1-2011

Improving Service through Just-in-Time Concept in a Dynamic Operational Environment

Kar Way TAN
Singapore Management University, karway.tan.2007@smu.edu.sg

Hoong Chuin LAU
Singapore Management University, hclau@smu.edu.sg

Na FU
Singapore Management University, nafu@smu.edu.sg

DOI: https://doi.org/10.1109/HICSS.2011.237

Improving service through Just-in-Time concept in a dynamic operational environment

Kar Way TAN, Na FU and Hoong Chuin LAU

School of Information Systems
Singapore Management University
80 Stamford Road Singapore 178902
{karway.tan.2007, na.fu.2007, hclau}@smu.edu.sg

Abstract

This paper is concerned with the problem of Just-In-Time (JIT) job scheduling in a dynamic environment under uncertainty to attain timely service. We provide an approach, based on robust scheduling concepts, to analytically evaluate the expected cost of earliness and tardiness for each job and also the project. In addition, we search for a schedule execution policy with the minimum robust cost such that for a given risk level ϵ, the actual realized schedule has (1 - ϵ) probability of completing with less than or equal to this robust cost. Our method is quite generic, and can be applied to JIT scheduling of jobs in a service company where earliness and tardiness are critical concerns.

1. Introduction

In service industry such as airlines, health-care and financial institutions, timely delivery of *product* (goods and services) is of utmost importance in gaining customers' confidence and goodwill. While JIT has been an established concept in lean manufacturing, it is also widely applicable in the service industry (where jobs should be completed to fulfill customers' orders in a timely fashion against a backdrop of dynamism and uncertainty in the environment). The notion of timeliness means that the jobs are completed neither too early (thereby minimizing holding cost and maximizing agility) nor too late (minimizing delays and customer waiting time).

In this paper, we study the problem of timely service delivery in an operational environment that is inherently uncertain. The broad research question is - can we obtain a policy on apriori (proactive) resource assignment to jobs that is robust in response to events such as a demand surge? We seek a rigorous modeling of uncertainty parameters and a computationally efficient approach to find solutions that are robust against such uncertainties. More precisely, we will look at a job scheduling context where we need to find robust solutions that are JIT. In Scheduling terminology, the JIT concept is often studied as the well-known earliness/tardiness (E/T) scheduling problems. A rich collection of scheduling problems in the service industry is given in [14].

While there is a rich literature in dealing with E/T problems (mostly in manufacturing contexts), they mainly deal with deterministic problems on single-machine or parallel machines. Baker and Scudder [1] provided a comprehensive survey on the different variations of deterministic E/T problems. In a deterministic setting, it is meaningful to seek a schedule (an assignment of start time of each task to a machine) that optimizes an objective function (such as makespan). However, in a dynamic operational environment particularly in a service company where we need to cope with the presence of uncertainties realistically, a schedule is potentially brittle against uncertainty (such as durational uncertainty).

To hedge against uncertainty, a widely adopted technique known as the Critical Chain was proposed by Goldratt [6] for project management. The idea was derived from Theory of Constraints. A critical chain is the longest sequence of both precedence and resource-dependent tasks in a project, and the idea is to insert *project buffer* time to the end of a critical chain to protect the project completion date, as well as *feeding buffer* time that hedges against resource contention of a task on a non-critical chain with a task on a critical chain should the former gets delayed. While buffering is an intuitive notion that is fairly simple to implement, the question is how much to buffer. In a typical critical chain approach, the size of the buffer increases linearly with the length of the chain with which it is associated, and in [7], it was reported that the resulting makespan based on critical chain can be twice as long compared to the scheme proposed by the authors' approach.

In Operations Research, stochastic programming and simulation techniques are commonly used to handle uncertainty - both of which require high computational budget typically (for large-scale problems) and rely heavily on statistical distributions. These are overcome with techniques from Artificial Intelligence and Economics. A survey by Herroelen and Leus [8] review these fundamental approaches for scheduling under uncertainty. More recently, the notion of robust optimization has been proposed that makes use of mild statistical information to find tractable solutions for optimization problems under uncertainty [3].

A recent approach that combines robust optimization with heuristic search to manage uncertainty proactively is based on the idea of robust local search [12]. In applying to project scheduling with resource constraints, [4], [5] proposed computationally efficient schemes for finding an *executable strategy* (or policy) that hedges against durational and resource uncertainties. More precisely, by making use of mild statistical information (means and variance) they proposed a computationally efficient scheme to generate a partial-order schedule (POS) (i.e. an execution policy) and a "robust" makespan guaranteeing that the schedule subsequently executed against all possible realizations of uncertainty will have a makespan of that value with a probability of at least $(1-\epsilon)$, where $\epsilon \in [0, 1]$.

In this paper, we adopt the idea proposed in the above-mentioned works to study the problem of E/T scheduling under durational uncertainty. We term our problem as Dynamic E/T Job Scheduling Problem. We believe this problem, though somewhat stylized in nature, is capable of modeling many of problems arising from job assignment in the service industry. We see many real-life business process scenarios that are plagued with both E/T and duration uncertainty considerations. One such application is in the production and delivery of perishable goods, such as flight catering business. A job in flight catering business includes multiple operations such as cooking for hot dish, cold dish, preparation of beverages and special meals, assembly of items and packing. An early completion of job results in holding costs and possible inaccuracy but a late completion of job will have severe impact on flight schedule, loss of customer satisfaction and potential disruption to other flights (if food is *borrowed* from another flight). The study of uncertainty is important as passenger load (paxload) is dynamic due to ticket sales and check-ins. The airlines update the caterer on the paxload as late as within 40 minutes before the flights departure. As paxload changes, the durations required to perform some operations such as assembly and packing could vary.

In our work, we aim to find a schedule policy with the *least robust cost* of E/T under duration uncertainty. Our contribution in this paper is three-fold. Firstly, under a stochastic setting, we provide an analytical method to evaluate the expected cost of E/T for each job as well as for all jobs. Secondly, we show that with the use of a executable strategy, we can gain greater visibility into a job completion time. Finally, we propose an efficient local search method to find a locally optimal executable policy under a given risk level ϵ.

The organization of the rest of the paper is as follows: Section 2 describes a business process of a service provider that motivates our work. Section 3 provides brief background information for the model and techniques that we would be applying in the proposed method. Section 4 provides a formal definition of the problem. Section 5 provides the detailed description about our approach to solve the problem. Section 6 shows the experimental results for our method. Finally, Section 7 presents a conclusion to this paper.

2. Motivating Case Study

The Dynamic E/T Job Scheduling Problem is motivated by business processes observed in real world. Typically, E/T concept is predominantly used in supply chain management to produce goods just-in-time to reduce inventory (holding) cost. However, here we present a business process as a case study that E/T scheduling can be applied and form an important aspect of providing quality services. For ease of illustration, simplified version of the actual process is provided here. We shall then use the case study as the basis for our proposed solution.

2.1. The Flight Catering Process

A airport terminal service provider receives orders from their customers (the various airlines) to supply in-flight food and beverages to the flights departing from the airport. The orders are made in advance to the service provider. Each order is a job that the provider must be able to fulfill before the estimated departure time (ETD) of the flights, i.e., the due date of each order. Each job consists of a series of operations (in sequence or in parallel) in order to complete the job. A simplified business process of the catering job is provided in Figure 1. In a job, the operations include preparation of ingredients, hot/cold dishes, special meals (e.g. vegetarian, diabetic meals), beverages, tray assembly and finally loading into truck or moved to staging area. Each operation within the job requires use of one or more types of limited resources and may also require more than one unit of each type of resource. For example,

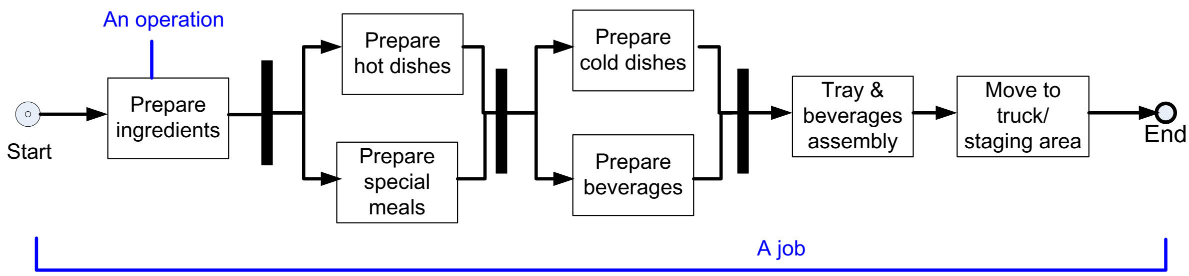

Figure 1. Flight catering business process as an example of a job

preparation of hot dishes may require multiple cooks and stoves.

Based on all the orders received, the provider plans and schedules the jobs in advance. However, such a process is subjected to highly dynamic business environment as the orders can be amended during the job execution due to changes in business demand. The preparation of food begins three days in advance based on initial planned order. However, the airline may amend the order as it approaches the ETD due to changes in passenger load (paxload). Paxload is highly dynamic in the last few days and even hours due to last minute bookings, cancellations and check-in information. The paxload information may arrive at the provider as late as 40 minutes before the ETD! Such amendment of orders affects the duration of the operations in the job and this is the source of uncertainty that we will deal with in this paper.

To further complicate matters, these jobs should complete as close to the due-dates as possible. Early completion of jobs imposes risk to freshness of the products and also occupies staging area and limited resources such as tray tower. Late completion of job is highly detrimental as it may cause flight delays and hence risking the loss of customer's goodwill. The challenge for the provider, not only is to be able to complete the jobs just-in-time but also have an executable strategy for handling jobs in this dynamic environment. The above discussion paves a need for our study of E/T job scheduling under duration uncertainty.

3. Preliminaries

We present in this section, background concepts that form the basis of our approach.

In a nutshell, we make use of the ideas of a Partial Order Schedule [13] and robust local search (proposed in [4], [5], [12]). Given a user-defined risk ϵ, the idea there is to seek a partial order schedule such that the makespan of the actual schedule that will be executed eventually against all uncertainty realizations is at most the robust makespan (denoted as G^*) with a guaranteed probability of at least $(1-\epsilon)$.

Given a set of activities (i.e., a set of comprising all the operations of all jobs) $a_1...a_n$, we can obtain an initial schedule S using a well-studied algorithm such as the serial schedule generation scheme [10] in solving the deterministic problem. In our setting, each activity has processing time $p_i = p_i^0 + \tilde{z}_i$ where p_i^0 is the deterministic mean processing time and $\tilde{z}_i$ represents the stochastic part with possibility of activity completing earlier or later than p_i^0. With an initial schedule S, resource assignment is performed on S to produce a partial order schedule (POS) x. Figure 2 shows an example of a POS. A POS can be represented by a graph where an activity is represented by a node and the edges represent the precedence constraints between the activities. In another words, x is a chain of activities that is both precedence (sequence within the jobs) and resource (due to resource constraints) feasible. Note that the POS does not have any start time associated with each activity (unlike a schedule), instead, it provides the executable strategy by explicitly expressing the logical sequence of activity execution on each machine (while keeping the start time flexible). Hence, in response to an event such as demand surge where the duration of an activity is realized (dynamically), and the POS will be used to guide the decision maker to choose which activity is to be executed next on which machine. In this sense, the POS is an executable strategy (or policy).

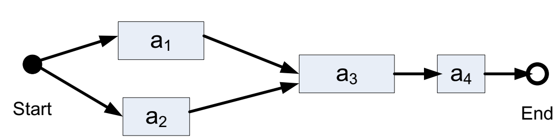

Figure 2. Example of a POS with 4 activities

Given the mean and variance of the duration random variable $\tilde{z}$ and the POS, we can analytically compute the expected makespan and variance of a given POS. Within a POS, the activities are chained either in series or in parallel. Consider a special case where each activity a_i is associated with duration uncertainty random variable $\tilde{z}_i$ that takes normal distribution $N(0, \sigma_i)$, then [5] provides a close-form formula to derive the mean and variance of the makespan for the POS x.

For activities that are in **serial** (e.g. a_3 and a_4 in Figure 2), the expected duration is given by

$$E[\sum_{i=1}^{n}(p_i^0 + \tilde{z}_i)] = \sum_{i=1}^{n} p_i^0 \quad (1)$$

and variance is given by

$$Var[\sum_{i=1}^{n}(p_i^0 + \tilde{z}_i)] = \sum_{i=1}^{n} \sigma_i \quad (2)$$

For m activities that are in **parallel** (e.g. a_1 and a_2 in Figure 2), the expected duration is given by

$$E[\max_{i=1...m} d_i^0 + \max_{i=1...m} \tilde{z}_i] = \max_{i=1...m} d_i^0 + E[\max_{i=1...m} \tilde{z}_i] \quad (3)$$

The variance can be given by

$$Var[\max_{i=1...m} d_i^0 + \max_{i=1...m} \tilde{z}_i] = Var[\max_{i=1...m} \tilde{z}_i] \quad (4)$$

Although mathematically it is difficult to derive an exact expression for $E[\max_{i=1...m} \tilde{z}_i]$ and $Var[\max_{i=1...m} \tilde{z}_i]$, [5] shows the details and proofs of how upper bounds can be obtained. Hence for a given POS x with random variable $\tilde{z}$, the makespan $G(x, \tilde{z})$ can be calculated based on the serial and/or parallel equations 1 to 4. Based on the one-sided Chebyshev's inequality, the schedule has at least $(1 - \epsilon)$ probability that it can be completed by the robust makespan value G^* where

$$G^* = E[G(x, \tilde{z})] + \sqrt{(1-\epsilon)/\epsilon}\sqrt{Var[G(x, \tilde{z})]} \quad (5)$$

Note that in the aforementioned works, the goal is to find a POS with the minimum robust makespan given ϵ. In this paper, we are concerned with finding a POS with the minimum robust cost of E/T.

4. Problem Definition

We formalize the Dynamic E/T Job Scheduling problem as follows: A project consists of n independent jobs j_1,...,j_n, where each job j represents an instance (or case) of the business process with a due date D_j (jobs have distinct due dates), a set of operations O_{jh}, where h_j denotes the hth operation of job j; and a precedence constraint PR matrix of size $O_{jh} \times O_{jh}$ to provide flexibility to allow parallel operations in the job. Each operation O_{jh} has processing time $p_{jh} \in \mathbb{R}^+$ and has resource requirements RQ_{jhk} for resource type R_k. The processing time is subject to uncertainty by a random variable $\tilde{z}$, i.e. $\tilde{p_{jh}} = p_{jh}^0 + \tilde{z}$. p_{jh}^0 denotes the deterministic duration.

We are given a set of k type of resources, which any of its members of type k are capable of executing the same task. Each type of resource is denoted by R_k. Each resource R_k has capacity RC_k. The resources are subjected to constraint $\sum_{jh \epsilon a_t} RQ_{jhk} \leq RC_k \ \forall t, k$, where resource usage for each resource type k does not exceed the capacity at any point of time during the execution of the set of activities in which a_t is the set of active activities at time t.

4.1. Cost of E/T of a job

We define a variable $\tilde{T}_j \in \mathbb{R}^+$ to represent the cost (or penalty) of earliness or tardiness that a job can incur if it is completed before or after its due date D_j. Let W_j and Y_j be the earliness and tardiness cost functions respectively and we assume that both functions can be estimated or obtained from historical data. Also, let $\tilde{C}_j$ be the random variable representing the completion time of job j, then:

1) If job j completes before its due date, then cost is $W_j(D_j - \tilde{C}_j)$ or
2) If job j completes on or after its due date, then cost is $Y_j(\tilde{C}_j - D_j)$.

Hence, we define for our paper that cost of job is

$$\begin{aligned} \tilde{T}_j &= P(\tilde{C}_j \leq D_j) * W_j(D_j - \tilde{C}_j) \\ &+ (1 - P(\tilde{C}_j \leq D_j)) * Y_j(\tilde{C}_j - D_j) \end{aligned} \quad (6)$$

In general, in the case when we have more than two cost functions and if t_i represents each discrete time-points where cost function changes, then we have

$$\tilde{T}_j^g = \sum_i P(t_{i-1} \leq \tilde{C}_j \leq t_i) * Cost(t_{i-1} \leq \tilde{C}_j \leq t_i) \quad (7)$$

Since each project contains j jobs, the E/T cost of a project $\tilde{T}$, is also the sum of the E/T costs of all jobs.

$$\tilde{T} = \sum_{j \in J} \tilde{T}_j \quad (8)$$

4.2. Objective Function

Following the methodology proposed in [12], by using one-sided Chebyshev's inequality, given ϵ, the robust E/T cost T^* is defined as:

$$T^* = E[\tilde{T}] + \sqrt{(1-\epsilon)/\epsilon}\sqrt{Var[\tilde{T}]} \quad (9)$$

Our goal is thus to find an optimal execution policy for the project with a given risk level $\epsilon \in [0, 1]$, i.e. the policy that yields the minimum robust E/T cost T^*_{min}, such that with probability at least $(1 - \epsilon)$, the actual realised cost is less than or equals to T^*_{min}.

4.3. Uniqueness of Our Problem

We highlight the uniqueness of our problem with respect to various standard problems in the literature.

With respect to Resource-Constrained Project Scheduling Problem (RCPSP), our problem includes additional structure of a job which links a series of activities (or operations) together and has a due date. Our problem also differs from a standard Job-Shop Scheduling (JSP) problem as the jobs has due dates, and each operation within the jobs may require one or more units of k types of resources. The standard JSP, on the other hand, involves only single unit of a single resource in each operation. In addition, our work differs from standard E/T scheduling (comprehensive survey found in [1]) as we takes into considerations, duration uncertainty in dynamic environment as compared to deterministic problems. Although scheduling under uncertainty has been studied in RCPSP and JSP (in [4], [5], [12] and [2] respectively), these works are concerned with completing project in shortest possible time and do not have earliness consideration that is important to service timeliness.

5. Proposed Approach to E/T Case Study

In this section, we show our proposed approach in solving the Dynamic E/T Job Scheduling problem for the case-study described in section 2. First, some assumptions/characteristics are given as follows.

- All activities are non-preemptive. Resources assigned to a specific operation work for the entire operation duration.
- Operations in all jobs are of equal priority. Search algorithm may place the operations in any sequence as long as precedence constraints between operations are preserved.

5.1. Overview

Figure 3 shows an overview of our approach to solve the Dynamic E/T Job Scheduling problem.

Table 1 presents our proposed local search algorithm that seeks to find a locally optimal POS. We first randomly form an initial activity list(AL) and obtain an initial schedule S (Line 01-02). Schedule S can be obtained either by a standard scheduling heuristic such as serial schedule generation scheme (no idle time inserted) or any E/T scheduling methods with inserted idle time [9]. With schedule S, forward-backward-chaining (FBC) is applied to obtain a POS for the schedule (Line 03). Lines 05-10 show how we obtain the expected E/T cost $E(\tilde{T})$ for the entire project. For each job j, we find the sub-POS

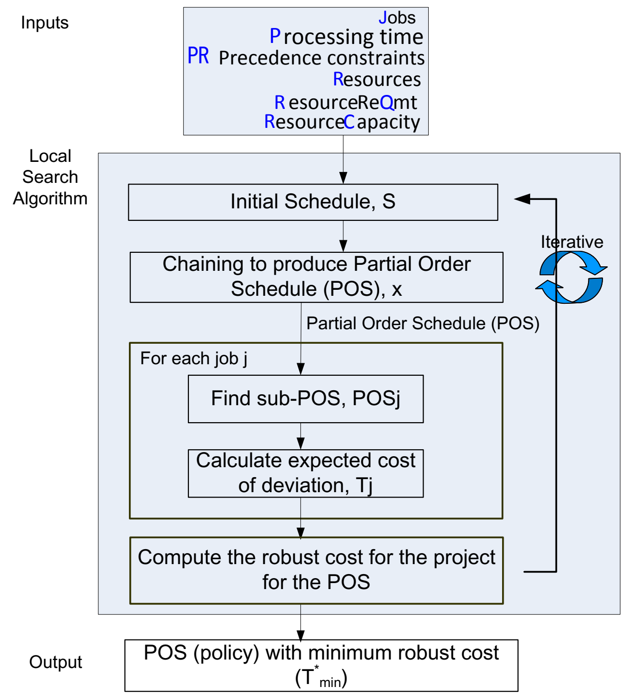

Figure 3. Overview of the approach to E/T job scheduling problem

involving the activities in the jobs (details in section 5.2). The use of sub-POS is required to compute the expected makespan and the expected cost of E/T of the job. The cost of E/T for job j is computed using earliness and tardiness functions(Line 08) and the expected E/T cost of the project is the sum of $E(\tilde{T}_j)$ for all jobs (Line 09). We then calculate the robust cost for the project T^*_{now} based on equation 9 (Line 11).

Line 12-22 is the local search algorithm in which we perform the search over *MaxIteration*. Line 13 to 16 show that we replace POS_{min} and T^*_{min} whenever we find one that gives a smaller robust cost of E/T. Lines $17-18$ shows the local search moves where two operations are randomly selected for swap to produce a new activity list AL'. We then repeat the search process using the new activity list AL' (Lines $19-21$). At the end of $MaxIteration$, we select the POS (execution policy) that provides the minimum robust E/T cost for the project.

5.2. Finding sub-POS

Our approach relies on use of sub-POS to find the expected makespan and the expected cost of E/T of each job. The sub-POS for job j (i.e., POS_j) consists of all the operations in the job as well as the other preceding operations from other jobs in the main POS that job

	Local Search Algorithm
01.	Randomly form an initial activity list AL
02.	Find a schedule, S randomly according to AL
03.	POS ← Forward-Backward-Chaining(S)
04.	$E(\tilde{T}) \leftarrow 0$
05.	For each job j
06.	$POS_j \leftarrow$ FindSubPOS(POS,j)
07.	$ExpMakespan \leftarrow$ ComputeExpectedMakespan(POS_j,$\tilde{z}$)
08.	$E(\tilde{T_j}) \leftarrow$ ComputeExpectedCostOfET(POS_j,Cost functions)
09.	$E(\tilde{T}) \leftarrow E(\tilde{T}) + E(\tilde{T_j})$
10.	endfor
11.	Compute robust cost T^*_{now} using Equation 9
12.	For $i \leftarrow 1$ to MaxIteration, do
13.	if $T^*_{now} \leq T^*_{min}$ then
14.	$T^*_{min} \leftarrow T^*_{now}$
15.	$POS_{min} \leftarrow POS$
16.	endif
17.	Randomly pick two operations a and b from AL
18.	Swap a and b in AL as AL'
19.	Find a schedule S' according to AL'
20.	POS ← FBC(S')
21.	Find T^*_{now} as per line $04 - 11$
22.	endFor
23.	Output a POS_{min} with T^*_{min}

Table 1. Local Search to find POS

j depends on due to sharing of the common pool of resources. Illustrating with an example, figure 4 shows an example of three jobs and its operations. Suppose the operations are translated to activities numbered 1 to 8 and has a possible POS represented in Figure 5. The sub-POS for jobs j_1, j_2 and j_3 consist of activities (1,4,7,2,8,3), (1,4,7,2,8,5,3,6), (1,4,7,2,8) respectively. This is obtained by using the last operation of each job and working backwards recursively to include any preceding activity (parent) in the POS to be included in the sub-POS. An algorithm for determining the sub-POS is shown in table 2.

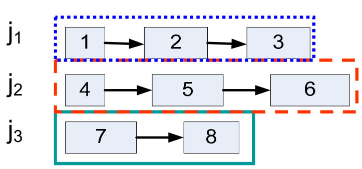

Figure 4. Example of three jobs

The use of sub-POS provides greater visibility to scheduling as one gain visibility to the dependencies on other operations (of another job) and one may also compute the expected makespan of each job according to equation 5. This is particularly important in applications whereby one would like to anticipate and be able to update airlines (i.e., customers) the status of the order and also be able to plan for the next chain of activities, e.g. dispatching service.

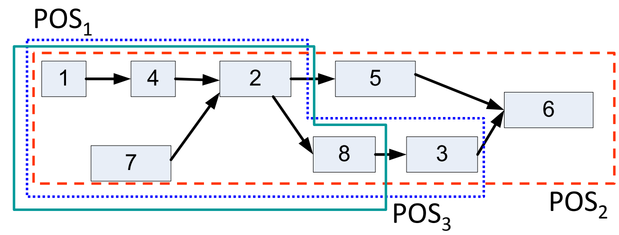

Figure 5. Example of sub-POS for three jobs in Figure 4

	Algorithm to find sub-POS for job j
01.	Given a POS and job j
02.	Find the activity a representing the last operation of job j
03.	Add a to POS_j
04.	Begin Function addParents(a)
05.	//parents refers to the predecessor(s) of a in a POS
06.	Vector $v \leftarrow$ get parents of a
07.	If v not empty
08.	For all element e in v
09.	add e to POS_j
10.	chain e to a
11.	addParents(e) //recursive
12.	endFor
13.	else //no more predecessor
14.	exit function addParents
15.	endIf
16.	endFunction
17.	Return POS_j

Table 2. Algorithm to find the sub-POS for job j

5.3. Compute Expected Cost of E/T for Each Job for Different Cases

The cost of E/T of a job depends on whether job completes before or after the job's due date, D_j and in the dynamic environment, the job's completion time $\tilde{C_j}$ is variable. We first consider a simple model where the cost functions are linear functions of the duration between the job's due date and completion time, such as given in Figure 6.

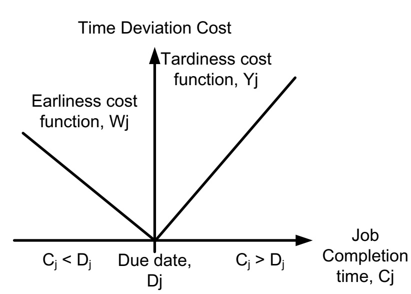

Figure 6. Linear cost functions

In such a situation, we have two mutually exclusive cases which either $\tilde{C}_j$ lies before or after due date. For a sub-POS with expected makespan of $E(\tilde{C}_j)$, by Equation 6 (with α and β as parameters to the cost functions), the expected E/T cost of each job, $E(\tilde{T}_j)$ is then given by

$$\begin{aligned} E(\tilde{T}_j) &= P(\tilde{C}_j \leq D_j) * \alpha(D_j - E(\tilde{C}_j)) \\ &+ (1 - P(\tilde{C}_j \leq D_j)) * \beta(E(\tilde{C}_j) - D_j) \end{aligned} \quad (10)$$

Similarly, the same method can be extended to other continuous cost functions such as quadratic earliness and tardiness cost functions. Also by Equation 6,

$$\begin{aligned} E(\tilde{T}_j) &= P(\tilde{C}_j \leq D_j) * \alpha(D_j - E(\tilde{C}_j))^2 \\ &+ (1 - P(\tilde{C}_j \leq D_j)) * \beta(E(\tilde{C}_j) - D_j)^2 \end{aligned} \quad (11)$$

And for the case when cost functions are step functions, such as the example given in Figure 7, then by Equation 7, we have:

$$\begin{aligned} E(\tilde{T}_j) &= P(\tilde{C}_j < t_1) * Cost(\tilde{C}_j < t_1) \\ &+ \sum_{i=2}^{5} P(t_{i-1} \leq \tilde{C}_j \leq t_i) * Cost(t_{i-1} \leq \tilde{C}_j \leq t_i) \\ &+ (1 - P(\tilde{C}_j < t_5)) * Cost(\tilde{C}_j > t_5) \end{aligned} \quad (12)$$

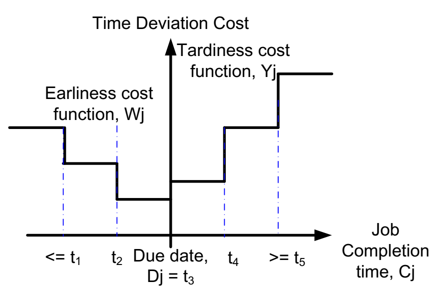

Figure 7. Step cost functions

5.4. Computing Robust Cost for the Project

Having computed the E/T cost for each job, the cost for the entire project $\tilde{T}$ is then given by the sum of these random variables. Assuming the expected value and variance of $\tilde{T}$ can be computed (we will show in the experiments that these quantities are computed via simulation since they are difficult to compute analytically), the goodness of an execution policy (represented by the POS) for a given risk level ϵ is given by T^* based on Equation 9. Accordingly, if we utilize this quantity as the fitness function within a search algorithm, we can obtain an execution policy (POS) with the minimum robust cost. In our work, since the search algorithm is a local search, we obtain a locally optimal POS with an upper bound of the minimum robust cost.

6. Experimental Results

In this section we will present experimental results obtained from applying our proposed local search algorithm to solve the Dynamic E/T Scheduling problem. We execute our algorithm against problem instances derived from the benchmark sets for RCPSP/max J10 as specified in the PSPLib[11]. We used the activity durations (with slight modifications), resource requirements and resource capacities from the benchmark sets. For each instance, we added the layer of jobs, added due dates for each jobs and changed the time-lag parameters to suit the requirements of our tests. For each test instance, we have 4 jobs and each job has either 2 or 3 operations. Each operations requires 1 or more units of k types of resources. 5 different types of resources are used.

For each operations O_{jh}, we set the expected processing time p_{jh} of the stochastic duration as the corresponding deterministic duration given by the benchmarks and assume that duration uncertainty $\tilde{z}_{jh}$ is normally distributed with mean 0. We run the algorithm across 4 increasing levels of risk, ϵ = {0.01, 0.5, 0.1, 0.2} and 4 different local search iterations, $MaxIteration$ = {50, 100, 250, 500}. To reduce the possible effect of random factors during the search process on final results, we average over 5 random executions for each problem instance.

In our tests, we used linear costs functions similar to one shown in Figure 6 where factors α and β (see Equation 10) are set to 0.5 and 0.8 respectively. We used simulation technique to obtain the probabilities required for computation of expected cost for each job. We achieved this by providing 100 actual realizations of the instance for each POS generated by the algorithm. The expected cost of the project $E(\tilde{T})$ is then obtained from summation of expected cost of the jobs and the variance of the E/T cost of the project $Var(\tilde{T})$ is derived through the simulation. The robust E/T cost T^* for each POS is then calculated based on Equation 9.

Firstly, we present the results of sensitivity test on the effect of the risk level ϵ on the robust cost. Figure 8 shows 5 selected instances from our test results with 250 local search iterations. We observe that the robust cost decreases as the level of risk increases. This is important to the planner as a decision maker on which strategy to adopt to improve the service quality in his organization, i.e., an policy that yields lower robust cost by taking a higher risk or vice versa. The experiments also show that this results is consistent with different instances of the problems.

Next, we evaluate the effectiveness of our algorithm by executing the experiment using different cycles of local search. Figure 9 shows 5 selected instances from our test results with $\epsilon = 0.2$. The results show that with 250 iterations of local search, the algorithm is able to provide a good convergence to minimum robust cost.

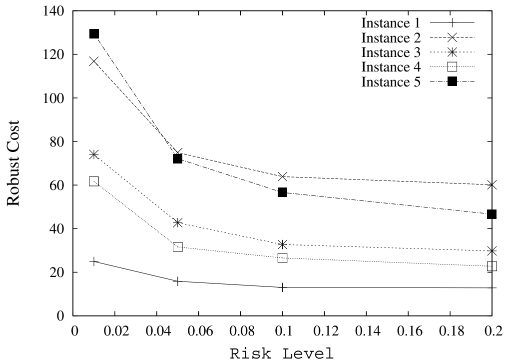

Figure 8. Selected instances of robust cost versus risk level

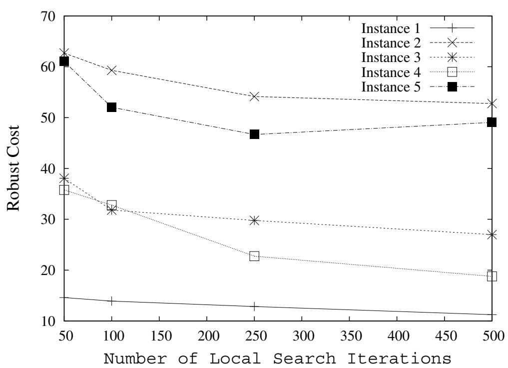

Figure 9. Selected instances showing the ability of algorithm to converge to a minimum robust cost

Finally, we observe that the POS that yields the minimum robust cost is not necessarily the same POS that yields the minimum expected cost of the project. We observe that the POS giving the minimum robust cost may have a higher expected cost compared to another policy with the minimum expected cost. This highlights the value of robust considerations. Suppose the risk level is 0.2. Our approach yields a policy that gives the cost of at most T^*_{min} with probability of at least 0.8. However, if we were to search for a policy with minimum expected cost of the project, if the variance of the project cost is high, we may end up with higher total cost over all realizations of the project. As such, the policy provided by our approach provides a better hedge against adversity and hence valuable to the decision makers.

7. Conclusion

We have shown in this paper, an approach to evaluate the robust cost of earliness and tardiness for a JIT project with duration uncertainty. We discussed that E/T job scheduling problem under uncertainty is an important consideration as it has many possible applications in real-life business processes in improving service delivery. We proposed a local search algorithm in search for a POS with minimum robust E/T cost for all jobs in the project. By using the concept of partial order schedule (POS), we also provide an executable strategy as a guidance to react to uncertainty. In addition, we show that we could add visibility to service provider by having the ability to compute the job completion time for each job.

This work provides opportunities for future work. It would be interesting to explore the possibilities of grouping of jobs that are similar in nature. In the example of flight catering business, to allow the operations to fulfill the orders that requires similar products to be processed together. This has interesting real-life application that may result in cost savings from economies of scale and reduce machine set-up cost if any. We could also extend our work to include more studies on how to improve the search algorithm by applying meta-heuristics such as tabu-search.

References

[1] Kenneth R. Baker and Gary D. Scudder. Sequencing with earliness and tardiness penalties: A review. *Operation Research*, 38(1):22–36, Jan-Feb 1990.

[2] J. Christopher Beck and Nic Wilson. Proactive algorithms for job shop scheduling with probabilistic durations. *J. Artif. Int. Res.*, 28(1):183–232, 2007.

[3] A. Ben-Tal and A. Nemirovski. Robust optimization - methodology and applications. *Math. Prog. Series B*, 92:453–480, 2002.

[4] Na Fu, Hoong Chuin Lau, and Fei Xiao. Generating robust schedules subject to resource and duration uncertainties. *International Conference on Automated Planning and Scheduling (ICAPS)*, 2008.

[5] Na Fu, Pradeep Varakantham, and Hoong Chuin Lau. Towards finding robust execution strategies for rcpsp/max with durational uncertainty. *International Conference on Automated Planning and Scheduling (ICAPS)*, 2010.

[6] Eli M. Goldratt. *Critical Chain*. The North River Press, Great Barrington, 1997.

[7] Willy Herroelen and Roel Leus. On the merits and pitfalls of critical chain scheduling. *Journal of Operations Management*, 19(5):559 – 577, 2001.

[8] Willy Herroelen and Roel Leus. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165(2):289–306, 2005.

[9] John J. Kanet and V. Sriharan. Scheduling with inserted idle time: Problem taxonomy and literature review. *Operations Research*, 48(1):099–110, 2000.

[10] Rainer Kolisch and Sonke Hartmann. Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research*, 174(1):23–37, October 2006.

[11] Rainer Kolisch, Christoph Schwindt, and Arno Sprecher. Benchmark instances for project scheduling problems. pages 197–212, 1998.

[12] Hoong Chuin Lau, Thomas Ou, and Fei Xiao. Robust local search and its application to generating robust schedules. In Mark S. Boddy, Maria Fox, and Sylvie Thibaux, editors, *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 208–215. AAAI, 2007.

[13] Nicola Policella, Stephen F. Smith, Amedeo Cesta, and Angelo Oddi. Generating robust schedules through temporal flexibility. *ICAPS*, pages 209–218, 2004.

[14] Christos Voudouris, G. Owusu, R. Dorne, and D. Lesaint. *Service Chain Management*. Springer-Verlag, Berlin, 2008.