Towards Finding Robust Execution Strategies for RCPSP/max with Durational Uncertainty

Na Fu, Pradeep Varakantham, Lau Hoong Chuin
School of Information Systems, Singapore Management University, Singapore
na.fu.2007@phdis.smu.edu.sg, pradeepv@smu.edu.sg, hclau@smu.edu.sg

Abstract

Resource Constrained Project Scheduling Problems with minimum and maximum time lags (RCPSP/max) have been studied extensively in the literature. However, the more realistic RCPSP/max problems – ones where durations of activities are not known with certainty – have received scant interest and hence are the main focus of the paper. Towards addressing the significant computational complexity involved in tackling RCPSP/max with durational uncertainty, we employ a local search mechanism to generate robust schedules. In this regard, we make two key contributions: (a) Introducing and studying the key properties of a new decision rule to specify start times of activities with respect to dynamic realizations of the duration uncertainty; and (b) Deriving the fitness function that is used to guide the local search towards robust schedules. Experimental results show that the performance of local search is improved with the new fitness evaluation over the best known existing approach.

Introduction

Much research in scheduling has assumed problems with deterministic durations. In real-world scheduling problems, unexpected external events such as manpower unavailability, weather changes, etc. lead to delays or advances in completion of activities, which can in turn have a significant impact on the completion of the overall project. There has been a growing interest to account for such data uncertainty (Rodríguez et al. 2009; Herroelen and Leus 2005) while providing optimized schedules. This paper also focusses on this important issue of duration uncertainty in scheduling problems.

From the computational perspective, stochasticity adds a great deal of complexity to the underlying deterministic problems. For example, for the infinite-resource project scheduling problem in which every processing time has two possible discrete values, the problem of computing the expected makespan (or any point on the cumulative distribution of the optimal makespan), is #P-hard (Hagstrom. 1988; Möhring 2001). It has also been shown that for the scheduling problem $1|stoch\, p_j; d_j = d|E[\sum w_j U_j]$, the problem of computing a policy (i.e. execution strategy) maximizing the probability that some job completes exactly at the deadline is PSPACE-hard (Dean, Goemans, and Vondrák 2004).

The concrete problems of interest in this paper are the Resource Constrained Project Scheduling Problems with minimum and maximum time lags (RCPSP/max), which are of great importance in manufacturing, logistics and project management. Though these problems have been shown to be NP-Hard (Bartusch, Mohring, and Radermacher 1988), local search based techniques (Demeulemeester. and Herroelen. 2002) have achieved great success in solving these problems. Taking a cue from this, we also employ local search techniques to solve RCPSP/max with durational uncertainty. More precisely, we focus on the issues of computing a $robust$ makespan with proven probability of success and a local search method for deriving a policy that will achieve the *robust makespan*.

A popular approach for tackling durational uncertainty in scheduling problems is to employ a hybrid of proactive and reactive methods (Vonder, Demeulemeester, and Herroelen 2007). In this approach, a robust baseline schedule is computed offline initially. This baseline schedule is then modified (if required) during execution reactively based on the occurrence of external events. In this paper, we focus only on the computation of a robust baseline schedule and assume that this schedule can then be modified (if necessary) by any of the existing reactive procedures.

There have been approaches that have addressed the computation of this robust baseline schedule. A recent approach (Beck and Wilson 2007) provides techniques to compute this robust baseline schedule from a risk management perspective, where durations of activities are modeled as random variables. Given a level of risk $0 < \varepsilon \leq 1$, they compute a schedule with minimal (probabilistic) makespan, which has a probability of successful execution of at least $1 - \varepsilon$ over all realizations of the durational uncertainty. The main idea there was to compute a lower bound for the minimal probabilistic makespan by solving a deterministic problem. However, it considers problems where the underlying scheduling problem is a Job Shop Scheduling Problems (JSP), which is a much simpler problem than RCPSP/max.

Another approach is to compute an execution strategy (also known commonly as schedule policy) such as a Partial Order Schedule (POS). This approach was adopted by (Lau,

Ou, and Xiao 2007), which combines techniques from robust optimization with classical local search to compute a POS that minimizes the *robust makespan*, V^*. The robust makespan is one where the probability that the actual realized makespan of the schedule derived from the POS does not exceed it is greater than $1-\varepsilon$, over all realizations of uncertainty. The key idea there was to apply the segregated linear decision rule (Chen et al. 2008) to represent start times of activities and then compute an upper bound for the robust makespan, which is in turn used for guiding the local search to find a POS.

The use of the segregated linear decision rule provides for computing the fitness measure quickly. However, this advantage comes at a price in that the fitness measure is a loose upper bound (which has the side effect of guiding the search into bad regions). In this paper, we address the issue of loose upper bounds with two key contributions: (a) we propose a new general non-linear decision rule; and (b) derive the fitness measure of a POS irrespective of the durational uncertainty distributions with the new decision rule. By using the new fitness measure to guide the local search, we were able to show a definitive improvement in performance through our experiments.

Background

Notationally, a random variable is denoted by $\tilde{x}$, bold face lower case letters like **x** represent vectors and the corresponding tilde case such as $\tilde{\mathbf{x}}$ denote a vector with variables as elements.

RCPSP/max with Durational Uncertainty

The RCPSP/max problem (Bartusch, Mohring, and Radermacher 1988) is defined over a set of N activities $\{a_1, a_2..., a_N\}$. Each activity a_i has an expected processing time of d_i^0 and requires r_{ik} units of resource type k ($k = 1, 2..., K$). C_k is the constant resource capacity for resource type k. Each activity has a start time of $st(a_i)$ and end time of $et(a_i)$, such that $st(a_i) + d_i^0 = et(a_i)$. In addition, there exist a set E of temporal constraints between various activity pairs $\langle a_i, a_j \rangle$ of the form $st(a_j) - st(a_i) \in [T_{ij}^{min}, T_{ij}^{max}]$ called minimum and maximum time lags between the start times of two related activities. Note that in the deterministic case, additional temporal constraints that designate time lags between the end time and the start time of two related activities can be equally transformed in the general start-start form.

Durational uncertainty corresponding to each activity a_i is modeled using a random variable $\tilde{z}_i$. Therefore, the duration of an activity is specified as: $\tilde{d}_i = d_i^0 + \tilde{z}_i$, where $\tilde{z}_i$ can correspond to any probability distribution, with an expected value of 0 and standard deviation σ. In this paper, we assume that these random variables, $\{\tilde{z}_i\}$, corresponding to durational uncertainty are independent of each other. Note that for RCPSP/max instances where the end-start time lag constraint exists, the temporal constraint can be violated during execution. This is due to duration uncertainty, which causes the end time of activities to be uncertain and uncontrollable. Thus, as in RCPSP/max literature (Bartusch, Mohring, and Radermacher 1988), we only handle the general start-start form of time lag constraints in this paper.

Given the level of risk, ε, the goal is then to find a resource and time feasible schedule (an assignment of start times to activities) that minimizes the robust makespan (makespan is the completion time of the last activity).

Partial Order Schedule

A **Partial Order Schedule** (**POS**) (Policella et al. 2004) is a set of activities, which are partially ordered such that any schedule with total activity order that is consistent with the partial order is a resource and time feasible schedule. Mathematically, 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. Within a POS, each activity retains a set of feasible start times, which provides the flexibility to respond to unexpected disruptions. To construct a POS, typically a feasible schedule is first obtained using a simple greedy heuristic. Consequently, a chaining procedure is used to construct the POS. It should be noted the time lag constraints of RCPSP/max have already been incorporated in the underlying temporal constraints network which corresponds to a Simple Temporal Problem (Dechter, Meiri, and Pearl 1991) based on which the POS is constructed. Hence, we do not need to model time lags in the decision rule expressions henceforth.

Robust Local Search Framework

(Lau, Ou, and Xiao 2007) integrated techniques from robust optimization (Chen et al. 2008) with the classical local search and proposed a Robust Local Search Framework to tackle the RCPSP/max problem under duration uncertainty, summarized as follows.

Segregated Linear Decision Rule In the segregated linear decision rule (SLDR) proposed by (Chen et al. 2008), each random variable $\tilde{z}$ is represented by two segregated random variables $\tilde{z}^+$ and $\tilde{z}^-$:

$$\begin{aligned} \tilde{z} &= \tilde{z}^+ - \tilde{z}^- \\ \tilde{z}^+ &= max\{\tilde{z}, 0\} \\ \tilde{z}^- &= max\{-\tilde{z}, 0\}. \end{aligned} \tag{1}$$

Thus, the uncertain processing time or duration of an activity is composed of three components: its mean d^0, lateness $\tilde{z}^+$ (i.e. $max\{\tilde{d} - d^0, 0\}$), and earliness $\tilde{z}^-$ (i.e. $max\{d^0 - \tilde{d}, 0\}$)

$$\tilde{d} = d^0 + \tilde{z}^+ - \tilde{z}^-. \tag{2}$$

For a normally distributed duration, i.e. $\tilde{z} \sim N\{0, \sigma\}$, the respective values of mean and variance for the segregated variables can be summarized as:

$$\begin{aligned} E[\tilde{z}^+] = E[\tilde{z}^-] &= \frac{\sigma}{\sqrt{2\pi}} \\ Var[\tilde{z}^+] = Var[\tilde{z}^-] &= \frac{(\pi - 1)\sigma^2}{2\pi}. \end{aligned} \tag{3}$$

In the context of scheduling, activities are either connected in series or in parallel, and hence the actual start times

of activities can be obtained by performing either a *sum* or *maximum* on a set of random variables. Under SLDR (Chen et al. 2008), upper bounds on both the sum and maximum of random variables are derived as linear functions of segregated variables.

- Serial Activities In the case of a project network involving k activities, any two of which have either precedence constraints in between or competing for the same resource units, a solution in the form of POS requires these activities be executed in series. Thus, the end time of a serial k-activity project in SLDR is expressed as:

$$\tilde{V}_k(\tilde{\mathbf{z}}^+, \tilde{\mathbf{z}}^-) = \sum_{i=1}^{k} (d_i^0 + \tilde{z}_i^+ - \tilde{z}_i^-). \tag{4}$$

- Parallel Activities Consider activities that are executed concurrently, the upper bound of the end time of a parallel k-activity project network in SLDR is represented by a linear function of the positive segregated components of duration perturbations:

$$\tilde{V}_k(\tilde{\mathbf{z}}^+, \tilde{\mathbf{z}}^-) \leq \max_{i=1,\ldots k} \{d_i^0\} + \sum_{i=1}^{k} \tilde{z}_i^+. \tag{5}$$

Robust Fitness Function In (Lau, Ou, and Xiao 2007), the robust optimization problem for scheduling was defined as *finding the minimum value V^* and a POS **x**, such that we have a prescribed probability guarantee that the realized makespan of the feasible schedule instantiated by the POS does not exceed V^**. In other words, computing the minimum V^* such that the following probability bound is guaranteed:

$$P(\tilde{V}(\mathbf{x}, \tilde{\mathbf{z}}) \leq V^*) \geq 1 - \varepsilon, \tag{6}$$

where $\tilde{V}(\mathbf{x}, \tilde{\mathbf{z}})$ represents the actual makespan variable of a feasible schedule derived from POS **x** with uncertainty parameters $\tilde{\mathbf{z}}$, whose value varies with respect to the decision rule as shown above.

From the one-sided Chebyshev's Inequality, the above robust optimization problem can be reformulated as:

$$\begin{array}{ll} \min & V^* \\ s.t. & E[\tilde{V}(\mathbf{x}, \tilde{\mathbf{z}})] + \sqrt{\frac{1-\varepsilon}{\varepsilon}} \sqrt{Var[\tilde{V}(\mathbf{x}, \tilde{\mathbf{z}})]} \leq V^*. \end{array} \tag{7}$$

Thus, the fitness function to be used in the local search can be obtained as a function of the expected value and variance of the adjustable variable (i.e.$\tilde{V}(\mathbf{x}, \tilde{\mathbf{z}})$). The fitness value finally returned is termed as the robust makespan.

Definition 1. *Given a level of risk $0 < \varepsilon \leq 1$ and the adjustable function $\tilde{V}(\mathbf{x}, \tilde{\mathbf{z}})$ expressed under Segregated Linear Decision Rule, the robust fitness function $f(\mathbf{x}, \tilde{\mathbf{z}}, \varepsilon)$ of a POS* **x**, *is defined as*

$$f(\mathbf{x}, \tilde{\mathbf{z}}, \varepsilon) = E[\tilde{V}(\mathbf{x}, \tilde{\mathbf{z}})] + \sqrt{\frac{1-\varepsilon}{\varepsilon}} \sqrt{Var[\tilde{V}(\mathbf{x}, \tilde{\mathbf{z}})]} \tag{8}$$

Robust Local Search Algorithm To efficiently solve problem of finding the POS that minimizes the robust makespan, (Lau, Ou, and Xiao 2007) proposed to employ the above robust fitness function in a traditional local search algorithm. The input to this algorithm are an RCPSP/max problem instance, mean and variance values of the segregated variables of data perturbations and the level of risk, while the output is the POS with a locally minimal robust value.

In the algorithm, local search is conducted on activity lists. An activity list is considered as the sequence of all activities that satisfy the non-negative minimal and maximal time lag constraints, and it is obtained by employing a greedy heuristic. Subsequently, each activity is scheduled sequentially based on its order position in the activity list. After finding a feasible schedule, a POS will be generated by applying the chaining procedure proposed by (Policella et al. 2004). Finally, each POS is evaluated by obtaining its fitness value, and the solution with the lowest robust objective value is the best solution.

The key to the local search is the local moves between activity lists in the space of all activity lists. In this case, it is performed as follows: When the activity list results in an infeasible schedule, the local move will randomly push ahead the first activity which cannot be scheduled in the current activity list. When the activity list results in a feasible schedule, the local move will randomly pick two activities and swap them in the current activity list, while satisfying the non-negative minimal time lag constraints. In order to explore more different activity lists, a probability set at 0.01 is introduced to accept the move to an activity list which leads to an infeasible schedule.

Comments on Segregated Linear Rule Compared with other linear decision rules (Ben-Tal and Nemirovski 2002), the superiority of SLDR (Chen et al. 2008) lies in the ability to linearly express an upper bound on a subset of random variables by dissecting each uncertainty into its positive and negative components (see Eqn 2), which is helpful to decide the start time of each activity (and also eventually the makespan which is represented as the start time of the final dummy activity). Unfortunately, this decision rule increases tractability and scalability at the expense of losing accuracy.

General Non-linear Decision Rule

As mentioned earlier, the SLDR was unable to provide tight upper bounds on robust makespan, due to the restriction on linear decision rules. In this section, we define a general non-linear decision rule(GNLDR), which not only provides tight upper bounds but is also efficiently computable and used effectively for local search.

For clarity and comparison purposes, we use $\tilde{G}$ to denote the start time instead of $\tilde{V}$ used by (Lau, Ou, and Xiao 2007). Given the mean and variance values of duration uncertainty, we first provide the definition of this new decision rule, i.e., the start time representation for activities in serial or in parallel. Furthermore, we compute the robust fitness value by computing the mean and variance values of the adjustable variable, $\tilde{G}(\mathbf{x},\tilde{z})$. To compare results with the pre-

vious method, we also investigate the specific case where duration perturbation for each activity a_i is normally distributed, i.e. $\tilde{z}_i \sim N(0, \sigma_i)$.

Serial Activities

In GNLDR, we represent the end time of the serial *k*-activity project network as the sum of all stochastic durations:

$$\tilde{G}_k(\tilde{\mathbf{z}}) = \sum_{i=1}^{k}(d_i^0 + \tilde{z}_i). \tag{9}$$

Therefore, in this case, we have the same representation as the SLDR.

Since $\{\tilde{z}_i\}_{i=1,...k}$ are random variables with zero mean, we can then calculate the expected value as:

$$E[\sum_{i=1}^{k}(d_i^0 + \tilde{z}_i)] = \sum_{i=1}^{k} d_i^0. \tag{10}$$

Because $\{\tilde{z}_i\}$ are assumed to be independent of each other, the variance value is computed by the following expression:

$$Var[\sum_{i=1}^{k}(d_i^0 + \tilde{z}_i)] = \sum_{i=1}^{k} Var[\tilde{z}_i], \tag{11}$$

and under normal distribution where $\tilde{z}_i \sim N(0, \sigma_i)$, we have

$$Var[\sum_{i=1}^{k}(d_i^0 + \tilde{z}_i)] = \sum_{i=1}^{k} \sigma_i^2. \tag{12}$$

Note that the expressions for expected value and variance in the case of serial activities are identical to the ones used in (Wu., Brown., and Beck 2009).

Parallel Activities

In the parallel case, for ease of explanation, we begin by considering two activities to be executed in parallel and then extend the analysis to multiple parallel activities.

Two Parallel Activities The new decision rule to represent the starting time of an activity, which will begin after the completion of two parallel activities is defined as:

$$\tilde{G}_2(\tilde{\mathbf{z}}) \leq max\{d_1^0, d_2^0\} + max\{\tilde{z}_1, \tilde{z}_2\}. \tag{13}$$

Note that we tighten the bound in Eqn **??** by replacing $\tilde{z}_1^+ + \tilde{z}_2^+$ with $max\{\tilde{z}_1, \tilde{z}_2\}$.

Towards computing the robust fitness value, we now derive the expressions for expected value and variance of the adjustable variable, i.e., the RHS term of Eqn 13 in this case. Firstly, for the expected value:

$$\begin{aligned} &E[max\{d_1^0, d_2^0\} + max\{\tilde{z}_1, \tilde{z}_2\}] \\ &\quad = max\{d_1^0, d_2^0\} + E[max\{\tilde{z}_1, \tilde{z}_2\}]. \end{aligned} \tag{14}$$

In the general case, it is difficult to derive an exact expression for $E[max\{\tilde{z}_1, \tilde{z}_2\}]$ and hence, we provide an upper bound.

Proposition 1. *Expected value for the maximum of two general distributions, $\tilde{z}_1$ and $\tilde{z}_2$ is less than* $\frac{1}{2}(E[\tilde{z}_1]+E[\tilde{z}_2])+\frac{1}{2}\sqrt{Var[\tilde{z}_1] + Var[\tilde{z}_2] + (E[\tilde{z}_1])^2 + (E[\tilde{z}_2])^2}$

Proof. We begin by considering the following two equalities:

$$\begin{aligned} max\{\tilde{z}_1, \tilde{z}_2\} + min\{\tilde{z}_1, \tilde{z}_2\} &= \tilde{z}_1 + \tilde{z}_2 \\ max\{\tilde{z}_1, \tilde{z}_2\} - min\{\tilde{z}_1, \tilde{z}_2\} &= |\tilde{z}_1 - \tilde{z}_2| \end{aligned}$$

We now sum the above two equalities.

$$max\{\tilde{z}_1, \tilde{z}_2\} = \frac{1}{2}(\tilde{z}_1 + \tilde{z}_2 + |\tilde{z}_1 - \tilde{z}_2|), \tag{15}$$

Thus, we can now compute the expected value of the maximum using the following equation.

$$E[max\{\tilde{z}_1, \tilde{z}_2\}] = \frac{1}{2}(E[\tilde{z}_1] + E[\tilde{z}_2] + E|\tilde{z}_1 - \tilde{z}_2|) \tag{16}$$

In addition, by using the definition of variance, we obtain:

$$Var|\tilde{z}_1 - \tilde{z}_2| = E(\tilde{z}_1 - \tilde{z}_2)^2 - (E|\tilde{z}_1 - \tilde{z}_2|)^2 \geq 0,$$

Therefore,

$$E|\tilde{z}_1 - \tilde{z}_2| \leq \sqrt{E(\tilde{z}_1 - \tilde{z}_2)^2}.$$

Since expected values for data perturbation can never be negative during computation of robust fitness value, we can further bound $E|\tilde{z}_1 - \tilde{z}_2|$ as:

$$\begin{aligned} &E|\tilde{z}_1 - \tilde{z}_2| \\ &\leq \sqrt{Var[\tilde{z}_1] + Var[\tilde{z}_2] + E(\tilde{z}_1)^2 + E(\tilde{z}_2)^2}. \end{aligned}$$

Substituting this expression into Eqn 16 yields the bound

$$\begin{aligned} &E[max\{\tilde{z}_1, \tilde{z}_2\}] \\ &\leq \tfrac{1}{2}(E[\tilde{z}_1] + E[\tilde{z}_2]) + \\ &\quad \tfrac{1}{2}\sqrt{Var[\tilde{z}_1] + Var[\tilde{z}_2] + (E[\tilde{z}_1])^2 + (E[\tilde{z}_2])^2}. \end{aligned} \tag{17}$$

Hence the proof. ■

In the special case where $\{\tilde{z}_i\}$ $(i = 1, ...k)$ are normally and identically distributed, i.e. $\tilde{z}_i \sim N(0, \sigma)$, we know from (E.Clark 1961) that there is a closed form representation for the expected value of the maximum when $k = 2$ and $k = 3$, respectively:

$$E[max\{\tilde{z}_1, \tilde{z}_2\}] = \frac{\sigma}{\sqrt{\pi}}$$

$$E[max\{\tilde{z}_1, \tilde{z}_2, \tilde{z}_3\}] = \frac{3\sigma}{\sqrt{2\pi}}$$

Now we focus on deriving expressions for variance of the maximum of two general distributions, i.e., $Var[max(\tilde{z}_1, \tilde{z}_2)]$.

Proposition 2. *Variance for the maximum of two general distributions, $\tilde{z}_1$ and $\tilde{z}_2$ is less than* $Var(\tilde{z}_1) + Var(\tilde{z}_2) + \frac{1}{2}(E(\tilde{z}_1))^2 + \frac{1}{2}(E(\tilde{z}_2))^2$

Proof. From Eqn 15, we have

$$\begin{aligned} &Var[max(\tilde{z}_1, \tilde{z}_2)] \\ &= \tfrac{1}{4}Var[\tilde{z}_1 + \tilde{z}_2 + |\tilde{z}_1 - \tilde{z}_2|] \\ &\leq \tfrac{1}{2}(Var[\tilde{z}_1 + \tilde{z}_2] + Var|\tilde{z}_1 - \tilde{z}_2|). \end{aligned} \tag{18}$$

Firstly, we consider the following two equations.

$$\begin{aligned} Var|\tilde{z}_1 - \tilde{z}_2| &= E(\tilde{z}_1 - \tilde{z}_2)^2 - (E|\tilde{z}_1 - \tilde{z}_2|)^2 \\ Var(\tilde{z}_1 - \tilde{z}_2) &= E(\tilde{z}_1 - \tilde{z}_2)^2 - (E(\tilde{z}_1 - \tilde{z}_2))^2 \end{aligned}$$

Subtracting the second from the first yields

$$Var|\tilde{z}_1 - \tilde{z}_2| = Var(\tilde{z}_1 - \tilde{z}_2) + (E(\tilde{z}_1 - \tilde{z}_2))^2 - (E|\tilde{z}_1 - \tilde{z}_2|)^2.$$

Now, we substitute this expression into the third term of Eqn 18 to obtain:

$$Var[max(\tilde{z}_1, \tilde{z}_2)] \leq Var(\tilde{z}_1) + Var(\tilde{z}_2) + \frac{1}{2}(E(\tilde{z}_1) - E(\tilde{z}_2))^2 - \frac{1}{2}(E|\tilde{z}_1 - \tilde{z}_2|)^2. \quad (19)$$

When no specific distribution about duration perturbation is known, we can obtain a bound for $Var[max(\tilde{z}_1, \tilde{z}_2)]$ as:

$$Var[max(\tilde{z}_1, \tilde{z}_2)] \leq Var(\tilde{z}_1) + Var(\tilde{z}_2) + \frac{1}{2}(E(\tilde{z}_1))^2 + \frac{1}{2}(E(\tilde{z}_2))^2. \quad (20)$$

Hence the proof. ■

It is interesting to consider the special case when both random variables are normally distributed. We first state the following lemma[1].

Lemma 01. *If X is normally distributed $X \sim N(0, \sigma)$, then $Y = |X|$ is half-normally distributed, with*

$$E(Y) = \sigma\sqrt{\frac{2}{\pi}}. \quad (21)$$

Under normal distribution $\tilde{z}_i \sim N(0, \sigma_i)$, since $\tilde{z}_1 - \tilde{z}_2$ is also normally distributed, and $\tilde{z}_1 - \tilde{z}_2 \sim N(0, \sigma_1 + \sigma_2)$, we can conclude from Lemma 01 that $|\tilde{z}_1 - \tilde{z}_2|$ follows half-normal distribution with

$$E|\tilde{z}_1 - \tilde{z}_2| = (\sigma_1 + \sigma_2)\sqrt{\frac{2}{\pi}}. \quad (22)$$

Thus, if we substitute this expression into Eqn 19, we can express an upper bound on the variance value for the maximum duration perturbation of two activities, when $\tilde{z}_i \sim N(0, \sigma_i)$ as :

$$Var[max(\tilde{z}_1, \tilde{z}_2)] \leq (1 - \frac{1}{\pi})(\sigma_1^2 + \sigma_2^2) - \frac{2}{\pi}\sigma_1\sigma_2. \quad (23)$$

Multiple Parallel Activities Extending from two to k ($k > 2$) parallel activities, the completion time can be upper bounded by:

$$\tilde{G}_k(\tilde{\mathbf{z}}) \leq \max_{i=1,...k}\{d_i^0\} + \max_{i=1,...k}\{\tilde{z}_i\} \quad (24)$$

In the following, we first compute the variance value of the above RHS term and then use a similar procedure to compute the expected value. The basic expression for variance of RHS is:

$$Var[\max_{i=1,...k}\{d_i^0\} + \max_{i=1,...k}\{\tilde{z}_i\}] = Var[\max_{i=1,...k}\{\tilde{z}_i\}] \quad (25)$$

To obtain the value of $Var[\max_{i=1,...k}\{\tilde{z}_i\}]$ for general probability distributions, we take advantage of the analysis provided for the two-parallel-activity case above. The following steps outline the overall idea:

(a) Firstly, we group the activity set $\{a_1, ..., a_k\}$ into a couple set $\{C_1, ..., C_{\lceil \frac{k}{2} \rceil}\}$, where each element $C_j (j = 1, ...\lceil \frac{k}{2} \rceil)$ contains two different activities $C_j = \{a_{j1}, a_{j2}\}$ chosen from the activity set. Note that when k is an odd, the final element in the couple set contains just one activity.
(b) For each couple C_j, we apply the maximum operator on duration perturbations of involving activities. Denote $\tilde{c}_j = max\{\tilde{z}_{j1}, \tilde{z}_{j2}\}$, where $\tilde{z}_{j1}$ and $\tilde{z}_{j2}$ are duration perturbations of the two activities involved in C_j, then $Var(\tilde{c}_j)$ can be calculated based on the expression for the two-parallel-activity case.
(c) Then we have $\max_{i=1,...k}\{\tilde{z}_i\} = \max_{j=1,...\lceil \frac{k}{2} \rceil}\{\tilde{c}_j\}$. (Note again just one activity is contained in $C_{\lceil \frac{k}{2} \rceil}$ when k is odd). Then, we can build another couple set from $\{C_1, ..., C_{\lceil \frac{k}{2} \rceil}\}$, and the same method from steps (1) and (2) above is used to compute $Var[\max_{j=1,...\lceil \frac{k}{2} \rceil}\{\tilde{c}_j\}]$ based on Eqn 20 and/or Eqn 23.

To generate the couple set $\{C_1, ..., C_{\lceil \frac{k}{2} \rceil}\}$ for k activities in parallel, we have $\frac{k(k-1)}{2}$ different ways of grouping the activities. Each of these groupings can lead to different levels of tightness of derived robust makespan. The problem to find the grouping technique which provides the best robust fitness value is an open question. Instead, we are trying to provide a good grouping technique under normal distribution $\tilde{z}_i \sim N(0, \sigma_i)$ by solving the following optimization problem:

$$\max_t \sum_{j=1,...\lfloor \frac{k}{2} \rfloor} \sigma_{j1}\sigma_{j2} \quad (26)$$

where t denotes the grouping technique and is also the decision variable; the size of solution space is $\frac{k(k-1)}{2}$; $\{C_j\}$ is the couple set constructed from the activity set under grouping method t; σ_{j1} and σ_{j2} are the standard deviations of data perturbation for durations of activities contained in C_j.

Proposition 3. *The solution t^* to the optimization problem of Eqn 26 is obtained by ordering the k activities in a non-increasing order of their variance values and then grouping all two nearest activities according to the order, i.e. $C_j = \{a_{j1}, a_{j2}\}$, where $j = 1, ...\lfloor \frac{k}{2} \rfloor$ and the standard deviations are in the following order:*

$$\sigma_{11} \geq \sigma_{12} \geq \sigma_{21} \geq \sigma_{22} \geq, ...\sigma_{\lfloor \frac{k}{2} \rfloor 1} \geq \sigma_{\lfloor \frac{k}{2} \rfloor 2}. \quad (27)$$

Proof. Suppose we have another grouping method t', in which all elements in the couple set are the same as under t^* except two couples [2] where the ordering is different, i.e., $C_m = \{a_{m1}, a_{n2}\}$ and $C_n = \{a_{m2}, a_{n1}\}$ ($m \neq n$), where $C_m = \{a_{m1}, a_{m2}\}$ and $C_n = \{a_{n1}, a_{n2}\}$ under t^*. Without loss of generality, assume $m > n$ and from Eqn 27, we have

$$\sigma_{m1} \geq \sigma_{m2} \geq \sigma_{n1} \geq \sigma_{n2}. \quad (28)$$

[1] This can be found in statistics texts, and found online at http://en.wikipedia.org/wiki/Half-normal_distribution

[2] It should be noted that if there is an ordering change in only one couple, then the method still produces the same solution because within a couple the variance computation does not consider the order.

Since t' is supposed to provide a solution which is no less (defined in Eqn 26) than t^*, i.e.
$\sigma_{11}\sigma_{12} + ... + \sigma_{m1}\sigma_{n2} + ... + \sigma_{n1}\sigma_{m2} + ... + \sigma_{\lfloor \frac{k}{2} \rfloor 1}\sigma_{\lfloor \frac{k}{2} \rfloor 2}$
$\geq$
$\sigma_{11}\sigma_{12} + ... + \sigma_{m1}\sigma_{m2} + ... + \sigma_{n1}\sigma_{n2} + ... + \sigma_{\lfloor \frac{k}{2} \rfloor 1}\sigma_{\lfloor \frac{k}{2} \rfloor 2}$.
Therefore, we have

$$\sigma_{m1}\sigma_{n2} + \sigma_{n1}\sigma_{m2} \geq \sigma_{m1}\sigma_{m2} + \sigma_{n1}\sigma_{n2},$$

which is equivalent to: $(\sigma_{m1} - \sigma_{n1})(\sigma_{n2} - \sigma_{m2}) \geq 0$.
This contradicts Eqn 28 (except the case where all standard deviations are equal, in which case mixing the order does not affect anything). Thus, there exists no such t' which is different from t^* by at least two couples and has better objective value. The general case that t' has multiple (more than two) couples different from t^* can be easily derived from to this case (and is omitted due to space constraints).

Hence the proof. ■

As for analyzing the expected value $E[\max_{i=1,...k}\{\tilde{z}_i\}]$, we apply the same procedure employed to calculate the variance, i.e., based on the group solution returned by the above optimization problem, we first calculate the expected value for each couple and then, get the final bound following Eqn 17.

Extended Robust Local Search

In this section, we derive a new fitness function under our proposed GNLDR, and present how the fitness function can used to guide the local search towards finding robust execution strategies (POS).

Fitness Function under General Non-linear Rule

Under GNLDR, we know how the actual start times of the activities are to be set with respect to dynamic realizations of uncertainty and a POS. The makespan for the POS, which is also the start time of the final dummy activity a variable, can as well be constructed(or upper bounded). Moreover, the mean and variance values of the makespan (or its upper bound) can be obtained in terms of given mean and variance values of duration perturbation, according to different ways of connection between activities analyzed above.

Given a $0 < \varepsilon \leq 1$, our goal is to find the minimum robust makespan G^* and a policy (i.e. POS) such that we have $1 - \varepsilon$ probability guarantee that the actual realized makespan instantiated by the policy does not exceed G^*. Based on the similar analysis in (Lau, Ou, and Xiao 2007), we can also derive the robust fitness function under our proposed non-linear decision rule which will be used to guide the local search towards robust strategies:

Definition 2. *Given a level of risk* $0 < \varepsilon \leq 1$ *and the adjustable function* $\tilde{G}(\mathbf{x}, \tilde{\mathbf{z}})$ *expressed under General Non-linear Decision Rule, the robust fitness function* $g(\mathbf{x}, \tilde{\mathbf{z}}, \varepsilon)$ *of a POS* $\mathbf{x}$, *is defined as*

$$g(\mathbf{x}, \tilde{\mathbf{z}}, \varepsilon) = E[\tilde{G}(\mathbf{x}, \tilde{\mathbf{z}})] + \sqrt{\frac{1-\varepsilon}{\varepsilon}}\sqrt{Var[\tilde{G}(\mathbf{x}, \tilde{\mathbf{z}})]} \quad (29)$$

Algorithm Design

To search for better schedules, we provide a minor modification to local search algorithm provided by Lau *et al.* (Lau, Ou, and Xiao 2007). Given the RCPSP/max problem instance, mean and variance values of duration perturbations, level of risk prescribed by the planner, this algorithm will also return the POS with the minimal robust makespan G^*.

We perform local search on the neighborhood set of activity lists and apply the same heuristic to generate the initial solution as in (Lau, Ou, and Xiao 2007). Where we depart from that work in our algorithm is that we design a fitness-based neighborhood generation mechanism, rather than randomly picking the activity list from a randomly generated neighborhood. The idea is, that for the current activity list, five candidate neighbors will be generated randomly and local move will pick the one which results in a POS with the best objective value. To explore different activity lists, we set a higher probability 0.8 to move from an activity list which results in a feasible schedule to one which lead to an infeasible schedule.

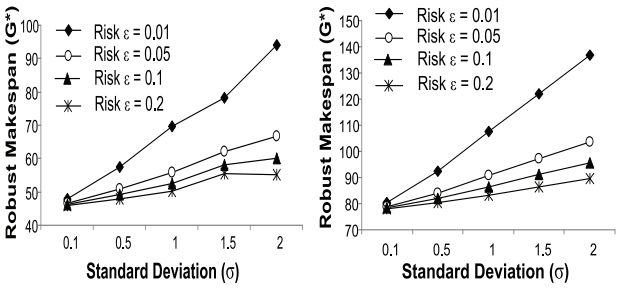
Experimental Evaluation

We tested on 3 benchmark sets for RCPSP/max: J10, J20 and J30 as specified in the PSPLib (Kolisch, Schwindt, and Sprecher 1998). Each set contains 270 problem instances (denoted from PSP1 to PSP270) with duration range for each activity as [1,10]. The maximum number of activities for J10, J20, J30 are 10, 20, 30, respectively. For each benchmark set, we record the average results over the solved problem instances included in that set. Our code was implemented in C++ and executed on a Core(TM)2 Duo CPU 2.33GHz processor under FedoraCore 11 (Kernel Linux 2.6.29.4-167.fc11.i586) with a main memory of 1004MB.

For each activity a_i, we set the expected value d_i^0 of the stochastic duration as the corresponding deterministic duration given by the benchmarks, and assume that duration uncertainty is normally distributed, i.e. $\tilde{z}_i \sim N(0, \sigma)$. We run our algorithm across 5 different duration variabilities $\sigma = \{0.1, 0.5, 1, 1.5, 2\}$ and 4 increasing levels of risk $\varepsilon = \{0.01, 0.05, 0.1, 0.2\}$ with the maximal number of iterations for local search set to 1000. To reduce the possible effect of random factors during the search process on final results, we average over 10 random executions for each problem instance.

Factors affecting G^*

We first present the results of applying our local search guided by our proposed GNLDR, and study how the robust makespan is affected by the level of risk ε and the standard deviation σ of duration uncertainty. This subsection also illustrates that our new algorithm retains the desirable properties of the algorithm by (Lau, Ou, and Xiao 2007). The algorithm will return an execution strategy (denoted as POS_{G^*}) with the robust makespan G^*. We observe that J30 has the same trend as for J10 and J20, thus missing due to space constraints. Figure 1 gives the results for variation in duration standard deviation and level of risk for J10 and J20. We



(a) Results of J10 (b) Results of J20

Figure 1: G^* over different values of σ and ε.

observe that:

(a) As the level of risk ε increases, the robust makespan G^* decreases. Clearly, the lower risk that the planner is willing to take, the higher robust value of the generated execution strategy, and our method is capable of quantifying the tradeoff, which can help the planner to decide on the desired strategies. Thus, the customer-oriented property of the approach of (Lau, Ou, and Xiao 2007) is retained in our method.

(b) As the degree of duration variability σ increases, the robust makespan G^* increases, and the value becomes more sensitive to σ when the level of risk is constrained to be to a small value (e.g. $\varepsilon = 0.01$).

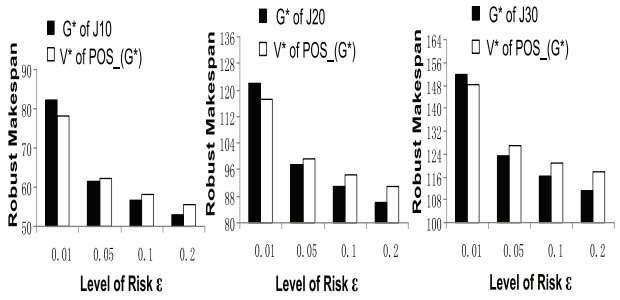


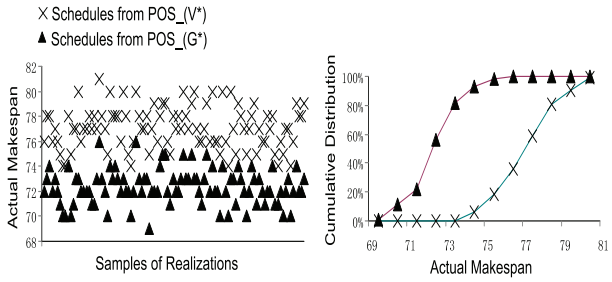
Figure 2: Comparison of G^* and $V^*_{POS_{G^*}}$ under $\sigma = 1.5$.

G^*-guided Local Search vs V^*-guided Local Search

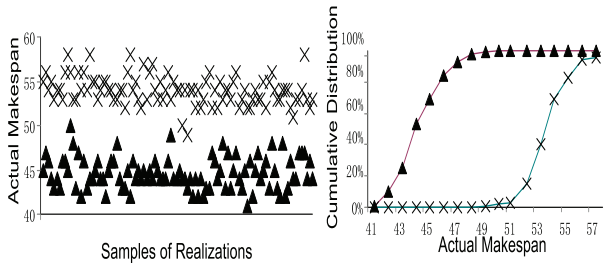
In our next experiment, we compare G^* and V^* corresponding to POS_{G^*}. This is to check whether the new robust fitness value provided by the new decision rule provides tighter bounds on robust makespan values. As can be noted from Figure 2, except for very small values of ε (= 0.01), our new decision rule is able to provide better robust makespan values. For very small values of epsilon, the robust makespan value provided by SLDR is better. However, for slightly higher values of ε (from 0.05), G^* dominates and the range of domination increases for higher values of ε.

Next, in Figure 3, we compare the quality of the execution strategies obtained by our scheme, against those produced by the search method guided by the fitness evaluation function V^* (Lau, Ou, and Xiao 2007). More precisely, we wish to obtain insights on and compare the distributions of the actual makespans of schedules derived using these methods.

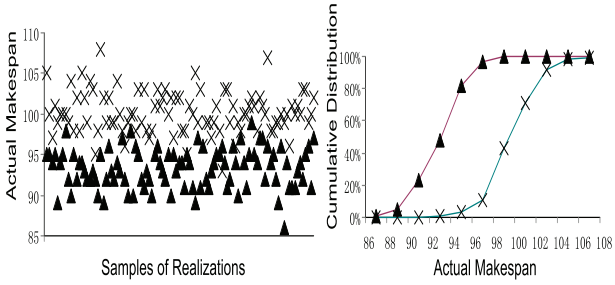
For this purpose, we generate a set of 100 samples of realizations of durational uncertainty and test with all 270 instances of each benchmark set with different levels of risk $\varepsilon = 0.2$, $\varepsilon = 0.1$ and $\varepsilon = 0.05$ to obtain the respective POS, and then compute the actual makespans of schedules derived from the respective POS under the given realization samples. This difference was observed across the board in all examples of the three sets for all values of ε except 0.01. Due to the same trend, we randomly selected three problem instances from the each benchmark set and summary the results in Figure 3. Figure 3 also compares the cumulative frequency distributions of the actual makespans. We observe that our method provided far better realized makespans - both in absolute terms, as well as distributionally.



(a) Results of PSP50 from J10 under $\varepsilon = 0.2$



(b) Results of PSP45 from J20 under $\varepsilon = 0.1$



(c) Results of PSP45 from J30 under $\varepsilon = 0.05$

Figure 3: Comparison of Actual Makespans.

Related Work

In scheduling with duration uncertainty, one key approach is to execute a baseline schedule that is buffered against uncer-

tainty (Aytug et al. 2005). However, baseline schedules may become brittle in face of unpredictable execution dynamics and can quickly get invalidated. On the other hand, Partial Order Schedules (POS) defined in (Policella et al. 2004) can retain temporal flexibility whenever the problem constraints allow it and can often absorb unexpected deviation from predictive assumptions. In (Rasconi., Cesta., and Policella 2010), different methods of generating POS are compared in terms of robustness, flexibility and fluidity of the resulting schedules.

For a good survey of works on project scheduling under uncertainty, one may refer to (Herroelen and Leus 2005). As for the Job Shop Scheduling Problems, a recent work in (Rodríguez et al. 2009) modeled uncertain durations as fuzzy numbers and improved local search methods to solve the problem.

Robust Optimization has been an active topic in Operations Research. (Chen et al. 2008) proposed tractable decision rule models to solve linear stochastic optimization problems. To overcome the inherent computational challenge of this decision rule model in solving large-scale problems, (Lau, Ou, and Xiao 2007) integrated robust optimization techniques into local search and proposed Robust Local Search framework. This framework was then extended with additional consideration of unexpected resource breakdowns in (Fu, Lau, and Xiao 2008), so that POS can absorb both resource and duration uncertainty.

Conclusion

Given a level of risk $0 < \varepsilon \leq 1$ chosen by the planner, we concerned with the problem of finding the minimum $(1-\varepsilon)$-guaranteed makespan (i.e. Robust Makespan) and a policy (i.e. POS) such that when uncertainty is dynamically realized, the execution policy will result in a solution whose value is as good as robust makespan. We first put forward a new decision rule utilized in scheduling to help specify the start times for all activities with respect to execution policy and dynamic realizations of data uncertainty. Based on the decision rule, new fitness function was then derived to evaluate robustness, which was finally integrated into a local search framework to produce the solution with robust makespan. Experimental results illustrate the improved performance of local search with the new fitness evaluation, which provider tighter bounds on robust makespan and better partial order schedules compared to the existing method.

References

Aytug, H.; Lawley, M. A.; McKay, K.; Mohan, S.; and Uzsoy, R. 2005. Executing production schedules in the face of uncertainties: A review and some future directions. In *European Journal of Operational Research*, volume 165(1), 86–110.

Bartusch, M.; Mohring, R. H.; and Radermacher, F. J. 1988. Scheduling project networks with resource constraints and time windows. *Ann. Oper. Res.* 16(1-4):201–240.

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

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

Chen, X.; Sim, M.; Sun, P.; and Zhang, J. 2008. A linear decision-based approximation approach to stochastic programming. *Oper. Res.* 56(2):344–357.

Dean, B. C.; Goemans, M. X.; and Vondrák, J. 2004. Approximating the stochastic knapsack problem: The benefit of adaptivity. In *FOCS*, 208–217.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artif. Intell.* 49(1-3):61–95.

Demeulemeester., E. L., and Herroelen., W. S. 2002. *Project scheduling : a research handbook*. Kluwer Academic Publishers, Boston.

E.Clark, C. 1961. The greatest of a finite set of random variables. *Oper. Res.* 9(2):145–162.

Fu, N.; Lau, H. C.; and Xiao, F. 2008. Generating robust schedules subject to resource and duration uncertainties. In *ICAPS*, 83–90.

Hagstrom., J. N. 1988. Computational complexity of pert problems. *Networks* 18:139–147.

Herroelen, W., and Leus, R. 2005. Project scheduling under uncertainty: Survey and research potentials. In *European Journal of Operational Research*, volume 165(2), 289–306.

Kolisch, R.; Schwindt, C.; and Sprecher, A. 1998. *Benchmark Instances for Project Scheduling Problems*. Kluwer Academic Publishers, Boston. 197–212.

Lau, H. C.; Ou, T.; and Xiao, F. 2007. Robust local search and its application to generating robust schedules. In *ICAPS*, 208–215.

Möhring, R. H. 2001. Scheduling under uncertainty: Bounding the makespan distribution. In *Computational Discrete Mathematics*, 79–97.

Policella, N.; Smith, S. F.; Cesta, A.; and Oddi, A. 2004. Generating robust schedules through temporal flexibility. In *International Conf. on Automated Planning and Scheduling (ICAPS)*, 209–218.

Rasconi., R.; Cesta., A.; and Policella, N. 2010. Validating scheduling approaches against executional uncertainty. *Journal of Intelligent Manufacturing* 21(1):49–64.

Rodríguez, I. G.; Vela, C. R.; Puente, J.; and Hernández-Arauzo, A. 2009. Improved local search for job shop scheduling with uncertain durations. In *ICAPS*.

Vonder, S.; Demeulemeester, E.; and Herroelen, W. 2007. A classification of predictive-reactive project scheduling procedures. *J. of Scheduling* 10(3):195–207.

Wu., C. W.; Brown., K. N.; and Beck, J. C. 2009. Scheduling with uncertain durations: Modeling - robust scheduling with constraints. *Computers and Operations Research* 36:2348–2356.