how it can be applied on any scheduling algorithm including heuristics and meta-heuristics. The chapter describes how the methodology used the program slicing to decompose the algorithm and used the interval programming, interval arithmetic and interval logic to extend the algorithm to be compatible with interval input parameters.

The fourth chapter shows how the proposed methodology reduced the impact of the bounded uncertainty on Bratley's algorithm. The chapter explained the unextended numerical-based algorithm, and then it explained how to extend it according to the proposed methodology that is explained in chapter three. The effect of the interval programming is emphasised by many experiments, which used the extended interval-based algorithm and compared the results with the numerical-based algorithm.

Chapter five shows how the proposed methodology reduced the negative impact of bounded uncertainty on the objective function of Hodgson's algorithm. The chapter started by describing an example of Hodgson's algorithm usages, such as scheduling the aircraft landings. The review of the uncertainty, scheduling and scheduling under uncertainty have been stated in sections 5.3 and . The problem formulation has been discussed in the section 5.4. Section 5.5 describe in details the proposed methodology and how it is applied to Hodgson's algorithm. The rest of the chapter explains the experiments, the results and discuss the criteria affecting them.

The sixth chapter explains another scheduling algorithm, which is McNaughton's algorithm. First, the standard non-extended numerical-based algorithm is stated, then the proposed methodology is applied, then the extended interval-based algorithm is stated. Many experiments have been performed to compare the results in terms of optimality and uncertainty between the two algorithms. The experiments, results and conclusion showed up that the extended interval-based algorithm reduced the total uncertainty with minor effect on optimality.

Chapter seven states the conclusion of the whole research, it also discusses the different criteria affecting the proposed methodology and states the conclusion of the research. Chapter seven also states in details the potential of the proposed research and how it can be extended in the future work.

Chapter 2

Scheduling under Uncertainty

2.1 Uncertainty Review

2.1.1 Overview

Uncertainty has many definitions according to the context. The most common ones in the engineering sector are defined by Mavris and Delaurentis [34] as "uncertainty is the lack of knowledge (either in information or context), that causes model-based predictions to deviate from reality" or "as any unpredictable change in the runtime environment causing disruption in the execution and the results of the system" by [35] or as "the dissimilarity between the amount of information required to execute a task and the amount of information already infatuated by some distribution" [36].

The concept of uncertainty is discussed and stated first by Socrates and Plato. Philosophers doubted whether scientific knowledge, no matter how elaborate, sufficiently reflected reality (Kant, 1783) [37]. They realised that the insight into the mysteries of nature is directly proportional to the awareness of the limits of the knowledge of how "things as such" are (Kant, 1783) [37, 38].

According to the context of decision making in scheduling process, which is an optimisation problem, the *uncertainty* can be defined as "the dissimilarity between the expected values at setup time and the actual values at runtime that may happen because of lack knowledge or misleading information or error in the model-based predictions".

Many optimisation problems have been solved mathematically, which gave an impression that the result is accurate and certain. Meanwhile many problems cannot be modelled mathematically due to complexity issues. Such issue created the need for modelling, simulation, numerical analysis and statistical solutions. Those techniques gave good enough results to solve the problem with a specific margin of error. The existence of such error margin changed the objective function to be multiple objective first is to optimise the cost function and the second is to minimise the error margin.

Recently, the concept of the uncertainty became well known, because of the nature of the new problems such as human will or the weather simulation or even any systematic disruption in factory machine such as electricity cut-offs. The causes of the problem can be parametric at the setup time or sudden at the runtime. The usual mathematical modelling, statistical modelling or numerical analysis cannot deal with such problems effectively to eliminate such uncertainty or its root causes. The current target now is to estimate it and to minimise its impact on the objective function.

According to Van der Sluijs [39], the fundamental and permanent nature of some uncertainties includes the epistemic and the unpredictable uncertainty (aleatory) and ambiguous uncertainty [40–42].

Epistemic happens because of the lack of knowledge, it is also considered as cognitive or informational uncertainty. In such case, the understanding of the problem exist and knowable but it is incomplete, which is called in management "Known unknown".

On the other hand, unpredictable uncertainty is also known as variable, Ontological or

ontic uncertainty. It describes an unknowable or unexpected action or behaviour. The unpredictable uncertainty is related to the randomness of nature similar to society dynamics, human behaviour, economic, cultural changes or technological disruptions [43].

Ambiguity is defined as the multiple understanding for the same term, which causes confusion to the decision maker. To consider all possible understandings the cost will be so high. Denying some of those understandings increases the risk of excluding the correct decision or choosing the wrong answer. Such ambiguity happens because of the usual variations of the values, goals or interests. It may also happen because of epistemic differences regarding the best way to manage the system functions [44–47].

The uncertainty concept has been classified according to multiple criteria similar to the causes, the uncertainty nature, formulations and the impact. Those classifications will be discussed in detail in this chapter within the following sections.

Every uncertainty problem has its own characteristics and has its context and scenarios, which can be decision making problem, future prediction, history analysis, human behaviour, society trends, information retrieval or natural phenomena. Many researchers discussed specific problems considering their uncertainty, its nature and the available information that can be used to minimise the impact of uncertainty.

One of the main sources of system uncertainty is the nature including climate changes, animal behaviour, water currents and ecosystems [48–50]. The behaviour of the natural systems has limited predictability because most of the factors are not controllable and does not have a well-known model similar to the solar system [48]. Natural system has cyclic drift uncertainty, where a specific factor change at runtime, then it propagates to all other environmental factors, which may affect again the first factor. The limited knowledge and control caused all modelling and simulation techniques for the natural systems to have big margin of error as well as great uncertainty [48].

Object	Nature		
	Unpredictability (unpredictable system behavior)	Incomplete knowledge - Lack of information - Unreliable information - Lack of theoretical understanding - Ignorance	Multiple knowledge frames - Different and/or conflicting ways of understanding the system - Different values and beliefs - Different judgement about the seriousness of the situation, growth potential of problems, priority of actions or interventions
Natural system - Climate impacts - Water quantity - Water quality - Ecosystem	Unpredictable behavior of the natural system, e.g., How will climate change affect weather extremes?	Incomplete knowledge about the natural system, e.g., What are reliable measurements of water levels?	Multiple knowledge frames about the natural system, e.g., Is the main problem in this basin the water quantity or ecosystem status?
Technical system - Infrastructure - Technologies - Innovations	Unpredictable behavior of the technical system, e.g., What will be the sideeffects of technology X?	Incomplete knowledge about the technical system, e.g., To what water level will this dike resist?	Multiple knowledge frames about the technical system, e.g., Should dikes be built or flood plains created?
Social system - Organisational context - Actors - Economic aspects - Political aspects - Legal aspects	Unpredictable behavior of the social system, e.g., How strong will actors' reactions be at the next flood?	Incomplete knowledge about the social system, e.g., What are the economic impacts of a flood for the different actors?	Multiple knowledge frames about the social system, e.g., Should water markets be introduced to deal with water scarcity or negotiation platforms?

Figure 2.1: Types of uncertainty as stated by Brugnach *et al.* [40] including examples and the nature of each example as incomplete knowledge or multiple knowledge frames

Uncertainty and Risk Analysis

Uncertainty is highly correlated with risks. Some researchers claim that uncertainty is a cause of the risk. Other researchers say they correlation does not imply causality and both of uncertainty and risk happens because of the same reason, which is the lack of information or lack of control on runtime factors. Such lacks lead the planner or the decision maker to state assumptions to be able to take the decision. The assumption by its nature is not guaranteed to be correct as it may fail to match the reality; such failure is the cause of the risks [51].

Risk analysis process is to estimate risk, its causes and find out how to avoid or minimise such risk. In the case of risks resulting from uncertainty, the decision maker can minimise the risk by minimising the uncertainty. National legislation, industry standards and company guidelines often require that, if possible, a quantitative evaluation of the uncertainties should be presented as part of the analysis results [52]

Mehr, *et al.* defined risk as uncertainty [53]. The degree of risk is measured by the probable variation of actual experience from expected experience. The lower the probable percentage of variation, the smaller the risk. Magee stated that "The uncertainty of the happening of an unfavourable contingency has been termed risk. Risk is present when there is a chance of loss" [54]. He also mentioned that "The various factors contributing to the uncertainty are termed hazards. Ordinarily there are many separate hazards that contribute to the chance or possibility of loss that attach to any particular object or person. The sum total of the hazards constitutes the risk", which imply that uncertainty by itself does not necessarily lead to risks [55].

Uncertainty Propagation

Uncertainty propagation is another challenge with uncertainty where uncertainty causes more uncertainties or increases the margin of uncertainty. Uncertainty propagation happens in reality where a small change causes bigger change in a way similar to butterfly effect or a growing snowball. The weather forecasting is a good example, where the uncertainty of temperature causes uncertainty of atmospheric pressure, which causes uncertainty in winds strength and direction, which might have clouds and cause rain, which will reduce the atmosphere temperature [56–58].

Uncertainty propagation can also happen in modelling and simulation according to the used equation or algorithm. For example, bounded uncertainty is formulated as numerical

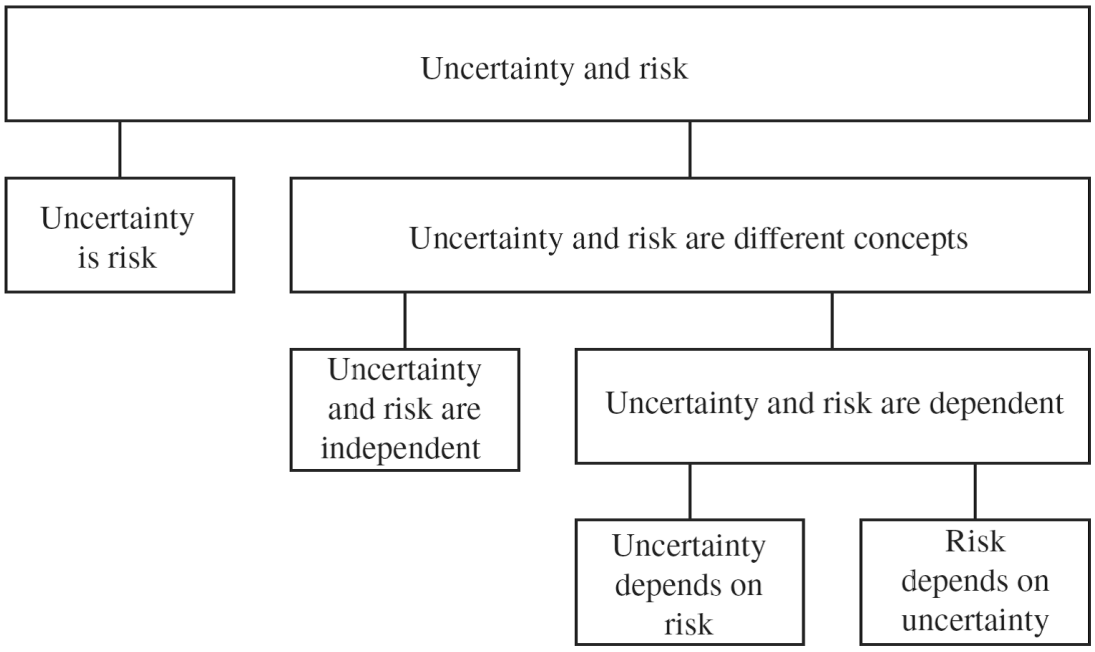

Figure 2.2: The classification of the relation between uncertainty and risks considering the different views regarding the relation between them including independence, correlation and causality

intervals $I = [a, b]$ if the equation uses I_n the interval will be $[a_n, b_n]$ and the distance between a_n and b_n will be extended. Such extension between the boundaries will increase the margin of the risk and the difficulty of decision making [59].

Researchers proposed many techniques to solve uncertainty propagation most of them are either preventive or corrective. The preventive techniques eliminate or minimise the source of the uncertainty if possible by stating assumptions that make the uncertain variable non-expandable. For example, to take the mean in case of statistical uncertainty or taking the upper or the lower bound in the case of bounded uncertainty. Although such methodology solves the propagation issue and gives accurate results, the probability of

giving incorrect results is high. [60–63]

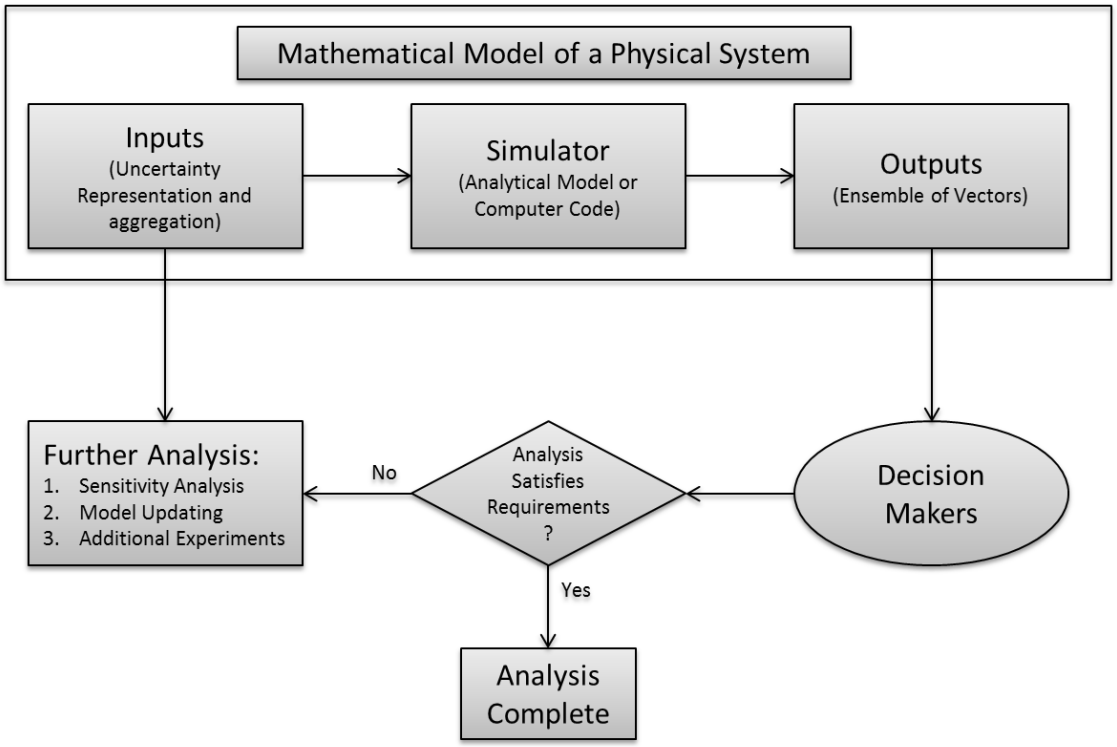

Figure 2.3: The propagation of uncertainty through mathematical models within a decision support context as stated by Oberkampf *et al.* [64]

The corrective techniques track the performance of the algorithm at the run time and monitor the uncertain variables to check if the uncertainty increases or not. In case the uncertainty increased, the system or the algorithm should be tuned to maintain an acceptable level of uncertainty. The figure 2.3 explains the propagation of uncertainty through mathematical models in a decision support context and how it can be solved using the iterative tuning [65–68].

2.1.2 Uncertainty Management Strategies

Many researchers developed and still developing different strategies, methodologies and techniques to deal with the uncertainty challenges, which vary according to the problem nature and context. The proposed solutions are grouped into multiple categories; each category can solve specific problems within specific domains.

First category has one strategy that is to ignore the uncertain factor implicitly or explicitly for the time being, it is called wait and see approach. Such technique can be complemented with thinking and implementing instant solution when the problem happens in the runtime. Though this is not a desirable solution but it is used for the rare events similar to catastrophic force majeure including volcanos, earth quakes, wars and nuclear explosions [40].

Second category is to generate the missing knowledge causing the uncertainty and this includes three strategies. First is the uncertainty assessment that is usually used within scientific, academic and research domains in order to get better understanding for the uncertainty within the researched problem. Uncertainty assessment includes uncertainty identification, uncertainty classification, uncertainty quantification, uncertainty propagation and uncertainty prioritization. [40, 43, 44, 69, 70].

Another strategy of knowledge generation is to reduce the epistemic uncertainty. This is can be applied by gathering more related data or by developing uncertainty indicators or by performing multiple experiments with different data or configurations. Quantitative modelling and simulation and qualitative assessment and expert opinions help in reducing epistemic uncertainty [40, 42–44, 71].

The third strategy for generating knowledge as solution for the uncertainty is to study different scenarios in a way similar to the what-if analysis. Such study considers the performance of different solutions and strategies within various possible future scenarios that

may happen. The main consideration with this solution is the number of possible scenarios with complicated problems similar to scheduling and combinatorial. [40, 42, 72].

The third category of uncertainty management strategies is to interact with uncertainty causes and to try to eliminate them. This strategy is useful with human related systems as human being is one of the considerable uncertainty causes in many systems. This category involves communicating uncertainties, persuasive communication, dialogical learning, negotiation and oppositional mode of action [39, 40, 42, 44, 46, 73–81].

Fourth category is to cope with the uncertainty by preparing for the worst case scenario. The target of such strategy is to limit any possible negative consequences, which is called "damage control". The main drawback with such scenario is the high cost of the wasted opportunities. The coping strategy is usually used in critical decision similar to surgery where any risk may waste a human being life [40, 75, 82].

Table 2.1: A list of uncertainty management strategies with their description according to Raadgever *et al.* [83]

Category	Strategy	Description
Ignoring	Ignoring strategy	Implicitly or explicitly ignoring uncertainty for the time being. This wait and see approach may be complemented with thinking up and implementing strategies in the timeframe of an unfolding potentially damaging event.

Knowledge Generation	Uncertainty assessment	Strategy often used in the academic world to get a better grip of uncertainty. E.g. uncertainty identification, uncertainty classification, uncertainty quantification, model uncertainty propagation and uncertainty prioritisation.
	Reduction of epistemic uncertainty	Strategy used to reduce epistemic uncertainty by developing indicators and monitors, data gathering, experimentation, quantitative assessment modelling, qualitative assessment, integrated tools and expert opinions.
	Scenario study	The performance of the alternative strategy is tested under several consistent and plausible pictures of how the future may unfold.
Interaction	Communicating uncertainties	communicating uncertainties from scientists to other actors in a policy debate allows other actors to co assess the quality of technical expertise and co produce the relevant evidence. Communication may also be aimed at raising awareness among actors.
	Persuasive communication	Convince others by presenting your perspective as attractive and worthwhile.

	Dialogical learning	Understanding one another's perspectives better through open dialog and by encouraging learning on all sides. Several authors advocate forms of dialogical learning as good strategy to reduce ambiguity, as it may lead to mutual understanding, trust and support for management action, or at least reduce resistance against actions. It may require a well founded process design and the involvement of facilitators and mediators.
	Negotiation	Reaching a mutually beneficial and integrative agreement that makes sense from multiple perspectives.
	Oppositional modes of action	Distancing and avoiding each other or trying to impose your perspective upon others by force.
Coping strategies	Preparing for the worst	Limiting potential negative consequences (controlling damage) of the worst case scenario, which means being conservative or precautious.

	Adopting robust solutions	Adopt strategies that perform well under multiple scenarios. This may mean adopting multiple measures or diversifying solutions to ensure that one or more will be effective under each of the possible scenarios.
	Developing resilience	Developing "the capacity of a system to absorb recurrent disturbances, such as natural disasters, so as retreating essential structures, processes and feedbacks".
	Adopting flexible solutions	Choosing flexible management strategies, which can be adapted to future changes. This may include adopting measures that are feasible within the timeframe of an unfolding potentially damaging event and that prevent or mitigate damage.

2.1.3 Uncertainty Classification and Taxonomy

Uncertainty is categorised usually according to its predictability into aleatory and epistemic. Aleatory is the unpredictable uncertainty, which happen because of sudden disruption. It is known within the project management context as "unknown unknown" as the planner or the decision maker cannot even realise that such uncertainty exist or such disruption may happen. On the other hand, Epistemic uncertainty is predictable as it results because of either missing or confusing knowledge. Epistemic uncertainty is described

within project management context as “known unknown”, where the missing information is already known to be missing, which lead to extra risks [84].

Aleatory uncertainty represents the variations related to the environment or the physical system under study [85–91]. Aleatory uncertainty problems are usually formulated using probability distributions, especially when there exist enough experimental data. The main challenge with probabilistic aleatory uncertainty is when the distribution is random or uniform as the first provides confusing information and the second distribution does not provide any useful information to deal with them. Aleatory uncertainty is objective, stochastic and usually irreducible.

Epistemic uncertainty is a predictable uncertainty that happens because of missing the information describing the system or the environment. Such term also refers to the missing knowledge and information in any phase or activity through the modelling, simulation or runtime process [92–95]. This implies that increasing the knowledge and information will lead to minimise the predicted uncertainty in system performance.

As lack of knowledge is the main reason for the epistemic uncertainty, many researchers tackled this issue from different perspectives similar to reliability analysis, design optimisation, risk assessment and decision making. Epistemic uncertainty is usually described as subjective uncertainty, reducible uncertainty or state of the knowledge uncertainty.

The term “Epistemic quantities” refers to the variables that have fixed values at the analysis, modelling and simulation time, but this fixed value is unknown. Sometimes, epistemic uncertainty is used to refer to the parametric uncertainty or the scientific uncertainty in the model - also known as model uncertainty because of their limited data, Information and knowledge.

Epistemic uncertainty can be a parametric uncertainty, where the estimates of the parameters are not known exactly or accurately. Epistemic uncertainty can also be model

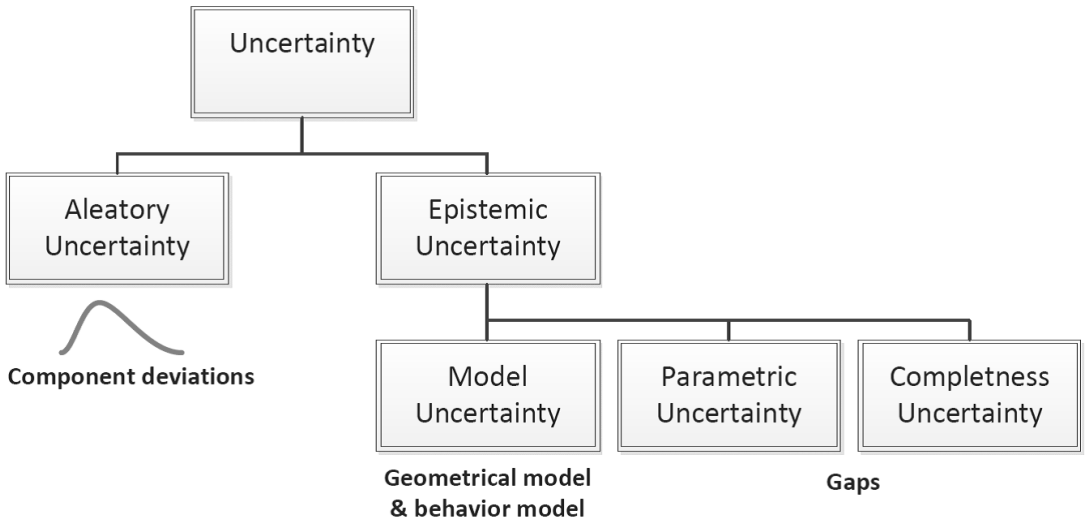

Figure 2.4: Classification of uncertainty according to its predictability

incompleteness, non-determinism in applying the model, or vagueness in engineering estimates, and ambiguity in the interpretation of results produced by a model [64].

Although epistemic uncertainty is totally different than aleatory uncertainty according to definition, in reality there exist common area and some similarities, which make the differentiation between them is a bit difficult. Both of aleatory and epistemic uncertainty are concerned with parameters, but in epistemic case the parameter is predictable. So, the researchers of the probability risk assessment proposed to divide the epistemic uncertainty into three main categories:

1. Model uncertainty, which can be structural uncertainty or behavioural uncertainty. Structural uncertainty is usually formulated using geometric modelling and behavioural uncertainty is usually formulated using algebraic models. Model uncertainty can be categorised to; (1) Fuzziness or vagueness of the model understanding (i.e., does the

model include or cover all the variables that can affect its results significantly), (2) Vagueness of the characteristics of the model, which refers to the relational uncertainties and the used descriptions in the model. Sometimes, the variables are known and well described in the model, but the relationships between them are not well stated.

2. Parametric uncertainty, where the parameters are known and predictable. It includes possible inaccuracies that happen because of the small datasets. It also includes uncertainties in the judgments of the experts regarding the parameter values, especially when the recorded data are limited.

3. Completeness uncertainty, where some of the significant properties, parameters or relationships are not included or not studied in the uncertainty analysis or the risk analysis. Completeness uncertainty is different than modelling uncertainty as it occurs at early stages in the uncertainty analysis or risk analysis. Completeness uncertainty is usually categorised to (1) Contributor uncertainties that consider if all serious risks and important accident have been covered and (2) Relationship uncertainty that check if the relations between all contributors and all variables have been considered.

2.1.4 Sources of Uncertainty

Uncertainty occurs in systematic models, mathematical models and the experimental measurements according to the problem nature and its context. The source of the uncertainty can be any factor affecting the reliability, the behaviour or the results of the model. The main sources of the uncertainty are parameter uncertainty, model structural uncertainty,

algorithmic uncertainty or experimental uncertainty. Some researchers consider the parametric variability and interpolation uncertainty, but parametric variability is considered as predictable parameter uncertainty and consider the interpolation uncertainty as algorithmic uncertainty.

Parametric Uncertainty

The uncertainty of the input parameters causes model uncertainty in terms of stability, reliability or accurateness of the results. The term parameter can refer to input parameter or configuration parameter or dependency parameter. The most famous example of such uncertainty is Heisenberg uncertainty where physicist cannot determine the speed of the electron and its location in same time. Another example is the exact speed at specific time for any wind or water current dependent system similar to aircrafts, wind farms, sail boats and free-fall experiment of a feather [96].

The main issue with parametric uncertainty is the uncertainty propagation and divergence. In other terms, if the uncertainty propagates as indicated in figure 2.3 by affecting the other variables in the system and converting them from certain to uncertain, it will lead to either instability of the system or inaccurate results. Such instability and inaccuracy are function of the uncertainty magnitude as well as the propagation factor. Also, if the uncertainty of the input parameter diverges along with the system performance, the final result will be very uncertain, which will cause high risk or incorrect decision [97–102].

As most of the complex systems in engineering, physics, business and economy are modelled using mathematics; most of the parameters are formulated as variables that affect the model with different levels [103, 104]. The effect of every parameter in the model behaviour and results is usually measured using sensitivity analysis [105, 106].

Sensitivity analysis help in determining, which parameters need more research to increase the knowledge base. Sensitivity analysis also helps in determining which parameters are useless or not effective for the model behaviour and results. Sensitivity analysis also detects to which extent the parameter uncertainty affects the output uncertainty. It also measures the correlation between input parameters and output, which help in understanding and developing the model [105–109].

Sensitivity analysis are applied through five main steps as follows: (1) Formulating the problem and building the model; (2) Defining the parameter variables and determine the dependent and independent ones; (3) Assigning probability density function to every input variable; (4) Build up a matrix using random sampling methods a in order to calculate the output vector; (5) Evaluate the effect and the correlation of each input/output pair [96, 110, 111].

Sensitivity analysis can be applied using many techniques according to the constraints and the settings of the problem and the model. The most frequently used techniques are:

1. One variable at time (OAT)
2. Local methods
3. Regression analysis
4. Variance-based methods
5. Screening

The main constraints and settings controlling which method to be selected are (1) Computational expenses (2) Correlated inputs (3) Linearity versus non-linearity (4) Model interactions (5) Multiple outputs and (6) The given data.

As there exist many approaches to execute the sensitivity analysis, most of them have been developed to solve a specific problem with specific constrains. According to the

mentioned constrains and the stated techniques, the process of sensitivity analysis consists of the following four steps:

1. Quantify the uncertainty in each input (e.g. ranges, probability distributions). Note that this can be difficult and many methods exist to elicit uncertainty distributions from subjective data.

2. Identify the model output to be analysed (the target of interest should ideally have a direct relation to the problem tackled by the model).

3. Run the model a number of times using some design of experiments that is dictated by the method of choice and the input uncertainty.

4. Using the resulting model outputs, calculate the sensitivity measures of interest.

In some cases this procedure will be repeated, for example in high-dimensional problems where the user has to screen out unimportant variables before performing a full sensitivity analysis.

Model Structural Uncertainty

Model uncertainty happens due to inadequacy or biasness or discrepancy in the structure of the model. Such uncertainty happens due to missing knowledge regarding the components of the system and the relation between these components. The structural uncertainty depends on the ability of the system to describe the different real life situations accurately.

The main sources of uncertainty in modelling, simulation, prediction or forecasting have been stated to be (1) Input parameters (2) Model conceptual structure and (3) runtime disruptions. Some researchers stated uncertainty in the model context, model assumptions, expert judgment and indicator choice [43].

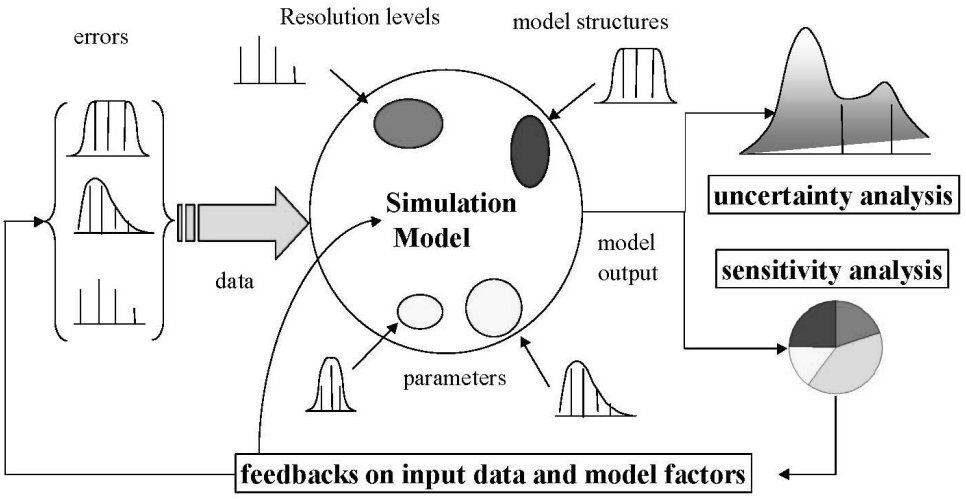

Figure 2.5: Sensitivity analysis and uncertainty analysis within a simulation model

The model is defined by Refsgaard *et al.* as an abstraction and conceptual interpretation of a real soft or hard structure [112]. Any mismatch between the real structure and the modelled one and any incompleteness in the model usually cause uncertainty in terms of model stability, reliability, performance and output.

The main issue facing researchers in modelling uncertain reality is the lack of useful observations in time and space, the transportation, the environment and eco systems are good examples. To work around this issue, researchers use techniques similar to extrapolating the unobserved features, forecast or predict the future behaviour as in eco systems [113–117].

Assessing structural uncertainty can be performed using several strategies according to the nature of uncertainty that can be incompleteness, inadequacy or lack of information. Such strategies are categorised to interpolation, which is used when the needed data exist

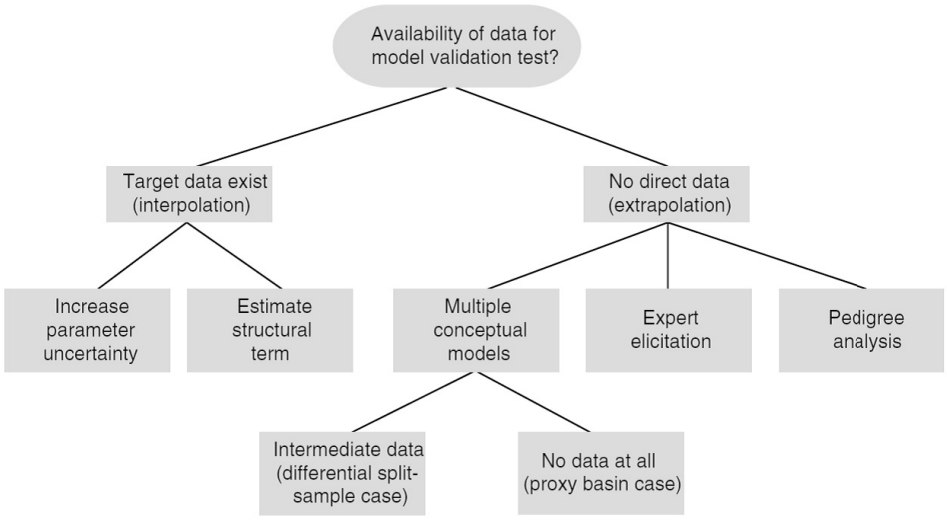

Figure 2.6: The uncertainty of the model structure according to data availability

or the extrapolation that is used when there is no enough data. Each of the two technique cover specific cases as declared in figure 2.6

Klemes defined a test scheme, where the interpolation is used when the split-sample test is applicable [118, 119]. He stated that extrapolation is used in case there is no data for the studied output variable, which is known as proxy-basin test.

Uncertainty interpolation is used when there is enough historical data to predict the missing data with a relatively small margin of error. In such case, a calibration is performed using a sample of the existing data to make sure that the model parameters are optimised. Afterwards the predictions are compared against the rest of the existing data, as a step to measure the deviations. Such deviations are considered the model's conceptual error. This

error can be used later on to re-optimise the model structure parameters.

There are many techniques or strategies to apply interpolation to solve structural or conceptual uncertainty. One of them increases the parameter uncertainty to a level equals to the omitted error of the model structure similar to Van Griensven and Meixner [120].

Another approach is estimating the structural uncertainty and its effects on the predictions of the model. Such approach assumes that the uncertainties from different sources are additive, regardless the difference of the nature or the causes. Such assumption is debatable as the combination of uncertainties is non-linear due to relation between variables, input parameters, configuration and system structure. Such relations can be interaction, dependencies and correlations especially if the system is based on artificial intelligence or heuristics.

The second assumption is that the differences between the predicted values and the observed values are always result of a structural error and excluding other types of uncertainty similar to parametric errors or runtime disruptions and excluding the error in the reality observation. Radwan *et al.* [121], uses statistical analysis and of the residuals between the model prediction and the observation to estimate the total predictive uncertainty.

Vrugt *et al.* [122] present another stochastic approach that is based on a simultaneous parameter optimisation and data assimilation with an ensemble Kalman filter. By specifying values for measurement error and a so-called stochastic forcing term, representing structural uncertainty, they are able to estimate the dynamic behaviour of the model structure uncertainty.

Uncertainty extrapolation is used when the model structure error cannot be estimated because of the shortage of the needed data. Extrapolation of the model can be applied by analysing other similar conceptual models that have been used for similar system structure. Then develop, build, generate and tune the old model, taking into consideration the expert

elicitation and pedigree analysis.

Multiple conceptual models approach is also called scenario based modelling, alternative or competitive conceptual modelling. Such approach uses a readymade model and upgrades it according the considered scenarios when upgrading the conceptual model. The uncertainties of the model input and the model parameters should be analysed, then measure the deviation between the predicted results of the model and the actual results to estimate the model structure uncertainty.

The idea of using alternative or competing candidate model structures is used in modelling the water quality [123]. It has also been used in flood forecasting by Butts *et al.* [124] and In groundwater modelling-based on different geological interpretations [125–127]. The alternative modelling is also used frequently in the climate change modelling [128].

Van Straten and Keesman [129] noted that good performance at the calibration stage does not guarantee correctly predicted behaviour, due to non-stationarity of the underlying processes in space or time. Butts *et al.* [124] found that exploring an ensemble of model structures provides a useful approach in assessing simulation uncertainty. Hjberg and Refsgaard [130] concluded that the larger the degree of extrapolation, the more the underlying conceptual model dominates over the parameter uncertainty and the effect of calibration.

Expert elicitation can be used as a supporting method for alternative conceptual modelling to analyse uncertainty. It is a structured process to elicit subjective judgments and ideas from experts. It is widely used in uncertainty assessment to quantify uncertainties in cases where there is no or too few direct empirical data available to infer uncertainty. Usually the subjective judgment is represented as a probability density function reflecting the experts degree of belief.

Expert elicitation aims to specify uncertainties in a structured and documented way, ensuring the account is both credible and traceable to its assumptions. Typically it is applied in situations where there is scarce or insufficient empirical material for a direct quantification of uncertainty. An example with use of expert elicitation to estimate probabilities of alternative conceptual models is given by Meyer *et al.* [131].

Expert elicitation can also be used to generate ideas for alternative causal structures (conceptual models) that govern the behaviour of a system. Techniques used in decision analysis include group model building [132] and the hexagon method [133] but these techniques usually aim to achieve consensus. From the point of view of model structure uncertainty, these elicitation techniques can perhaps be used to generate alternative conceptual models.

Pedigree analysis idea came from Funtowicz and Ravetz [134], who note that statistical uncertainty in terms of inexactness does not cover all relevant dimensions of uncertainty, including the methodological and epistemological dimensions.

To promote a more differentiated insight into uncertainty they propose to extend good scientific practice with five qualifiers for quantitative scientific information: numeral unit, spread, assessment, and pedigree (NUSAP).

By adding expert judgment of reliability (assessment) and systematic multi-criteria evaluation of the processes by which numbers have been produced (pedigree), NUSAP has extended the statistical approach to uncertainty (inexactness) with the methodological (unreliability) and epistemological ignorance dimensions.

By providing a separate qualification for each dimension of uncertainty, it enables flexibility in their expression. Each special sort of information has its own aspects that are key to its pedigree, so different pedigree matrices using different pedigree criteria can be used to qualify different sorts of information.

Early applications of pedigree analysis of environmental models have focused on parameter pedigree, using proxy representation, empirical basis, methodological rigor, theoretical understanding and validation as pedigree criteria. Later on, pedigree analysis has been extended to assessment of model assumptions and problem framing [135, 136].

Runtime or Behavioural Uncertainty

Another type of uncertainty is the runtime uncertainty or behavioural uncertainty. This happens when the system is running and sudden unexpected event happens that affects the performance of the system. Such event is called disruption or disturbance in engineering domain and is called exception in computer science and information technology domain. One example is when a lightning strikes electricity tower, it is totally unexpected but it can affect factories, hospitals and electric trains. Another example is when a problem is solved using cloud computing and the internet connection is down.

Such unexpected events can be solved using reactive or corrective actions after the accident happens as it is so difficult to predict if such disruption can happen and when. Recent researchers uses hybrid technique where they try to predicate if such event will happen and get prepared for when it happens. The most famous example for the hybrid techniques is the ambulance, where they take specific spots within the city so they can access any location then go to the hospital within a very short time. The density of the ambulance cars and the distance between them enhances the performance in terms of the needed time to transfer the patient. The runtime uncertainty become more challenging when the event are not frequently happening such as the tsunami and the cyclone.

Other Uncertainty Sources

Algorithmic Uncertainty usually refers to the numerical uncertainty, which comes from numerical errors and numerical approximations per implementation of the computer model. Many models are too complicated to be solved exactly, accurately and complete. For example the finite element method or finite difference method may be used to approximate the solution of a partial differential equation, which causes numerical errors. Other examples are numerical integration and infinite sum truncation that are necessary approximations in numerical implementation.

Experimental Uncertainty happens because of the observation error, which comes from the variability of experimental measurements. The experimental uncertainty is inevitable and can be noticed by repeating a measurement for many times using exactly the same settings for all inputs/variables.

Completeness Uncertainty describes the missing parts in the process of modelling and simulation, which can be missing structural component or missing parameter or missing variable. Such uncertainty happens because of the lack of information or optional neglection of some details due to the cost of money or time. Increasing the level of modelling details causes facing more uncertainty and more accurate results.

The most well-known example of completeness uncertainty is the calculation precision in terms of the number of decimal points that will be used in the calculation. Increasing the number of decimal points for the variables will result in a very accurate output but with relatively huge computation and processing time. If the variables are approximated to integer numbers, the algorithm will be faster with easy calculation but with inaccurate results.

Some researchers consider the completeness uncertainty as structural uncertainty. This is true to some extent, but modelers should consider the completeness uncertainty of the

variables, parameters, environmental factors and processing. This implies that structural completeness uncertainty is a special case of the general completeness uncertainty.

Scheduling is a decision making process for allocating limited amount of resources to multiple tasks or jobs within a specific environment and constraints. If the available resources are much more the needed amount by all tasks at any time, there will be no need for scheduling. The main target of scheduling is to allocate a specific resource to each job at a specific time slot, in order to optimise the objective function. Scheduling is needed in many domains similar to manufacturing, supply chain, queuing and traffic control.

The resources and tasks can take many different forms including machines in a production line, airport gates or runways, construction crew, CPU in a computing. The tasks or jobs can be operations in an operating room, assembly in production process, landings or takeoffs of aircrafts in an airport, milestones in a development project, running computer instructions in parallel environment, and so on. Every job can have a specific priority, release time, actual starting time and a due date. The objective function of the scheduling process can also take many different forms. One objective can be minimising the total completion time of all tasks or jobs. Another objective can be minimising the total number of tasks or jobs completed after their planned due-date.

Most of the manufacturing systems, production systems, supply chains, transportation, traffic, processing units and information technology environments depend heavily on sequencing and scheduling algorithms.

2.2 Scheduling Review

2.2.1 Scheduling Problems Formulations

Scheduling is described technically as the process of ordering a set of tasks or jobs in a sequence that optimises the objective function or the cost function. Every job has multiple properties including the release time, processing time, due date, deadline, dependency and pre-emption. Some or all of the job properties are known before generating the schedule and using it in run time. The environment of the problem is usually stated in the problem formulation as well. Scheduling environment describes the machines properties and the constraints. Every scheduling problem must have an objective function that can be single or multi objective. The triplet of the parameters, environment and the objective function are usually described as $\alpha \mid \beta \mid \gamma$.

Every scheduling problem has a number of jobs; each job has a set of properties that describe the job itself or the relation between the job and its peers. The main properties describing the job are the release date, processing time, due date and the weight. Such properties are usually known at setup time, but sometimes they change at runtime due to some disturbance or disruption. Such change causes the system to be dynamic rather static.

The second factor describing the scheduling problem is the environment in terms of number of machines and constraints. The environment of the problem can have single or multiple machines, sequential or parallel, same speed or different speed related or unrelated machines and always up or temporary working. The constraints can be on the job level or on the machine level. The constraints can be the pre-emption, precedence, sequence dependent setup times, batch processing and eligibility criteria.

Scheduling objective is the cost function to be optimised as minimisation, maximisation or constraint satisfaction. The solution of the problem can be optimum if it is solvable

in time, which happens if the problem is computationally Turing reducible in polynomial time. Any problem can be Turing reducible if it is deterministic polynomial (P) or non-deterministic polynomial (NP) excluding the non-deterministic polynomial complete problems and the non-deterministic polynomial hard problems.

The solution of the scheduling problem can be optimal if it is solved using heuristics. Heuristics and meta-heuristics are used when the problem is not solvable in time because of being NP-Complete or NP-hard. Such techniques finds a feasible solution for the problem, but the solution is not necessarily the best or the optimum. Scheduling objective functions vary according to input parameters and constraints. It can be any of the following make-span, maximum lateness, total completion time, total weighted completion time, total tardiness, total weighted tardiness and weighted number of tardy jobs.

2.2.2 Scheduling Classification and Taxonomy

Sequencing and scheduling problems can be formulated in different ways and can be solved using different techniques. Many classifications for scheduling techniques have been proposed by many researchers in order to find out the best formulation and technique for any scheduling problem [137–140]. One of the most recognised and accepted ways to classify scheduling problems is the number of machines, which affects sequential and parallel classification too. Another classification criterion is the behaviour of machines, which specifies if the problem is deterministic or stochastic. A third criterion is the behaviour of the parameters, which make the problem either static or dynamic [137, 141].

Scheduling problems also can be classified based on its complexity and what is the best technique to solve such problems. The main classes are polynomial solvable problems and nondeterministic polynomial hard problems (NP-hard). polynomial solvable problems can be solved using polynomial time algorithms. NP-hard problems can be solved using

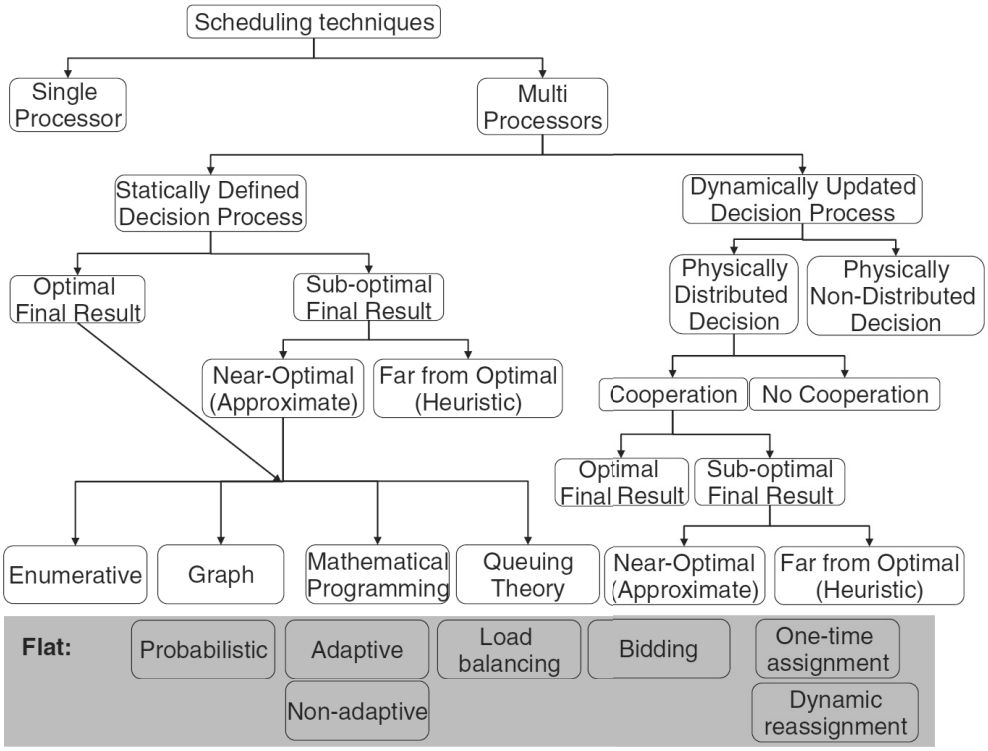

Figure 2.7: The taxonomy of the scheduling techniques for the distributed resource management scheduling problem consists of interacting classes and policies presented by Casavant and Kuhl [142]

either the exact approaches including branch and bound and dynamic programming or the approximation and heuristic approaches. The figure 2.8 explains the complexity-based classification of scheduling problems. The table 2.9 refers to some scheduling problems, which are solvable in polynomial time, including the problem formulation as $\alpha \mid \beta \mid \gamma$ as well as the algorithm complexity and the references. The table 2.10 states some scheduling problems that are nondeterministic polynomial and solved using exact methods, heuristics

and meta heuristics.

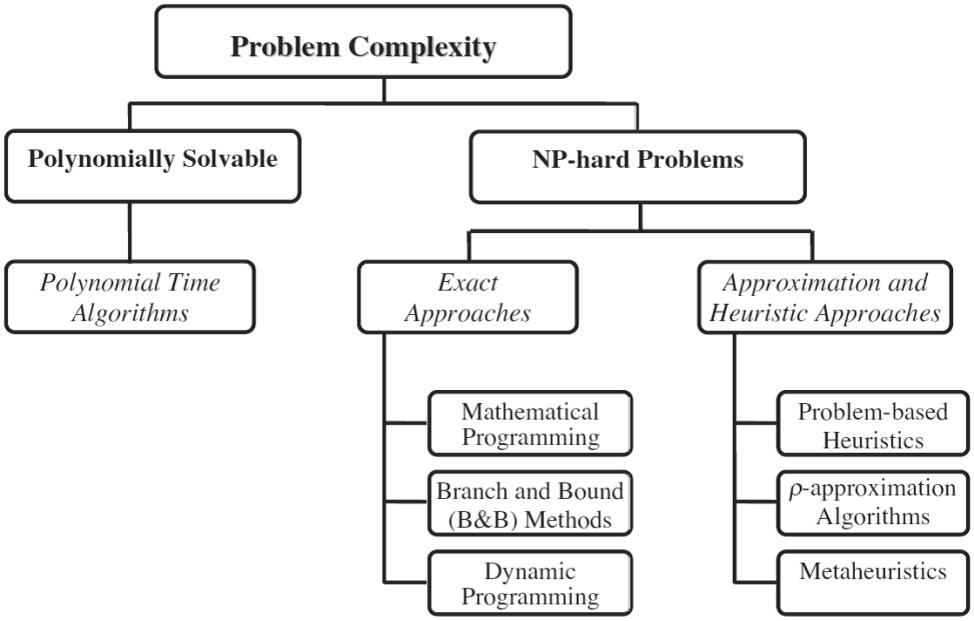

Figure 2.8: Classification of the scheduling problems based on the complexity and the solution techniques

Deterministic and Stochastic Scheduling

The behaviour of the machine at the runtime is one of the criteria used for classifying scheduling problems, as it can be either predictable or unpredictable. The predictable behaviour of the machine along the schedule life time makes the problem deterministic unless there exit some constraints that increases the complexity. If the behaviour is not predictable the scheduling problem needs a stochastic solution [143–145].

Reference	Problem classification	Algorithm complexity
Garey and Johnson (1975)	$P2\|res\cdots, p_i = 1\|C_{\max}$	$O(n^3)$
Blazewicz (1978)	$P\|res1{\cdot}1, r_i, p_i = 1\|C_{\max}$	$O(n)$
Blazewicz (1979)	$P\|res1{\cdot}1, p_i = 1, r_i, d_i\|\varnothing$	$O(n^2)$
Blazewicz et al. (1983)	$Q2\|res1{\cdot}{\cdot}, p_i = 1\|C_{\max}$	$O(n\log n)$
Blazewicz and Ecker (1983)	$P\|res\lambda\sigma\delta, p_i = 1\|C_{\max}$	$O(\log n)$
Blazewicz et al. (1987)	$P\|res1{\cdot}1\|\sum C_i$	$O(n^3)$
Blazewicz et al. (1989)	$P\|res\cdots, \text{types} = R, p_i \leqslant p\|C_{\max}$	$O(n^{R(p+1)})$
	$Pm\|res\cdots, \text{types} = R\|C_{\max}$	$O(n^{R(m+1)})$
Brucker and Kramer (1996)	$P\|res\cdots, \text{types} = R, p_i \leqslant p\|\sum w_iC_i, \sum T_i$	$O(R(p+u)n^{Rp} + R^2pn^{R(p+2)})$
	$P\|res\cdots, \text{types} = R, p_i \leqslant p, r_i\|C_{\max}, \sum C_i$	$O(R(p+u)n^{Rp} + R^2pn^{R(p+2)+1})$
	$P\|res\cdots, \text{types} = R, p_i \leqslant p\|\sum w_iU_i$	$O(R(p+u)n^{Rp} + R^2pn^{R(p+2)})$
	$Pm\|res\cdots, \text{types} = R\|\sum w_iC_i, \sum w_iU_i, \sum T_i$	$O(R^m\ mu + m(m+R)R^{m+1}\ n^{R(m+1)})$
	$Pm\|res\cdots, \text{types} = R, r_i\|C_{\max}, \sum C_i$	$O(R^m\ mu + m(m+R)R^{m+1}\ n^{R(m+1)+m})$
Daniels et al. (1996)	$PD2\|res1{\cdot}{\cdot}\|C_{\max}$	$O(n\log n)$
	$PD\|res1{\cdot}{\cdot}, Stc\|C_{\max}$ (Static PMFRS)	$O(nb(n+m))$
Kovalyov and Shafransky (1998)	$P\|res1{\cdot}1, p_i = 1\|C_{\max}$	$O(1)$
	$Q\|res1{\cdot}1, p_i = 1, nmit\|C_{\max}$	$O(m\log m)$
Ventura and Kim (2000)	$P\|res1{\cdot}1, p_i, d_i = d\|TAD$	$O(n^4)$
Ventura and Kim (2003)	$P\|res1{\cdot}1, r_i, p_i = 1, d_i\|TADD$	$O(n^4)$
Kellerer and Strusevisch (2003)	$PD2\|res111\|C_{\max}$	$O(n)$
Kellerer and Strusevisch (2004)	$PD2\|res1{\cdot}{\cdot}\|C_{\max}$	$O(n\log n)$
	$PD2\|res211\|C_{\max}$	$O(n)$

Figure 2.9: Polynomially solved scheduling problems as stated by Edis 2013 [14]

Deterministic Scheduling Deterministic scheduling is used for the problems that can be solved at design time, using a sequence of known steps, in order to find an optimal or near-optimal solution. Such problems do not have any have infrequent mapping of the setup time on the single or the multiple machines at runtime [137, 145].

Deterministic scheduling uses stepwise techniques and branch and bound algorithms, it does not use any randomization techniques. The stepwise branch is divided into the pruning of solution space and the iterative classes. The branch and bound approach is considered as a recursive solution, it can be adaptive or rigid.

An example of the pruning techniques is the partitioning technique that divides the overall problem into smaller sectors based on specific rules, for example, the operations are

Reference	Problem classification	Solution method
Krause et al. (1973)	$P\|res1\cdot\cdot, p_i = 1\|C_{max}$	Polynomial-time (4/3)-approximation algorithm
Blazewicz et al. (1993)	$P2\|res1\cdot\cdot, p_i = 1\|L_{max}$	Problem-based heuristic algorithms
	$P2\|res\cdot\cdot\cdot, p_i = 1\|L_{max}$	A branch and bound algorithm
Daniels et al. (1996)	$PD\|res1\cdot\cdot, Int\|C_{max}$ (Dynamic PMFRS)	A branch and bound algorithm
		A static-based heuristic (SBH)
Daniels et al. (1997)	$PD\|res1\cdot\cdot, Int\|C_{max}$ (Dynamic PMFRS)	A tabu search heuristic
		A static-based tabu search heuristic
Srivastav and Stangier (1997)	$P\|res\cdot\cdot1, r_i, p_i = 1\|C_{max}$	A polynomial-time approximation algorithm
Daniels et al. (1999)	$P\|res1\cdot\cdot, Stc\|C_{max}$ (Static UPMFRS)	A decomposition heuristic (using SBH)
		A tabu search heuristic based on SBH
Olafsson and Shi (2000)	$PD\|res1\cdot\cdot, Int\|C_{max}$ (Dynamic PMFRS)	A heuristic, named Nested Partitions (NP)
Kellerer and Strusevisch (2003)	$PDm\|res111\|C_{max}$	O(mn) Group Technology approximation alg.
	$PD3\|res111\|C_{max}$	O(n) heuristic approximation algorithm
	$PD4\|res111\|C_{max}$	O(n) heuristic approximation algorithm
	$PDm\|res111\|C_{max}$	PTAS
Li et al. (2003)	$PD\|res1\cdot\cdot\|C_{max}$	A genetic algorithm (GA)
Ventura and Kim (2003)	$P\|res\cdot\cdot\cdot, r_i, p_i = 1, d_i\|TADD$	Lagrangian-based heuristic algorithm
Kellerer and Strusevisch (2004)	$PD\|res\lambda11\|C_{max}$	An O(nm min $\{n, m\}$) greedy approximation alg.
	$PDm\|res\lambda11\|C_{max}$	PTAS
Grigoriev et al. (2005)	$R\|res1\cdot\cdot, Int\|C_{max}$ (Dynamic UPMFRS)	$(4 + 2\sqrt{2})$-approximation algorithm
	$PD\|res1\cdot\cdot, Int\|C_{max}$ (Dynamic PMFRS)	$(3 + 2\sqrt{2})$-approximation algorithm
Kumar et al. (2005)	$R\|res1\cdot\cdot, Int\|C_{max}$ (Dynamic UPMFRS)	4-approximation algorithm
Grigoriev and Uetz (2006)	$PD\|res1\cdot\cdot, Lin\|C_{max}$ (Dynamic PMFRS)	A quadratic IP-based $(3 + \varepsilon)$-approximation algorithm
Grigoriev et al. (2006)	$R\|res1\cdot\cdot, Int\|C_{max}$ (Dynamic UPMFRS)	3.75-approximation algorithm
Ruiz-Torres and Centeno (2007)	$P\|res1\cdot\cdot, Stc\|C_{max}$ (Static UPMFRS)	Problem-based heuristic algorithms
Ruiz-Torres et al. (2007)	$Q\|res1\cdot\cdot, Stc\|\sum U_i$ (Static UPMFRS)	Five different problem-based heuristic algorithms
Edis et al. (2008)	$P\|res1\cdot2, M_i, p_i = 1\|\sum C_i$	Lagrangian and problem-based heuristic alg.
Kellerer (2008)	$P\|res1\cdot\cdot, Int\|C_{max}$ (Dynamic UPMFRS)	$(3.5 + \varepsilon)$-approximation algorithm
Kellerer and Strusevisch (2008)	$PD2\|res111, Bi\|C_{max}$ (Dynamic PMFRS)	A pseudo-polynomial-time dynamic prog. alg.
	$PD2\|res111, Bi\|C_{max}$ (Dynamic PMFRS)	FPTAS
	$PD\|res111, Bi\|C_{max}$ (Dynamic PMFRS)	Polynomial-time (3/2)-approximation algorithm
	$PD\|res1\sigma\sigma, Int\|C_{max}$ (Dynamic PMFRS)	Polynomial-time $(3 + \varepsilon)$-approximation alg.
	$PDm\|res111, Bi\|C_{max}$ (Dynamic PMFRS)	PTAS
Grigoriev and Uetz (2009)	$PD\|res1\cdot\cdot, Int\|C_{max}$ (Dynamic PMFRS)	A (nonlinear) IP-based $(3 + \varepsilon)$-approximation alg.
Sue and Lien (2009)	$P\|res1\cdot\cdot, Stc\|C_{max}$ (Static UPMFRS)	Problem-based heuristic algorithms
Edis and Ozkarahan (2011)	$P\|res1\cdot2, M_i\|C_{max}$	IP, CP and IP/CP combined approach
Edis and Oguz (2011)	$R\|res1\cdot\cdot\|\sum C_i$	IP, CP, Lagrangian-based CP approach
Xu et al. (2011)	$P2\|res111, Bi\|C_{max}$ (Dynamic UPMFRS)	FPTAS
Edis and Ozkarahan (2012)	$P36\|res1\cdot2, M_i\|C_{max}$	IP/IP and IP/CP sequential approaches
Edis and Oguz (2012)	$PD\|res1\cdot\cdot, Int\|C_{max}$ (Dynamic PMFRS)	IP/CP sequential approach
	$R\|res1\cdot\cdot, Int\|C_{max}$ (Dynamic UPMFRS)	IP/CP sequential approach

Figure 2.10: A set of complicated scheduling problems, which are solved using exact, approximation, heuristic and meta-heuristic techniques as stated by Emrah B. Edis 2013 [14]

partitioned to critical path and noncritical path ones [146]. The Integer Linear Programming (ILP) is an example of the iterative techniques. ILP is used to analyse the constraints

through polyhedral theory that determines the scheduling polytope to be solved [147]. The solver class provides the actual solver of the formulated problem, such as a Mixed Integer Linear Programming Solver (MILP) [148]. The mentioned techniques are efficient in solving the scheduling problems that are sufficiently decreased in size, have a largely linear solution space, and require a guarantee on optimality.

The adaptive class defines the recursive techniques, which are capable of self-training and self-learning. They start from an initial state and they indicate when and how the applied process is modified and when it converges to the near-optimal solution. For example, some adaptive deterministic techniques use neural networks [149], which are composed of the net topology, the node characteristics, and a set of training rules indicating the initial weights and their adaptation [150].The performance of each technique varies according to the problem to be solved [151] and according to the effectiveness of the deterministic technique's nature to this type of problem [152].

Many of the scheduling problems are considered deterministic similar to total completion time, number of tardy jobs, total earliness and lateness, maximum lateness, multiple objectives on one machine, makespan with sequence dependent setup time, batch processing, flowshop, jobshop and openshop problems [153, 154].

Stochastic Scheduling Scheduling problem is considered stochastic if its behaviour is not predictable at run time due to changes in the task parameters or machine performance. The stochastic techniques start from an initial state and use probabilities and randomness to determine the next state, which is potentially closer to the near-optimal solution. The next state generation iterates until some termination criteria are met. These usually indicate that the obtained solution has reached the near-optimal one or that the search for a better solution is too costly, whereas the gained quality is too low; accordingly, the process terminates. The way of generating subsequent states and which objective function

is minimised determines the quality of the obtained solution. The main techniques used for stochastic scheduling are tabu search, simulated annealing, Pareto-optimal and genetic algorithms [137].

The need for stochastic models arises with the lack of knowledge either in the input parameters or the constraints. Such lack of knowledge can happen in the processing times, the release dates or the due dates. The unknown variable is usually represented as random variable of some distribution. The actual estimate of the variable (e.g. the processing time) becomes known by the processing completion; the actual value of the start date or the completion time becomes known only time when the task is finished. Distributions and density functions of the random variable for the uncertain variables may take many forms. The describing function can be either probability or membership or mathematical, it can be represented as continuous function over specific intervals or as a discrete function with concentration at specific points, which imply that the describing function cannot be differentiated by time [155, 156].

Stochastic scheduling problems includes problems with arbitrary processing, priority queues, work conservation, Poisson releases, parallel make-span, parallel total completion time, parallel make-span with pre-emption, parallel total completion time with pre-emption, flow shop with variable storage and flow-shop with blocking.

Although stochastic models are used for any scheduling problem with instability of the behaviour, it is usually used for the instability in machine or processor performance due to any sudden disruption or disturbances. Meanwhile the fluctuations in the input parameters are usually handled in the phase of the data preparation [157–159].

Static and Dynamic Scheduling

The scheduling system is called static when the number of the available jobs does not change by time. Sequencing and scheduling problems depend heavily on sorting algorithms, which implies the static problems will be easier to solve as the ranking and ordering process will be done once and does not need to be changed by every new job appear in the job list [160].

Although most of the scheduling problems in real life should be modeled as dynamic problems, static models can be used to formulate the dynamic problem after simplifying it by relaxing some constraints. Static models contributes frequently in finding the heuristic principles that can help in the dynamic situations.

Many researchers simplify the complicated dynamic scheduling problems to be static problems with a relaxed environment as a step to find the complicated solutions. They use it as a heuristic base to provide a base plan that can be dynamically updated later [160, 161]. For example, airport schedules are planned in advance in a static way, but the sequence of departures and arrivals is subject to dynamic scheduling in reality.

The system is called dynamic, when new jobs appear over time, or the properties of the jobs change through time. Dynamic environments usually face inevitable unpredictable real-time events, which may cause a change in the scheduled plans. Such change in the plans makes a previously feasible schedule turn into infeasible. Dynamic problems are usually correlated with uncertainty, so it is categorised to three main categories: completely reactive scheduling, predictive-reactive scheduling, and robust pro-active scheduling [155, 162–166].

Real-time scheduling events, also known as schedule disturbances, have been categorised to two main categories as follows [164, 166–168]:

- Job-based: sudden urgent jobs, cancelled jobs, changes of due-dates, early or late

arrival of any job, change in priorities, change in weights and changes in processing time.

- Resource-based: machine breakdown, unavailability or failure of any tool, operator absence, loading limits, delay in the arrival and shortage of materials.

Dynamic scheduling can be solved using many techniques according to the problem formulation. One of the ways to solve dynamic scheduling problem is to decompose it into a series of static problems that can be solved by using classical scheduling algorithms. Another approach is to use heuristics such as the right-shift scheduling [162, 169, 170], the match-up scheduling [171, 172] and the dispatching rule-based rescheduling by [173, 174]. A third approach is to use meta-heuristics including tabu-search, simulated annealing and genetic algorithms [162, 175, 176]. A fourth approach is to use multi-agent technique with its different architectures similar to autonomous and mediator architectures, [177–180]. Finally, to use artificial intelligence techniques such as knowledge-based systems, neural networks, case-based reasoning, fuzzy logic, and Petri nets.

Sequential and Parallel Scheduling

Scheduling problems are classified according to the number of the machines used and the relations between those machines. The scheduling problem can be of one or multiple machines. Most of the scheduling problems consist of multiple machines. Nevertheless many scheduling problems are formulated using a single machine as a step to solve the scheduling problems with multiple machines. Such simplification technique is useful with dynamic and stochastic scheduling problems [181, 182].

Most of the multiple-machine-based scheduling problems can be either sequential or parallel. Sequential scheduling can be represented as a queue with a single path of dependencies were machine number $n + 1$ depends on machine number n. The main concerns

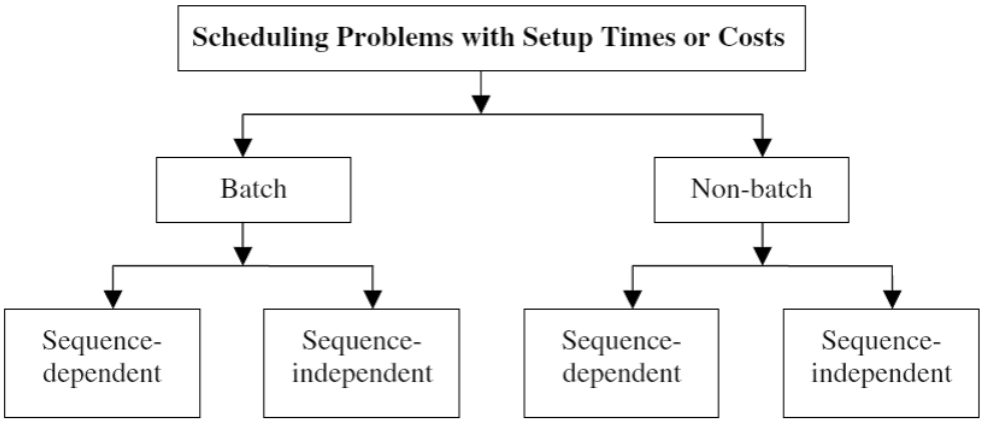

Figure 2.11: Classification of scheduling problems based on setup time and cost [181]

with sequential scheduling is the release time, due-date and the deadline as well as any runtime disturbances. A sequence usually corresponds to a permutation of the n jobs or the order, in which jobs are to be processed on a given machine. One of the well-known problems with sequential processing is to optimise the traveling sales man problem, other sequential processing problems include the *sequence dependent setup times*, *flowshop with sequence dependent setup times*, *flowshop with dependencies*, *jobshop with sequence dependent setup times*, *jobshop with dependencies* and *batch processing* [141, 181].

On the contrary, parallel scheduling has multiple paths of execution and possible dependencies and different rate of production. Some other factors affect parallel scheduling similar to load balancing, pre-emption, dependencies and batch processing.

Parallel machines can be:

- Parallel identical machines
- Parallel machines with different speeds

- Unrelated machines in parallel

Every parallel scheduling problem is identified by a set of characteristics that determine how to formulate the problem and what is the most suitable technique to solve it [182]:

1. Number of added resources and their types.
2. Effect of the extra resources on the processing times for every machine, considering that:
    - The total processing time decreases by increasing amount of the additional resources allocated to every single machine (speeding-up)
    - The resource requirements of the jobs are fixed with a known priori.
3. How the resources will be allocated, static or dynamic:
    - Static: Each machine can use a predetermined amount of the static extra resource.
    - Dynamic: Different machines can use the additional resources during the schedule by switching the allocation.
4. Job-machine assignment that can be either pre-specified or unspecified

2.2.3 Scheduling Challenges

Scheduling problems face many challenges that make finding the optimum solution is difficult rather impossible. Such challenges lead many problem solvers to find the optimal solutions, which are good enough with acceptable known margin of error. The main challenges facing the scheduling problems are the computational complexity and the uncertainty. Computational complexity makes the problem unsolvable in time or solvable with

a margin of error using heuristics. Uncertainty of the scheduling problem creates uncertainty of the output, extra computational complexity, additional risks, additional cost or all of them combined.

Computational Complexity

Scheduling problems have different formulations according to the objective formulation, the given parameters and the environment variables and constraints. The complexity challenge appears with mathematical solutions in formulating the complicated problems. Another kind of complexity happens with branch and bound algorithms, where the algorithm has high order of magnitude leading to huge processing time. The third type of complexity is in predicting the solution behaviour in stochastic and agent-based problems

Scheduling problem can be solved using standard mathematics similar to linear algebra or integer programming. Mathematical techniques can find the optimum solutions but it cannot solve all scheduling problems. Formulating the complicated scheduling problems mathematically is difficult, especially if we considered the precedence, constraints, preemption and the dependencies.

Scheduling problems can also be solved using branch and bound algorithms including state space search and heuristics. The problem with branch and bound solution is transforming the problem to mathematical combinatorial in the basic cases where the branch selection and pruning is either depth first or breadth first. Mathematical combinatorial takes very long time for processing if the number of tasks or jobs is more than 20 as its number of iteration is estimated by factorial (20) and its complexity curve grows rapidly in what is known by combinatorial explosion. In other words the order of magnitude of the standard branch and bound algorithms is exponential, assuming worst case scenario. Branch and bound algorithms have been combined with heuristics that can determine which branches

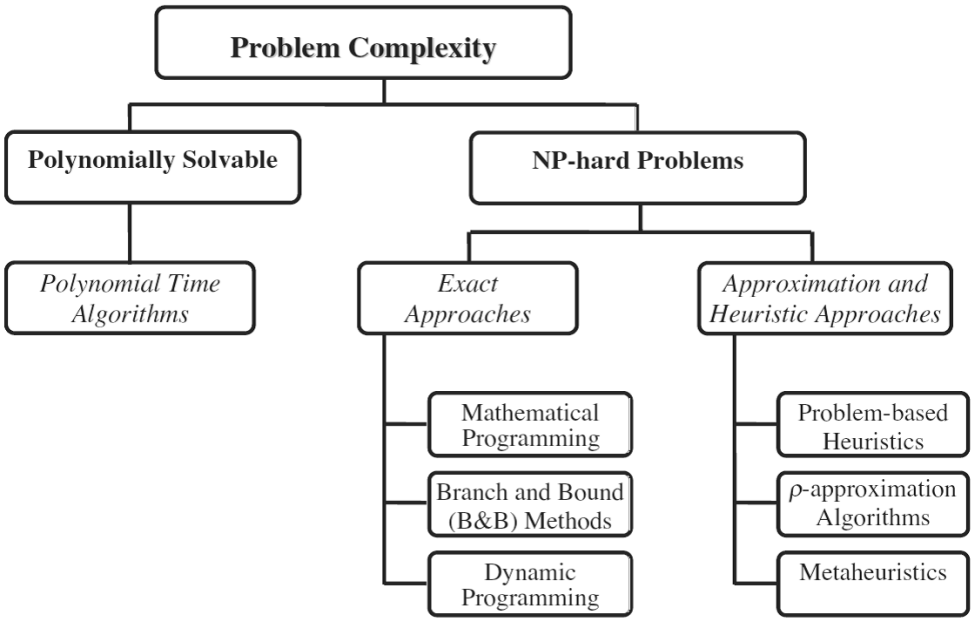

Figure 2.12: Classification of the scheduling problems based on the complexity and the solution techniques

are more likely to have the solution and which branches to be pruned. Merging heuristics with branch and bound algorithms make the solution optimal rather than optimum.

The third approach to solve scheduling problems is the meta-heuristics techniques and machine learning. This approach includes genetic algorithms, neural networks, simulated annealing and agent-based programming. The challenge with this approach is the difficulty of predicting the system behaviour, estimating the order of magnitude and training the system. Such techniques need lots of training with different datasets that can be either real or generated, provided that the data covers the real problem scenario or its distribution.

Problem description	Additional characteristics	Reference
Job availability		
$1/\mathrm{ST}_{si,b}/\sum(w_i)C_j$		Potts and Kovalyov (2000)
$Pm/ST_{si,b}/\sum C_j$	m is constant	Potts and Kovalyov (2000)
$P/\mathrm{ST}_{si,b},p_j=p,C_j\leqslant d_j/-$	Equal setup times s, s is not a multiple of p	Brucker et al. (1998)
$Q_m/\mathrm{ST}_{si,b},p_j=p,C_j\leqslant d_j/-$	Equal setup times s, s is not a multiple of p, constant m	Brucker et al. (1998)
$F2/\mathrm{ST}_{si,b}/C_{\max}$	Machine independent setup times	Kleinau (1993)
$^{*}F2/\mathrm{ST}_{si,b}/C_{\max}$		Kleinau (1993)
$O2/\mathrm{ST}_{si,b}/C_{\max}$	Machine independent setup times	Kleinau (1993)
$^{*}O2/\mathrm{ST}_{si,b}/C_{\max}$		Kleinau (1993)
$^{*}O2/\mathrm{ST}_{si,b}/C_{\max}$	Group technology assumption	Blazewicz and Kovalyov (2002)
Batch availability		
$1/\mathrm{ST}_{si,b}/\sum(w_i)C_j$		Cheng et al. (1994)
$1/\mathrm{ST}_{si,b}/\sum w_jC_j$	One family, batch setup and processing times are equal to the maximum of job setup and processing times in the batch	Dang and Kang (2004)
$1/\mathrm{ST}_{si,b}/\sum C_j$	One family, resource dependent processing times	Ng et al. (2003a,b)
$1/\mathrm{ST}_{si,b}/L_{\max}$	One family, bounded batch sizes	Cheng and Kovalyov (2001)
$1/\mathrm{ST}_{si,b}/\sum U_j$	One family, bounded batch sizes	Cheng and Kovalyov (2001)
$1/\mathrm{ST}_{si,b},p_j=p/\sum C_j$	One family, bounded batch sizes	Cheng and Kovalyov (2001)
$P/\mathrm{ST}_{si,b}/\sum C_j$	One family	Cheng et al. (1996)
Single server problems		
$Pm,S/\mathrm{ST}_{si,b}/\sum U_j$	Constant $m\geqslant 4$	Brucker et al. (2002)
$P,S/\mathrm{ST}_{si,b},p_j=p,r_j/L_{\max}$	Unit setup times	Brucker et al. (2002)
$P,S/\mathrm{ST}_{si,b},p_j=p,r_j/\sum w_jC_j$	Unit setup times	Brucker et al. (2002)
$P,S/\mathrm{ST}_{si,b},p_j=p,r_j/\sum w_jU_j$	Equal setup times	Brucker et al. (2002)
$P,S/\mathrm{ST}_{si,b},p_j=p,r_j/\sum w_jT_j$	Equal setup times	Brucker et al. (2002)

Figure 2.13: Open scheduling problems by Allahverdi [181]. These problems were reported as having unknown computational complexity in the latest literature. Open problems with respect to strong NP-hardness are marked with an asterisk, assuming a reasonable encoding scheme (see Garey and Johnson, 1979) for each problem, i.e., if the problem formulation explicitly states that there are k parameters equal to a, all these parameters are encoded with two numbers k and a.

Scheduling under Uncertainty

As mentioned in the section 2.1, uncertainty is a challenge facing every decision maker. Two factors affect the uncertainty of the problem. The first factor is the level of details that

is covered by the problem formulation including the parameters, environment, constraints and the objective. Such factor can be relaxed to some extent according to decision maker needs. The second factor is the problem nature in terms of dynamicity, stochasticity and parallelism. Dynamicity changes the parameter values at run time. Stochasticity looks for best solution through a set of local optimal values. Parallelism causes the same effect of dynamicity or stochasticity if the outputs of the machines are not predictable.

Most of the researchers try to solve scheduling problems under uncertainty constraints either by preventive or reactive techniques. Preventive techniques are used for scheduling with predictive uncertainty, it targets eliminating the cause of uncertainty or minimising its impacts on the results or the cost functions. Reactive techniques are used with dynamic and stochastic scheduling; it tries to rebuild the schedule at the runtime in order to recover the uncertainty consequences.

Scheduling problems are concerned with generating the schedule at the setup time and revising the schedule at the run time [183]. Scheduling generation is used as a predictive mechanism that specify the plan baseline in terms of the start and completion times of every task or job, according the given requirements and constraints. On the other hand, schedule revision is considered reactive or corrective mechanism, as it monitors the schedule at the runtime and changes it when unexpected events happen [156].

Karabuk and Sabuncuoglu divided the scheduling approaches into online and offline scheduling [184]. Scheduling is offline when all the available jobs are scheduled at once for the whole future plan. Meanwhile, the scheduling is online if the resources are allocated at the run time once the decision is needed.

Reactive scheduling is the process of modifying or changing the generated schedule at the run time in order to adapt with unexpected event, it is also known as corrective scheduling. The mentioned disruptive events can be machine breakdown, rush order arrivals or order cancelation. Reactive scheduling is usually used with unpredictable uncertainty, where

there is no information prior to runtime detection of the uncertain variables that might be used for protective action [156].

On the other hand, preventive scheduling can minimise the uncertainty resulting from the parametric changes similar to processing times, prices and products demand. Predictive uncertainty can be achieved using historical data and forecasting techniques, by deriving the missing information that describe the behaviour of the uncertain parameters. Such predictions or forecasting can take the form of parameter ranges, statistical distributions, fuzzy membership or stochastic distribution.

Preventive scheduling builds the baseline plan considering the predictive uncertainties, such plan should be realised in runtime assuming no disruption or disturbances happened. Preventive scheduling is considered the basis for the plan support activities as the commitments are stated according to the generated schedule. For preventive scheduling, the following approaches are distinguished: stochastic-based approaches, robust optimisation methods, fuzzy programming methods, sensitivity analysis and parametric programming methods [156].

Most of the scheduling techniques assume the completeness and the accuracy of the information used to solve the problem. The recent researchers target generating a feasible baseline schedule, optimising the objective function and considering the uncertainty and the risk factors. The current contributions in scheduling under uncertainty are sparse due to the different natures and formulations of uncertainties as declared in section 2.1.

Many techniques have been used to minimise the impact of uncertainty on scheduling optimisation. The most suitable varies according to the scheduling problem nature and the uncertainty nature. The most frequently used techniques are stochastic scheduling, dynamic programming, fuzzy scheduling, robust optimisation and sensitivity analysis [155].

1. Stochastic scheduling: The problem is formulated as a multi-stage decision process,

then select the activities to be started at random decision points through time according to the scheduling policies such policies are based on the observed past experience and the priori knowledge of the processing time distributions. Such technique has the disadvantage that it cannot explicitly generate a pre-schedule that can be used as the baseline plan for making advance commitments.

2. Dynamic programming: It is used to solve stochastic multimode problems by determining the resource allocation vectors for the project activities in order to minimise total expected cost. Dynamic programing relies on the assumption that uncertainty is inherent in the work contents of the tasks rather than their duration.

3. Fuzzy scheduling: It uses membership functions describing the activity or the task duration, which needs historical data to build the membership function at the beginning. Such fuzzy uncertainty is captured as an instance belongs to a fuzzy set or member of a specific fuzzy range. Fuzzy scheduling can be used to develop the baseline schedules as preventive action; it can also be used for reactive scheduling after developing the set of fuzzy rules describing what to do in case disruption happened.

4. Proactive robust scheduling: It is a redundancy-based techniques similar to buffer insertion approach, which is the fundamental ingredient of Goldratt's critical chain methodology (Goldratt, 1997). Such methodology did not give great results due to the severe oversimplifications, meanwhile it acted as eye-opener for other techniques. The generation of robust multi-resource baseline schedules in combination with efficient and effective reactive schedule repair mechanisms constitutes a viable area of future research. Whereas numerous reactive scheduling mechanisms have been developed and tested in real-time machine scheduling environments, the field is in need for further research aimed at their implementation and validation in a project scheduling environment.

5. Sensitivity analysis has been frequently used in the area of scheduling under uncertainty to seek answers of question similar to "What if . . . ?" or to support the decisions based on the what-if scenario analysis. It has been also used to measure the effect of the parameters and the input variables on the system in terms of uncertainty and constraint satisfaction, which will help to simplify the problem as a step to solve it.

2.3 Preventive Scheduling under Bounded Uncertainty

2.3.1 Combinatorial approach

The problem of scheduling uncertainty can be solved using the combinatorial approach by considering all the possible combinations of the starting points and ending points of every uncertain interval parameter, then calculate the minimum or the maximum displacements [7,8].

The combinatorial approach assumes that the displacement surface is a monotonic function the input interval parameters [7]. The displacement surface consists of a 2D line due to the uncertainty of one parameter only. The curve increases monotonically between the lower bound to the upper bound of the parameter interval. This implies that if $f(a_1, a_2, \ldots, a_N)$ represents a monotonic displacement surface with a specific degree of freedom as a function of N uncertain parameter $a_i (i = 1, 2, \ldots, N)$ then

$$f_{min} \leq f(a) \leq f_{max} \tag{2.1}$$

where

$$f_{min} = \min_{r=1,2,\ldots,2^N} (f_r) \tag{2.2}$$

$$f_{max} = \max_{r=1,2,\ldots,2^N} (f_r) \tag{2.3}$$

$$f_r = f(a_1^i, a_2^j, \ldots, a_N^k), \quad i = 1, 2, \quad j = 1, 2, \quad k = 1, 2; \tag{2.4}$$

and

$$a_m^{(l)} = \begin{cases} \max(a_m), & \text{if } l = 1. \\ \min(a_m)1, & \text{if } l = 2. \end{cases} \tag{2.5}$$

A recent approach uses the same concept of combinatorial approach is the fuzzy finite element method [6] where the magnitude of the uncertainty is defined using the α-cut, in which the interval bounds for the ith uncertain input parameter a_i are re-stated as:

$$\alpha a_{0i} \ + \ (1-\alpha) \ \min(a_i) \ \leq a_i \leq \alpha a_{0i} \ + \ (1-\alpha) \ \max(a_i) \tag{2.6}$$

where α is a parameter that belongs to the interval [0,1] and is used to define the uncertainty degree. a_{0i} is the crisp value of the uncertain parameter.

The main challenge with the combinatorial approach is the tradeoff between the problem formulation and the solution complexity. The simple scheduling problems with small uncertainty margin and monotonic displacement can be solved using such approach in non-deterministic polynomial time. Unfortunately, many of the scheduling problems are not simple and the uncertainty margin is not small and the displacement functions are not always monotonic, which imply the solution become NP-hard. So, there is a need for other techniques to solve the scheduling problem in optimal rather optimum way.

2.3.2 Interval Perturbation

Interval perturbation analysis depends on calculating the possible change to the displacement that happens when a small change occurs to the uncertain parameter. The value of the ith parameter is described by a_{0i}. Calculating the impact of perturbing the uncertain parameter on the cost function requires to formulate the objective function considering the displacement of the perturbed parameter. Such displacement is represented in terms of the displacement of the unperturbed variable (i.e. when $a = a_0$).

The perturbed displacement is represented as $u_0 + \Delta u = u(a_0 + \Delta a)$, such that u_0 is the displacement of the unperturbed variable. The vector denoting the change to the objective function is Δu, and Δa is the vector denoting the change of the vector of the uncertain input parameter [185].

Interval calculations can be used to manipulate the results to obtain approximate upper and lower bounds for each entry of the objective vector Δu [186]. The main disadvantage of this approach is that the interval computations performed are based on the numerical form and take no account of the actual structure. Consequently, the uncertainties resulting from the interactions between the input parameters and the scheduling variables are neglected. This implies the existence of uncertainty regarding the absolute accuracy of the calculations. Despite this, the method has been shown to be useful in producing accurate bounds for a number of simple linear systems [186, 187].

Qiua studied interval perturbation for anti-optimisation of the structures with large non-random parameters [187]. He found that the interval perturbation around the midpoint of the interval can minimise the impact of the input parameter uncertainty on the model structure [188]. he also concluded that when uncertainty of the interval parameter is large, the perturbation technique is not effective as the interval parameter is not infinitesimal quantity [187].

Guo-jian introduced the concept of using sub interval perturbation for the finite element analysis with uncertain interval parameters. Guo-jian found that the subinterval perturbation method is effective by numerical simulation [189]. The speed of computing is fast when the number of elements, which have interval parameters is few and high precision is gained. But for the large number of elements, which have interval parameters, how to improve computational efficiency remains an interesting problem.

The research of McWilliam [185], Qiua [187] and Guo-jian [189] showed up that interval perturbation is good for simple problems with linear formulation and small uncertainty

ranges. Meanwhile many of the scheduling problems are complex, non linear and use heuristic and meta-heuristic techniques, where the interval perturbation is not useful or effective, especially if the range of uncertainty is huge.

2.3.3 Mixed Integer Linear Programming

MILP is used to solve linear programing scheduling problems that are constrained in terms of coefficients or the objectives as inequalities. MILP can solve the scheduling problems under bounded uncertainty constraints as long as the problem can be formulated as system of linear equations. On the other hand it cannot solve scheduling problems, which have been formulated mathematically as non-linear problems or which have been solved using heuristics and meta-heuristics and this leads to us MINLP. Some researchers tried to solve nonlinear scheduling problems under uncertainty using MINLP but they did not achieve optimal schedule with uncertain output. No body provided a feasible methodology to formulate the problem or solve it in non deterministic polynomial time.

The MILP optimisation that is formulated in Pantelides *et al.* (1995) [190] generates schedules where the outcome of any scheduling decision can be determined exactly. Robert Gonzalez used operating parameters, such as station processing times, are specified to take a single value [191]. Because this scheduling information is known in advance, the schedule with the best outcome can be identified. This resulting schedule will generally have very good utilisation of resources.

However, these schedules are only predictive in nature. That is, they are created assuming the world will realize precisely the processing and travel times and resource availabilities given in the problem statement. Sometimes, the operational environments are dynamic,

with potential unanticipated variations and events, which make the generated schedule dependent on predicted outcomes may be unreliable. Furthermore, in our scheduling formulation several assumptions are embedded in the solution.

Xiaoxia Lin *et al.* proposed a robust optimisation approach to solve the problem of scheduling under bounded uncertainty [192]. This approach uses MILP as extension for the robust optimisation methodology used for linear programming problems that is developed by Ben-Tal and Nemirovski (2000) [193]. Lin *et al.* showed that MILP approach produces robust solutions, which are immune to bounded uncertainties in both the coefficients and right-hand-side parameters of the inequality constraints. The application of this approach covered many problems of scheduling with uncertain parameters similar to processing times, market demands, and/or prices of products and raw materials.

The standard mixed-integer linear programming (MILP) problem is formulated as:

$$\begin{aligned} &\min_{x,y}/\max_{x,y} c^T x + d^T y \\ &\ such\ that \\ &Ex + Fy = e \\ &Ax + By \leq p \\ &\overline{x} \leq x \leq \underline{x} \\ &y = 0, 1 \end{aligned} \tag{2.7}$$

Where the uncertainty arises from both the coefficients and the right-hand-side parameters of the inequality constraints, namely, a_{lm}, b_{lk} and p_l. The objective function is constrained by the feasibility of the following inequality.

$$\sum_m a_{lm} x_m + \sum_k a_{lk} y_k \leq p_l \tag{2.8}$$

The optimal solution of an MILP program may become infeasible, that is, one or more constraints are violated substantially, if the nominal data is slightly perturbed. The objective here is to develop a robust optimisation methodology to generate reliable solutions

to the MILP program, which are immuned against uncertainty. This robust optimisation methodology was first introduced for Linear Programming (LP) problems with uncertain linear coefficients by Ben-Tal and Nemirovski (2000) [193] and is extended in this work to MILP problems under uncertainty. Two types of uncertainty are addressed: (i) bounded uncertainty and (ii) bounded and symmetric uncertainty.

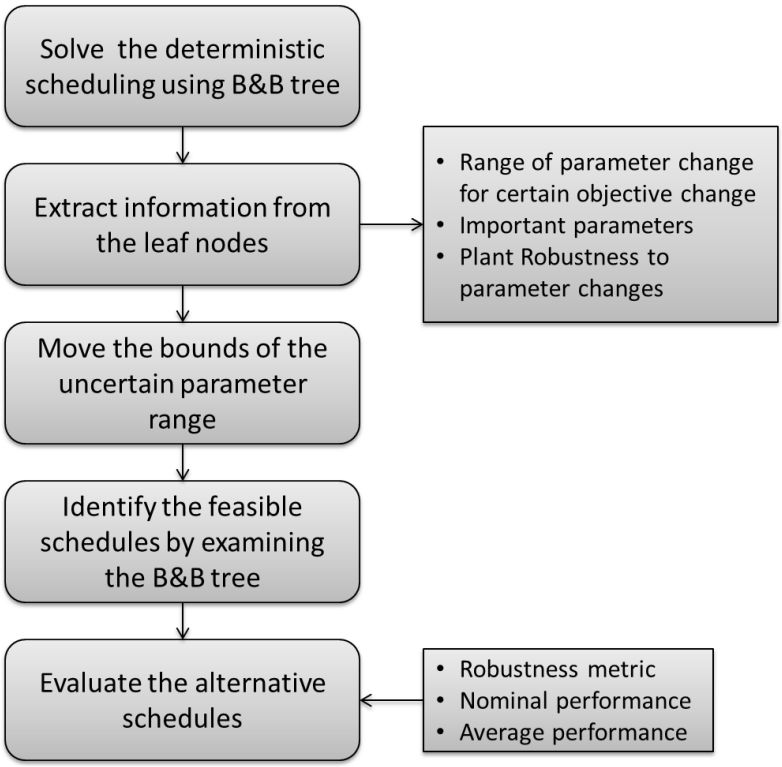

Figure 2.14: Flowchart for using sensitivity analysis for minimising uncertainty as described by Jia and Ierapetritou, 2004 [194]

Discrete-time models results in difficult MILP problems in terms of complexity and

processing time. Many techniques have been proposed targeting to increase the efficiency of the solution according to the problem nature [195]. Such techniques include:

1. Reformulation that reduce the gap between the optimal solution and its linear programming relaxation counterpart, for example, Yee and Shah (1998) [196] reformulated the constraints of the batch-sizing according to the variable aggregation/disaggregation;

2. Adding cut constraints, which are redundant but reduce the region of integer infeasibility, such as those proposed by Dedopoulos and Shah (1995) [197].

3. Interfering the branch and bound solution algorithm, for instance, Shah *et al.* (1993) [198] developed techniques to reduce the size of the relaxed LP and perform post analysis of the solution at each node of the branch and bound tree, Dedopoulos and Shah (1995) [197] proposed techniques to fix variables to values implied during the branch and bound procedure.

4. Decomposing the large and complex problems into smaller sub-problems, for example, Bassett *et al.* (1996) [199] proposed many approaches for time-based decomposition and Elkamel *et al.* (1997) [200] developed an algorithm that uses both of temporal decomposition and spatial decomposition.

2.3.4 Quadratic Programming

Quadratic programming is a special case of non-linear programming, where the objective function f is quadratic and the constraints are represented in the functions h and g [202, 203]. The h function represents the equality constraint and the g function represents the inequality constraint. Both of the constraint functions h and g are linear in $x \in R^n$. The quadratic programming problem are usually formulated as

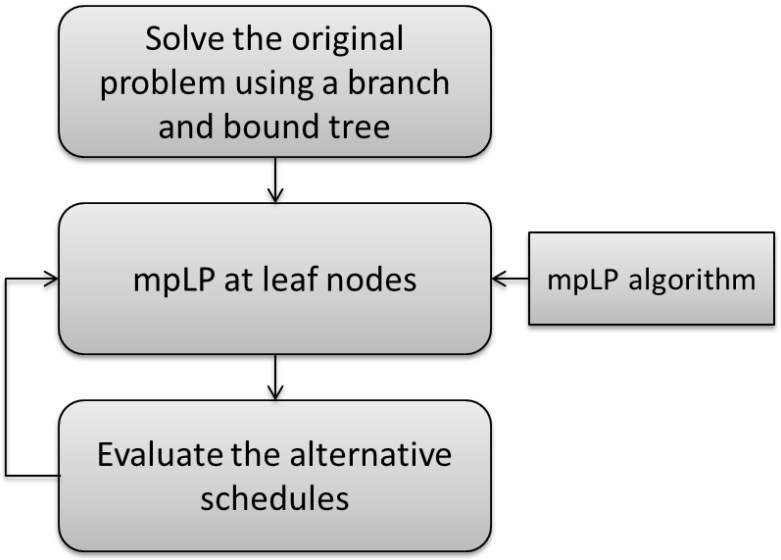

Figure 2.15: Uncertainty analysis using multi parametric linear programming by Jia and Ierapetritou [201]

$$\begin{aligned} &Minimise\ f(x) = 1/2x^T Bx - x^T b \\ &\quad over\ x \in R^n \\ &\quad subject\ to\ A_1\ x\ =\ c, A_2\ x\ \leq\ d, \\ &\quad wehere \\ &\quad B\ \in\ R^{n\times n}\ is\ symmetric, \\ &\quad A_1\ \in\ R^{m\times n},\ A_2\ \in\ R^{p\times n} \end{aligned} \tag{2.9}$$

Although quadratic programing can find the optimal solution within the interval constraints, it cannot neutralise the uncertainty of the other parameters affecting the model such as B, b, A_1, A_2. One of the solutions to optimise a quadratic problem with uncertain parameters is to extend the parameters to be interval-based, which is another extension to the proposed work but for another algorithm. The new formulation of the quadratic problem after the interval extension will be as follows:

$$
\begin{aligned}
Minimise\ f(x) = 1/2\ & x^T[B^-, B^+]x - x^T[b^-, b^+] \\
& over\ x \in R^n \\
& subject\ to\ [A_1^-, A_1^+]x = [c^-, c^+], [A_2^-, A_2^+]x \leq [d^-, d^+], \\
& where \\
& [B^-, B^+] \in R^{n \times n}\ is\ symmetric, \\
& [A_1^-, A_1^+] \in R^{m \times n}, [A_2^-, A_2^+] \in R^{p \times n}
\end{aligned}
\tag{2.10}
$$

Quadratic programming optimises interval-based constraints but does not optimise interval-based parametric uncertainty.

2.4 Summary

Scheduling under uncertainty is a challenging problem as it increases the difficulty of decision making, it also increases the complexity to the problem formulation and solution and sometimes it makes the problem not solvable in time. Tackling this problem requires to understand the uncertainty challenge as well as the scheduling problems.

This chapter stated in the first main section a general review on uncertainty, which is defined as “the dissimilarity between the amount of information required to execute a task and the amount of information already infatuated by some distribution”. Uncertainty propagation happens when uncertain parameter affects another parameter to make it uncertain as well. Although uncertainty is highly correlated with risks, some researchers argue that such correlation does not imply causality.

The first section also stated the most frequently used strategies to manage uncertainties, which can be (1) Ignoring the uncertainty, (2) Generate the missing knowledge, (3) Interact with existing uncertainty, (4) Coping with the uncertainty and its impact. Every strategy has multiple techniques to deal with uncertainty within different contexts.

The uncertainty classification or taxonomy is important to solve any uncertainty related problem, it has been discussed in section 2.1.3. Uncertainty can be categorised to either aleatory or epistemic. Aleatory uncertainty is unpredictable, but can be represented by statistics. On the other hand, the epistemic uncertainty is the result of the missing information, but it is predictable to some extent. Epistemic uncertainty is classified according to its predictability to model uncertainty, parametric uncertainty and completeness uncertainty.

The sources and the causes of the uncertainty in any system should be identified as a preliminary step to solve the problem, either by eliminating the root cause or by minimising its impact. The different sources and causes are mentioned in details in section 2.1.4.

Understanding the scheduling problems and techniques is the second step to resolve the challenge of scheduling under uncertainty. Section 2.2.1 stated the formulation of the scheduling problems including the given input parameters, the objective function and the environmental conditions and constraints. The formulation of the scheduling problem affects directly the possibility of solving it in a deterministic polynomial time rather being NP-complete or NP-Hard.

Classifying the scheduling problem helps in determining the best technique to solve the problem as well as solving the problem under uncertainty constraint. Any scheduling problems is usually classified according to one of three main criteria, as it can be deterministic or stochastic problem, static or dynamic, single machine or multiple machines, which can be either sequential or parallel. All the mentioned criteria affect the formulation of the problem and complexity of the solution especially after adding external constraints, such as the parametric uncertainty.

Scheduling challenges are stated in 2.2.3 as the problem complexity and the problem uncertainty. The problem complexity can be solved by relaxing the constraints or by using heuristics and meta-heuristics to find the near optimal rather optimal solution. The

uncertainty challenge creates the dilemma of selecting the optimal value with high uncertainty or the non-optimal value with low uncertainty. The uncertainty challenge implies that minimising the uncertainty impact will make the decision making much easier.

Studying the uncertainty and the scheduling was an important step to build a robust methodology that can minimise the impact of the interval-based bounded uncertainty on the scheduling algorithm in a preventive way. Meanwhile, there exist few techniques used to solve the same problem such as mixed integer linear programing and interval perturbation and combinatorial approach, these approach are stated in details in section 2.3.

Chapter 3

Proposed Methodology

3.1 Overview

As the literature review declared, minimizing the negative effect of the scheduling uncertainty is a challenging issue. Many researchers developed different approaches to solve such a problem. Every approach is used for a specific problem with a specific uncertainty nature with different input parameters. The proposed methodology is considered a preventive scheduling techniques, which uses interval programming including interval arithmetic [204–208], interval algebra [209, 210] and interval logic [211–215] to extend the scheduling algorithms to be able to minimise the impact of bounded uncertainty.

The proposed methodology creates a new algorithm out of the old algorithm. The new algorithm will be able to minimise the impact of bounded uncertainty on the objective function. The ability to minimise the uncertainty impact varies according to the nature of the problem and the nature of the algorithm. So, The proposed methodology consists of three main phases, uncertainty analysis, algorithm extension and feasibility verification. The first phase is to formulate the uncertainty objective function and to analyse the standard numerical-based algorithm. This phase is called algorithm uncertainty analysis. The second phase is to define how the numerical-based, logical-based and conditional instructions

will be extended to accept the interval-based, interval-logic and interval-inequalities-based instructions. Then extend the algorithm to use them. The third phase is to verify that the algorithm give more certain results than the numerical-based one. The verification can be performed using either real collected data from the problem domain or it can be performed using generated data with a specific distribution that match the nature of the data in a specific problem. Once the three phases are implemented and the algorithm is extended, the extended algorithm can be used to generate the optimum schedule. It is important to notice that there is no optimal schedule in case of uncertain input parameters, which imply that the solution will be optimum rather optimal. The three phases are declared in figure 3.1.

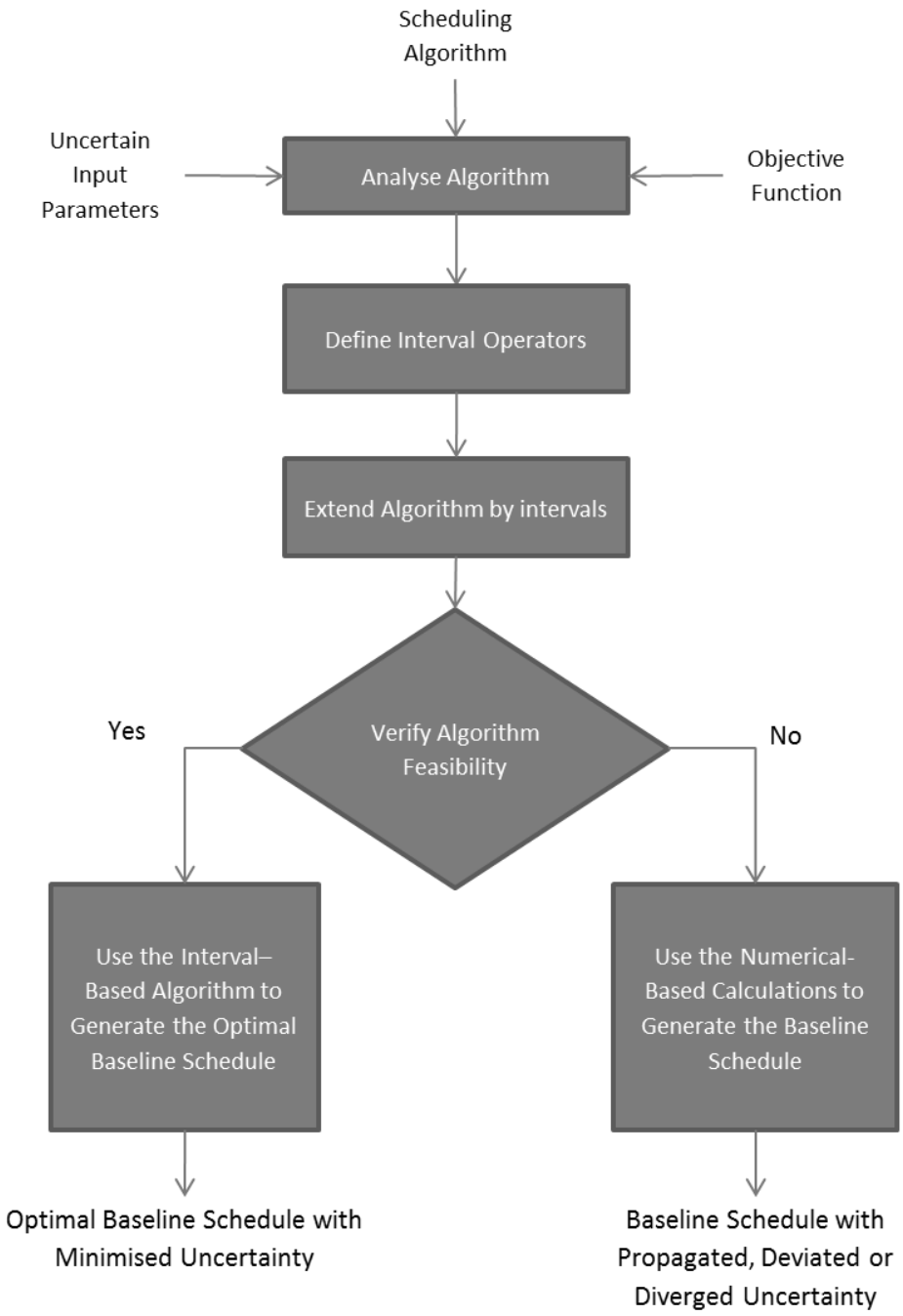

Figure 3.1: The flow chart illustrates the five steps of the proposed methodology, starting by analyzing the uncertainty of the algorithm, then check the feasibility of using such algorithm and finally generate the schedule using either interval programming or numerical programming.

3.2 Formulate Uncertainty Objective Function

Minimizing the impact of uncertainty on the objective function can be formulated as a composite function consisting of uncertainty function of the objective function similar to the formulation of Zhang *et al.* for real-time optimisation [216] and Williams for adaptive management [217]. Such formulation is stated in the equations 3.1, 3.2. Let $f(x)$ represents the objective function to be optimised and $u(f(x))$ represents the uncertainty function to be minimised. $u(f(x))$ will be calculated as function of $f(x)$ using different estimates of the input $x, x^-, x^+, [x^-, x^+]$ and $((x^- + x^+))/2$. It will consider the difference between the estimates of the objective function at the upper and the lower bounds during the calculation.

$$u_b(f(x)) = |f(x^+) - f(x^-)| \tag{3.1}$$

$$u_I(f(x)) = |f([x^-, x^+])^+ - f([x^-, x^+])^-| \tag{3.2}$$

The proposed premises are: In case $u_I > u_b$ then interval programming will produces more certain results for this scheduling algorithm. Otherwise; then bisecting the interval or approximating it numerically will be able to produce more certain results for this scheduling algorithm, where $u_I < u_b$. Such premises will be validated in the next 3 chapters, which are papers explaining the concept and the implementation of interval programming on three different scheduling algorithms The stated objective function will be estimated using a significant amount of data that can be either real life data or generated data that match a specific probability density function. The input data should be analyzed to its main properties. Such properties should be correlated with the output to know the main criteria affecting the impact of input uncertainty on the objective function.

3.3 Define Interval Operators

Scheduling algorithms vary according to the objective function and the input parameters. Such variation implies that the interval relations used for a specific algorithm will not give the least uncertain output estimation with another algorithm. To estimate the output of the algorithm with minimum uncertainty, the interval relations should be defined, takin in consideration the main three factors affecting any interval, which are the lower bound, upper bound and the distance between them. The nature of the relations between different intervals should be considered, which can be disjoint, overlapping or containing intervals. Control statements are important component of any algorithm including the scheduling ones as it is part from any sorting algorithm and it determines which task to be allocated first. Control statements are the instructions that control the path of the execution of the other instruction at the run time. Control statements include the conditional statements, the loops and runtime instruction allocation in case of parallel processing. Any change in the order of the tasks can make a huge difference in the estimate of the objective function. Conditional statements use Boolean logical operators to compare any two numbers as inequality. Boolean logical operators do not work with interval variables or values, which require to use either interval algebra by Allen [30, 31] or to define inequality operators using different relations between the upper bounds, lower bounds and distances.

The nature of every algorithm determines how to estimate the inequality logic. It depends on how far the uncertain variable affects the objective function. The implementation of the algorithm as either a critical or relaxed problem is another factor that determines the estimate of the inequality. For example, if the algorithm is implemented in the medical domain such as an intensive care unit, then it should assume the worst case scenario even though it may cost more than average in order to eliminate any risk of a human life. If the same algorithm is implemented to schedule the resources of a personal computer, the

Table 3.1: The definitions of relations between any two intervals are defined by Allen's algebra. The timeline examples represent the relation between the lower bound of the first interval and the lower bound of the second interval and the relation between the upper bound of the first interval and second interval, where $x^- < x^+$ and $y^- < y^+$.

Basic Relation	Operator	Example	End points
x precedes y y preceded by x	p $p^{-}1$	xxxx yyyy	$x^+ < y^-$
x meets y y met by x	m $m^{-}1$	xxxxx yyyyy	$x^+ < y^-$
x overlaps y y overlapped by x	o $o^{-}1$	xxxxxxx yyyyyy	$x^- < y^- < x^+$, $x^+ < y^+$
x during y y includes x	d $d^{-}1$	xxxx yyyyyyyyy	$x^+ < y^+$, $x^+ < y^+$
x starts y y started by x	s $s^{-}1$	xxxx yyyyyyyyy	$x^- = y^-$, $x^+ < y^+$
x finishes y y finished by x	f $f^{-}1$	xxxxx yyyyyyyyy	$x^+ = y^+$, $x^- > y^-$
x equals y	$\equiv$	xxxxxxxxx yyyyyyyyy	$x^- = y^-$, $x^+ = y^+$

algorithm may assume the best case scenario considering the probability of deadlocks is minor and its danger is limited.

Allen [30, 31] stated all the possible relations between any two intervals and called it *Interval Algebra*. The definition of such relations depends on the numerical relation between the lower bound of the first interval versus the lower bound of the second interval and the same for the upper bound. Table 3.1 explains the definitions of interval relations according to Allen's algebra. This table shows that any relation between two intervals can be classified into one of five categories as explained in the next five items and in the figures 3.2, 3.3, 3.4, 3.5.

1. Disjoint interval: where the upper and the lower bounds of the first interval are less than the upper and the lower bounds of the second interval. In such case the first interval is smaller than the second interval always.

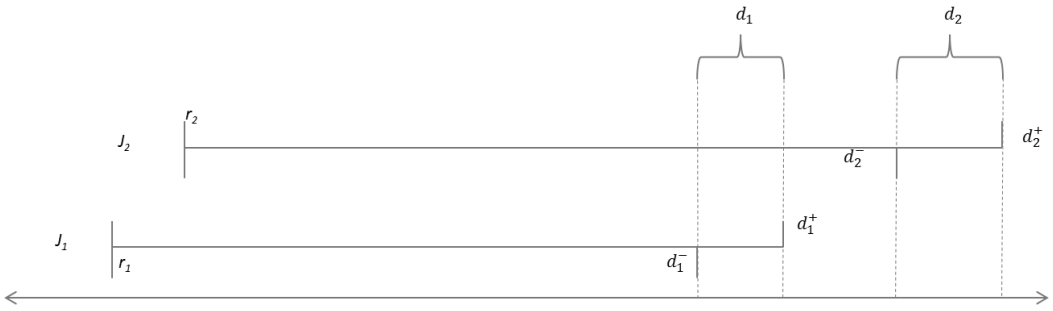

Figure 3.2: A plot for two jobs relative to the number line, where the release time (r_i) is certain and numerical-based. Meanwhile, the deadline is uncertain and interval-based $d_1 = [d_1^-, d_1^+]$. In this figure the deadline intervals are disjoint as $d_1^+ < d_2^-$

2. Overlapping intervals: where the upper bound of the first interval is between the lower bound and the upper bound of the second interval. In such case, there exist two factors controlling the inequality estimate. The first is the percentage of the overlap. The second factor is the nature of the problem. For example if the urgency of the task grows exponentially, then the first interval takes higher priority and if the first interval has a logarithmic curve then second interval will be considered as earlier interval.

3. Equal intervals: where the upper bound of the first interval equals the upper bound of the second interval and the same for the lower bounds.

4. Containment intervals: where the lower bound of the first interval is between the upper and lower bounds for the second interval. The upper bound is between the lower and upper bound of the second interval. This case is complicated because the point estimate of the first interval can be larger or smaller than the point estimate of the second interval. There exist three factors controlling the priority of the two

Figure 3.3: A plot for two jobs relative to the number line, where the release time (r_i) is certain and numerical-based. Meanwhile, the deadline is uncertain and interval-based $d_1 = [d_1^-, d_1^+]$. In this figure the deadline intervals are disjoint as $d_2^- < d_1^+ < d_2^+$

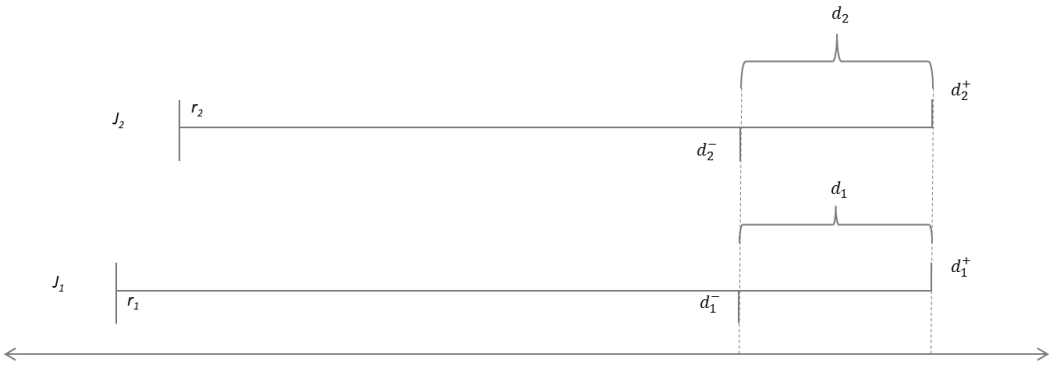

Figure 3.4: A plot for two jobs relative to the number line, where the release time (r_i) is certain and numerical-based. Meanwhile, the deadline is uncertain and interval-based $d_1 = [d_1^-, d_1^+]$. In this figure the deadline intervals are disjoint as $d_1^+ = d_2^+$ and $d_1^- = d_2^-$

intervals:

(a) The ratio between the distance of the first interval and the second interval, which is estimated by d_1/d_2.

(b) The location of the interval inside.

(c) The nature of the problem, urgency curve (linear, exponential, logarithmic or normal)

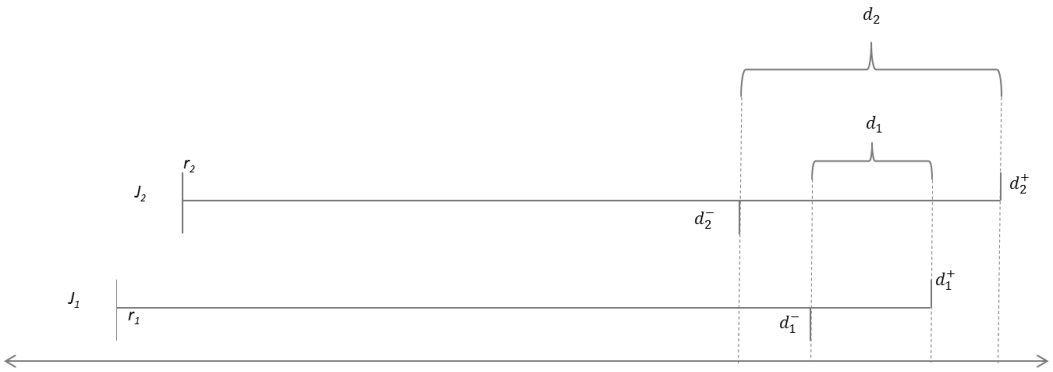

Figure 3.5: A plot for two jobs relative to the number line, where the release time (r_i) is certain and numerical-based. Meanwhile, the deadline is uncertain and interval-based $d_1 = [d_1^-, d_1^+]$. In this figure the deadline intervals are disjoint as $d_2^- < d_1^- < d_1^+ < d_2^+$

5. The meeting relation is approximated to the overlap relation and the starting and finishing relations are both approximated to containment relations.

For any two disjoint intervals, the interval with minimum upper and lower bound comes first because the upper bound of the first interval is less than the lower bound of the second interval ($I_1^+ < I_2^-$). This implies that in case the first interval action happened as late as it

can, it will be preceding the action of the second interval even if it occurred as early as it can.

For any two overlapping intervals, the nature of the problem and the urgency curve of every interval should be considered. If urgency curve increases by time, the first interval gets higher priority in sorting, which means it will be considered as the smaller number even if the interval started after. On the Other hand if the urgency curve decays of time, the latest interval gets higher priority. For linear functions, the priority can be estimated by comparing the slopes. For the non-linear curves, the priority can be estimated using numerical integration between the lower bound of the interval and the upper bound.

3.4 Extending Scheduling Algorithms

Scheduling algorithms consist of a sequence of primitive operations including arithmetic and logic operations as well as conditional statements and loops. In order to upgrade the algorithm, the steps that affect the uncertainty of the objective function should be specified then changed as needed, which can be applied in three main steps as follows: The first step is to specify if the instruction code should be changed. It is important to make sure that the code of this instruction is uncertain and is affecting the objective function. In case of arithmetic and logic operations, uncertainty of the instruction can happen if it depends on or uses an uncertain an input parameter or an uncertain variable or an uncertain estimate of a function. The uncertainty of the code can also occur due to conditional statement or loops that depend on the estimate of uncertain variables. Although uncertian input parameters affect the uncertainty of instruction, it is not necessary to affect the estimate of the objective function as the caculations can take a path that uses accurate values due to some conditional statement.

The second step is to change every uncertain instruction in a way that replace every

numerical arithmetic and Boolean logic operation with an interval arithmetic and interval logic operation in order to cope with interval programming. Such replacement can be classified to three different categories, each of them has different nature and different exetension equations. The categories and their extension are stated as follows different categories are stated as follow:

1. The most frequently used arithmetic instruction is the assignment operation, where a variable is assigned a specific value (e.g. x=5). Assignment operation can also be assigning a variable to another variable, which is implemented by copying the value of the right hand side to the left hand side. Such assignment process can be classified to four categories according to the nature of the left side and the right side, which can be either a number or an interval as follows:

   (a) Assign numerical value to numerical variable. E.g; $X = 4$.

   (b) Assign numerical value to interval variable: Let X be an interval variable where $X = [x^-, x^+]$ and y is a numerical variable, where $y = 5$. The assignment operation will be as follows: $X = y \Rightarrow x^- = y \; and \; x^+ = y \Rightarrow X = [y, y] X = [5, 5]$

   (c) Assign interval value to numerical variable Let X be a numerical variable where $x = 5$ and y is a numerical variable, where $Y = [y^-, y^+] = [2, 12]$. The assignment operation will be as follows: $x = (y^- + y^+)//2 \Rightarrow x = (2 + 12)//2 \Rightarrow x = 14$

   (d) Assign interval value to interval variable Let X be an interval variable where $X = [x^-, x^+] = [4, 9]$ and Y is an interval variable where $Y = [y^-, y^+] = [6, 14]$. The assignment operation will be as follows: $X = Y \Rightarrow x^- = y^- \; and \; x^+ = y^+ \Rightarrow X = [x^-, x^+] = [y^-, y^+] \Rightarrow X = [6, 14]$

2. Arithmetic operation is frequently needed in the calculation of the objective function that will be optimised. It can be addition, subtraction, multiplication or division, where all are defined by Moore [218]. In case one of the variables is numerical, the numerical variable should be converted to interval variable or vice versa. Such conversion depends on the result variable, in which the result of the calculation will be assigned. The following equations shows how Moore defined the arithmetic equations:

   Let I_1 *and* I_2 *be two interval variables where* $I_1 = [I_1^-, I_1^+]$ *and* $I_2 = [I_2^-, I_2^+]$

   (a) Addition:
   $$I_1 \oplus I_2 = [I_1^- + I_2^-, I_1^+ + I_2^+] \tag{3.3}$$

   (b) Subtraction:
   $$I_1 \ominus I_2 = [I_1^- - I_2^-, I_1^+ - I_2^+] \tag{3.4}$$

   (c) Multiplication:
   $$\begin{aligned} I_1 \otimes I_2 = [&Min(I_1^- \times I_2^-, I_1^- \times I_2^+, I_1^+ \times I_2^-, I_1^+ \\ &\times I_2^+), Max(I_1^- \times I_2^-, I_1^- \times I_2^+, I_1^+ \times I_2^-, I_1^+ \\ &\times I_2^+)] \end{aligned} \tag{3.5}$$

   (d) Division:
   $$\begin{aligned} I_1 \oslash I_2 = &[I_1^-, I_1^+] \times [1/I_2^-, 1/I_2^+] \; such\ that\ 0 \\ &\in [I_2^-, I_2^+] \end{aligned} \tag{3.6}$$

3. Conditional statements depend on the logical estimate of the inequality relation. Such estimate can be calculated either using interval algebra of Allen [30, 31], which is used for temporal reasoning, which is stated in table 3.1. It can also be estimated using the custom interval logical operators that are defined in the previous section.

The third step is to add observers to every uncertain instruction. These observers estimate the uncertainty distance before and after the instruction using each of interval bisection and interval arithmetic. The observer then select the instruction that will make the distance between the upper bound and the lower bound of the result of this instruction as small as possible.

3.5 Verifying Algorithm Feasibility

Interval programming affects every scheduling algorithm in a different way. Some algorithms give more certain estimate of the objective function than numerical calculations. Other algorithms produce the estimate of the objective function with similar certainty or less certainty than the numerical calculations.

To verify the usefulness of the interval programming in minimizing the impact of bounded uncertainty of the input data it should be compared to another technique dealing with the same problem, which is the numerical calculation in terms of certainty of the results and the complexity of the algorithm. Certainty of the results is estimated using the objective function stated in the problem formulation section in equations 3.7 and 3.8

$$u_b(f(x)) = |f(x^+) - f(x^-)| \tag{3.7}$$

$$u_I(f(x)) = \mid f([x^-, x^+])^+ - f([x^-, x^+])^- \mid \tag{3.8}$$

3.5.1 Complexity Feasibility

The complexity of the algorithm can be measured using one of the orders of magnitude according to Bachmann-Landau notations. These notations include the big Omicron (Big O), big Omega, big Theta, small Omicron and the small Omega. The most frequently used one is the big Omicron (Big O) as it describes the limiting behaviour of a function when

the argument tends towards a particular value or infinity using simpler functions similar to constant functions, linear, logarithmic function and exponential function. Such limiting behaviour is considered the worst case scenario. Big O notation is used to classify algorithms according to their response to changes in the input size, in terms of the processing time or working space requirements. $f(n)O(g(n))$, where f is bounded above by g (up to constant factor) asymptotically, where $|f(n)||g(n)|.k$ for some constant positive number k

If the certainty of the algorithm is considered regardless the complexity, interval perturbation will produce results with the minimum uncertainty. Interval perturbation tries all possible combinations of the probable estimates of every interval. The only problem with interval perturbation is its NP-hard complexity as it uses combinatorial optimisation, which can work on a small number of intervals but its complexity grows exponentially with large number of intervals and the distance between lower bound and upper bound of each interval.

3.5.2 Measuring Uncertainty using Different Datasets

In order to know how each algorithm responds to the interval programming extension, the upgraded algorithm have to be tested the using different datasets and the objective function should be calculated using lower bound, upper bound and interval values. Compare the results and calculate the uncertainty measure. To guarantee that the extended algorithm outperforms its benchmark, which is the numerical bisection or the multiple calculation, the tests should fulfill as much as possible from the following conditions:

1. The size of the data set should be statistically significant, which can be randomly generated as in chapter 4 or collected data from practical problem logs as in chapter 5.

2. The dataset should represent the nature of the problem data, which can be uniform

distribution as in chapter 4 or normal distributions as in chapter 5.

3. The dataset should cover all interval relations mentioned by Allen [30, 32].

4. The tests should be performed on multiple independent datasets and measure uncertainty estimates.

5. The uncertainty estimates should be correlated with different factors affecting the results including the number of overlaps, interval distances and containments.

Afterwards, if the total uncertainty of the cost function using interval programming was lower than the total uncertainty estimates using numerical calculations and Boolean logic, then the extended interval-based algorithm is better for generating the baseline schedule.

3.6 Summary

This chapter explained in details the new methodology that minimises the impact of interval-based bounded uncertainty on the objective function of any preventive scheduling algorithm. The methodology consists of five sequential steps, starting by decomposing the algorithm into a sequence of effective instructions using program slicing techniques.

The second step is to analyse the interval relations, which can be disjoint, overlapped, containing, equal or meeting each other. Afterwards, define the interval operators according to the preceding interval relation analysis.

Extend the scheduling algorithm by replacing every numerical operation in the effective instructions with interval operation using the interval arithmetic, which is defined by Moore. The assignment operations and variable allocation also changes according to the instruction nature as indicated in section3.4. The Boolean-logic operations will also be replaced with the new interval-logic operators as defined in section 3.4 .

To verify the feasibility of the extended algorithm, the objective function should be tested on multiple datasets that represent the nature of the problem. The feasibility test measures the uncertainty of the final estimates of the objective function and compares it to the final estimates of the non-extended numerical-based algorithm, which uses the upper bound and lower bound as separate values. The uncertainty of the interval-based algorithm should be less than the non-extended algorithm. The uncertainty of the objective function is measured by a distance function between the upper bound and the lower bound of the final estimate.

The extended interval-based algorithm can be used to generate the baseline predictive schedule after verifying that it minimise the uncertainty of the objective function. Otherwise, the numerical algorithm can be used. It is important to analyse the criteria affecting the marginal efficiency of the extended algorithm to identify the exact situations to use it. Such criteria can be the number of intervals, the relation between intervals, the total distance between the upper bound and the lower bound.

The next three chapters explain how the proposed methodology is applied to three different algorithms to create the extended versions in order to minimise the uncertainty. The next chapters also explains the performance of the experiments using different datasets, they also show up the results, analyse them and explain the different criteria affected such results.

Chapter 4

Minimising the Impact of Bounded Uncertainty on Bratley's Algorithm

Hossny, A.; Nahavandi, S.; Creighton, D.;, "Minimizing Bounded Uncertainty Impact on Scheduling with Earliest Start and Due-date Constraints via Interval Computation", in Emerging Technology and Factory Automation, 2012. ETFA 2012. International Conference on, September 2012.

Bounded uncertainty is a major challenge to real life scheduling as it increases the risk and cost depending on the objective function. Bounded or interval-based uncertainty provides limited information describing its nature. It provides only the upper and the lower bounds without information in between, in contrast to probability distributions and fuzzy membership functions. Bartley's algorithm is usually used for scheduling with the constraints of earliest start and due-date, which is formulated as $1 \mid r_j,\ d_j \mid C_{max}$. The proposed research uses interval computation to minimise the impact of bounded uncertainty of processing times on Bratley's algorithm. It minimises the uncertainty of the estimate of the objective function. The proposed concept is to do the calculations on the interval values and approximate the end result instead of approximating each interval then performing numerical calculations. This methodology gives a more certain estimate of the objective

function.

4.1 Introduction

Uncertainty is a major challenge in applying scheduling to real life problems. Scheduling algorithms optimise the cost function theoretically, as it assumes that input data are exact and accurate, which leads to exact estimate for the cost function. Such assumption leads to more risk and more cost as the parametric data in reality are not always accurate, where the initial estimate of the data can be inaccurate and the parametric data can vary at the runtime. To increase the accuracy percentage, uncertainty reasons and nature should be identified and eliminated, otherwise the impact on the objective function should be minimised.

Bratley's algorithm is used to build the schedule with objective function minimizing the maximum completion time C_{max} and given input includes the release time r_j of each task and the due-date d_j of each task [219]. Bratley's algorithm is a branch and bound algorithm that uses backtracking a tree of all solution, which makes the problem NP-hard to find the best solution [220].

This research aims to minimise the impact of bounded uncertainty of the processing time parameter for each task, which is formulated as mathematical interval where $I = [I^-, I^+]$, such that I^- represents the lower bound and I^+ represents the upper bound, taking into consideration that the probability distribution inbetween is unknown, which force us to consider it uniform to ensure equal chance to all values between lower and upper bounds.

The algorithm is implemented using interval computation [221] including interval arithmetic [218] and interval algebra [33], which is considered as mathematical uncertainty handling technique [222]. Section 2 introduces scheduling uncertainty basic concepts and scheduling uncertainty. Section 3 explains the interval computation and shows how to use it to minimise scheduling uncertainty. Section 4 explains the experiments and results. Section

5 will discuss conclusions.

4.2 Scheduling Uncertainty

Uncertainty has many definitions varying according to the context, in engineering domain uncertainty can be defined as the dissimilarity between the expected values at setup time and the actual values at runtime that occasionally happen because of lack knowledge or misleading information or error in the model based predictions. Uncertainty can be reduced using different techniques according to the problem nature, which can be the numerical approximation, the interval mathematics, the probability distribution or the fuzzy presentation [222].

Minimizing the uncertainty impact on scheduling process and cost function have been addressed by many researchers proposing different techniques, which implement one of the previously mentioned methodologies according to problem nature. These techniques are classified into two main classes; first is preventive scheduling, which can predict the possible reasons of uncertainty and eliminate them, second is reactive or corrective scheduling where the reason of uncertainty is sudden and not predictable which means to wait till the problem happens and try to fix it at runtime by rebuilding the scheduling partially or totally.

4.2.1 Preventive Scheduling

Preventive scheduling is to generate a flexible schedule that can deal with uncertainty before it happens according to a previously known uncertain input data. It is also known as proactive or protective scheduling. The perfect scenario is to absorb the uncertainty of the input data to evaluate the cost function exactly. Such scenario is not always applicable due to the problem nature as the optimality of the schedule and the accuracy of the cost function

estimate are negatively correlated with uncertainty of the data. So, preventive scheduling tries to minimise the possible deviation of the optimal schedule leading to more accurate cost estimate with smaller margin of error.

Preventive scheduling has been handled by five main techniques. First is stochastic scheduling, which transforms the original deterministic scheduling model into stochastic model treating the uncertainties as stochastic variables [223]. Second is robust optimisation method, which focuses on obtaining preventive schedules that minimise the effects of disruptions on the performance measure, and it ensures that the preventive schedules maintain a high level of performance [224].

A third preventive scheduling technique is the fuzzy programming, which uses heuristic search for scheduling optimisation. It considers random parameters as fuzzy numbers and constraints are treated as fuzzy sets. Some constraint violation is allowed and the degree of satisfaction of a constraint is determined as the membership function of this constraint [225].

A fourth technique is sensitivity analysis, which analyses the deterministic solutions to determine the importance of different parameters and constraints and determine the range of parameters, in which the optimal solution remains unchanged [194].

4.2.2 Reactive Scheduling

Reactive scheduling is a process of modifying the created schedule at runtime to adapt sudden change in production environment due to uncontrollable factors causing performance uncertainty, such as disruptive events or machine breakdowns. Such type of uncertainty results from lack of information that describes the runtime parameters prior to real execution of the schedule. So, the preventive techniques are not useful in this case [226].

Many researchers have tackled reactive scheduling that can be summarised in two ways.

First is to minimise the effect of disruptions on the objective function after their occurrences. Second is to recreate the scheduling instantly, which requires resolving the computational complexity issue as many of scheduling algorithms are infeasible or not applicable to be recomputed instantly because of their combinatorial nature that makes the problem complexity NP-Complete or NP Hard [227]. Rebuilding the schedule at runtime can be achieved partially by fixing or repairing the disrupted part of the schedule if possible. Otherwise the schedule should be rebuilt totally. Such decision depends on the schedule nature and its level of complication and dependency [227].

4.3 Proposed Methodology

The proposed methodology is to use interval programming including interval arithmetic and interval algebra as alternative to numerical calculations within the scheduling algorithm to find less uncertain sequence of tasks to achieve more accurate cost function.

4.3.1 Interval Programming

Interval programming is used to solve the interval-based problems, where the numerical value is not known exactly and the only available information is the upper and lower bounds. Interval programming includes interval arithmetic, interval logic and interval temporal relations.

Interval arithmetic is first introduced by Moore to solve system of equations [218]. It simply uses the interval as a replacement of the number in all calculations and where the numerical value i belongs to the interval I where $I = [I^-, I^+]$, some representations use the notation $I = [i - \Delta i, i + \Delta i]$ but this refers to known mean for the uncertainty range and exact error margin, which is not always available. Interval arithmetic defined a set

of arithmetic operations as basis for any interval-based equations or algorithms, such that operations of additions, subtraction, multiplication and division are redefined as follows:

$$I_1 \oplus I_2 = [I_1^- \ + \ I_2^- \ , \ I_1^+ \ + \ I_2^+] \tag{4.1}$$

$$I_1 \ominus I_2 = [I_1^- \ - \ I_2^- \ , \ I_1^+ \ - \ I_2^+] \tag{4.2}$$

$$\begin{gathered} I_1 \otimes I_2 = [\min(I_1^- \times I_2^-, I_1^- \times I_2^+, I_1^+ \times I_2^-, I_1^+ \times I_2^+), \\ Max(I_1^- I_2^-, I_1^- I_2^+, I_1^+ I_2^-, I_1^+ I_2^+)] \end{gathered} \tag{4.3}$$

$$\begin{gathered} I_1 \oslash I_2 = [I_1^-, I_1^+][1/(I_2^-), 1/(I_2^+)] \\ \text{such that } \ 0 \notin [I_2^-, I_2^+] \end{gathered} \tag{4.4}$$

Implementing the interval arithmetic to algorithms determines to which extent the uncertainty will be decreased. The operations that can decrease the interval range should be applied at early stages of the solutions. This is mentioned by Maiumder and Rao [228] and named as interval optimisation.

Interval temporal relations are first identified by James Allen in what is called interval algebra or Allen's algebra [31] [30]. Interval algebra identifies relations between time intervals using 13 basic operations represents the combination of relations between the lower bounds and upper bounds of the two intervals.

4.3.2 Applying Interval Programming to Scheduling Algorithms

Any scheduling algorithm consists of a set of numerical operations and set of logical comparisons, in addition to some algorithmic steps such as assignment operation. To apply the

Table 4.1: Mapping Allen's relations between the intervals $[x^-, x^+]$ and $[y^-, y^+]$ to standard numerical relations.

x^-y^-	x^+y^+	Allen's Description	Logical Relation
$<$	$<$	Less	$\prec$
$<$	$<$	Overlap	$\prec$
$<$	=	Finishes	$\prec$
$<$	$>$	Contains	$\prec$
=	$<$	starts	$\prec$
=	=	equal	$\equiv$
=	$>$	Starts	$\succ$
$>$	$<$	Contained	$\succ$
$>$	=	Finishes	$\succ$
$>$	$>$	Overlap	$\succ$
$>$	$>$	Larger	$\succ$

interval programming to any scheduling algorithm, every arithmetic operation should be replaced by interval arithmetic and every logical comparison should be replaced by interval algebraic comparison.

Although interval relations have been identified by Allen's algebra, it cannot be used directly as replacement to standard numerical comparison in normal algorithms, which lead to the need to mapping Allen's interval relations to numerical relations. Different mappings have been tried targeting to minimise the uncertainty of the cost function until the best mapping is found as stated in table 4.1

4.3.3 Using Interval Programming to minimise Uncertainty Impact on Bratley's Algorithm

Bratley *et al.* proposed an algorithm to find a feasible schedule for non-preemptive tasks by minimizing the maximum completion time with given release time and due-date of each task, which is formulated as $1 \mid r_j,\ d_j \mid C_{max}$. The algorithm is classified as branch and bound, which depends on building a tree of solutions and iterate through it till find the

best solution, which lead on worst case to nondeterministic polynomial hard (NP-hard) complexity, specifically at worst case is $O(n.n!\,)$.

Bratley's algorithm starts with empty schedule, by every step it visits new node and add a task to a partial schedule, then uses the pruning techniques to determine when the searching process should be stopped. So the current branch should be pruned in one of two cases:

1. Adding the node causes missing the due-date.
2. Finding a feasible schedule at the current path.

4.4 Experiment and Results

In order to measure the effectiveness of the proposed methodology, the objective function of such scheduling algorithm should be calculated using the uncertain data inputs. Such calculation is first performed using the numerical-based algorithm, then performed using the extended interval-based algorithm. As the objective function is to minimise the C_{max}, So, the cost function will be calculated by the equation 4.5

$$(\max_{i=0}^{n} C_i^- + \max_{j=0}^{n} C_j^+)/2 \tag{4.5}$$

Although the cost function looks as average of maximum completion time, it will vary a lot according to uncertainty complication, which can be disjoint intervals or overlapping intervals or even containing intervals. Disjoint intervals does not affect the schedule anyhow, which make $\max C_i$ equals $\max C_j$ and the cost function calculates the average between the lower and the upper bound, on contrary overlapping and containing intervals build different schedules causing $\max C_i$ not equal $\max C_j$.

The interval-based extended algorithm is tested using 40 different data sets. Each dataset consists of multiple tasks that vary between 10 and 30 tasks. Each task has some

certain and accurate properties such as the release time and the deadlines. The processing time is the uncertain property of each task, where the processing time is estimated as an interval with lower bound and upper bound $[p_i^-, p_i^+]$. The data is generated randomly using normal and uniform distributions and the processing time intervals are generated randomly using the uniform distribution.

The target is to build a schedule that guarantees that all tasks complete processing before the deadline. Meanwhile, the uncertainty of the completion time should be minimised, which is estimated by the distance between the upper and the lower bounds of the completion time.

Table 4.2 lists 12 different datasets with their associated properties including the number of overlaps between intervals, number of containments, total input uncertainty, the output uncertainty using interval programming and the output uncertainty using numerical calculations. The input uncertainty is estimated by the equation $\sum p_j^+ - p_j^-$. The total uncertainty output is estimated by the equation $\sum C_j^+ - C_j^-$. The table shows that 30% of the data sets gave more certain results with different percentage. The marginal enhancement of the certainty varies between 2% and 13% according to every dataset's nature.

Table 4.2: The effect of interval programming on the schedule as tested on 12 randomly generated datasets.

Set Number	Number of Overlaps	Number of Containments	Total Input Uncertainty	Interval Output Uncertainty	Numerical Output Uncertainty	Enhancement Ratio
1	21	28	510	431	498	13%
2	20	18	466	405	459	11%
3	14	26	386	386	386	0%
4	23	24	562	562	562	0%
5	30	16	628	628	642	2.2%
6	24	20	623	623	623	0%
7	16	6	313	313	313	0%
8	12	6	268	236	260	8.9%
9	17	20	428	428	442	3.3%
10	24	24	613	613	613	0%
11	14	20	332	332	332	0%
12	20	16	631	631	631	0%

An example dataset is listed in table 4.3 including the values of the properties of each task, including the release-date numerical values, deadline numerical values and processing time interval values. The release-date is randomly generated using the uniform distribution with maximum value of 100. The deadline is also generated randomly using uniform distribution with maximum value of 3,000. The lower bound of the processing time interval is randomly generated using the uniform distribution, where the value resides between a minimum value of 50 and a maximum value of 150. The upper bound of the processing time

interval is also generated randomly using the uniform distribution, where its value resides between a minimum value of 100 and a maximum value of 250.

The example dataset has been scheduled using the unmodified Bratley's algorithm and the extended interval-based Bratley's algorithm. The numerical-based results are illustrated in figure 4.1 and the interval based results are illustrated in figure 4.2. As declared in table 4.3 and the figures 4.1 4.2. The two algorithms minimised the maximum completion time and achieved the scheduling objective by ensuring that all tasks complete their processing before their deadlines. Meanwhile, the extended interval-based algorithm produced a schedule with less uncertain estimates than the numerical-based algorithm, in terms of the start and end time for every task and the total time for all tasks completion.

Table 4.3: The effect of interval programming on the schedule as tested on 12 randomly generated datasets.

Task Name	Release Time	Deadline	Processing Lower Bound	Processing Upper Bound	Numerical-based Completion Time	Interval-based Completion Time
$T1$	41	2878	63	129	[104, 1156]	[104, 170]
$T2$	72	743	116	214	[535, 648]	[400, 741]
$T3$	93	883	137	212	[807, 860]	[760, 1427]
$T4$	99	2637	78	165	[973, 1836]	[838, 1592]
$T5$	99	468	135	186	[285, 353]	[973, 1778]
$T6$	90	1399	58	139	[670, 1295]	[623, 1215]
$T7$	87	1154	88	167	[895, 1027]	[565, 1076]
$T8$	81	2452	77	168	[612, 1671]	[477, 909]
$T9$	56	518	66	149	[419, 434]	[284, 527]
$T10$	42	2628	114	208	[218, 1503]	[218, 378]

The completion time is uncertain and formulated as intervals as it equals to the sum of the processing time and the release date. The completion intervals of the listed tasks overlapped each other for 33 times and contained each other for 6 times. The total input

uncertainty was estimated to be 805 units of time. The numerical-based calculations produced uncertainty distance of 863 units of time. Meanwhile the interval-based algorithm produced uncertainty distance of 805 units of time, which is less than the numerical-based by 58 units of time with marginal enhancement of 7.2%.

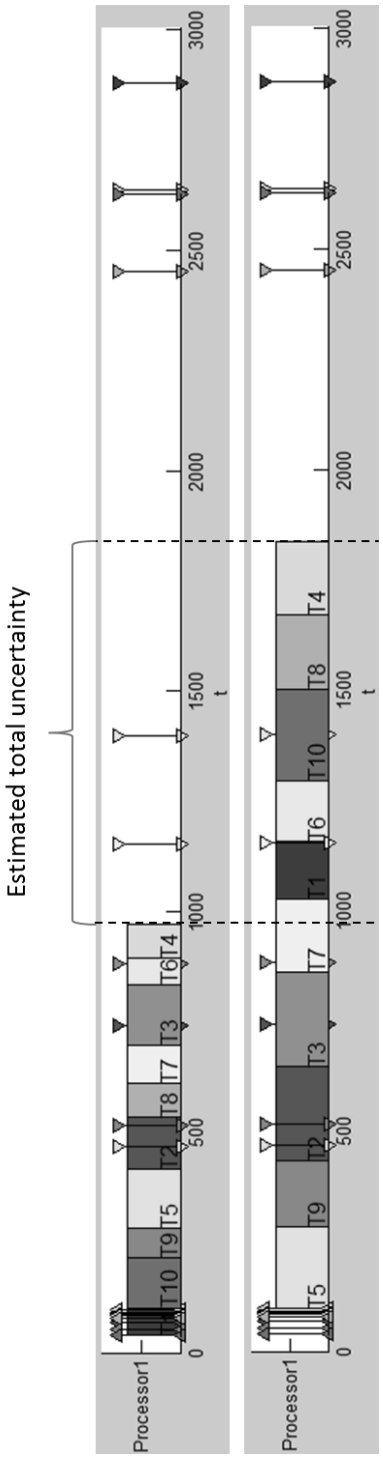

Figure 4.1: Scheduling 4 uncertain tasks based on interval computation assuming upper bound

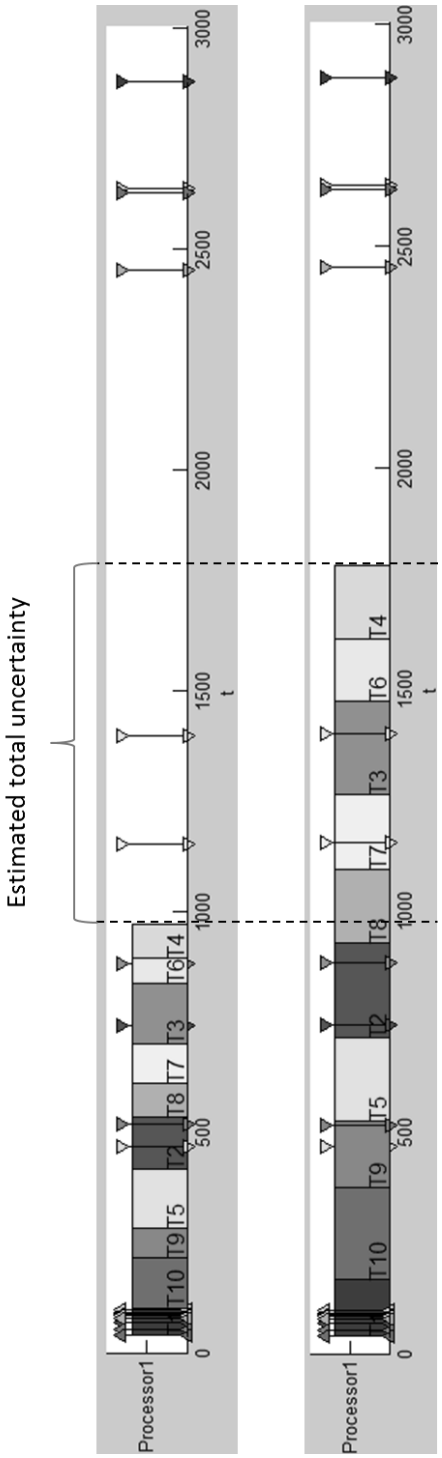

Figure 4.2: Scheduling 4 uncertain tasks based on intervals average (numerical approximation)

4.5 Conclusion

Scheduling under the constraints of the earliest start time and due-date is one of the famous problems that uses the branch and bound method. The problem is classified according to its complexity as NP-Hard problem, which make the MILP, quadratic programming and combinatorial solutions not applicable. Bratley's algorithm is used to solve such problem assuming the data is certain and accurate, but the challenge is how to solve the problem with uncertain data.

This chapter shows how the proposed methodology is implemented to Bratley's algorithm. First, the algorithm is analysed, then the interval relations are stated and mapped to Allen's relations. The algorithm is extended later to implement interval arithmetic rather numerical calculation and implement the interval logic rather than Boolean logic.

Multiple experiments have been performed using multiple randomly generated datasets. The results showed that using interval programming gives more certain results than using numerical calculations such as midpoint bisection or upper-lower bounds calculations. The total uncertainty of the results has been measured according to the distance function between the upper and lower bounds of the final estimate of the objective functions as stated in the experiments section. The extended algorithm outperformed the standard algorithms in 30% of the cases with marginal added certainty of 7.2%. The performance of both algorithms was equal for the remaining 70%. In other words, the standard algorithm did not outperform the extended algorithm at all.

Chapter 5

Minimising the Impact of Bounded Uncertainty on Hodgson's Algorithm

Hossny, A.; Nahavandi, S.; Creighton, D.;, "Using Interval Programming to Minimise the Impact of Bounded Uncertainty on Hodgson's Scheduling Algorithm", submitted to Journal of IEEE Systems, March 2014 (under Review).

Uncertainty of data is an important issue in sequencing and scheduling as it increases costs and risks. Such issues are usually handled by either preventive or reactive techniques depending on the problem nature. Bounded uncertainty provides only the upper and lower bounds around each value without any probability density function or fuzzy membership function in between. The lack of information between lower and upper bound makes such probabilistic and fuzzy techniques not useful. This research describes how to use interval programming to minimise the impact of bounded uncertainty on the schedule. It finds a sequence of tasks that optimises the cost function to be more accurate and less uncertain. The proposed methodology used interval arithmetic and logic instead of numerical arithmetic and Boolean logic. It introduces the concept that by applying interval calculations on input parameters then approximating the end result, The result will be more optimal and accurate than approximating the input values before performing the calculations. The proposed

methodology is applied on Moore-Hodgson's algorithm, the total number of delayed tasks is decreased by 12%. These results emphasised that the main factor affecting the optimality is the total number of overlaps per data set.

5.1 Introduction

Most of real life schedules have a percentage of uncertainty depending on the nature of the problem and its context. One of the reasons of the schedule uncertainty is the inaccuracy in input parameters such as release time. Another reason is the run-time disturbances, such as sudden machine breakdown. These inaccuracies or disturbances lead to possible changes in the sequence or the schedule of tasks. Such changes cause that the cost function cannot be estimated exactly and the associated error function cannot be stated accurately, or can be stated with large margin. Thus, decision making becomes more difficult.

To overcome the uncertainty problem, decision makers have one of two options: the first is to assume an optimistic scenario with no problems; the margin of risk is increased. The other option is to assume the worst case will happen, requiring extra costs to cover the expected problems. There are two types of scheduling uncertainty: the first is predictable such as parametric inaccuracy and can be handled by preventive actions. The second is unpredictable as it occurs in runtime; it should be handled by an/the instant reaction of re-scheduling, known as reactive scheduling.

This research aims to minimise the effect of parametric uncertainty on the schedule, where each parameter is known to be inside a numerical interval with known upper and lower bounds. The lack of information that describes the bounded uncertainty leads to the need for interval programming including interval arithmetic and interval algebra. Applying interval programming in calculations then averaging the final result gives more accurate results than averaging input data before performing the calculation. The following example

describes the proposed way to deal with the bounded uncertainty.

Example of the proposed methodology:

1. Assume $x = [I_1^-, I_1^+], y = [I_2^-, I_2^+]$
2. Calculate the cost function using interval algebra: $Z = f([I_1^-, I_1^+], [I_2^-, I_2^+])$
3. Averaging the interval (bisecting): $result = Z/2$

Section 2 shows up a motivating example of the bounded uncertainty and how it affects the schedules. Section 3 introduces the concepts of uncertainty and scheduling uncertainty. Section 4 states the formal formulation of the problem using numerical calculations and interval programming. Section 5 explains the proposed methodology using interval programming and shows how it minimises the total uncertainty of the schedule. Section 6 describes the experiments and results. Section 7 summarises the research contribution into the conclusion and discusses the contribution.

5.2 Motivating Example

Aircraft landing is a good example for bounded uncertainty where the arrival times are not known accurately. Such inaccuracy results from the uncontrollable factors such as sudden changes in weather or the unexpected issues at the departure airport. According to Tavakkoli-Moghaddam [229], the landing process starts by the appearance of the aircraft at the sky within the range of the radar of the airport. The aircraft sends its information including the flight number, altitude and speed to the air traffic control tower. Then, the controller tells the aircraft which runway to use and when exactly to land. Figure 1 displays the holding pattern and primary and secondary stacks and how they are merged into one queue as a primary step for landing.

Figure 5.1: The solid lines describe how the aircraft act in non-emergency situations when they arrive to the airport while the runway is not yet available. When an aircraft arrives, it contacts the tower asking for landing permission. The tower evaluates the situation of all aircrafts then prioritises them, formerly it tells the aircraft to join a specific stack at a specified level until it receives the permission to land. In most of airports there exist 2 runways but only one is used for landing. The aircraft landing queue is represented in 2 stacks because of aerodynamics constraints.

If the airport is crowded or the runway is busy and the aircraft cant land now, it joins a previously defined holding pattern that keeps it circling around the airport until the traffic controller tower sends it the signal allowing it to land. Such a holding pattern takes an oval shape with one side above the runway and the other side away of it. The major airports with multiple runways use multiple holding patterns with minimum intersection between them. A recently appeared aircraft joins the holding pattern associated with its runway until

it gets order of landing from the air traffic controller. The controller decides which aircraft should land when taking in consideration the emergency cases and the cost of the delays.

As landing of aircrafts is formulated as a scheduling problem with the issue of uncertainty; it can be minimised using either preventive or reactive techniques or hybrids of them. The preventive technique uses time ranges or windows to be sure that a specific aircraft will arrive within its time frame and can never come before it considering distance, speed and fuel consumption and it will never come after it because of rerouting or departure issues. The reactive scheduling considers the current situation of each aircraft including fuel availability, medical emergencies and delay penalties. In reality, airports build the landing schedule using hybrid of preventive and reactive techniques. The preventive technique allocates a time frame for each aircraft that is expected to arrive. The time frame states the earliest landing time, the targeted landing time and the latest possible landing time. The reactive technique is applied inside each frame according to the instant need of each aircraft depending on the priority criteria. Each time frame has many flights that are expected to arrive within its range. Sometimes the allocated arrival time frames overlap with each other. Table 1 shows landing schedule in the airport of Heathrow, which have been used by Beasly *et al.* in [230] and [231].

5.3 Uncertainty Review

The definition of uncertainty varies according to the application domain, which can be industrial [232], information technology [233] or managerial [234]. In the scheduling context, uncertainty has been defined as the dissimilarity between the expected values at setup time and the actual values at runtime. This uncertainty exists because of lack of knowledge or misleading information or error in the model-based predictions.

Dealing with uncertainty first requires an understanding of the nature of problem then

Table 5.1: Example of the data used by Beasly *et al.* [230] and [231] air traffic scheduling research. Appearance time is the release time, earliest landing is the lower bound of the landing interval and latest landing is the upper bound of landing interval.

Appearance Time (Mins)	Earliest Landing (Mins)	Target Landing (Mins)	Latest Landing (Mins)	Penalty for Early Landing	Penalty for Late Landing
54	129	155	559	10	10
120	195	258	744	10	10
14	89	98	510	30	30
21	96	106	521	30	30
35	110	123	555	30	30
45	120	135	576	30	30
49	124	138	577	30	30
51	126	140	573	30	30
60	135	150	591	30	30

representing it in a model that can minimise the uncertainty or neutralise its effect in the problem. Bandemer discussed four ways to model uncertainties in engineering domains: these are (1) numerical approximation, (2) interval mathematics (3) probability theory and (4) fuzzy theory [222].

The available information that describes the uncertain data can determine which model should be used. If the only known information is the range around the values with upper and lower bounds where the distribution in between is uniform, then interval mathematics techniques will be the most effective. In some cases the probability distribution can be extracted for the values inside the interval as confidence intervals or prediction intervals. In such cases probabilistic techniques will better minimise the uncertainty. If the uncertainty has a known membership function, the fuzzy theory and its techniques can be used. It is possible to use hybrids of these techniques according to the nature of the problem.

Interval programming has been used in many optimisation problems under uncertainty. Sun et al [235] used interval fuzzy programming to build a model for environmental management under uncertainty. Jin *et al.* [236] used dual interval programming to allocate

water irrigation under uncertainty. Fan *et al.* [237] introduced robust interval linear programming (RILP) as enhancement for interval linear programming (ILP) to maintain the robustness of engineering and environmental problems within uncertainty constraints. Liu et al. used interval linear programming to build risk explicit model to optimise the nutrient-reduction for a lake within uncertainty [238]. Li and Huang introduced a method using interval-based possibilistic programming minimise the cost of waste management under uncertainty [239].

5.4 Problem Formulation

Scheduling a set of tasks according to their due-dates with single resource constraint has been solved using Moore-Hodgson's algorithm [218] based on numerical or time point estimate. To minimise the impact of bounded uncertainty, the problem is formulated based on time intervals instead of time points, and then the algorithm is extended to do interval computation including interval calculation and interval logic. The Algorithm in figure 1 explains the detailed steps of the algorithm using numerical calculations and Boolean logic.

Data: $J = set\ of\ all\ tasks$;

$JS = \phi,\ set\ of\ scheduled\ tasks$;

$JN = \phi,\ set\ of\ delayed\ tasks$;

$p_i =\ processing\ time\ of\ task\ number\ i$;

$tp = 0,\ total\ processing\ time$;

$d_i =\ due\ date\ of\ task\ number\ i$;

$n =\ number\ of\ tasks$;

Result: a set of scheduled tasks that maintain the cost function optimal.

```
Sort all tasks with due date such that d_1 < ... < d_n;
for i = 1 ... n do
    tp = tp + p_i ;
    JS := JS ∪ J_i ;
    if tp > d_i then
        k = find longest task in JS ;
        JS = JS − J_k ;
        JN = JN ∪ J_k ;
    end
end
```

Algorithm 1: Moore-Hodgson's algorithm minimises the number of delayed tasks according to earliest due-date; it applies numerical calculations and Boolean logic on the time-point estimates.

For the aircraft landing problem, the due-date represents the targeted landing time and the single resource represents the runway in the airport and the task processing represents the landing time. Most of aircrafts arrive either earlier or later than the targeted landing time, which causes the holding pattern stack. This makes building the schedule based on

the landing time frame will be more useful than building it based on the exact time point for decision maker, especially if the possible deviations at reality is taken into consideration. This proactive technique that is known as preventive scheduling too is used to handle predictable uncertainty situations.

The landing sequence problem using exact estimate and excluding any possible uncertainty can be formulated as follows:

Let J set of all aircrafts waiting to land of size N
x_i Landing task for aircraft $i \; \forall \; x_i \in J$ & $0 < i < N$
t_i Targeted time to land
a_i Actual time to land
E_i Early time to land with no penalty
L_i Late time to land with no penalty

The objective function is to minimise total number of delayed tasks. For example, the aircrafts waiting to land in the holding pattern because of the unavailability of the runway. It is formulated using numerical estimate as:

$$\min Z = \sum_{i=1}^{N} f(x_i) where f(x_i) = \begin{cases} 1, & \text{if } a_i > L_i. \\ 0, & \text{if } a_i < L_i \; . \end{cases} \tag{5.1}$$

The algorithm that is used to optimise this objective function is called Moore-Hodgson's algorithm [218]. It is working on two main steps, assuming the set of tasks is named J:

1. Determine the subset of tasks that can be processed on time and name it JS.

2. Build the schedule from the subsets Js and JN where JN is the set of tasks that cannot be processed on time $(J = JSJN)$

To build the schedule based on time intervals, the objective function is formulated as:

$$\min Z = \sum_{i=1}^{N} f(x_i) where f(x_i) = \begin{cases} 1, & \text{if } [a_i, a_i] after [E_i, L_i]. \\ 1, & \text{if } [a_i, a_i] inside [E_i, L_i]. \\ 0, & \text{if } [a_i, a_i] before [E_i, L_i]. \end{cases} \tag{5.2}$$

5.5 Proposed Methodology

The proposed methodology is to upgrade Hodgson's algorithm to use interval variables, interval arithmetic and interval logic instead of numerical variables, arithmetic and logic. Such upgrade will minimise the impact of uncertain inputs on the estimate of the cost function by making it more accurate. This methodology is called interval programming that is used to minimise the impact of bounded uncertainty of input parameters. It gave better results varying according to the problem nature.

First the interval arithmetic and interval algebra are explained then the relational operators are defined to be used in extending this algorithm to support interval programming including arithmetic and logic.

5.5.1 Interval Arithmetic

Interval arithmetic has been introduced by Moore-Hodgson [218] and became a widely used technique to represent mathematical uncertainty where the numerical value is not

known accurately but the only known values are the upper and lower bounds. Moore proposed the definitions of interval arithmetic operations in 1. 1 to solve system of equations [228], where it uses the interval as a replacement of the number in all calculations, where the numerical value i belongs to the interval I such that $I = [I^-, I^+]$, some representations use the notation $I = [i-i, i+i]$ but this refers to known mean for the uncertainty range and exact error margin, which is not always true.

Interval arithmetic defined a set of arithmetic operations as basis for any interval-based equations or algorithms, such that operations of additions, subtraction, multiplication and division are redefined as follows:

$I_1 \oplus I_2 = [I_1^- + I_2^-, I_1^+ + I_2^+](1)$

$I_1 \ominus I_2 = [I_1^- - I_2^-, I_1^+ - I_2^+](2)$

$I_1 \otimes I_2 = [min(I_1^- I_2^-, I_1^- I_2^+, I_1^+ I_2^-, I_1^+ I_2^+), Max(I_1^- I_2^-, I_1^- I_2^+, I_1^+ I_2^-, I_1^+ I_2^+)](3)$

$I_1 \oslash I_2 = [I_1^-, I_1^+][1/(I_2^-), 1/(I_2^+)](4)$ such that $0[I_2^-, I_2^+]$.

As interval arithmetic maintain the properties of the operations such as associativity and commutativity of addition and distributivity of multiplication. Therefore it can be applied to matrices as well.

$[A][B] = [C] = c_i j(p \times r)$ (7)

Where $[A] = a_i j = [a_i j^-, a_i j^+](p \times q)$

and $[B] = b_i j = [b_i j^-, b_i j^+](q \times r)$

and the elements of the matrix $[C]$ are given by:

$C_i j = \sum(k = 1)^q \; a_i k b_k j; \; i = 1, 2, , p \; and \; j = 1, 2, \ldots, r(8)$

Implementing the interval arithmetic to algorithms helps determining to which extent the uncertainty will be decreased. This implies that applying the operations that decreases the intervals' range should be performed at early stages of the solutions. This is mentioned by Maiumder and Rao [228] and named as interval optimisation.

To apply interval arithmetic in the algorithm, the numerical value of every due-date (d_i)

Table 5.2: The definitions of relations between any two intervals are defined by Allen's algebra. The timeline examples represent the relation between the lower bound of the first interval and the lower bound of the second interval and the relation between the upper bound of the first interval and second interval, where $x^- < x^+$ and $y^- < y^+$.

Basic Relation	Operator	Example	End points
x precedes y y preceded by x	p p^-1	xxxx yyyy	$x^+ < y^-$
x meets y y met by x	m m^-1	xxxxx yyyyy	$x^+ < y^-$
x overlaps y y overlapped by x	o o^-1	xxxxxxx yyyyyy	$x^- < y^- < x^+$, $x^+ < y^+$
x during y y includes x	d d^-1	xxxx yyyyyyyyyy	$x^+ < y^+$, $x^+ < y^+$
x starts y y started by x	s s^-1	xxxx yyyyyyyyyy	$x^- = y^-$, $x^+ < y^+$
x finishes y y finished by x	f f^-1	xxxxx yyyyyyyyyy	$x^+ = y^+$, $x^- > y^-$
x equals y	$\equiv$	xxxxxxxxxx yyyyyyyyyy	$x^- = y^-$, $x^+ = y^+$

will be replaced with a due-date interval ($[d_i^-, d_i^+]$) and replace every landing processing time value (p_i) to its interval representation $[p_i, p_i]$. This will change the step number 3 to be as follows:

$$tp = tp \oplus P_i \Longrightarrow tp^- = tp^- + p_i, tp^+ = tp^+ + p_i$$

5.5.2 Interval Algebra and Temporal Relations

Interval relations are first introduced by James Allen in what is called interval algebra or Allen's algebra [32, 240, 241]. Interval algebra defines relations between any two intervals in timely manner using 13 operations depending on the combination of numerical relations between the start-time and end-time of the two time intervals as illustrated in table 6.1.

The proposed methodology used Allen's algebra for the inequality comparison in Hodgson's Algorithm as it applies interval programming to solve the sequencing and the scheduling problem as follows:

$If(tp \succeq d_i) \longrightarrow If(d_i\ precedes\ tp\ \vee tp\ meets\ d_i\ \vee\ tp\ starts\ d_i)$

Although Allen's algebra provided a good contribution in time related problems such as relating dates in data mining and in historical text analysis in the NLP domain, but it does not provide explicit operations for less, greater or equal relations for numerical intervals. It is not expressive mathematically and it will not be easy to be used for future algorithms or problems. This lead to define the meaning of such relations using the interval components of lower bound, higher bound and distance between them as in the following section.

5.5.3 Defining Non-Temporal Interval Relations

In order to solve the optimisation problems using interval formulation, The definitions of the non-temporal interval relations including the equality and inequality operators are stated in the definiitions 5.5.1, 5.5.2 and 5.5.3.

Definition 5.5.1

For any two real intervals $I_1 = [I_1^-, I_1^+]$ and $I_2 = [I_2^-, I_2^+]$ I_1 is less than I_2 if $I_1^- < I_2^-$ and $I_1^+ < I_2^+$ I_1 is less than I_2 if $I_1^- = I_2^-$ and $I_1^+ < I_2^+$ I_1 is less than I_2 if $I_1^- > I_2^-$ and $I_1^+ < I_2^+$ where $I_1^-, I_1^+, I_2^-, I_2^+ R$

Definition 5.5.2

For any two real intervals $I_1 = [I_1^-, I_1^+]$ and $I_2 = [I_2^-, I_2^+]$ $I_1 = I_2$ if and only if $I_1^- = I_2^-$ and $I_1^+ = I_2^+$ where $I_1^-, I_1^+, I_2^-, I_2^+ R$,

Definition 5.5.3

Otherwise,I_1 is larger than I_2

Such definitions are built by stating all combinations of Allen's operators and testing them against a dataset of 1110 tasks then calculate the cost function. Then Allen's operators are mapped back to the relational numerical inequality operators considering the lower bound and upper bound of right hand side and left hand side.

These definitions are stated for this specific scheduling problem and it may vary according to the nature of the problem and the dominance of each component of the interval, which are the start time (lower bound), the end time (upper bound) and the range between them. For the mentioned scheduling problem, this definition gave the best results.

Sometimes the arithmetic and logic operations are applied to an interval and a number, which leads to convert one of the operands to the format of other operand either by converting interval into number or converting number into interval. It was preferred to convert number into interval as ageneralisation, where the lower bound equals to upper bound with range equals zero.

Definition 5.5.4

Definition 5.5.4 for any number xR,it can be represented as the interval $X = [x, x]$

After applying the stated definitions on Hodgson's algorithm that is illustrated in 2, the new algorithm will be as follows:

Data: $J = set\ of\ all\ tasks$;

$JS = \phi,\ set\ of\ scheduled\ tasks$;

$JN = \phi,\ set\ of\ delayed\ tasks$;

$p_i =\ processing\ time\ of\ task\ number\ i$;

$tp = 0,\ total\ processing\ time$;

$d_i =\ due\ date\ of\ task\ number\ i$;

$n =\ number\ of\ tasks$;

Result: a set of scheduled tasks that maintain the cost function optimal.

Sort all tasks with due date such that $d_1 < \cdots < d_n$;

for $i = 1 \ldots n$ **do**

  $tp = tp + p_i$;

  $JS := JS \cup J_i$;

  if $tp > d_i$ **then**

    $k = find\ longest\ task\ in\ JS$;

    $JS = JS - J_k$;

    $JN = JN \cup J_k$;

  end

end

Algorithm 2: Moore-Hodgson's algorithm minimises the number of delayed tasks according to earliest due-date; it applies numerical calculations and Boolean logic on the time-point estimates.

5.6 Experiments and Results

The proposed experiments compare the interval programming approach with the standard numerical approach. As the formulation of the objective function is upgraded to use intervals, the algorithm is upgraded to implement the interval arithmetic and logic. The two algorithms are applied on the same dataset and calculate the two objective functions and relate them with the factors affecting the intervals including overlapping, containment and interval ranges.

The data used in this experiment consist of 13 scheduled datasets of 300 aircraft waiting for landing permission. This dataset is used by Beasly in scheduling aircraft landing in static case [230] and in solving the displacement problem and scheduling aircraft landings dynamically [231]. This dataset can be downloaded online from the OR library. The three tables 5.3, 5.4 and 5.5 show few records of the dataset that have been used in this research and Beasly's research as well. The three tables 5.3, 5.4 and 5.5 displays 10 sets, each one contains 30 tasks waiting to be processed, i.e. aircrafts waiting to land in the holding pattern as declared in section 5.2. The table states total number of overlaps, total number of containment happened for each set and summation of interval distances.

The experiments are implemented using the Torsche toolbox, which is developed by Sucha et al. [241]. The proposed modification on Hodgson's algorithm is implemented by a cloning the toolbox then changing the code to cope with the interval arithmetic and logic instead of numerical arithmetic and Boolean logic. We applied same experiments on both algorithms, the one that is implemented by Torsche and the newly developed one, and then the results are compared to find out the optimality of the cost function, which is defined by the total number of late tasks.

The main factors affecting the schedule optimality within bounded uncertainty are: (1) Relations between any two intervals, which can be disjoint, overlapped or contained. (2)

Table 5.3: The ten datasets used in the experiment each have 30 aircraft waiting to land before the deadline. The datasets contain different arrival times and deadlines for each aircraft, which affects the number-of-overlaps, number-of-containment, and total-uncertainty-range. Such differences determine number of late tasks, and margin of enhancement achieved by interval programming.

Set Number	Number of Overlaps	Late Tasks Using Intervals	Late Tasks Using Bisecting	optimised Delays	Enhancement Ratio
1	176	11	13	2	15%
2	177	10	13	3	23%
3	174	9	13	4	30%
4	182	10	13	3	23%
5	49	2	5	3	60%
6	26	1	1	0	0%
7	24	0	0	0	0%
8	34	0	0	0	0%
9	179	9	13	4	30%
10	158	6	10	4	40%

Table 5.4: The ten datasets used in the experiment each have 30 aircraft waiting to land before the deadline. The datasets contain different arrival times and deadlines for each aircraft, which affects the number-of-overlaps, number-of-containment, and total-uncertainty-range. Such differences determine number of late tasks, and margin of enhancement achieved by interval programming.

Set Number	Number of Containments	Late Tasks Using Intervals	Late Tasks Using Bisecting	optimised Delays	Enhancement
1	22	11	13	2	15%
2	24	10	13	3	23%
3	28	9	13	4	30%
4	10	10	13	3	23%
5	34	2	5	3	60%
6	0	1	1	0	0%
7	0	0	0	0	0%
8	20	0	0	0	0%
9	20	9	13	4	30%
10	64	6	10	4	40%

Table 5.5: The ten datasets used in the experiment each have 30 aircraft waiting to land before the deadline. The datasets contain different arrival times and deadlines for each aircraft, which affects the number-of-overlaps, number-of-containment, and total-uncertainty-range. Such differences determine number of late tasks, and margin of enhancement achieved by interval programming.

Set Number	Summation of Interval Distances	Late Tasks Using Intervals	Late Tasks Using Bisecting	optimised Delays	Enhancement
1	9,063	11	13	2	15%
2	9,694	10	13	3	23%
3	9,685	9	13	4	30%
4	9,454	10	13	3	23%
5	6,479	2	5	3	60%
6	4,935	1	1	0	0%
7	4,881	0	0	0	0%
8	6,550	0	0	0	0%
9	9,689	9	13	4	30%
10	10,693	6	10	4	40%

Distance between lower bound and upper bound of each interval. So, the correlation factor between the number of overlapping intervals and the optimised delays is calculated and it output a score of 0.81. The number of delayed tasks using interval-based algorithm is compared versus the number of delayed tasks using numerical-based algorithm considering the number of overlaps happened in each set and the result is represented in figure 5.2. It is noticeable that the number of delayed flights jumps up then down for the number of overlaps 80, this happens because flight delays are not dependent on number of overlaps 100% because correlation is less than 1 and there exist other factors affects the delays.

The second factor affecting the schedule optimality within bounded uncertainty is the containment of the interval. So, the correlation between this factor and the optimised delays is calculated and it gave a score of 0.65. The number of delayed tasks using interval-based algorithm is compared against the number of delayed tasks using numerical-based algorithm considering the number of containment happened in each set and the result is

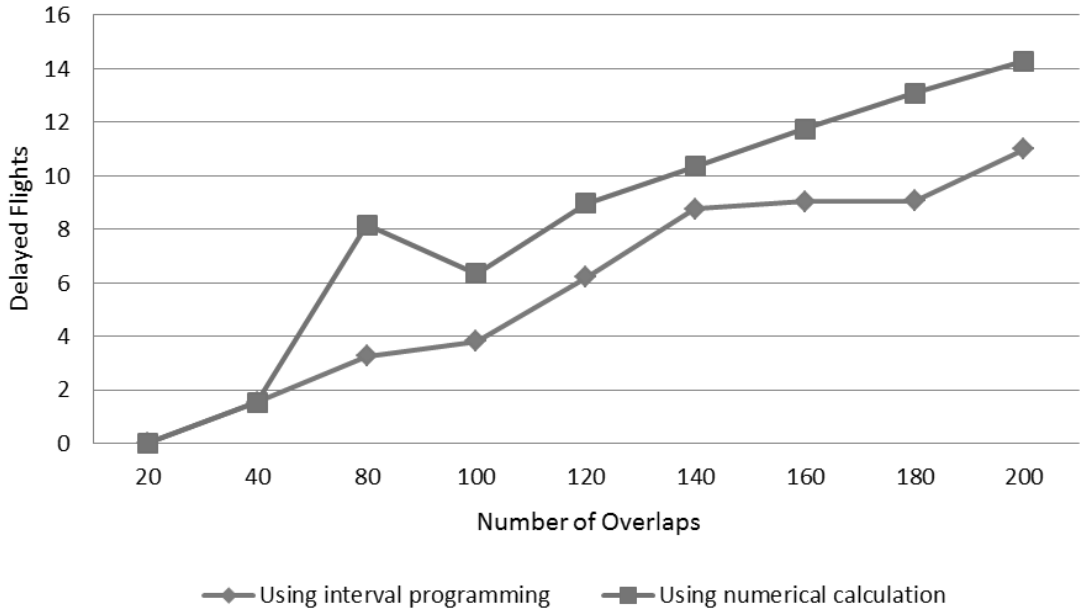

Figure 5.2: This chart illustrates the relation between the number of delayed aircraft and the number of overlaps in each dataset. The red diamonds represent the number of delayed flights resulting from approximating the data numerically before applying the algorithm. The blue triangles represent the number of delayed flights resulting from applying the interval-based algorithm then approximate the final result. The red and blue lines describe the trend of delayed flights and indicate that they are directly proportional with the number-of-overlaps per each dataset.

represented in figure 5.3. According to the figure and correlation, it is difficult to claim any strong relation between number of intervals contained and number of delayed tasks.

The third factor affecting the schedule optimality within bounded uncertainty is the uncertainty range, which is calculated by summing up the distance between the upper and lower bound for each interval using the equation $_(i = 1)^n I_i^+ - I_i^-$. By calculating the

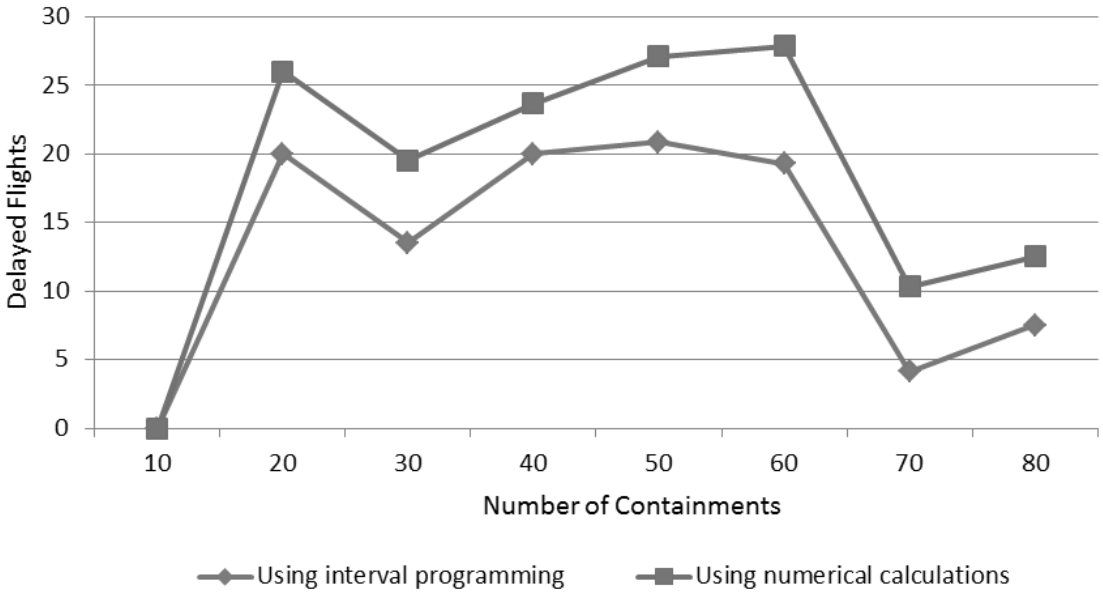

Figure 5.3: This chart illustrates the relation between the number of delayed aircraft and the number of containments in each dataset. The red diamonds represent the number of delayed flights resulting from approximating the data numerically before applying the algorithm. The blue triangles represent the number of delayed flights resulting from applying the interval-based algorithm then approximate the final result. The red and blue lines indicate the trend of delayed flights and emphasise that they are independent of the number-of-containments per each dataset.

correlation between this factor and the optimised delays and it gave a score of 0.86. The number of delayed tasks using interval-based algorithm is compared against the number of delayed tasks using numerical-based algorithm considering the uncertainty range in each set and the result is represented in figure 5.4. The figure shows that the number of delayed flights decreased for the range of uncertainty of 8000 against the trend of the curve, this happened because flight delays are not dependent on ranges totally 100% as the correlation

is less than 1 and there exist other factors affects the delays.

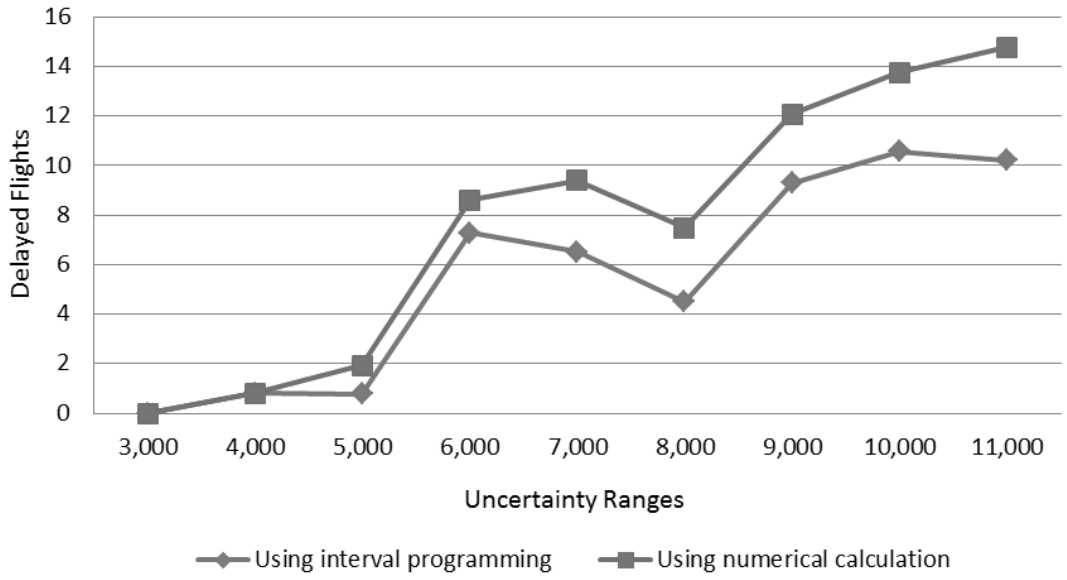

Figure 5.4: This chart illustrates the relation between the number of delayed aircraft and the uncertainty ranges. The red diamonds represent the number of delayed flights resulting from approximating the data numerically before applying the algorithm. The blue triangles represent the number of delayed flights resulting from applying the interval-based algorithm then approximate the final result. The red and blue lines indicate the trend of delayed flights and emphasise that they are directly proportional and highly correlated with the uncertainty-range per each dataset.

The results illustrated in figures 5.2, 5.3 and 5.4 showed up that interval programming gives better results as long as the total uncertainty range or the number of overlaps increases as they both represents possible dependency between consecutive intervals. The results also showed that interval programming gives better results as total uncertainty range increases

especially that uncertainty range and overlaps are highly correlated. Another conclusion is that interval programming cannot use the number of intervals that is contained in other ones because the trend of the curve is not stable and the correlation factor is very low. Finally, the interval programming does not give better results than bisecting or approximation in case of disjoint intervals as the order of the tasks will be exactly the same as if the numerical average of lower or upper bound of the interval are used.

It is important to mention that number of delayed flights estimated using interval programming was always less than the number of delayed flights estimated by numerical approximation. The experiments showed that the factors affecting the number of delayed aircraft are the total uncertainty range and the number of overlaps, which are dependent on each other. By calculating the correlation score between them, the score was 0.95. So, the total uncertainty range can be used to minimise the number of delayed tasks.

For the aircraft landing problem, the other factors should be covered including emergency cases, freezing time, the stability of the weather above the airport for the stacked holding pattern and the readiness of the airport to receive new aircraft landings. These factors are out of the scope of this paper, but by relaxing these constraints and considering just the appearance time in the sky, the late flights should have higher probability to arrive within small interval remaining even if it overlaps with another interval. This will give possibility to resolve such uncertainty using Bayesian probability techniques instead of interval-based techniques, which is more accurate and easier.

5.7 Conclusion

Uncertainty is considered a major challenge for scheduling process from the engineering and decision making perspectives. Eliminating the root causes of data uncertainty is not

always possible. So, the effect of uncertainty should be minimised by increasing the accuracy without affecting the optimised objective function negatively. Bounded uncertainty has a special nature as it is identified only by its lower and upper bounds without any information describing what is in between them. In contrast; probabilistic uncertainty has its probability density function and fuzzy uncertainty has its membership function. Bounded uncertainty is usually handled by calculating the average to be used in decision making calculations where the approximation is applied before the calculation.

This research minimises the impact of bounded uncertainty on the optimality of the objective function. It proposed a new preventive scheduling technique that uses interval programming including interval arithmetic and interval algebra, especially that the probability density function and fuzzy membership functions are not available. The proposed technique starts by performing the calculation of interval input data then averaging-bisecting the final output. The used case study to emphasise the validity of the proposed technique is the problem of "minimizing the number of tardy jobs", as it is usually solved on numerical basis using Moore-Hodgson's algorithm. The the nature of the problem is considered, the objective function is reformulated to be interval-based and the algorithm is promoted to use interval arithmetic and interval logic, and then tested the algorithm on the interval data to achieve the objective function.

The experiments ad results showed that the effectiveness of the interval programming increases proportionally with the total number of overlaps and the total uncertainty distance inside the dataset. It is important to mention that the every task inside each dataset was of different distances between lower and upper bounds. The proposed methodology is applied on the scheduling problem of minimizing the number of tardy jobs and it achieved 12% less number of tardy jobs than by using the average of the intervals of the cost function.

Scheduling uncertainty happens in airports different stages including aircraft landing, aircraft takeoff and passenger queues. It also happens in medical field in allocating patients

to surgical operating rooms, usage of intensive care rooms, emergency rooms and normal medical queues and allocating different patients to different doctors and resources. The scheduling uncertainty problem and its optimisation is important issue for countries trying to apply medical care for all people. Scheduling uncertainty affects many other domains including logistics, transportation, traffic, assembly lines, supply chains and chemical plants. Minimizing the impact of bounded uncertainty in theory opens the door to apply it in all other domains and on different algorithms, which will be the future prospects for this research.

Chapter 6

Minimising Impact of Bounded Uncertainty on McNaughton's Algorithm

Hossny, A.; Nahavandi, S.; Creighton, D.;, "Minimizing Bounded Uncertainty Impact on McNaughton's Scheduling Algorithm via Interval Programming", in IEEE international conference on Systems Man and Cybernetics, 2013. IEEE-SMC 2013. Manchester, UK, 13-16 Oct. 2013.

Uncertainty of data affects decision making process as it increases the risk and the costs of the decision. One of the challenges in minimizing the impact of the bounded uncertainty on any scheduling algorithm is the lack of information, as only the upper bound and the lower bound are provided without any known probability or membership function. On the contrary, probabilistic uncertainty can use probability distributions and fuzzy uncertainty can use the membership function.

McNaughton's algorithm is used to find the optimum schedule that minimises the makespan taking into consideration the preemption of tasks. The challenge here is the bounded inaccuracy of the input parameters for the algorithm, namely known as bounded

uncertain data. This research uses interval programming to minimise the impact of bounded uncertainty of input parameters on McNaughton's algorithm, it minimises the uncertainty of the cost function estimate and increase its optimality. This research is based on the hypothesis that performing the calculations on interval values then approximate the end result will be produce more accurate results than approximating each interval input then apply the numerical calculations.

6.1 Introduction

McNaughton's algorithm is used to solve the scheduling problem with a set of independent tasks with known processing times [242]. These tasks have to be allocated to parallel identical processors in order to minimise total schedule length by minimising the maximum completion time C_{max} . This algorithm considers the preemption constraint of tasks and produces the optimal schedule. The objective function is defined as the maximum of the two values: $max(p_j)$; $(p_j)/m$, such that m is the number of processors.

Uncertainty is a major challenge to applying scheduling to real life problems similar to job shop scheduling of automotive industry. Scheduling algorithms optimise the cost function theoretically, as it assumes that input data are exact and accurate, which leads to exact estimate for the cost function without any change of values at runtime. Such assumption leads to more risk and more cost as the parametric data in reality is often not so accurate. The initial estimate of the data may not be 100% accurate and the parametric data may vary at the run time. To increase the accuracy percentage, the possible reasons and nature of the uncertainty should be identified, then either to eliminate them or minimise their impact.

The uncertainty challenge of scheduling algorithms affects different scheduling applications in different domains including the mining industry [243, 244] [3], liquefied natural

gas industries [245] , medical follow-up, surgery resource allocation, electricity and power uncertainties in wind farms , telecom networks [246] and logistics [247].

The traditional way to minimise the impact of bounded uncertainty is to approximate the interval of the uncertain parameter to the lower bound or upper bound or to bisect the interval to its average, then use the approximated numerical value for the calculations. Figure 6.1 represents how old systems minimise the impact of bounded uncertainty.

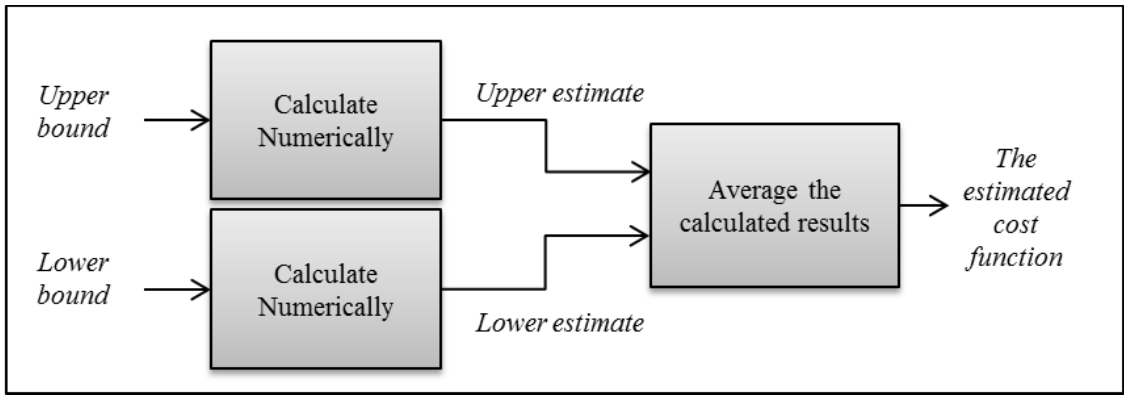

Figure 6.1: A flowchart for approximation input parameters before calculation the objective function

This research aims to minimise the impact of bounded uncertainty of the processing time parameter for each task, which is formulated as mathematical interval, where $I = [I^-, I^+]$, such that I^- represents the lower bound and I^+ represents the upper bound taking into consideration that the probability distribution is not known inside, which force us to consider it uniform to ensure equal chance to all values between lower and upper bounds. The algorithm is implemented using interval computation [248] including interval arithmetic [218] and interval algebra [33], which is considered as mathematical uncertainty handling technique [222] . Figure 6.2 describes how the proposed sequence of steps will

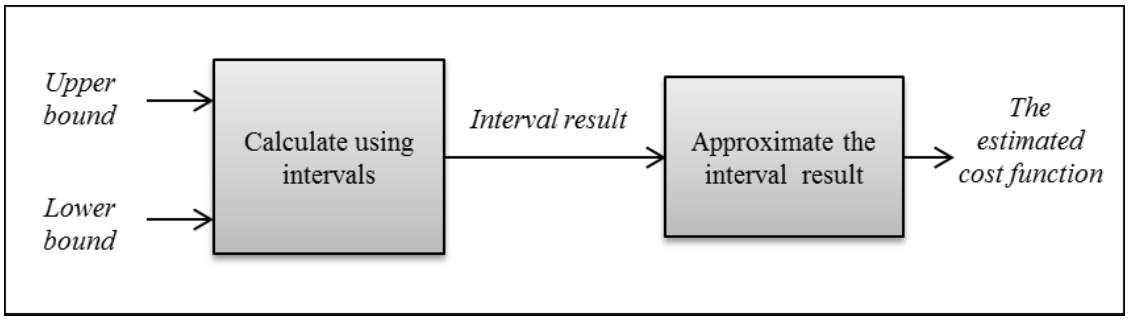

Figure 6.2: A flowchart for calculating using intervals before approximating the end results

minimise the impact of bounded uncertainty.

Section 2 introduces scheduling uncertainty concepts and techniques. Section 3 explains the proposed methodology including interval programming, arithmetic and logic and how to use them to minimise the uncertainty impact on McNaughton's algorithm. Section 4 describes the experiments and results. Section 5 closes with conclusions.

6.2 Proposed Methodology

The proposed methodology is to use interval programming including interval arithmetic and interval algebra as alternative to numerical computation in the scheduling algorithms to find less uncertain sequence of tasks to achieve better cost function.

6.2.1 Interval programming

Interval programming is a widely used technique to represent mathematical uncertainty where the numerical value is not known exactly and the only available information is the upper and lower bounds. Interval programming uses interval arithmetic, interval logic and

interval temporal relations. Interval arithmetic was first introduced by Moore to solve system of equations [218] . It simply uses the interval as a replacement of the number in all calculations and where the numerical value i belongs to the interval I where $I = [I^-, I^+]$. Some representations use the notation $I = [i - \varepsilon, i + \varepsilon]$ but this refers to known mean for the uncertainty range and an exact error margin, which is not always available.

Interval arithmetic defines a set of arithmetic operations as basis for any interval-based equations or algorithms, such that operations of additions, subtraction, multiplication and division are redefined as follows:

$$I_1 \oplus I_2 = [I_1^- + I_2^-, I_1^+ + I_2^+] \tag{6.1}$$

$$I_1 \ominus I_2 = [I_1^- - I_2^-, I_1^+ - I_2^+] \tag{6.2}$$

$$I_1 \otimes I_2 = [min(I_1^- I_2^-, I_1^- I_2^+, I_1^+ I_2^-, I_1^+ I_2^+), Max(I_1^- I_2^-, I_1^- I_2^+, I_1^+ I_2^-, I_1^+ I_2^+)] \tag{6.3}$$

$$\begin{array}{l} I_1 \oslash I_2 \ = [I_1^-, I_1^+][1/(I_2^-), 1/(I_2^+)] \\ such\ that\ 0 \notin [I_2^-, I_2^+]. \end{array} \tag{6.4}$$

As interval arithmetic maintain the properties of the operations such as associativity and commutativity of addition and distributivity of multiplication. Therefore it can be applied to matrices as well.

$$\begin{array}{l} [A][B] \ = [C] = c_{ij}(p \times r) \\ Where\ [A] \ = a_{ij} = [a_{ij}^-, a_{ij}^+](p \times q) \\ and\ [B] \ = b_{ij} = [b_{ij}^-, b_{ij}^+](q \times r) \end{array} \tag{6.5}$$

and the elements of the matrix $[C]$ are given by:

Table 6.1: The definitions of relations between any two intervals are defined by Allen's algebra. The timeline examples represent the relation between the lower bound of the first interval and the lower bound of the second interval and the relation between the upper bound of the first interval and second interval, where $x^- < x^+$ and $y^- < y^+$.

Basic Relation	Operator	Example	End points
x precedes y y preceded by x	p p^-1	xxxx yyyy	$x^+ < y^-$
x meets y y met by x	m m^-1	xxxxx yyyyy	$x^+ < y^-$
x overlaps y y overlapped by x	o o^-1	xxxxxxx yyyyyy	$x^- < y^- < x^+$, $x^+ < y^+$
x during y y includes x	d d^-1	xxxx yyyyyyyyyy	$x^+ < y^+$, $x^+ < y^+$
x starts y y started by x	s s^-1	xxxx yyyyyyyyyy	$x^- = y^-$, $x^+ < y^+$
x finishes y y finished by x	f f^-1	xxxxx yyyyyyyyyy	$x^+ = y^+$, $x^- > y^-$
x equals y	$\equiv$	xxxxxxxxxx yyyyyyyyyy	$x^- = y^-$, $x^+ = y^+$

$C_{ij} = \sum(k = 1)^q a_{ik}\, b_{kj} \;\; i = 1, 2, \ldots, p \; and \; j = 1, 2, \ldots, r$

The implementation of interval arithmetic in algorithms determines ts, which extent the uncertainty will be decreased, as executing the operations that decrease the interval range should be applied at early stages of the solutions. This is mentioned by Maiumder and Rao [228] in what is called interval optimisation.

Interval temporal relations were first identified by James Allen in interval algebra or Allen's algebra [30, 31]. Interval algebra identifies relations between time intervals using 13 basic operations representing the combination of relations between the lower bounds and upper bounds of the two intervals as represented in table 6.1.

Interval arithmetic achieved good results in uncertainty reduction for many problems. However in some cases it increases the solutions interval range, which makes such solutions less useful as it increases the uncertainty rather decreasing it. Such increases depend

on the mathematical expression of the problem such as when interval variables occur many times in same equation and interval variables depend on other variables in same equation. In such case, any change in any interval variable will propagate to all other variables increasing their range. Such problem have been mentioned by Maiumder and Rao [228] and they proposed the truncation method to deal with it by [249] . Same problem is discussed by Muscolino and Sofi naming it the dependency phenomena [250] and proposed the generalised interval arithmetic [251] or affine arithmetic [252, 253] as a solution.

6.2.2 Applying Interval Programming to Scheduling Algorithms

Any scheduling algorithm consists of a set of numerical operations and set of logical comparisons in addition to some algorithmic steps such as the assignment operation. In order to apply the interval programming to any scheduling algorithm, every numerical arithmetic operation should be replaced by interval arithmetic operation and every logical comparison should be replaced by interval algebraic comparison.

Although interval relations have been identified by Allen's algebra but it cannot be used directly as replacement to standard numerical comparison in normal algorithms, which lead to the need of mapping Allen,s interval relations to numerical relations. Different mappings have been tried targeting to minimise the uncertainty of the cost function until the best mapping is found as stated in table 6.2.

Table 6.2: Mapping Allen's relations between the intervals $[x^-, x^+]$ and $[y^-, y^+]$ to standard numerical relations.

x^-y^-	x^+y^+	Allen's Description	Logical Relation
$<$	$<$	Less	$\prec$
$<$	$<$	Overlap	$\prec$
$<$	=	Finishes	$\prec$
$<$	$>$	Contains	$\prec$
=	$<$	starts	$\prec$
=	=	equal	$\equiv$
=	$>$	Starts	$\succ$
$>$	$<$	Contained	$\succ$
$>$	=	Finishes	$\succ$
$>$	$>$	Overlap	$\succ$
$>$	$>$	Larger	$\succ$

6.2.3 Using Interval Programming to Minimise Uncertainty Impact on McNaughton's algorithm

McNaughton introduced his algorithm to find an optimal solution for the problem of scheduling with deadline and loss functions [242]. This algorithm finds a feasible schedule for preemptive tasks on multiple processors by minimizing the maximum completion time with given processing time and preemption constraint, which is formulated as $P|pmtn|C_{max}$ with $m-1$ preemptions at most. The algorithm creates the schedule machine by machine rather than task by task over time. Bo Chen *et al.* [254] used McNaughton's algorithm for online preemptive scheduling, Schmidt upgraded the algorithm to build schedule with limited machine availability [255], Hoogeveen also developed a new algorithm for preemptive scheduling in a two-stage multiprocessor flow shop as McNaughton's algorithm is NP hard in such case [256].

Data: $n := Number\ of\ all\ operations$;

$t_j := Processing\ times\ for\ all\ jobs$;

$m := Number\ of\ parallel\ processors$;

Result: a set of scheduled tasks on multiple processor with minimum completion time.

1 Find the makespan using the formula: $M^* = \max 1/m \sum_(j = 1)^n t_j, \max(t_j)$

2 Set processor number i=1.

3 Set remaining time on current processor , R= M^*

4 set the job index ,j=1

5 **for** $j = 1 \ldots n$ **do**

6 **if** $t_i < R$ **then**

7 $Processor(i)\ =\ Job(j)\ calculate\ completion\ C_j\ on\ processor\ i$

$R = R - t_j$ **if** $R = 0$ **then**

8 $i = i + 1,\ R = M^*$

9 **end**

10 **end**

11 **else**

12 $Schedule\ part\ of\ job\ j\ [R - C(j-1)]\ units\ of\ time\ on\ processor\ i$

$i = i + 1\ job(j).rt = t_j - [R - C_(j-1)]\ processor\ i =\ job(j).rt$

$compute\ its\ complete\ time\ on\ processor\ i$

$Update\ R = M^* - t_j - [R - C_(j-1)]$

13 **end**

14 $j := j + 1$

15 **end**

Algorithm 3: McNaughton's algorithm minimises the total completion time ; it applies numerical calculations and Boolean logic on the time-point estimates.

The source of uncertainty in this algorithm is the possible change of the processing time of every task while execution. For every t_j variable or any dependent variable including R or C_j or C_{max} will be formulated as intervals according to table 6.3.

Table 6.3: mapping algorithm variables to interval variables

Numerical Variable	Interval Variable
t_j	$[t_j^-, t_j^+]$
M^*	$[M^(*-), M^(*+)]$
R	$[R^-, R^+]$
C_j	$[C_j^-, C_j^+]$
C_max	$[C_max^-, C_max^+]$

After extending the algorithm, every arithmetic or Boolean variable will be replaced with interval-based variable according to the table 6.3 . The extended algorithm is stated in algorithm 4

Data: $n := Number\ of\ all\ operations$;

$[t_j^-, t_j^+] := Processing\ intervals\ for\ all\ jobs$;

$m := Number\ of\ parallel\ processors$;

Result: a set of scheduled tasks on multiple processor with minimum completion time.

1 Find the makespan using the formula:

$[M^-, M^+]^* = \max 1/m \sum_(j = 1)^n [t_j^-, t_j^+], \max[t_j^-, t_j^+]$

2 Set processor number i=1.

3 Set remaining time on current processor , [Rˆ-,Rˆ+]= [Mˆ-,Mˆ+]ˆ*

4 set the job index ,j=1

5 **for** $j = 1 \ldots n$ **do**

6 **if** $[t_i^-, t_i^+] < [R^-, R^+]$ **then**

7 $Processor(i) \Leftarrow Job(j)\ calculate\ completion\ [C_j^-, C_j^+]\ on\ processor\ i$
$[R^-, R^+] = [R^-, R^+] - [t_i^-, t_i^+]$ **if** $[R^-, R^+] = 0$ **then**

8 $i = i + 1,\ [R^-, R^+] = [M^-, M^+]^*$

9 **end**

10 **end**

11 **else**

12 $Schedule\ part\ of\ job\ j\ [[R^-, R^+] - C(j-1)]\ units\ of\ time\ on\ processor\ i$
$i = i + 1\ job(j).rt = t_j - [[R^-, R^+] - C_(j-1)]\ processor\ i = \ job(j).rt$
$compute\ its\ complete\ time\ on\ processor\ i$
$Update\ [R^-, R^+] = [M^-, M^+]^* - [t_j^-, t_j^+] - [[R^-, R^+] - C_(j-1)]$

13 **end**

14 $j := j + 1$

15 **end**

Algorithm 4: McNaughton's algorithm minimises the maximum completion time on all processors; it applies interval-based calculations and logic .

6.3 Experiments and Results

To verify that interval programming can provide more certain results than numerical approximation, a set of experiments have been done using dataset gathered from automotive production facility. The experiment is to build three schedules for the same uncertain input data, where the processing time of every task is an interval with known upper and lower bounds. The first schedule is built using the lower bound value, then apply the numerical algorithm. The second schedule is built using the upper bound value, then apply the numerical algorithm. The third schedule is built by implementing interval-based algorithm then bisecting the end result.

The objective function is stated as minimising the uncertainty of the maximum completion time of the whole schedule by minimizing the average of completion distance end result that is formulated by the equation 6.6:

$$Z = (\max_{i=0:n} C_i^- + \max_{j=0:n} C_j^+)/2 \tag{6.6}$$

The relation between intervals affects the complexity of the uncertainty. Such relation can be disjoint, overlapping or containing. The complexity of the interval-based uncertainty is measured by number of overlaps and containment as the disjoint does not affect the algorithm anyhow. Disjoint intervals does not affect the schedule anyhow, because $\max_{i=0:n} C_i^-$ equals $\max_{j=0:n} C_j^+$ and the cost function calculates the average between them. On the contrary, the overlapping and containing intervals build different schedules because $\max_{i=0:n} C_i^-$ does not equal to $\max_{j=0:n} C_j^+$.

Our experiment is performed using a Matlab tool box named Torsche that is developed by Sucha *et al.* [241]. The toolbox is upgraded to support interval programming, arithmetic and logic. This instance of our experiments used 4 similar processors and 10 tasks with uncertain processing times, which are represented in table 6.4.

The experiments, the stated examples and the results analysis showed up the following:

Table 6.4: Scheduling 10 tasks with uncertain processing times on 4 exact processors

Task	Lower Bound	Upper Bound	Distance
T_1	11	15	4
T_2	21	35	14
T_3	9	25	16
T_4	4	16	12
T_5	5	23	18
T_6	33	45	12
T_7	12	24	12
T_8	22	36	14
T_9	25	38	13
T_10	20	38	18

1. Building the schedule on the lower bound minimises the total schedule time, but with high probability of delays due to uncertainty of the input data as declared in figure 6.3.

2. Building the schedule using the upper bound minimises the probability of delays, but gives very long total schedule time as declared in figure 6.3.

3. Building the schedule using interval programming gives more flexibility for task allocation on processors, but it does not guarantee the minimum total completion time of the schedule as declared in figure 6.3.

As declared in the table 6.4, figure 6.3 and figure 6.3, the estimate of the objective function according to the numerical estimate is 140 units of time on all processors. On the other hand, the estimate of the objective function using the interval programming is 130. The uncertainty is reduced by 7.1% of all uncertainty time.

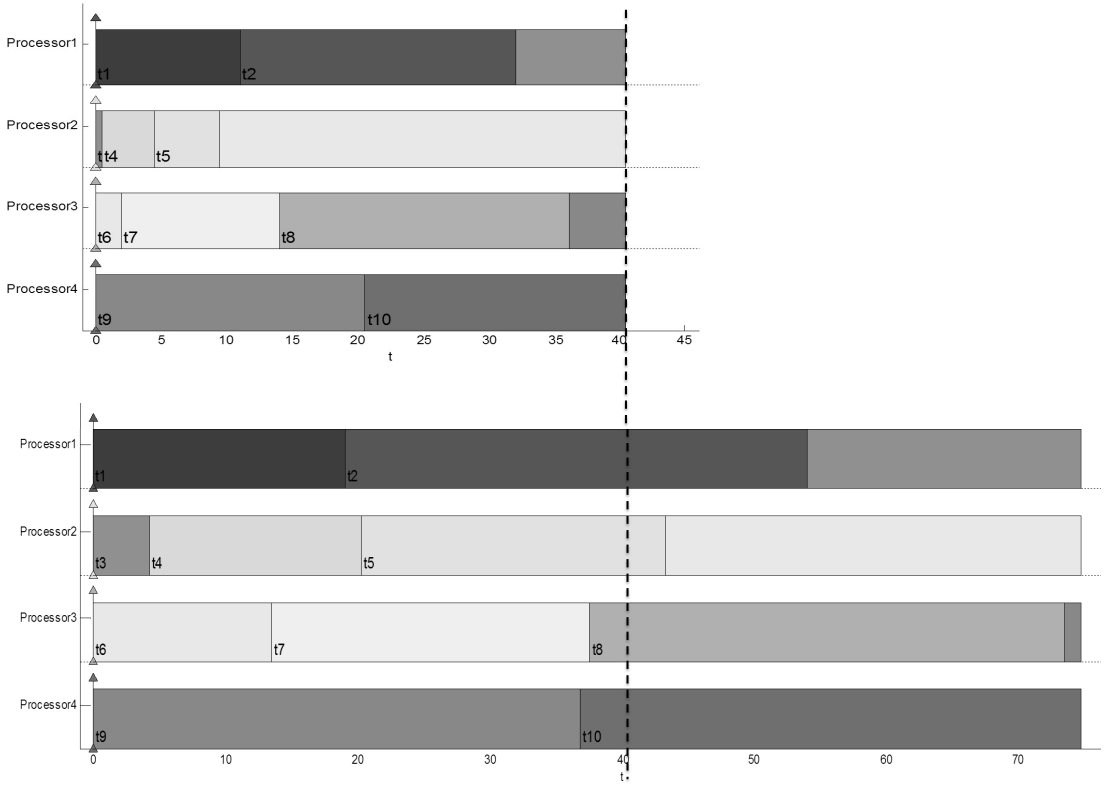

Figure 6.3: Scheduling 10 tasks based on numerical estimates of the lower and upper bound , the total uncertainty distance between lower bound and upper bound of the final estimate of the objective function = 4 * 35 = 140 units of time

6.4 Conclusion

The uncertainty of data is a challenge for scheduling algorithms, especially the problems that use the heuristic or the meta-heuristic techniques to find the optimal solution. McNaughton's algorithm is a one of the heuristic algorithms that is used to minimise the total completion time on multiple machines with the preemption constraints. The job shop

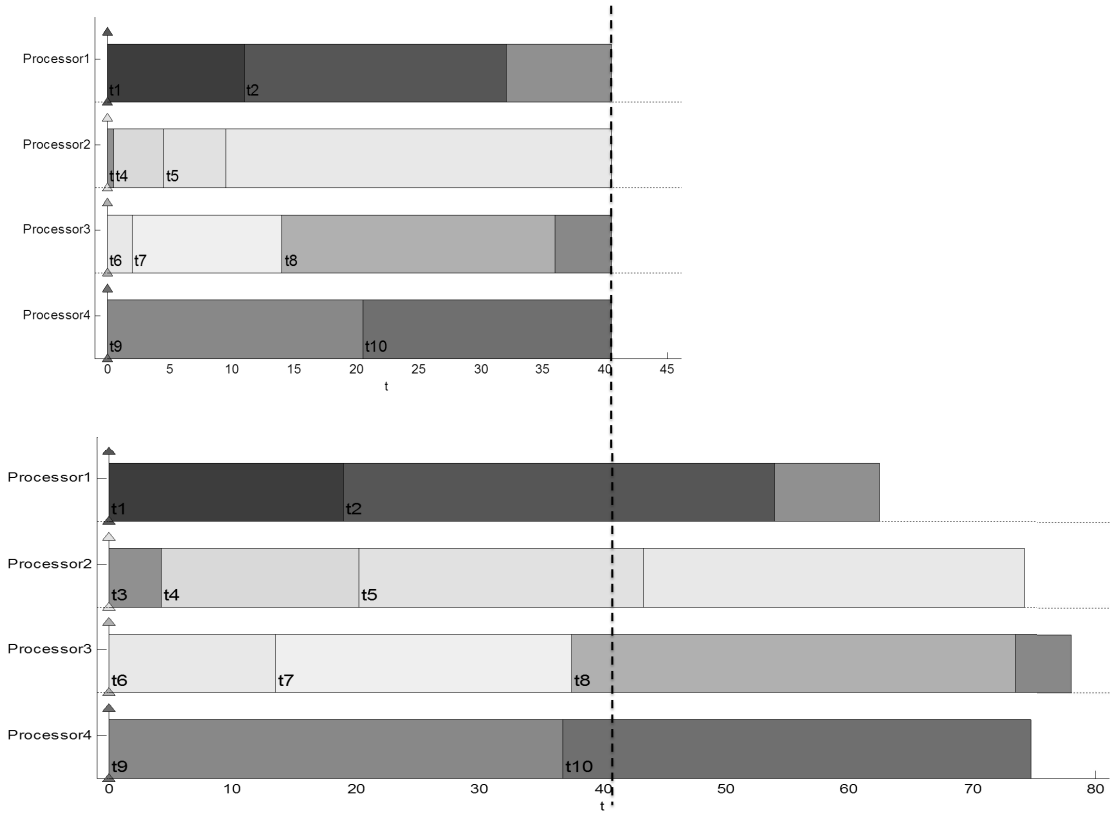

Figure 6.4: Scheduling 10 tasks based on interval programming estimates, the total uncertainty distance between lower bound and upper bound of the final estimate of the objective function = = 25 + 35 + 40 + 35 = 130 units of time

scheduling in automotive production facility is a good example of such a problem.

In this chapter, McNaughton's scheduling algorithm has been extended to apply the interval programming, in order to reduce the impact of the bounded uncertainty of the input parameters. According to the proposed methodology, all the arithmetic and logic operations in the algorithm have been replaced with interval arithmetic and interval logic.

After performing multiple experiments, the results showed that using interval programming in estimating the objective function gives more accurate and less uncertain results than using numerical calculation directly such as average or median. Interval programming is more applicable than interval perturbation for a large number of tasks as it does not have an exponential order of magnitude.

The extended McNaughton's algorithm did not achieve great marginal enhancements in terms of optimality, as C_{max} of the interval-based algorithm is equal to C_{max} of the original algorithm. Meanwhile, the extended algorithm achieved less uncertainty and better utilisation of the first processor, which outperform the original algorithm.

Chapter 7

Conclusions and Discussion

The uncertainty challenge is common and affects many domains including engineering, decision making, chemistry, physics, biology, medicine, ecology and environment. The uncertainty has multiple definitions, taxonomies, causes, implications and techniques. They all vary according to the problem domain, nature and formulation. One of the challenging categories of uncertainty that affects many domains is the bounded uncertainty. Bounded uncertainty is defined by its lower bound and upper bound and its main challenge is the lack of information between its bounds, as it does not have any descriptive function similar to probability density function or fuzzy membership function.

The bounded uncertainty of the parameters of the scheduling problem affects many domains significantly due to the uncontrollable factors. One of the domains is the scheduling of the aircraft take-off, arrival, holding patterns and landing in airports. The airports and aircraft scheduling depends on the weather, the crew, the passengers, the baggage system and the runway availability, which are uncertain factors. Such factors forced aircraft schedules to be stated as time intervals even though the official statements give a specific time.

Another domain that faces the bounded uncertainty challenge is the scheduling of the chemical plants. The main reasons for that are the micro chemical reactions that happen

during the process as well as the physical changes in the reaction environment that may affect the reaction back. Such uncertainty is considered cyclic drift uncertainty and it is difficult to measure all the micro reaction and physical changes to every molecule in the process.

The domain of renewable energy also faces the bounded uncertainty challenge, which includes the wind farms, water currents and solar panels. The source of uncertainty in such domain is the environmental variations, which are neither controllable nor predictable. This domain is a great example of uncertainty propagation as the energy generation is not certain and depends on the predictions of the weather, which is not certain as well.

Although this problem has been tackled using mixed integer linear programming and quadratic programming, the two techniques needed a mathematical formulation as a system of linear equations. MILP and quadratic programming cannot eliminate or minimise the uncertainty for heuristic algorithms or meta-heuristics techniques such as neural networks and genetic algorithms.

Two other techniques are used to resolve the parametric uncertainty challenge within the scheduling problems. The first is the numerical approximation to the average, upper or lower bounds of the interval parameter. Such approximation causes the drawback of the value drifting away of the actual value. The second technique is to calculate the objective function using different input values that belong to the interval of the input parameter. The draw back with multiple calculations is the uncertainty propagation.

Some other techniques have been suggested to solve the parametric uncertainty problem for scheduling algorithms similar to combinatorial approach and interval perturbation, but they both face the problem of NP-hard complexity, which make the solution infeasible.

The proposed research introduced a new methodology to minimise the impact of bounded uncertainty on the scheduling problem and its solution. The challenge of bounded uncertainty is the lack of information describing the uncertainty nature between the upper bound

and the lower bound of the interval parameter.

The proposed methodology extends any scheduling algorithm to be able to use the interval input parameter as variable that will be used through the execution. The execution of the algorithm uses the interval arithmetic as replacement for numerical arithmetic and uses the interval logic instead of Boolean logic. The output of the algorithm is an interval value, where the distance between the upper and lower bounds can be zero, which is the ultimate certainty as the upper and lower bounds are equal. The distance of the interval output variable represents the uncertainty of the output and can be used to measure if the uncertainty increased or decreased comparable to the uncertainty of the inputs.

The proposed methodology, which is based on interval programming, consists of five main steps and uses four techniques. The first step is to analyse and decompose the algorithm into a set of instruction and determine, which instructions are affected by the uncertain input parameters and which instructions affect the objective function result.

The second step is to define the interval operators that will be used in extending the algorithm. Those operators can be arithmetic or logical. The arithmetic operators are usually defined using Moore's interval arithmetic, but it can also be defined using other techniques similar to affine arithmetic, which is generalised interval arithmetic.

The logic operators can be defined using interval temporal algebra that is introduced by James Allen, it is also known as Allen's algebra. The main issue with Allen's algebra is the number of operators, which are seven in contrary of the Boolean operators, which are just three. The variation between the number of operators requires mapping Allen's algebra to Boolean logic. The other option is to define new logical operators that express the nature of the problem.

The third step is to extend the scheduling algorithm by replacing the instructions that have been selected in the algorithm analysis and the decomposition phase with new modified instructions. The new instructions are modified version of the old instructions after

applying the interval arithmetic instead of numerical arithmetic and after applying interval logic rather Boolean logic.

The fourth step is to verify the effectiveness of applying the new methodology on a specific algorithm, using multiple realistic or generated data sets. Such verification is measured by comparing the total uncertainty of the objective function using interval programming versus the total uncertainty after calculating the objective function using the lower bound subtracted from the objective function using the upper bound. Estimating the total uncertainty of the objective function using interval programming is calculated in terms of the distance between the upper bound and the lower bound of the end result.

Finally, the fifth step is to use the new extended algorithm to generate the new baseline schedule. That is why it is considered as protective or preventive technique for scheduling under uncertainty.

To emphasise the effectiveness of the proposed methodology, it has been applied on three algorithms solving three different scheduling problems of different objectives, environments and constraints. The first is Hodgson's algorithm that solves the problem of minimising the total number of delayed tasks. This algorithm is considered a single processor heuristic that is formulated as $1 \mid\mid \sum U_j$.

The second is Bartley's algorithm that uses branch and bound method to minimise the maximum completion time of all jobs, taking the release time and the deadlines as constrains. Such problem is a branch and bound NP-hard problem that is formulated as $1 \mid r_j\ ,\ \sim dj \mid Cmax$.

The third is McNaughton's algorithm that is used for minimising the maximum completion time of all tasks on multiple identical parallel processors, considering the pre-emption constraint. The problem is formulated as $P \mid pmtn \mid C_max$ and is calculated by finding the maximum of the two values $\max(p_j)$; and $(\sum p_j)/m$, where m is the number of processors.

The results of the three experiments using different sets of data emphasized that the

proposed methodology can minimise the impact of uncertainty on the objective function. The estimate of the objective function using interval arithmetic and logic have resulted in more certain results and less distance between upper bound and lower bound than the estimate of the objective function using multiple calculation.

The results of the experiments showed that the marginal minimisation of the uncertainty is affected by two factors. The first factor is the sum of all distances between the upper bound and the lower bound of the interval parameter. The more distance between upper and lower bounds causes more marginal minimisation of the uncertainty. The second factor is the relation between interval parameters, which can be disjoint intervals or overlapping intervals or containing/contained intervals. The more number of overlaps increases the marginal minimisation of the uncertainty as well. It is important to mention that the total distance between upper and lower bounds is highly correlated with the number of overlaps taking in consideration that correlation does not imply causality.

7.1 Advantages and Limitations of Interval Programming

The proposed methodology has many advantages. First advantage is minimising the total uncertainty as declared in the multiple experiments in chapters 6, 7 and 8. Second advantage is the applicability on all algorithms including heuristics and meta-heuristics. Third, the uncertainty does not deviate towards the upper or the lower bound due to applying the calculations on a specific number. Fourth, the uncertainty does not propagate through the function, especially through loops or exponential calculations.

Another advantage of using the proposed approach is the ability to execute it using numerical non-interval inputs. Interval programming converts the numerical values to an interval starting and ending at the same value with distance equals zero. Such conversion is called the point estimate.

The main limitation of interval programming was the containment relation between intervals, where one of the intervals contains or include the other one totally. the lower bound of the wide interval is less than the lower bound of the narrow interval and the upper bound of the wide interval is larger than upper bound of the narrow interval. The results of the estimating the objective function varies according to the used bound. After performing a set of experiments on four different algorithms using real data and generated data, The result analysis showed that the correlation between the number of containments in the dataset and the marginal enhancement of the uncertainty is 60% which is very low.

7.2 The Trade-off between Optimality and Uncertainty

The main target of solving scheduling problems is to find the optimal solution, but having an optimal solution with uncertainty implies potential risks and hidden costs. On the other hand having certain and accurate estimates for the objective function problem can result in non-optimal solution. The challenge is to find an optimal solution with certain and accurate results, especially if the input is uncertain by nature. This problem is similar to Heisenberg uncertainty principle in physics that states "the more precisely the position of some particle is determined, the less precisely its momentum can be known, and vice versa". Figure 7.1 illustrates an example of a function with global optimal solution at the point s_1 with wide uncertainty range and non-optimal solution at point s_2 with narrow solution range.

The proposed methodology minimised the impact of the uncertain variables on the output variables by tracking the behaviour of the uncertain variables through execution time. The next step is to find the optimal estimate of the objective function that guarantees the most near optimal estimate with minimum uncertainty margin.

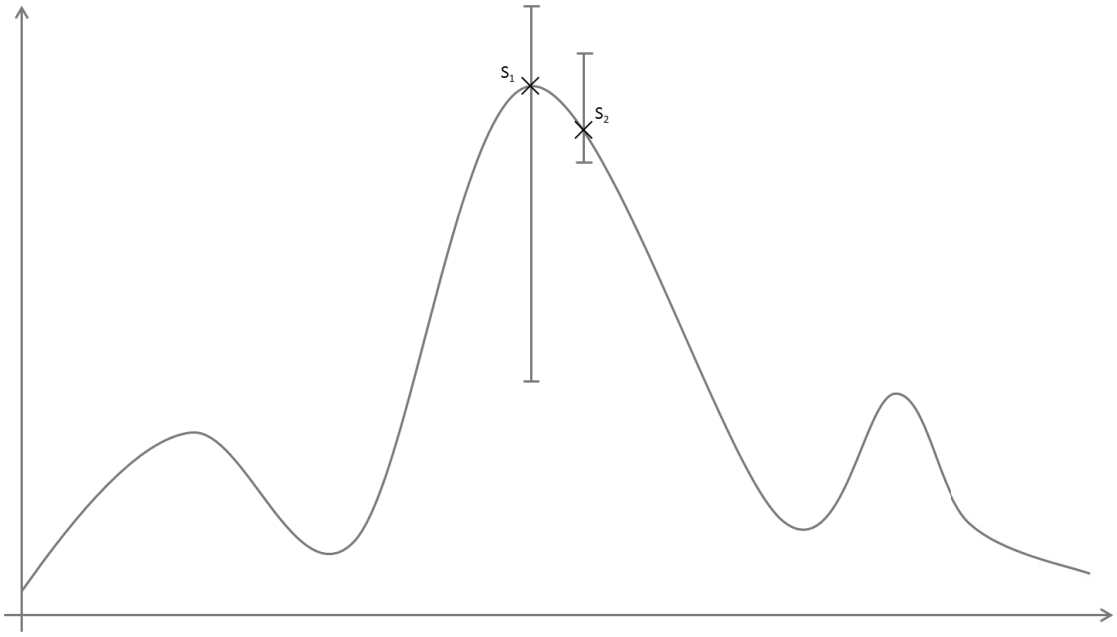

Figure 7.1: Illustration of the possible tradeoff between optimality and uncertainty

7.3 Considerations for Applying Interval Programming

The interval programming should be used when the parameters are uncertain and the uncertainty is bounded and represented as numerical intervals. The numerical intervals has just the lower and upper bound without any known information between them similar to a probability density function or the fuzzy membership function.

The interval programming technique is better to be used when the number of tasks is huge and the interval size is big and the sensitivity is high as it minimises the uncertainty of the results. Whenever the number of the tasks is relatively small, the combinatorial approach is more suitable, as it will find the optimal result and it will not face the complexity issue. The suitable number of tasks for combinatorial approach varies according to the

problem nature and its order of magnitude; it is usually less than 15 tasks for most of the problems.

In case the number of tasks is relatively high and the interval size of each task is relatively small, it will be more optimal to use the interval perturbation combined with sensitivity analysis. The benefit of interval perturbation is that it will find the near optimal solution with minimum uncertainty without facing the complexity issue resulting from the range of perturbation inside each interval. It is important to notice that the complexity is function of the perturbation step, which can be in terms of integers, or fractions.

7.4 Conclusions

This research studied multiple definitions, taxonomies and understandings of uncertainty and how it affects decision making as well as optimisation techniques. It also discussed the proposed approaches to minimise the negative impact of bounded uncertainty on scheduling process and what are the limitation of each technique. The proposed research introduced a novel methodology to minimise the impact of uncertainty on scheduling algorithm and emphasised its usefulness and applicability on three different algorithms of different natures in terms of problem formulation, solution technique, number of processors and constraints.

This research concludes that interval programming is useful to minimise bounded uncertainty for scheduling algorithms, especially for the heuristic-based and meta-heuristic-based solution. The solution achieves less uncertainty than other uncertainty handling approaches, within feasible processing time and acceptable accuracy, especially when the number of tasks is large and the interval distance of each task is wide.

7.5 Future Work

The proposed methodology can be extended to consider the urgency function, the weights, the confidence and the prediction intervals. In case the urgency function of the problem is known, the schedule can be weighted by calculating the area under the curve using numerical integration. In case urgency curve increases by time, the first interval gets higher priority in sorting, which means it will be considered as the smaller number even if the interval started after. On the other hand if the urgency curve decays of time, the latest interval gets higher priority. The priority can be estimated by comparing the slope for linear functions. For the non-linear curves, the priority can be estimated using numerical integration between the lower bound of the interval and the upper bound.

The numerical integration can be estimated using many methods, which vary in the performance and the accuracy. One method is the Implicit Midpoint Rule that is also known as rectangle rule. Such rule takes the average of x_i and $x_{(i+1)}$ to create a sequence of rectangles, each rectangle intersects the curve in the middle of the top edge [257]. The midpoint rule method assumes the interpolating function to be a polynomial of degree zero, which passes through the point $(x_i + x_{i+1})/2$ as stated in the equation 7.1 and declared in figure 7.2.

$$\int_{d^-}^{d^+} f(x)dx \approx (d^+ - d^-)f((d^- + d^+)/2) \tag{7.1}$$

Another method for numerical integration is the trapezoidal rule method, where the interpolating function is polynomial of the first degree. The trapezoidal method intersects the function curve in the starting point, middle point and the end point as declared in figure 7.3. The interval of the integration can be divided to a sequence of subintervals to provide more accurate estimate. The smaller the size of subintervals, the larger their count is. The standard trapezoidal rule method is represented in the equation 7.2 and the composite trapezoidal rule is described in the equation 7.3, which is also known as the extended rule

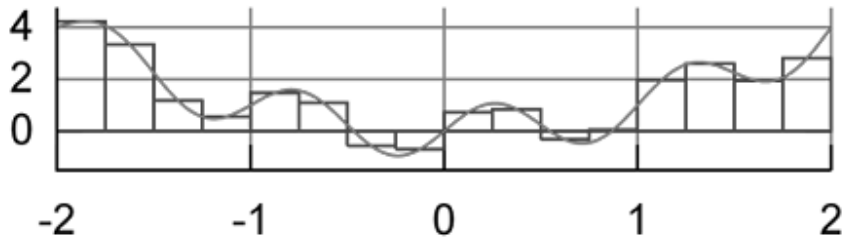

Figure 7.2: Numerical integration using midpoint rule

or the iterated rule.

$$\int_{d^-}^{d^+} f(x)dx \approx (d^+ - d^-)(f(d^-) + f(d^+))/2 \tag{7.2}$$

$$\int_{d^-}^{d^+} f(x)dx \approx \frac{(d^+ - d^-)}{2}\left(\frac{f(d^-)}{2} + \sum_{k=1}^{n-1} f\left(a + k\frac{(d^+ - d^-)}{n}\right) + \frac{(f(d)^+)}{2}\right) \tag{7.3}$$

where the subintervals have the form $[kh, (k+1)h]$, with $h = (ba)/n$ and $k = 0, 1, 2, \ldots, n1$.

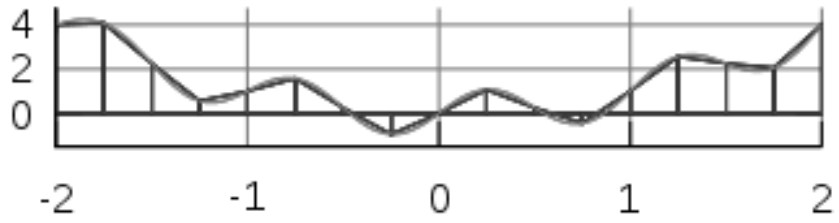

Figure 7.3: Numerical integration using trapezoidal rule

According to the stated equations numerical integration, the inequality relations can be defined as follows:

Definition 7.5.1

For any two real intervals $I_1 = [I_1^-, I_1^+]$ *and* $I_2 = [I_2^-, I_2^+]$ I_1 *is less than* I_2 *if* $\int_{I_1^-}^{I_1^+} f(x)dx < \int_{I_2^-}^{I_2^+} f(x)dx$ *and* $I_1^+ < I_2^+$ *where*$I_1^-, I_1^+, I_2^-, I_2^+ R$

Definition 7.5.2

For any two real intervals $I_1 = [I_1^-, I_1^+]$ *and* $I_2 = [I_2^-, I_2^+]$ $I_1 = I_2$ *if and only if* $I_1^- = I_2^-$ *and* $I_1^+ = I_2^+$ *where* $I_1^-, I_1^+, I_2^-, I_2^+ R$

Definition 7.5.3

Otherwise,I_1 *is larger than* I_2 *(11)*

The following set of figures indicates different combinations of the relations between two intervals with different urgency curves. Such urgency functions affect the estimate of the integration of the interval, which affect the priority of the intervals in the inequality situation.

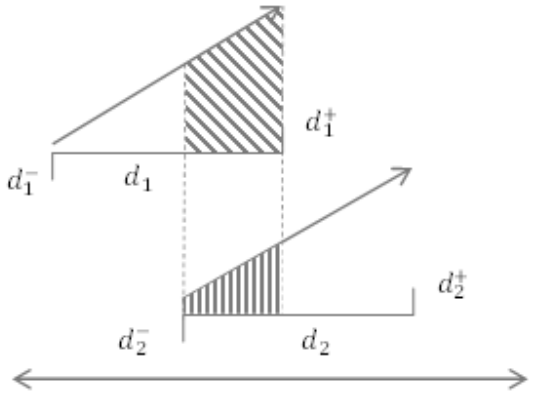

Figure 7.4: a plot for two intervals representing two uncertain deadlines for two jobs. Each job has its own urgency curve that varies with time. Deadline d1 has great urgency factor by its end where deadline d2 has small urgency factor by its beginning. Using the numerical integration of the areas under the curves imply that d1 $>$ d2

As long as the algorithm is under execution, the uncertainty increases or decreases by processing according to the arithmetic and logic operations, which are implemented using a specific instruction at a specific slot of time. Such operation is called uncertainty convergence and uncertainty divergence. One of the possible extensions to keep the uncertainty at the same level or converging and to stop any possible divergence is to put functional observers that monitor every change in the runtime variables and try to keep them with minimum uncertainty. Another extension for the proposed methodology is to integrate it with sensitivity analysis, in order to final, which changes within each interval can lead to more optimal solution with the same uncertainty level. The same concept of applying program slicing, defining interval operators and extending the algorithms can be applied to

probabilistic-based and fuzzy-based uncertainty. Probabilistic approach will need to consider the density function, mean and variance and consider how they will affect the uncertainty of the objective function. The Fuzzy approach will need to consider the membership function and the fuzzy rules and how they can affect the uncertainty and the optimality of the objective function. The fuzzy approach will be useful for meta-heuristic solutions.

References

[1] J. R. Montoya-Torres, E. Gutierrez-Franco, and C. Pirachicn-Mayorga, "Project scheduling with limited resources using a genetic algorithm," *International Journal of Project Management*, vol. 28, no. 6, pp. 619 – 628, 2010.

[2] D. Golenko-Ginzburg and A. Gonik, "Stochastic network project scheduling with non-consumable limited resources," *International Journal of Production Economics*, vol. 48, no. 1, pp. 29 – 37, 1997.

[3] G. Ulusoy and L. zdamar, "A framework for an interactive project scheduling system under limited resources," *European Journal of Operational Research*, vol. 90, no. 2, pp. 362 – 375, 1996.

[4] S. Perez-Canto and J. C. Rubio-Romero, "A model for the preventive maintenance scheduling of power plants including wind farms," *Reliability Engineering and System Safety*, vol. 119, no. 0, pp. 67 – 75, 2013.

[5] G. M. Kopanos and E. N. Pistikopoulos, "Reactive Scheduling of micro Combined Heat and Power Systems via Multiparametric Programming," in *23rd European Symposium on Computer Aided Process Engineering*, A. Kraslawski and I. Turunen, Eds., vol. 32 of *Computer Aided Chemical Engineering*, pp. 277 – 282. Elsevier, 2013.

[6] R. Azizipanah-Abarghooee, T. Niknam, F. Bavafa, and M. Zare, "Short-term

scheduling of thermal power systems using hybrid gradient based modified teachinglearning optimiser with black hole algorithm," *Electric Power Systems Research*, vol. 108, no. 0, pp. 16 – 34, 2014.

[7] N. Meskens, D. Duvivier, and A. Hanset, "Multi-objective operating room scheduling considering desiderata of the surgical team," *Decision Support Systems*, vol. 55, no. 2, pp. 650 – 659, 2013.

[8] A. Jebali, A. B. H. Alouane, and P. Ladet, "Operating rooms scheduling," *International Journal of Production Economics*, vol. 99, no. 12, pp. 52 – 62, 2006.

[9] B. Cardoen, E. Demeulemeester, and J. Belin, "Operating room planning & scheduling: A literature review," *European Journal of Operational Research*, vol. 201, no. 3, pp. 921 – 932, 2010.

[10] A. A. Soukour, L. Devendeville, C. Lucet, and A. Moukrim, "A Memetic Algorithm for staff scheduling problem in airport security service," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7504 – 7512, 2013.

[11] Q. Liu, T. Wu, and X. Luo, "A space-time network model based on improved genetic algorithm for airport taxiing scheduling problems," *Procedia Engineering*, vol. 15, no. 0, pp. 1082 – 1087, 2011.

[12] M. Sam, A. DAriano, and D. Pacciarelli, "Rolling Horizon Approach for Aircraft Scheduling in the Terminal Control Area of Busy Airports," *Procedia - Social and Behavioral Sciences*, vol. 80, no. 0, pp. 531 – 552, 2013, 20th International Symposium on Transportation and Traffic Theory (ISTTT 2013).

[13] J. V. den Bergh, J. Belin, P. D. Bruecker, E. Demeulemeester, and L. D. Boeck, "Personnel scheduling: A literature review," *European Journal of Operational Research*, vol. 226, no. 3, pp. 367 – 385, 2013.

[14] E. B. Edis, C. Oguz, and I. Ozkarahan, "Parallel machine scheduling with additional resources: Notation, classification, models and solution methods," *European*

Journal of Operational Research, vol. 230, no. 3, pp. 449 – 463, 2013.

[15] J. M. Framinan and R. Ruiz, "Architecture of manufacturing scheduling systems: Literature review and an integrated proposal," *European Journal of Operational Research*, vol. 205, no. 2, pp. 237 – 246, 2010.

[16] P. Perez-Gonzalez and J. M. Framinan, "A common framework and taxonomy for multicriteria scheduling problems with interfering and competing jobs: Multi-agent scheduling problems," *European Journal of Operational Research*, vol. 235, no. 1, pp. 1 – 16, 2014.

[17] Z. Li and M. Ierapetritou, "Process scheduling under uncertainty: Review and challenges," *Computers & Chemical Engineering*, vol. 32, no. 45, pp. 715 – 727, 2008.

[18] C. N. Potts and M. Y. Kovalyov, "Scheduling with batching: A review," *European Journal of Operational Research*, vol. 120, no. 2, pp. 228 – 249, 2000.

[19] M. Bassett, P. Dave, F. D. III, G. Kudva, J. Pekny, G. Reklaitis, S. Subrahmanyam, D. Miller, and M. Zentner, "Perspectives on model based integration of process operations," *Computers & Chemical Engineering*, vol. 20, no. 67, pp. 821 – 844, 1996.

[20] E. Kondili, C. Pantelides, and R. Sargent, "A general algorithm for short-term scheduling of batch operations. {MILP} formulation," *Computers & Chemical Engineering*, vol. 17, no. 2, pp. 211 – 227, 1993.

[21] J. Hu and Z. Qiu, "Non-probabilistic convex models and interval analysis method for dynamic response of a beam with bounded uncertainty," *Applied Mathematical Modelling*, vol. 34, no. 3, pp. 725 – 734, 2010.

[22] C. Ramos, M. Martnez, J. Sanchis, and J. Herrero, "Robust and stable predictive control with bounded uncertainties," *Journal of Mathematical Analysis and Applications*, vol. 342, no. 2, pp. 1003 – 1014, 2008.

[23] W. Shuning, D. Jianshe, and H. Ping, "An extension of estimation theory on unknown but bounded uncertainty," *Control Engineering Practice*, vol. 3, no. 1, pp. 132 –, 1995.

[24] I. Elishakoff, R. Haftka, and J. Fang, "Structural design under bounded uncertaintyoptimisation with anti-optimisation," *Computers & Structures*, vol. 53, no. 6, pp. 1401 – 1405, 1994.

[25] K.-K. K. Kim and R. D. Braatz, "Computational complexity and related topics of robustness margin calculation using Mu theory: A review of theoretical developments," *Computers & Chemical Engineering*, 2013.

[26] Y. Liu, S. Lee, O. Kwon, and J. H. Park, "Delay-dependent exponential stability criteria for neutral systems with interval time-varying delays and nonlinear perturbations," *Journal of the Franklin Institute*, vol. 350, no. 10, pp. 3313 – 3327, 2013.

[27] M. Hifi and H. Mhalla, "Sensitivity analysis to perturbations of the weight of a subset of items: The knapsack case study," *Discrete optimisation*, vol. 10, no. 4, pp. 320 – 330, 2013.

[28] V. D. Blondel and J. N. Tsitsiklis, "A survey of computational complexity results in systems and control," *Automatica*, vol. 36, no. 9, pp. 1249 – 1274, 2000.

[29] P. Sucha, M. Kutil, M. Sojka, and Z. Hanzalek, "TORSCHE Scheduling toolbox for Matlab," in *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, 2006, pp. 1181–1186.

[30] J. Allen, "Temporal reasoning in plan management," *Fifth International Workshop on Temporal Representation and Reasoning - Proceedings*, pp. 2–2, 1998.

[31] A. Krokhin, P. Jeavons, and P. Jonsson, "Reasoning about temporal relations: The tractable subalgebras of Allen's interval algebra," *Journal of the ACM*, vol. 50, no. 5, pp. 591–640, 2003.

[32] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, 1983.

[33] T. Sunaga, "Theory of an interval algebra and its application to numerical analysis," *Japan Journal of Industrial and Applied Mathematics*, vol. 26, no. 2, pp. 125–143, 2009.

[34] D. N. Mavris and D. A. DeLaurentis, "A probabilistic approach for examining aircraft concept feasibility and viability," *Aircraft Design*, vol. 3, no. 2, pp. 79 – 101, 2000.

[35] S. Koh and S. Saad, "Design and implementation of ERP-controlled manufacturing in simulation," *Proceedings of the 2nd International Conference on Responsive Manufacturing, Gaziantep, Turkey*, pp. 345–350, 2002.

[36] J. Mula, R. Poler, J. Garca-Sabater, and F. Lario, "Models for production planning under uncertainty: A review," *International Journal of Production Economics*, vol. 103, no. 1, pp. 271 – 285, 2006.

[37] I. Kant, "Prolegomena (1783)," *Philosophy of Material Nature. Hackett Publishing Co., Indianapolis*, 1985.

[38] C. Tannert, H.-D. Elvers, and B. Jandrig, "The ethics of uncertainty. In the light of possible dangers, research becomes a moral duty," *EMBO reports*, vol. 8, no. 10, pp. 892, 2007.

[39] J. P. Van der Sluijs, "Uncertainty and complexity: the need for new ways of interfacing climate science and climate policy," *From climate change to social change. Perspectives on science–policy interactions. International Books Utrecht, Utrecht*, 2010.

[40] M. Brugnach, A. Dewulf, C. Pahl-Wostl, and T. Taillieu, "Toward a relational concept of uncertainty: about knowing too little, knowing too differently, and accepting not to know," *Ecology and Society*, vol. 13, no. 2, pp. 30, 2008.

[41] N. Isendahl, A. Dewulf, M. Brugnach, G. François, S. Möllenkamp, and C. Pahl-Wostl, "Assessing framing of uncertainties in water management practice," *Water resources management*, vol. 23, no. 15, pp. 3191–3205, 2009.

[42] P. Van der Keur, H. Henriksen, J. Refsgaard, M. Brugnach, C. Pahl-Wostl, A. Dewulf, and H. Buiteveld, "Identification of major sources of uncertainty in current IWRM practice. Illustrated for the Rhine basin," *Water Resources Management*, vol. 22, no. 11, pp. 1677–1708, 2008.

[43] W. E. Walker, P. Harremoës, J. Rotmans, J. P. van der Sluijs, M. B. van Asselt, P. Janssen, and M. Krayer von Krauss, "Defining uncertainty: a conceptual basis for uncertainty management in model-based decision support," *Integrated assessment*, vol. 4, no. 1, pp. 5–17, 2003.

[44] M. J. Arentsen, H. T. A. Bressers, and L. J. O'Toole, "Institutional and policy responses to uncertainty in environmental policy: A comparison of Dutch and US styles," *Policy Studies Journal*, vol. 28, no. 3, pp. 597–611, 2000.

[45] A. Dewulf, M. Craps, R. Bouwen, T. Taillieu, and C. Pahl-Wostl, "Integrated management of natural resources: dealing with ambiguous issues, multiple actors and diverging frames," *Water science and technology*, vol. 52, no. 6, pp. 115–124, 2005.

[46] J. Newig, C. Pahl-Wostl, and K. Sigel, "The role of public participation in managing uncertainty in the implementation of the Water Framework Directive," *European Environment*, vol. 15, no. 6, pp. 333–343, 2005.

[47] G. Raadgever, E. Mostert, N. Kranz, E. Interwies, and J. G. Timmerman, "Assessing management regimes in transboundary river basins: do they support adaptive management," *Ecology and Society*, vol. 13, no. 1, pp. 14, 2008.

[48] X. Morin and W. Thuiller, "Comparing niche-and process-based models to reduce prediction uncertainty in species range shifts under climate change," *Ecology*, vol. 90, no. 5, pp. 1301–1313, 2009.

[49] B. T. McClintock, J. D. Nichols, L. L. Bailey, D. I. MacKenzie, W. Kendall, A. B. Franklin, et al., "Seeking a second opinion: uncertainty in disease ecology," *Ecology letters*, vol. 13, no. 6, pp. 659–674, 2010.

[50] V. Polo, P. López, and J. Martín, "Uncertainty about future predation risk modulates monitoring behavior from refuges in lizards," *Behavioral Ecology*, vol. 22, no. 1, pp. 218–223, 2011.

[51] T. Nilsen and T. Aven, "Models and model uncertainty in the context of risk analysis," *Reliability Engineering & System Safety*, vol. 79, no. 3, pp. 309–317, 2003.

[52] A. Mosleh, N. Siu, C. Smidts, and C. Lui, "Model uncertainty: its characterization and quantification," in *Proceedings of Workshop*, 1994, vol. 1.

[53] R. I. Mehr, E. Cammack, and T. Rose, *Principles of insurance*, vol. 8, RD Irwin, 1980.

[54] J. Magee, "General Insurance, Richard D. Irwin," *Inc, Homewood, Illinois*, 1961.

[55] S. Samson, J. A. Reneke, and M. M. Wiecek, "A review of different perspectives on uncertainty and risk and an alternative modeling paradigm," *Reliability Engineering & System Safety*, vol. 94, no. 2, pp. 558–567, 2009.

[56] F. Pappenberger, K. Beven, N. Hunter, P. Bates, B. Gouweleeuw, J. Thielen, and A. d. Roo, "Cascading model uncertainty from medium range weather forecasts (10 days) through a rainfall-runoff model to flood inundation predictions within the European Flood Forecasting System (EFFS)," *Hydrology and Earth System Sciences*, vol. 9, no. 4, pp. 381–393, 2005.

[57] T. N. Palmer, "Predicting uncertainty in forecasts of weather and climate," *Reports on Progress in Physics*, vol. 63, no. 2, pp. 71, 2000.

[58] Y. He, F. Wetterhall, H. Cloke, F. Pappenberger, M. Wilson, J. Freer, and G. McGregor, "Tracking the uncertainty in flood alerts driven by grand ensemble weather predictions," *Meteorological Applications*, vol. 16, no. 1, pp. 91–101, 2009.

[59] D. Draper, "Assessment and propagation of model uncertainty," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 45–97, 1995.

[60] X. Gu and J. E. Renaud, "Implicit uncertainty propagation for robust collaborative optimisation," in *Proceedings of DETC01, ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, CD-ROM Proceedings*. Citeseer, 2001.

[61] P. Baraldi and E. Zio, "A combined Monte Carlo and possibilistic approach to uncertainty propagation in event tree analysis," *Risk Analysis*, vol. 28, no. 5, pp. 1309–1326, 2008.

[62] R. Jin, W. Chen, and A. Sudjianto, "Analytical metamodel-based global sensitivity analysis and uncertainty propagation for robust design," *SAE SP*, pp. 47–54, 2004.

[63] X. Yin, S. Lee, W. Chen, W. K. Liu, and M. Horstemeyer, "Efficient random field uncertainty propagation in design using multiscale analysis," *Journal of Mechanical Design*, vol. 131, pp. 021006, 2009.

[64] W. L. Oberkampf, J. C. Helton, C. A. Joslyn, S. F. Wojtkiewicz, and S. Ferson, "Challenge problems: uncertainty in system response given uncertain parameters," *Reliability Engineering & System Safety*, vol. 85, no. 1, pp. 11–19, 2004.

[65] L. Criscenti, G. Laniak, and R. Erikson, "Propagation of uncertainty through geochemical code calculations," *Geochimica et cosmochimica acta*, vol. 60, no. 19, pp. 3551–3568, 1996.

[66] V. J. Romero, "Characterization, costing, and selection of uncertainty propagation methods for use with large computational physics models," in *submitted for the Non-Deterministic Approaches Forum of the 42nd Structures, Structural Dynamics, and Materials (SDM) Conference in Seattle, WA*, 2001.

[67] K. Durga Rao, H. Kushwaha, A. K. Verma, and A. Srividya, "Quantification of epistemic and aleatory uncertainties in level-1 probabilistic safety assessment studies,"

Reliability Engineering & System Safety, vol. 92, no. 7, pp. 947–956, 2007.

[68] K. Farquar and A. Mosleh, "An approach to quantifying reliability-growth effectiveness," in *Reliability and Maintainability Symposium, 1995. Proceedings., Annual*. IEEE, 1995, pp. 166–173.

[69] J. Ascough, H. R. Maier, J. K. Ravalico, and M. Strudley, "Future research challenges for incorporation of uncertainty in environmental and ecological decision-making," *Ecological Modelling*, vol. 219, no. 3, pp. 383–399, 2008.

[70] J. Rotmans and M. B. van Asselt, "Uncertainty management in integrated assessment modeling: towards a pluralistic approach," *Environmental monitoring and assessment*, vol. 69, no. 2, pp. 101–130, 2001.

[71] A. J. Jakeman and R. A. Letcher, "Integrated assessment and modelling: features, principles and examples for catchment management," *Environmental Modelling & Software*, vol. 18, no. 6, pp. 491–501, 2003.

[72] L. Börjeson, M. Höjer, K.-H. Dreborg, T. Ekvall, and G. Finnveden, "Scenario types and techniques: towards a user's guide," *Futures*, vol. 38, no. 7, pp. 723–739, 2006.

[73] S. O. Funtowicz and J. R. Ravetz, "Science for the post-normal age," *Futures*, vol. 25, no. 7, pp. 739–755, 1993.

[74] J. A. Wardekker, J. P. van der Sluijs, P. H. Janssen, P. Kloprogge, and A. C. Petersen, "Uncertainty communication in environmental assessments: views from the Dutch science-policy interface," *Environmental science & policy*, vol. 11, no. 7, pp. 627–641, 2008.

[75] B. Wynne, "Uncertainty and environmental learning: reconceiving science and policy in the preventive paradigm," *Global environmental change*, vol. 2, no. 2, pp. 111–127, 1992.

[76] R. Bouwen, A. Dewulf, and M. Craps, "Participatory development of technology innovation projects: collaborative learning among different communities of practice,"

Anales de la Universidad de Cuenca, vol. 49, pp. 127–142, 2006.

[77] L. Hanssen, E. Rouwette, and M. M. Van Katwijk, "The role of ecological science in environmental policy making: from a pacification toward a facilitation strategy," *Ecology and Society*, vol. 14, no. 1, pp. 43, 2009.

[78] J. Koppenjan and E.-H. Klijn, *Managing uncertainties in networks: Public private controversies*, Routledge, 2004.

[79] E. M. Van Bueren, E.-H. Klijn, and J. F. Koppenjan, "Dealing with wicked problems in networks: Analyzing an environmental debate from a network perspective," *Journal of Public Administration Research and Theory*, vol. 13, no. 2, pp. 193–212, 2003.

[80] C. Leeuwis, "Reconceptualizing participation for sustainable rural development: towards a negotiation approach," *Development and change*, vol. 31, no. 5, pp. 931–959, 2000.

[81] B. Gray, "Freeze framing: The timeless dialogue of intractability surrounding Voyageurs National Park," *Making sense of intractable environmental conflicts: concepts and cases. Island Press, Washington, DC, USA*, pp. 91–125, 2003.

[82] A. Klinke and O. Renn, "A New Approach to Risk Evaluation and Management: Risk-Based, Precaution-Based, and Discourse-Based Strategies," *Risk Analysis*, vol. 22, no. 6, pp. 1071–1094, 2002.

[83] G. Raadgever, C. Dieperink, P. Driessen, A. Smit, and H. Van Rijswick, "Uncertainty management strategies: Lessons from the regional implementation of the Water Framework Directive in the Netherlands," *Environmental Science & Policy*, vol. 14, no. 1, pp. 64–75, 2011.

[84] J. Dantan, N. Gayton, A. Qureshi, M. Lemaire, and A. Etienne, "Tolerance Analysis Approach based on the Classification of Uncertainty (Aleatory/Epistemic)," *Procedia CIRP*, vol. 10, no. 0, pp. 287 – 293, 2013.

[85] F. O. Hoffman and J. S. Hammonds, "Propagation of uncertainty in risk assessments: the need to distinguish between uncertainty due to lack of knowledge and uncertainty due to variability," *Risk Analysis*, vol. 14, no. 5, pp. 707–712, 1994.

[86] J. C. Helton, "Treatment of uncertainty in performance assessments for complex systems," *Risk analysis*, vol. 14, no. 4, pp. 483–511, 1994.

[87] W. D. Rowe, "Understanding uncertainty," *Risk analysis*, vol. 14, no. 5, pp. 743–750, 1994.

[88] S. C. Hora, "Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management," *Reliability Engineering & System Safety*, vol. 54, no. 2, pp. 217–223, 1996.

[89] S. Ferson and L. R. Ginzburg, "Different methods are needed to propagate ignorance and variability," *Reliability Engineering & System Safety*, vol. 54, no. 2, pp. 133–144, 1996.

[90] H. C. Frey and D. S. Rhodes, "Characterizing, simulating, and analyzing variability and uncertainty: an illustration of methods using an air toxics emissions example," *Human and Ecological Risk Assessment*, vol. 2, no. 4, pp. 762–797, 1996.

[91] S. Rai, D. Krewski, and S. Bartlett, "A general framework for the analysis of uncertainty and variability in risk assessment," *Human and ecological risk assessment*, vol. 2, no. 4, pp. 972–989, 1996.

[92] G. W. Parry, "The characterization of uncertainty in probabilistic risk assessments of complex systems," *Reliability Engineering & System Safety*, vol. 54, no. 2, pp. 119–126, 1996.

[93] M. E. Paté-Cornell, "Uncertainties in risk analysis: Six levels of treatment," *Reliability Engineering & System Safety*, vol. 54, no. 2, pp. 95–111, 1996.

[94] A. C. Cullen and H. C. Frey, *Probabilistic techniques in exposure assessment: a handbook for dealing with variability and uncertainty in models and inputs*,

Springer, 1999.

[95] W. L. Oberkampf, K. V. Diegert, K. F. Alvin, and B. M. Rutherford, "Variability, uncertainty, and error in computational simulation," *American Society of Mechanical Engineers-Publications-Heat Transfer Division*, vol. 357, pp. 259–272, 1998.

[96] D. Hamby, "A review of techniques for parameter sensitivity analysis of environmental models," *Environmental Monitoring and Assessment*, vol. 32, no. 2, pp. 135–154, 1994.

[97] P. Carvalho, L. Ferreira, F. Lobo, and L. Barruncho, "Optimal distribution network expansion planning under uncertainty by evolutionary decision convergence," *International Journal of Electrical Power & Energy Systems*, vol. 20, no. 2, pp. 125–129, 1998.

[98] R. Le Roy, "Uncertainty, sensitivity, convergence, and rounding in performing and reporting least-squares fits," *Journal of molecular spectroscopy*, vol. 191, no. 2, pp. 223–231, 1998.

[99] E. M. Miller, "Risk, uncertainty, and divergence of opinion," *The Journal of Finance*, vol. 32, no. 4, pp. 1151–1168, 1977.

[100] J. L. Thorne and H. Kishino, "Divergence time and evolutionary rate estimation with multilocus data," *Systematic Biology*, vol. 51, no. 5, pp. 689–702, 2002.

[101] J. Wu and J. Cao, "Linear and nonlinear response functions of the Morse oscillator: Classical divergence and the uncertainty principle," *The Journal of Chemical Physics*, vol. 115, pp. 5381, 2001.

[102] M. E. Steiper, N. M. Young, and T. Y. Sukarna, "Genomic data support the hominoid slowdown and an Early Oligocene estimate for the hominoid–cercopithecoid divergence," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 49, pp. 17021–17026, 2004.

[103] A. Van Griensven, T. Meixner, S. Grunwald, T. Bishop, M. Diluzio, and R. Srinivasan, "A global sensitivity analysis tool for the parameters of multi-variable catchment models," *Journal of Hydrology*, vol. 324, no. 1, pp. 10–23, 2006.

[104] Y. Xia, "Learning about predictability: The effects of parameter uncertainty on dynamic asset allocation," *The Journal of Finance*, vol. 56, no. 1, pp. 205–246, 2001.

[105] J. C. Helton, J. D. Johnson, C. J. Sallaberry, and C. B. Storlie, "Survey of sampling-based methods for uncertainty and sensitivity analysis," *Reliability Engineering & System Safety*, vol. 91, no. 10, pp. 1175–1209, 2006.

[106] S. Marino, I. B. Hogue, C. J. Ray, and D. E. Kirschner, "A methodology for performing global uncertainty and sensitivity analysis in systems biology," *Journal of theoretical biology*, vol. 254, no. 1, pp. 178–196, 2008.

[107] A. Saltelli, K. Chan, and E. M. Scott, *Sensitivity analysis*, vol. 134, Wiley New York, 2000.

[108] J. C. Helton and F. Davis, "Sampling-based methods for uncertainty and sensitivity analysis," *Multimedia Environmental Models*, vol. 32, no. 2, pp. 135–154, 2006.

[109] J. E. Oakley and A. O'Hagan, "Probabilistic sensitivity analysis of complex models: a Bayesian approach," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 66, no. 3, pp. 751–769, 2004.

[110] J. E. Helton, *Simplified estimating for builders and engineers*, Prentice-Hall, 1985.

[111] W. W. Helton, *Around Home in Unicoi County*, WW Helton, 1986.

[112] J. C. Refsgaard, J. P. Van der Sluijs, J. Brown, and P. Van der Keur, "A framework for dealing with uncertainty due to model structure error," *Advances in Water Resources*, vol. 29, no. 11, pp. 1586–1597, 2006.

[113] M. B. Beck, "Environmental foresight and structural change," *Environmental Modelling & Software*, vol. 20, no. 6, pp. 651–670, 2005.

[114] I. G. Dubus, C. D. Brown, and S. Beulke, "Sources of uncertainty in pesticide fate modelling," *Science of the Total Environment*, vol. 317, no. 1, pp. 53–72, 2003.

[115] E. Usunoff, J. Carrera, and S. Mousavi, "An approach to the design of experiments for discriminating among alternative conceptual models," *Advances in Water Resources*, vol. 15, no. 3, pp. 199–214, 1992.

[116] I. Linkov and D. Burmistrov, "Model Uncertainty and Choices Made by Modelers: Lessons Learned from the International Atomic Energy Agency Model Intercomparisons," *Risk Analysis*, vol. 23, no. 6, pp. 1297–1308, 2003.

[117] S. Neuman and P. Wierenga, "A comprehensive strategy of hydrogeologic modeling and uncertainty analysis for nuclear facilities and sites. University of Arizona," Tech. Rep., Report NUREG/CR-6805, 2003.

[118] V. Klemeš, "Operational testing of hydrological simulation models," *Hydrological Sciences Journal*, vol. 31, no. 1, pp. 13–24, 1986.

[119] J. C. Refsgaard, "Towards a formal approach to calibration and validation of models using spatial data," *Spatial Patterns in Catchment Hydrology: Observations and Modelling*, pp. 329–354, 2001.

[120] A. Van Griensven and T. Meixner, "Dealing with unidentifiable sources of uncertainty within environmental models," in *Proceedings of the iEMSs International Congress*, 2004.

[121] M. Radwan, P. Willems, J. Berlamont, et al., "Sensitivity and uncertainty analysis for river quality modelling," *Journal of Hydroinformatics*, vol. 6, pp. 83–99, 2004.

[122] J. A. Vrugt, C. G. Diks, W. Bouten, J. M. Verstraten, B. Webb, N. Arnell, C. Onof, N. MacIntyre, R. Gurney, C. Kirby, et al., "Improved treatment of uncertainty in hydrological modelling.," in *Hydrology: science and practice for the 21st century. Proceedings of the British Hydrological Society International Conference, Imperial College, London, July 2004.* British Hydrological Society, 2004, pp. 389–397.

[123] M. Beck, "Uncertainty, System Identification, and the Prediction of Water Quality," in *Uncertainty and Forecasting of Water Quality*, M. Beck and G. Straten, Eds., pp. 3–68. Springer Berlin Heidelberg, 1983.

[124] M. B. Butts, J. T. Payne, M. Kristensen, and H. Madsen, "An evaluation of the impact of model structure on hydrological modelling uncertainty for streamflow simulation," *Journal of Hydrology*, vol. 298, no. 1, pp. 242–266, 2004.

[125] W. G. Harrar, T. O. Sonnenborg, and H. J. Henriksen, "Capture zone, travel time, and solute-transport predictions using inverse modeling and different geological models," *Hydrogeology Journal*, vol. 11, no. 5, pp. 536–548, 2003.

[126] L. Troldborg, *The influence of conceptual geological models on the simulation of flow and transport in Quaternary aquifer systems*, Environment & Resources, Technical University of Denmark, 2004.

[127] J.-O. Selroos, D. D. Walker, A. Ström, B. Gylling, and S. Follin, "Comparison of alternative modelling approaches for groundwater flow in fractured rock," *Journal of Hydrology*, vol. 257, no. 1, pp. 174–188, 2002.

[128] H. Visser, R. Folkert, J. Hoekstra, and J. De Wolff, "Identifying key sources of uncertainty in climate change projections," *Climatic Change*, vol. 45, no. 3-4, pp. 421–457, 2000.

[129] G. T. Van Straten and K. J. Keesman, "Uncertainty propagation and speculation in projective forecasts of environmental change: A lake-eutrophication example," *Journal of Forecasting*, vol. 10, no. 1-2, pp. 163–190, 1991.

[130] A. Hjberg and J. Refsgaard, "Model uncertainty parameter uncertainty versus conceptual models," *Water Science & Technology*, vol. 52, no. 6, pp. 177–186, 2005.

[131] P. D. Meyer, M. Ye, S. P. Neuman, and K. J. Cantrell, "Combined estimation of hydrogeologic conceptual model and parameter uncertainty," Tech. Rep., Pacific Northwest National Laboratory (PNNL), Richland, WA (US), 2004.

[132] J. A. Vennix, "Group model-building: tackling messy problems," *System Dynamics Review*, vol. 15, no. 4, pp. 379–401, 1999.

[133] A. M. Hodgson, "Hexagons for systems thinking," *European Journal of Operational Research*, vol. 59, no. 1, pp. 220–230, 1992.

[134] S. O. Funtowicz and J. R. Ravetz, *Uncertainty and quality in science for policy*, vol. 15, Springer, 1990.

[135] J. P. Van der Sluijs, M. Craye, S. Funtowicz, P. Kloprogge, J. Ravetz, and J. Risbey, "Combining quantitative and qualitative measures of uncertainty in model-based environmental assessment: the NUSAP system," *Risk Analysis*, vol. 25, no. 2, pp. 481–492, 2005.

[136] M. Craye, S. Funtowicz, and J. P. Van Der Sluijs, "A reflexive approach to dealing with uncertainties in environmental health risk science and policy," *International Journal of Risk Assessment and Management*, vol. 5, no. 2, pp. 216–236, 2005.

[137] A. Kritikakou, F. Catthoor, V. Kelefouras, and C. Goutis, "A Systematic Approach to Classify Design-time Global Scheduling Techniques," *ACM Computing Surveys*, vol. 45, no. 2, pp. 14:1–14:30, Mar. 2013.

[138] W. Herroelen, E. Demeulemeester, and B. Reyck, "A Classification Scheme for Project Scheduling," in *Project Scheduling*, J. Wglarz, Ed., vol. 14 of *International Series in Operations Research & Management Science*, pp. 1–26. Springer US, 1999.

[139] P. Brucker, A. Drexl, R. Mhring, K. Neumann, and E. Pesch, "Resource-constrained project scheduling: Notation, classification, models, and methods," *European Journal of Operational Research*, vol. 112, no. 1, pp. 3 – 41, 1999.

[140] J. Blazewicz, J. Lenstra, and A. Kan, "Scheduling subject to resource constraints: classification and complexity," *Discrete Applied Mathematics*, vol. 5, no. 1, pp. 11 – 24, 1983.

[141] M. Pinedo, *Scheduling: theory, algorithms, and systems*, Springer, 2012.

[142] T. L. Casavant and J. G. Kuhl, "A taxonomy of scheduling in general-purpose distributed computing systems," *IEEE Transactions on Software Engineering*, vol. 14, no. 2, pp. 141–154, 1988.

[143] R. Kolisch and R. Padman, "An integrated survey of deterministic project scheduling," *Omega*, vol. 29, no. 3, pp. 249 – 272, 2001.

[144] S. Tzafestas and A. Triantafyllakis, "Deterministic scheduling in computing and manufacturing systems: a survey of models and algorithms," *Mathematics and Computers in Simulation*, vol. 35, no. 5, pp. 397 – 434, 1993.

[145] A. Jain and S. Meeran, "Deterministic job-shop scheduling: Past, present and future," *European Journal of Operational Research*, vol. 113, no. 2, pp. 390 – 434, 1999.

[146] A. C. Parker, J. T. Pizarro, and M. Mlinar, "MAHA: a program for datapath synthesis," in *Proceedings of the 23rd ACM/IEEE Design Automation Conference*. IEEE Press, 1986, pp. 461–466.

[147] S. Chaudhuri, S. Blthye, and R. A. Walker, "A solution methodology for exact design space exploration in a three-dimensional design space," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 5, no. 1, pp. 69–81, 1997.

[148] N. Chabini and W. Wolf, "Unification of scheduling, binding, and retiming to reduce power consumption under timings and resources constraints," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 10, pp. 1113–1126, 2005.

[149] A. Hemani and A. Postula, "A neural net based self organising scheduling algorithm," in *Proceedings of the conference on European design automation*. IEEE Computer Society Press, 1990, pp. 136–140.

[150] R. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Magazine*, vol. 4, no. 2, pp. 4–22, 1987.

[151] A. Safir and B. Zavidovique, "Towards a global solution to high level synthesis problems," in *Proceedings of the conference on European design automation*. IEEE Computer Society Press, 1990, pp. 283–288.

[152] E. Tsang, "AIP scheduling techniques-A comparative study," *British Telecom Technology Journal*, vol. 13, no. 1, pp. 16–28, 1995.

[153] C.-Y. Lee, L. Lei, and M. Pinedo, "Current trends in deterministic scheduling," *Annals of Operations Research*, vol. 70, pp. 1–41, 1997.

[154] R. Jiminez-Peris, M. Patiño-Martínez, and S. Arévalo, "Deterministic scheduling for transactional multithreaded replicas," in *Proceedings The 19th IEEE Symposium on Reliable Distributed Systems, 2000. SRDS-2000.* IEEE, 2000, pp. 164–173.

[155] W. Herroelen and R. Leus, "Project scheduling under uncertainty: Survey and research potentials," *European journal of operational research*, vol. 165, no. 2, pp. 289–306, 2005.

[156] Z. Li and M. Ierapetritou, "Process scheduling under uncertainty: Review and challenges," *Computers & Chemical Engineering*, vol. 32, no. 4, pp. 715–727, 2008.

[157] M. Wosko, I. Moser, and K. Mansour, "Scheduling for Optimal Response Times in Queues of Stochastic Workflows," in *AI 2013: Advances in Artificial Intelligence*, pp. 502–513. Springer, 2013.

[158] A. Sturt and G. Strbac, "Efficient stochastic scheduling for simulation of wind-integrated power systems," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 323–334, 2012.

[159] C. G. Baslis and A. G. Bakirtzis, "Mid-term stochastic scheduling of a price-maker hydro producer with pumped storage," *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 1856–1865, 2011.

[160] D. Ouelhadj and S. Petrovic, "A survey of dynamic scheduling in manufacturing systems," *Journal of Scheduling*, vol. 12, no. 4, pp. 417–431, 2009.

[161] I. Ovacikt and R. Uzsoy, "Rolling horizon algorithms for a single-machine dynamic scheduling problem with sequence-dependent setup times," *The International Journal Of Production Research*, vol. 32, no. 6, pp. 1243–1263, 1994.

[162] S. V. Mehta, "Predictable scheduling of a single machine subject to breakdowns," *International Journal of Computer Integrated Manufacturing*, vol. 12, no. 1, pp. 15–38, 1999.

[163] G. E. Vieira, J. W. Herrmann, and E. Lin, "Predicting the performance of rescheduling strategies for parallel machine systems," *Journal of Manufacturing Systems*, vol. 19, no. 4, pp. 256–266, 2000.

[164] G. E. Vieira, J. W. Herrmann, and E. Lin, "Rescheduling manufacturing systems: a framework of strategies, policies, and methods," *Journal of Scheduling*, vol. 6, no. 1, pp. 39–62, 2003.

[165] H. Aytug, M. A. Lawley, K. McKay, S. Mohan, and R. Uzsoy, "Executing production schedules in the face of uncertainties: A review and some future directions," *European Journal of Operational Research*, vol. 161, no. 1, pp. 86 – 110, 2005.

[166] V. Suresh and D. Chaudhuri, "Dynamic schedulinga survey of research," *International Journal of Production Economics*, vol. 32, no. 1, pp. 53 – 63, 1993.

[167] P. P. Stoop and V. C. Wiers, "The complexity of scheduling in practice," *International Journal of Operations & Production Management*, vol. 16, no. 10, pp. 37–53, 1996.

[168] P. Cowling and M. Johansson, "Using real time information for effective dynamic scheduling," *European Journal of Operational Research*, vol. 139, no. 2, pp. 230 – 244, 2002.

[169] M. Yamamoto and S. Nof, "Scheduling/rescheduling in the manufacturing operating system environment," *International Journal of Production Research*, vol. 23, no. 4, pp. 705–722, 1985.

[170] R. O'Donovan, R. Uzsoy, and K. N. McKay, "Predictable scheduling of a single machine with breakdowns and sensitive jobs," *International Journal of Production Research*, vol. 37, no. 18, pp. 4217–4233, 1999.

[171] J. C. Bean, J. R. Birge, J. Mittenthal, and C. E. Noon, "Matchup scheduling with multiple resources, release dates and disruptions," *Operations Research*, vol. 39, no. 3, pp. 470–483, 1991.

[172] M. S. Akturk and E. Gorgulu, "Match-up scheduling under a machine breakdown," *European Journal of Operational Research*, vol. 112, no. 1, pp. 81–97, 1999.

[173] R. Ramasesh, "Dynamic job shop scheduling: a survey of simulation research," *Omega*, vol. 18, no. 1, pp. 43–57, 1990.

[174] C. Rajendran and O. Holthaus, "A comparative study of dispatching rules in dynamic flowshops and jobshops," *European Journal of Operational Research*, vol. 116, no. 1, pp. 156–170, 1999.

[175] G. Chryssolouris and V. Subramaniam, "Dynamic scheduling of manufacturing job shops using genetic algorithms," *Journal of Intelligent Manufacturing*, vol. 12, no. 3, pp. 281–293, 2001.

[176] M. T. Jensen et al., *Robust and flexible scheduling with evolutionary computation*, Ph.D. thesis, BRICS, Computer Science Department, University of Aarhus, 2001.

[177] H. V. D. Parunak, "Agents in overalls: Experiences and issues in the development and deployment of industrial agent-based systems," *International Journal of Cooperative Information Systems*, vol. 9, no. 03, pp. 209–227, 2000.

[178] P. Cowling, D. Ouelhadj, and S. Petrovic, "Multi-agent systems for dynamic scheduling," in *Proceedings of the nineteenth workshop of planning and scheduling of the UK*, 2000, pp. 45–54.

[179] R. W. Brennan and D. H. Norrie, "Evaluating the performance of reactive control architectures for manufacturing production control," *Computers in Industry*, vol. 46,

no. 3, pp. 235–245, 2001.

[180] W. Shen, D. Norrie, and J. Barthes, "Multi-agent systems for concurrent intelligent design and manufacturing. 2001," *New York: Taylor Francis Inc*, vol. 381, 2001.

[181] A. Allahverdi, C. Ng, T. E. Cheng, and M. Y. Kovalyov, "A survey of scheduling problems with setup times or costs," *European Journal of Operational Research*, vol. 187, no. 3, pp. 985–1032, 2008.

[182] E. B. Edis, C. Oguz, and I. Ozkarahan, "Parallel Machine Scheduling with Additional Resources: Notation, Classification, Models and Solution Methods," *European Journal of Operational Research*, 2013.

[183] I. Sabuncuoglu and M. Bayız, "Analysis of reactive scheduling problems in a job shop environment," *European Journal of operational research*, vol. 126, no. 3, pp. 567–586, 2000.

[184] I. Sabuncuoglu and S. Karabuk, "Rescheduling frequency in an FMS with uncertain processing times and unreliable machines," *Journal of Manufacturing Systems*, vol. 18, no. 4, pp. 268–283, 1999.

[185] S. McWilliam, "Anti-optimisation of uncertain structures using interval analysis," *Computers & Structures*, vol. 79, no. 4, pp. 421 – 430, 2001.

[186] Z. Qiu, S. Chen, and D. Song, "The displacement bound estimation for structures with an interval description of uncertain parameters," *Communications in Numerical Methods in Engineering*, vol. 12, no. 1, pp. 1–11, 1996.

[187] Z. Qiu and I. Elishakoff, "Antioptimisation of structures with large uncertain-but-non-random parameters via interval analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 152, no. 3-4, pp. 361–372, 1998.

[188] Z. Qiu, S. Chen, and D. Song, "The displacement bound estimation for structures with an interval description of uncertain parameters," *Communications in Numerical Methods in Engineering*, vol. 12, no. 1, pp. 1–11, 1996.

[189] G.-j. Shao and J.-b. Su, "Interval finite element method and its application on anti-slide stability analysis," *Applied Mathematics and Mechanics*, vol. 28, no. 4, pp. 521–529, 2007.

[190] C. Pantelides, M. Realff, and N. Shah, "Short-term scheduling of pipeless batch plants," *Chemical Engineering Research and Design*, vol. 73, no. A4, pp. 431–444, 1995.

[191] R. Gonzalez and M. J. Realff, "Operation of pipeless batch plants I. MILP schedules," *Computers & Chemical Engineering*, vol. 22, no. 78, pp. 841 – 855, 1998.

[192] X. Lin, S. L. Janak, and C. A. Floudas, "A new robust optimisation approach for scheduling under uncertainty:: I. Bounded uncertainty," *Computers & Chemical Engineering*, vol. 28, no. 67, pp. 1069 – 1085, 2004.

[193] A. Ben-Tal and A. Nemirovski, "Robust solutions of Linear Programming problems contaminated with uncertain data," *Mathematical Programming, Series B*, vol. 88, no. 3, pp. 411–424, 2000.

[194] Z. Y. Jia and M. G. Ierapetritou, "Short-term scheduling under uncertainty using MILP sensitivity analysis," *Industrial & Engineering Chemistry Research*, vol. 43, no. 14, pp. 3782–3791, 2004.

[195] C. A. Floudas and X. Lin, "Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review," *Computers & Chemical Engineering*, vol. 28, no. 11, pp. 2109 – 2129, 2004.

[196] K. Yee and N. Shah, "Improving the efficiency of discrete time scheduling formulation," *Computers and Chemical Engineering*, vol. 22, no. SUPPL.1, pp. S403–S410, 1998.

[197] I. Dedopoulos and N. Shah, "Optimal short-term scheduling of maintenance and production for multipurpose plants," *Industrial and Engineering Chemistry Research*, vol. 34, no. 1, pp. 192–201, 1995.

[198] N. Shah, C. Pantelides, and R. Sargent, "A general algorithm for short-term scheduling of batch operations-II. Computational issues," *Computers and Chemical Engineering*, vol. 17, no. 2, pp. 229–244, 1993.

[199] M. Bassett, J. Pekny, and G. Reklaitis, "Decomposition Techniques for the Solution of Large-Scale Scheduling Problems," *AIChE Journal*, vol. 42, no. 12, pp. 3373–3387, 1996.

[200] A. Elkamel, M. Zentner, J. Pekny, and G. Reklaitis, "A decomposition heuristic for scheduling the general batch chemical plant," *Engineering Optimisation*, vol. 28, no. 4, pp. 299–330, 1997.

[201] Z. Jia and M. Ierapetritou, "Uncertainty analysis on the righthand side for MILP problems," *AIChE Journal*, vol. 52, no. 7, pp. 2486–2495, 2006.

[202] P. T. Boggs and J. W. Tolle, "Sequential Quadratic Programming," *Acta Numerica*, vol. 4, pp. 1–51, 1 1995.

[203] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956.

[204] F. Benhamou and W. J. Older, "Applying interval arithmetic to real, integer, and Boolean constraints," *The Journal of Logic Programming*, vol. 32, no. 1, pp. 1 – 24, 1997.

[205] T. Duff, "Interval arithmetic recursive subdivision for implicit functions and constructive solid geometry," *SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 131–138, July 1992.

[206] T. Hickey, Q. Ju, and M. H. Van Emden, "Interval arithmetic: From principles to implementation," *Journal of the ACM*, vol. 48, no. 5, pp. 1038–1068, Sept. 2001.

[207] E. Hansen, "Interval Arithmetic in Matrix Computations, Part I," *Journal of the Society for Industrial and Applied Mathematics Series B Numerical Analysis*, vol. 2, no. 2, pp. 308–320, 1965.

[208] E. Hansen, "A generalized interval arithmetic," in *Interval Mathematics*, K. Nickel, Ed., vol. 29 of *Lecture Notes in Computer Science*, pp. 7–18. Springer Berlin Heidelberg, 1975.

[209] T. Sunaga, "Theory of an interval algebra and its application to numerical analysis," *Japan Journal of Industrial and Applied Mathematics*, vol. 26, no. 2-3, pp. 125–143, 2009.

[210] A. Krokhin, P. Jeavons, and P. Jonsson, "Reasoning about temporal relations: The tractable subalgebras of Allen's interval algebra," *Journal of the ACM*, vol. 50, no. 5, pp. 591–640, Sept. 2003.

[211] R. L. Schwartz, P. M. Melliar-Smith, and F. H. Vogt, "An interval logic for higher-level temporal reasoning," in *Proceedings of the second annual ACM symposium on Principles of distributed computing*, New York, NY, USA, 1983, PODC '83, pp. 173–186, ACM.

[212] L. K. Dillon, G. Kutty, L. E. Moser, P. M. Melliar-Smith, and Y. S. Ramakrishna, "A graphical interval logic for specifying concurrent systems," *ACM Transaction Software Engineering Methodology*, vol. 3, no. 2, pp. 131–165, Apr. 1994.

[213] Z. Chaochen and M. Hansen, "An Adequate First Order Interval Logic," in *Compositionality: The Significant Difference*, W.-P. Roever, H. Langmaack, and A. Pnueli, Eds., vol. 1536 of *Lecture Notes in Computer Science*, pp. 584–608. Springer Berlin Heidelberg, 1998.

[214] D. A. Randell, Z. Cui, and A. G. Cohn, "An interval logic for space based on c̈onnection," in *Proceedings of the 10th European conference on Artificial intelligence*, New York, NY, USA, 1992, ECAI '92, pp. 394–398, John Wiley & Sons, Inc.

[215] R. Schwartz, P. Melliar-Smith, and F. Vogt, "An interval-based temporal logic," in *Logics of Programs*, E. Clarke and D. Kozen, Eds., vol. 164 of *Lecture Notes in*

Computer Science, pp. 443–457. Springer Berlin Heidelberg, 1984.

[216] Y. Zhang, D. Monder, and J. F. Forbes, "Real-time optimisation under parametric uncertainty: a probability constrained approach," *Journal of Process Control*, vol. 12, no. 3, pp. 373 – 389, 2002.

[217] B. K. Williams, "Reducing uncertainty about objective functions in adaptive management," *Ecological Modelling*, vol. 225, no. 0, pp. 61 – 65, 2012.

[218] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to interval analysis*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2009.

[219] P. Bratley, P. Robillar, and M. Florian, "Scheduling with earliest start and due date constraint," *Naval Research Logistics Quarterly*, vol. 18, no. 4, pp. 511–518, 1971.

[220] E. L. Lawler, "Citation-Classic - Branch-and-Bound Methods - a Survey," *Current Contents//Social & Behavioral Sciences*, , no. 2, pp. 16–16, 1987.

[221] M. Di Loreto, M. Da, L. Jaulin, J. F. Lafay, and J. J. Loiseau, "Applied interval computation: A new approach for time-delays systems analysis," *Applications of Time Delay Systems*, vol. 352, pp. 175–197, 2007.

[222] H. Bandemer, *Mathematics of uncertainty : ideas, methods, application problems*, Studies in fuzziness and soft computing,. Springer, Berlin ; New York, 2006.

[223] Z. Li and M. G. Ierapetritou, "Process scheduling under uncertainty using multiparametric programming," *Aiche Journal*, vol. 53, no. 12, pp. 3183–3203, 2007.

[224] Z. K. Li and M. Ierapetritou, "Process scheduling under uncertainty: Review and challenges," *Computers & Chemical Engineering*, vol. 32, no. 4-5, pp. 715–727, 2008.

[225] M. Ierapetritou and Z. K. Li, "Modeling and managing uncertainty in process planning and scheduling," *optimisation and Logistics Challenges in the Enterprise*, vol. 30, pp. 97–144, 2009.

[226] Z. K. Li and M. G. Ierapetritou, "Robust optimisation for process scheduling under uncertainty," *Industrial & Engineering Chemistry Research*, vol. 47, no. 12, pp. 4148–4157, 2008.

[227] J. H. Ryu, V. Dua, and E. N. Pistikopoulos, "Proactive scheduling under uncertainty: A parametric optimisation approach," *Industrial & Engineering Chemistry Research*, vol. 46, no. 24, pp. 8044–8049, 2007.

[228] L. Maiumder and S. S. Rao, "Interval-based optimisation of aircraft wings under landing loads," *Computers & Structures*, vol. 87, no. 3-4, pp. 225–235, 2009.

[229] R. Tavakkoli-Moghaddam, M. Yaghoubi-Panah, and F. Radmehr, "Scheduling the sequence of aircraft landings for a single runway using a fuzzy programming approach," *Journal of Air Transport Management*, vol. 25, pp. 15–18, 2012.

[230] J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson, "Scheduling aircraft landings - The static case," *Transportation Science*, vol. 34, no. 2, pp. 180–197, 2000.

[231] J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson, "Displacement problem and dynamically scheduling aircraft landings," *Journal of the Operational Research Society*, vol. 55, no. 1, pp. 54–64, 2004.

[232] C. Piluso, J. Huang, Z. Liu, and Y. L. Huang, "Sustainability Assessment of Industrial Systems under Uncertainty: A Fuzzy Logic Based Approach to Short-Term to Midterm Predictions," *Industrial & Engineering Chemistry Research*, vol. 49, no. 18, pp. 8633–8643, 2010.

[233] S. Chowdhury, F. Gibb, and M. Landoni, "Uncertainty in information seeking and retrieval: A study in an academic environment," *Information Processing & Management*, vol. 47, no. 2, pp. 157–175, 2011.

[234] N. Lankton and J. Luft, "Uncertainty and industry structure effects on managerial

intuition about information technology real options," *Journal of Management Information Systems*, vol. 25, no. 2, pp. 203–240, 2008.

[235] Y. Sun, Y. P. Li, and G. H. Huang, "A queuing-theory-based interval-fuzzy robust two-stage programming model for environmental management under uncertainty," *Engineering optimisation*, vol. 44, no. 6, pp. 707–724, 2012.

[236] L. Jin, G. H. Huang, Y. R. Fan, X. H. Nie, and G. H. Cheng, "A hybrid dynamic dual interval programming for irrigation water allocation under uncertainty," *Water Resources Management*, vol. 26, no. 5, pp. 1183–1200, 2012.

[237] Y. R. Fan, G. H. Huang, and Y. P. Li, "Robust interval linear programming for environmental decision making under uncertainty," *Engineering optimisation*, vol. 44, no. 11, pp. 1321–1336, 2012.

[238] Y. Liu, R. Zou, and H. C. Guo, "Risk explicit interval linear programming model for uncertainty-based nutrient-reduction optimisation for the lake qionghai watershed," *Journal of Water Resources Planning and Management-ASCE*, vol. 137, no. 1, pp. 83–91, 2011.

[239] Y. P. Li and G. H. Huang, "An interval-based possibilistic programming method for waste management with cost minimization and environmental-impact abatement under uncertainty," *Science of the Total Environment*, vol. 408, no. 20, pp. 4296–4308, 2010.

[240] P. vanBeek and D. W. Manchak, "The design and experimental analysis of algorithms for temporal reasoning," *arXiv preprint cs/9601101*, 1996.

[241] P. Sucha, M. Kutil, M. Sojka, and Z. Hanzalek, "TORSCHE Scheduling toolbox for MATLAB," 4-6 Oct. 2006.

[242] R. McNaughton, "Scheduling with deadlines and loss functions," *Management Science*, vol. 6, no. 1, pp. 1–12, 1959.

[243] M. Galetakis, C. Roumpos, G. Alevizos, and D. Vamvuka, "Production scheduling of a lignite mine under quality and reserves uncertainty," *Reliability Engineering & System Safety*, vol. 96, no. 12, pp. 1611–1618, 2011.

[244] A. Lamghari and R. Dimitrakopoulos, "A diversified Tabu search approach for the open-pit mine production scheduling problem with metal uncertainty," *European Journal of Operational Research*, vol. 222, no. 3, pp. 642–652, 2012.

[245] E. E. Halvorsen-Weare, K. Fagerholt, and M. Rnnqvist, "Vessel routing and scheduling under uncertainty in the liquefied natural gas business," *Computers & Industrial Engineering*, vol. 64, no. 1, pp. 290–301, 2013.

[246] U. Rathnayake, M. Iftikhar, M. Ott, and A. Seneviratne, "Realistic data transfer scheduling with uncertainty," *Computer Communications*, vol. 34, no. 9, pp. 1055–1065, 2011.

[247] B. Vahdani, R. Tavakkoli-Moghaddam, M. Modarres, and A. Baboli, "Reliable design of a forward/reverse logistics network under uncertainty: A robust-M/M/c queuing model," *Transportation Research Part E-Logistics and Transportation Review*, vol. 48, no. 6, pp. 1152–1168, 2012.

[248] M. Di Loreto, M. Da, L. Jaulin, J. F. Lafay, and J. J. Loiseau, "Applied interval computation: A new approach for time-delays systems analysis," *Applications of Time Delay Systems*, vol. 352, pp. 175–197, 2007.

[249] S. S. Rao and L. Berke, "Analysis of uncertain structural systems using interval analysis," *Aiaa Journal*, vol. 35, no. 4, pp. 727–735, 1997.

[250] G. Muscolino and A. Sofi, "Stochastic analysis of structures with uncertain-but-bounded parameters via improved interval analysis," *Probabilistic Engineering Mechanics*, vol. 28, pp. 152–163, 2012.

[251] E. R. Hansen, *A generalized interval arithmetic*, vol. 29 of *Lecture Notes in Computer Science*, Chapter 2, pp. 7–18, Springer Berlin Heidelberg, 1975.

[252] N. S. Nedialkov, V. Kreinovich, and S. A. Starks, "Interval arithmetic, affine arithmetic, Taylor series methods: why, what next?," *Numerical Algorithms*, vol. 37, no. 1-4, pp. 325–336, 2004.

[253] L. H. de Figueiredo and J. Stolfi, "Affine arithmetic: concepts and applications," *Numerical Algorithms*, vol. 37, no. 1-4, pp. 147–158, 2004.

[254] B. Chen, A. vanVliet, and G. J. Woeginger, "An optimal algorithm for preemptive on-line scheduling," *Operations Research Letters*, vol. 18, no. 3, pp. 127–131, 1995.

[255] G. Schmidt, "Scheduling with limited machine availability," *European Journal of Operational Research*, vol. 121, no. 1, pp. 1–15, 2000.

[256] H. Hoogeveen, M. Skutella, and G. J. Woeginger, "Preemptive scheduling with rejection," *Mathematical Programming*, vol. 94, no. 2-3, pp. 361–374, 2003.

[257] C. F. Gerald and P. O. Wheatley, *Applied numerical analysis*, Pearson/Addison-Wesley, Boston, 7th Edition, 2004.