Optimal Stochastic Single–Machine–Tardiness Scheduling by Stochastic Branch-and-Bound

W.J. Gutjahr, A. Hellmayr and G.Ch. Pflug

Department of Statistics, Operations Research
and Computer Science
University of Vienna

Abstract. A stochastic branch–and–bound technique for the solution of stochastic single–machine tardiness problems with job weights is presented. The technique relies on partitioning the solution space and estimating lower and upper bounds by sampling. For the lower bound estimation, two different types of sampling ("within" and "without" the minimization) are combined. Convergence to the optimal solution (with probability one) can be demonstrated. The approach is generalizable to other discrete stochastic optimization problems. In computational experiments with the single–machine tardiness problem, the technique worked well for problem instances with a relatively small number of jobs; due to the enormous complexity of the problem, only approximate solutions can be expected for a larger number of jobs. Furthermore, a general precedence rule for the single–machine scheduling of jobs with uncertain processing times has been derived, essentially saying that "safe" jobs are to be scheduled before "unsafe" jobs.

Keywords. Scheduling theory, stochastic scheduling, single–machine tardiness problem, branch–and–bound.

1 Introduction

An important problem investigated in sequencing literature is the *single machine total tardiness problem* (SMTTP): A set of k jobs with indices $1, \ldots, k$ is to be processed on a single machine. To each job i, a processing time p_i and a due date d_i are assigned $(i = 1, \ldots, k)$. The aim is to find a permutation $x = (x(1), \ldots, x(k))$ of $\{1, \ldots, k\}$ minimizing the *total tardiness*

$$\sum_{j=1}^{k} [p_{x(1)} + \ldots + p_{x(j)} - d_{x(j)}]^+, \tag{1}$$

i.e., the sum of the tardiness values of the jobs if they are arranged in order x. Therein, $x(1)$ is the index of the first job to be processed, $x(2)$ is the index of the second job to be processed, etc. (For references, see French [7] and Lawler et al. [4]). In 1990, Du and Leung [6] proved the SMTTP to be NP-hard (in the ordinary sense).

According to the symbolic classification presented in [4], the SMTTP is the problem $1 || \sum T_i$. In the *weighted* case, where the jth term of (1) is multiplied by a factor $\gamma_{x(j)}$, we obtain the problem $1 \,||\, w_i T_i$ which is strongly NP–hard.

In many applications, especially in the area of project management, it is not adequate to consider processing times of jobs as well-known, deterministic quantities. There is uncertainty on the time job i will actually require, which suggests that this processing time should be considered as a *random variable* P_i. Hence we obtain the *stochastic* single machine total tardiness problem (SSMTTP):

$$\text{Minimize } \mathbf{E}\,(\sum_{j=1}^{k} [P_{x(1)} + \ldots + P_{x(j)} - d_{x(j)}]^+) \quad \text{for } x \in \mathcal{S}_k, \tag{2}$$

where $\mathcal{S}_k$ is the set of permutations of $\{1, \ldots, k\}$. The objective function is the expected total tardiness. We shall assume that the random variables $P_1, \ldots, P_k$ are independent. No special assumption on the distributions $D_1, \ldots, D_k$ of $P_1, \ldots, P_k$ shall be made, except that for each i, the expected value $\mathbf{E}(P_i)$ exists, and there is some means or other of *sampling* P_i according to D_i for simulation purposes. Our approach uses samples of the random variables P_i as the only information on the distributions D_i. In that, the range of applicability of our technique is broader than that of most other approaches in stochastic scheduling, where only specific distributions of processing times (usually exponential distributions) are considered (see Pinedo [14]).

Note that problem (2) requires the jobs to be scheduled in advance, i.e., the unexecuted jobs $x(j+1)$, …, $x(k)$ cannot be re–scheduled after the processing times $P_{x(1)}, \ldots, P_{x(j)}$ of the j already executed jobs have been observed ($1 \leq j < k$).

As in the deterministic case, a straightforward generalization of (2), and relevant for practical applications, consists in assigning *weights* $\gamma_1, \ldots, \gamma_k$ to the jobs $1, \ldots, k$. The value γ_i indicates the importance of finishing job i in time. In the computation of the total tardiness, the tardiness of job i is counted with the factor γ_i, such that (2) takes the form

$$\text{Minimize } \mathbf{E}\,(\sum_{j=1}^{k} \gamma_{x(j)}\, [P_{x(1)} + \ldots + P_{x(j)} - d_{x(j)}]^+) \quad \text{for } x \in \mathcal{S}_k. \tag{3}$$

Problem (3) can be classified as a *discrete stochastic optimization problem.* Recently, some special techniques for discrete stochastic optimization have been developed, e.g., stochastic simulated annealing (Gutjahr and Pflug [8]) or the stochastic branch-and-bound method (Norkin, Ermoliev and Ruszczynski [12]). The main intention of the present article is to investigate the applicability of stochastic branch-and-bound to stochastic scheduling problems, and we have decided to concentrate on the SSMTTP as a representative example of a hard problem of this class. Let us mention that stochastic scheduling problems have recently been attacked by other methods as well; interesting results can be found, e.g., in Birge's and Dempster's article [2].

The plan of the present paper is the following: In section 2, the stochastic branch-and-bound method will be recapitulated. Section 3 describes our application of this method to the

SSMTTP. In section 4, we develop an extension of the technique in [12]; this extension can also be used for other applications of stochastic branch-and-bound. Section 5 derives a general precedence rule for uncertain jobs. Section 6 presents experimental results.

2 Stochastic Branch-and-Bound

A general discrete stochastic optimization problem can be formulated as follows:

$$\text{Minimize } F(x) = \mathbf{E}(f(x,\omega)) = \int f(x,\omega)\mathbb{P}(d\omega) \quad \text{for } x \in \mathcal{X}, \tag{4}$$

where $\mathcal{X}$ is a finite set of possible decisions or actions and $\omega \in \Omega$ denotes the influence of randomness, formally described by a probability space $(\Omega, \Sigma, \mathbb{P})$. By $\mathbf{E}$, the expectation of a random variable is denoted; it can also be represented in integral form, as indicated above. $f(x,\omega)$ is any cost function, depending both on the decision variable x and on the random influence ω.

The stochastic branch-and-bound method for solving (4), as developed in [12], consists of

- partitioning the feasible set $\mathcal{X}$ into smaller subsets, and
- estimating lower and upper bounds of the objective function $F(x)$ within these subsets.

A key idea is to apply the well-known *lowest-bound rule* from deterministic branch-and-bound: At each step of the algorithm, the subset with minimum estimated lower bound is selected for a further partition into smaller subsets. In this way, "promising" subsets are investigated in more detail. Contrary to deterministic branch-and-bound, there is no definite step when the algorithm terminates with the exact solution. Instead, the computation can be aborted according to some stopping criterion selected by the user, yielding an approximate solution for (4).

For a more explicit description, let us denote by $\mathcal{X}^s$ $(s = 1, 2, \ldots)$ the current subsets into which the original set $\mathcal{X}$ has been divided. In total, the sets $\mathcal{X}^s$ form a partition $\mathcal{P}$ of $\mathcal{X}$. Correspondingly, the original problem (4) is divided into subproblems

$$\text{Minimize } F(x) = \mathbf{E}(f(x,\omega)) \quad \text{for } x \in \mathcal{X}^s,$$

where $\mathcal{X}^s \in \mathcal{P}$. Let us set $F^*(\mathcal{X}^s) = \min_{x \in \mathcal{X}^s} F(x)$.

The following assumptions are made:

1. There is a *lower bound function* L and an a *upper bound function* U, both of them mapping the set of subsets of $\mathcal{X}$ into the set $I\!R$ of real numbers, such that for all $\mathcal{X}^s \in \mathcal{P}$,

$$L(\mathcal{X}^s) \leq F^*(\mathcal{X}^s) \leq U(\mathcal{X}^s)$$

with $U(\mathcal{X}^s) = F(x')$ for some $x' \in \mathcal{X}^s$, and

$$L(\mathcal{X}^s) = F^*(\mathcal{X}^s) = U(\mathcal{X}^s)$$

if $\mathcal{X}^s$ is a singleton set.

2. There exist sequences $\xi^l(\mathcal{X}^s)$, $l = 1, 2, \ldots$, and $\eta^m(\mathcal{X}^s)$, $m = 1, 2, \ldots$, of *random estimates* of $L(\mathcal{X}^s)$ resp. $U(\mathcal{X}^s)$, such that

$$\xi^l(\mathcal{X}^s) \to L(\mathcal{X}^s) \text{ a.s. as } l \to \infty, \text{ and}$$

$$\eta^m(\mathcal{X}^s) \to U(\mathcal{X}^s) \text{ a.s. as } m \to \infty.$$

Note that $\xi^l(\mathcal{X}^s)$ and $\eta^m(\mathcal{X}^s)$ are random variables, while $L(\mathcal{X}^s)$ and $U(\mathcal{X}^s)$ are deterministic quantities.

With these assumptions and notations, we are now in the position to formulate the stochastic branch-and-bound algorithm:

Stochastic Branch-and-Bound:

$\mathcal{P}_0 := \{\mathcal{X}\}$; $\xi_0(\mathcal{X}) := \xi^{l_0}(\mathcal{X})$; $\eta_0(\mathcal{X}) := \eta^{m_0}(\mathcal{X})$;
set $r := 0$;
until stopcriterion satisfied
{ select the lowest-bound subset $\mathcal{Y}^r \in \text{argmin}\ \{\xi_r(\mathcal{X}^s) \mid \mathcal{X}^s \in \mathcal{P}_r\}$;
select an approximate solution $x^r \in \mathcal{X}^r \in \text{argmin}\ \{\eta_r(\mathcal{X}^s) | \mathcal{X}^s \in \mathcal{P}_r\}$;
if ($\mathcal{Y}^r$ is a singleton) set $\mathcal{P}_{r+1} := \mathcal{P}_r$;
else { construct a partition of the lowest-bound subset:
$\mathcal{P}'_r(\mathcal{Y}^r) = \{\mathcal{Y}^r_i \mid i = 1, \ldots, n_r\}$;
construct the new full partition:
$\mathcal{P}_{r+1} = (\mathcal{P}_r \setminus \{\mathcal{Y}^r\}) \cup \mathcal{P}'_r(\mathcal{Y}^r);$ }
set $r := r + 1$;
for (all subsets $\mathcal{X}^s \in \mathcal{P}_r$)
determine estimates $\xi_r(\mathcal{X}^s) = \xi^{l_r(\mathcal{X}^s)}(\mathcal{X}^s)$ and $\eta_r(\mathcal{X}^s) = \eta^{m_r(\mathcal{X}^s)}(\mathcal{X}^s)$;
}

In [12], the following convergence result is demonstrated:

Theorem 2.1. (Norkin, Ermoliev and Ruszczynski): Suppose the indices $l_r(\mathcal{X}^s)$ and $m_r(\mathcal{X}^s)$ are chosen in such a way that whenever a subset $\mathcal{X}'$ is element of $\mathcal{P}_r$ for infinitely many r, then

$\lim_{r\to\infty} l_r(\mathcal{X}') = \lim_{r\to\infty} m_r(\mathcal{X}') = \infty$ a.s. Then with probability one there exists a number r_0 such that for all $r \geq r_0$:

1. the lowest-bound subsets $\mathcal{Y}^r$ are singletons and contain optimal solutions only;

2. the approximate solutions x^r are optimal solutions.

3 Application to the SSMTTP

In this section, we return to the SSMTTP described in the introduction. Here, the cost function $f(x, \omega)$ of (4) is given by

$$f(x,\omega) = \sum_{j=1}^{k} \gamma_{x(j)} \, [P_{x(1)}(\omega) + \ldots + P_{x(j)}(\omega) - d_{x(j)}]^+,$$

and $\mathcal{X} = \mathcal{S}_k$. Before applying the stochastic branch-and-bound algorithm, we have to specify how the partitioning step is to be performed, and how estimates for lower and upper bounds are determined.

1. **Partitioning**

   A natural way of partitioning $\mathcal{X} = \mathcal{S}_k$ into subsets is the following: Let $\mathcal{X}_{s_1,\ldots,s_j}$ denote the set of all permutations $x \in \mathcal{S}_k$ with $x(1) = s_1, \ldots, x(j) = s_j$. Then, whenever a lowest-bound subset $\mathcal{X}_{s_1,\ldots,s_j}$ is to be exposed to a further partitioning during the execution of the algorithm, we do this by constructing the sets $\mathcal{X}_{s_1,\ldots,s_j,m}$, where $m \in \{1,\ldots,k\} \setminus \{s_1,\ldots,s_j\}$. The current hierarchy of partitions at iteration r of the algorithm can be represented by a tree, as it is also customary in deterministic branch-and-bound. Fig. 1 shows an example of such a tree.

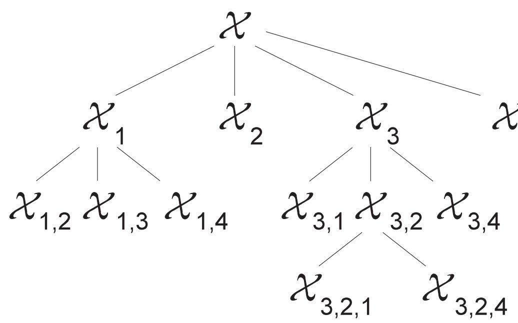

Fig. 1. A branch-and-bound tree for the SSMTTP ($k = 4$). The sets $\mathcal{X}_{3,2,1} = \{(3,2,1,4)\}$ and $\mathcal{X}_{3,2,4} = \{(3,2,4,1)\}$ are already singletons.

2. **Bound Estimation**

Lower bounds: For obtaining an estimate of a lower bound $L(\mathcal{X}^s)$ for $F(x) = \mathrm{E}(f(x, \omega))$, $x \in \mathcal{X}^s$, one may use interchange of minimization and expectation operator, as suggested in [12]: It can immediately be verified that

$$F^*(\mathcal{X}^s) = \min_{x \in \mathcal{X}^s} \mathrm{E}(f(x, \omega)) \geq \mathrm{E}(\min_{x \in \mathcal{X}^s} f(x, \omega)).$$

Thus, it is possible to choose

$$L(\mathcal{X}^s) = \mathrm{E}\left(\min_{x \in \mathcal{X}^s} f(x, \omega)\right).$$

The last expression, however, may be estimated by sampling: Draw N values $\omega_1, \ldots, \omega_N$ independently according to distribution $\mathbb{P}$. Each ω_ν gives rise to a specific sequence $P_1(\omega_\nu), \ldots, P_k(\omega_\nu)$ of processing times. For each ω_ν, determine

$$\min_{x \in \mathcal{X}^s} f(x, \omega_\nu)$$

by solving a (deterministic) SMTTP. Then

$$\frac{1}{N} \sum_{\nu=1}^{N} \min_{x \in \mathcal{X}^s} f(x, \omega_\nu)$$

is an unbiased estimate of $L(\mathcal{X}^s)$. Obviously, the variance of this estimate can be reduced to any desired degree by increasing the sample size N, such that condition 2 in section 2 on the lower-bound estimates is satisfied.

In section 4, we shall describe a further improvement of this technique.

Upper bounds: An upper bound $U(\mathcal{X}^p s)$ is given by the objective function value $F(x)$ in an arbitrary point $x \in \mathcal{X}^s$. For obtaining a *good* (i.e., low) upper bound, we can proceed as follows: We replace the SSMTTP by a corresponding SMTTP, setting $p_j = \mathrm{E}(P_j)$ $(j = 1, \ldots, k)$. (If $\mathrm{E}(P_j)$ is not known, p_j is chosen as an estimate of $\mathbf{E}(P_j)$, obtained by sampling.) The resulting SMTTP is approximately solved by heuristic optimization, which yields a specific $x^* \in \mathcal{X}^s$. (As to techniques for the heuristic solution of SMTTPs, see, e.g., Panwalkar et al. [13] and other articles cited therein.) This determines an upper bound $U(\mathcal{X}^s) = F(x^*)$. The value $F(x^*)$ is estimated by taking the average of $f(x^*, \omega_\nu)$ for samples $\omega_1, \ldots, \omega_N$ drawn independently according to $\mathbb{P}$. By successively increasing the value of N, we obtain a sequence of unbiased estimates for $U(\mathcal{X}^s)$ satisfying condition 2 in section 2 on the upper-bound estimates.

4 Improved Lower Bounds

While the upper bounds defined in section 3 can be expected to be sufficiently good, the gaps between the lower bounds $L(\mathcal{X}^s) = \mathrm{E}(\min_{x\in\mathcal{X}^s} f(x,\omega))$ and the function minima $F^*(\mathcal{X}^s) = \min_{x\in\mathcal{X}^s} F(x)$ may be, in some cases of application, too large to guarantee the efficiency of the algorithm. In this section, we try to improve these bounds by doing part of the sampling (with respect to different ω-values) already *within* the argument of the minimum operator in the expression $\min_{x\in\mathcal{X}^s} f(x,\omega)$. We call this *inner sampling*.

The results of this section do not only hold for the application to the SSMTTP, but for any application of the stochastic branch-and-bound method.

For any subset $\mathcal{Y} \subseteq \mathcal{X}$, we have

$$F^*(\mathcal{Y}) = \min_{x\in\mathcal{Y}} F(x) = \min_{x\in\mathcal{Y}} \int f(x,\omega)\mathbb{P}(d\omega).$$

Let us now define

$$G_n(\mathcal{Y}) = \int \min_{x\in\mathcal{Y}} \left(\frac{1}{n} \sum_{i=1}^{n} f(x,\omega_i) \right) \mathbb{P}(d\omega_1)\ldots\mathbb{P}(d\omega_n). \tag{5}$$

The last expression is the expectation of $\min_{x\in\mathcal{Y}} \left(\frac{1}{n}\sum_{i=1}^{n} f(x,\omega_i)\right)$ for n independent sample values ω_i. Obviously, $G_1(\mathcal{Y}) = \mathbf{E}(\min_{x\in\mathcal{Y}} f(x,\omega))$, which is just the lower bound $L(\mathcal{Y})$ of section 3. The following theorem shows that $G_n(\mathcal{Y})$ forms a sequence of lower bounds with increasing quality, that the gap to $F^*(\mathcal{Y})$ tends to zero, as $n \to \infty$, and that the choice of the weights $1/n$ in the definition of $G_n(\mathcal{Y})$ is optimal:

Theorem 4.1.

(i) $G_1(\mathcal{Y}) \le G_n(\mathcal{Y}) \le G_{n+1}(\mathcal{Y}) \le F^*(\mathcal{Y})$.

(ii) $\lim_{n\to\infty} G_n(\mathcal{Y}) = F^*(\mathcal{Y})$.

(iii) If $G_n(\mathcal{Y}) = G_{n+1}(\mathcal{Y})$ for some n, then $G_n(\mathcal{Y}) = G_{n+1}(\mathcal{Y}) = G_{n+2}(\mathcal{Y}) = \ldots = F^*(\mathcal{Y})$.

(iv) For $\alpha = (\alpha_1,\ldots,\alpha_n)$ with $\alpha_i \ge 0$, $\sum_{i=1}^{n} \alpha_i = 1$, let

$$G_n^{(\alpha)}(\mathcal{Y}) = \int \min_{x\in\mathcal{Y}} \left(\sum_{i=1}^{n} \alpha_i f(x,\omega_i) \right) \mathbb{P}(d\omega_1)\ldots\mathbb{P}(d\omega_n).$$

Then

$$G_n^{(\alpha)}(\mathcal{Y}) \le G_n(\mathcal{Y}),$$

i.e., $\alpha = (1/n,\ldots,1/n)$ gives the optimal weights.

Proof:

(i) First, we show $G_n(\mathcal{Y}) \le F^*(\mathcal{Y})$ for each $n \ge 1$:

$$\begin{aligned} F^*(\mathcal{Y}) &= \min_{x\in\mathcal{Y}} \frac{1}{n}\sum_{i=1}^{n}\int f(x,\omega_i)\mathbb{P}(d\omega_i) \\ &= \min_{x\in\mathcal{Y}} \int \frac{1}{n}\sum_{i=1}^{n} f(x,\omega_i)\mathbb{P}(d\omega_1)\ldots\mathbb{P}(d\omega_n) \\ &\ge \int \min_{x\in\mathcal{Y}} \frac{1}{n}\sum_{i=1}^{n} f(x,\omega_i)\mathbb{P}(d\omega_1)\ldots\mathbb{P}(d\omega_n) = G_n(\mathcal{Y}). \end{aligned}$$

Next, $G_n(\mathcal{Y}) \le G_{n+1}(\mathcal{Y})$ for each $n \ge 1$ is shown. Using

$$\frac{1}{n+1}\sum_{i=1}^{n+1} f(x,\omega_i) = \frac{1}{n+1}\sum_{i=1}^{n+1}\frac{1}{n}\sum_{j\le n+1, j\ne i} f(x,\omega_j),$$

one obtains

$$\min_{x\in\mathcal{Y}} \frac{1}{n+1}\sum_{i=1}^{n+1} f(x,\omega_i) \ge \frac{1}{n+1}\sum_{i=1}^{n+1}\min_{x\in\mathcal{Y}}\frac{1}{n}\sum_{j\le n+1, j\ne i} f(x,\omega_j).$$

Integration over $\mathbb{P}(d\omega_1)\ldots\mathbb{P}(d\omega_{n+1})$ yields:

$$\begin{aligned} G_{n+1}(\mathcal{Y}) &= \int \min_{x\in\mathcal{Y}} \frac{1}{n+1}\sum_{i=1}^{n+1} f(x,\omega_i)\mathbb{P}(d\omega_1)\ldots\mathbb{P}(d\omega_{n+1}) \\ &\ge \int \frac{1}{n+1}\sum_{i=1}^{n+1}\min_{x\in\mathcal{Y}}\frac{1}{n}\sum_{j\le n+1, j\ne i} f(x,\omega_j)\mathbb{P}(d\omega_1)\ldots\mathbb{P}(d\omega_{n+1}) \\ &= \frac{1}{n+1}\sum_{i=1}^{n+1}\int \min_{x\in\mathcal{Y}}\frac{1}{n}\sum_{j\ne i, j\le n+1} f(x,\omega_j)\mathbb{P}(d\omega_1)\ldots\mathbb{P}(d\omega_{n+1}) \\ &= \frac{n+1}{n+1}G_n(\mathcal{Y}) = G_n(\mathcal{Y}). \end{aligned} \tag{6}$$

(ii) By Lemma $A1$ in [15], p. 67, we know that

$$\frac{1}{n}\sum_{i=1}^{n} f(x,\omega_i) \to \int f(x,\omega)\mathbb{P}(d\omega)$$

a.s. uniformly on $\mathcal{Y}$, and therefore

$$\min_{x\in\mathcal{Y}} \frac{1}{n}\sum_{i=1}^{n} f(x,\omega_i) \;\to\; \min_{x\in\mathcal{Y}} \int f(x,\omega)\,\mathbb{P}(d\omega) \;\text{ a.s.}$$

By Lebesgue's theorem it follows that

$$\begin{aligned} G_n(\mathcal{Y}) &= \mathrm{E}\left(\min_{x\in\mathcal{Y}}\frac{1}{n}\sum_{i=1}^{n} f(x,\omega_i)\right) \;\to\; \mathrm{E}\left(\min_{x\in\mathcal{Y}}\int f(x,\omega)\mathbb{P}(d\omega)\right) \\ &= \mathrm{E}(F^*(\mathcal{Y})) = F^*(\mathcal{Y}). \end{aligned}$$

(iii) From (6), it follows that $G_{n+1}(\mathcal{Y}) = G_n(\mathcal{Y})$ entails

$$\begin{aligned}
&\int \min_{x\in\mathcal{Y}} \sum_{i=1}^{n+1} \frac{1}{n} \sum_{j\leq n+1, j\neq i} f(x,\omega_j)\mathbb{P}(d\omega_1)\ldots\mathbb{P}(d\omega_{n+1}) \\
=\ &\int \sum_{i=1}^{n+1} \min_{x\in\mathcal{Y}} \frac{1}{n} \sum_{j\neq i, j\leq n+1} f(x,\omega_j)\mathbb{P}(d\omega_1)\ldots\mathbb{P}(d\omega_{n+1}).
\end{aligned}$$

Since the integrand of the first integral is larger than or equal to the integrand of the second integral, this further implies that

$$\min_{x\in\mathcal{Y}} \sum_{i=1}^{n+1} \sum_{j\leq n+1, j\neq i} f(x,\omega_j) = \sum_{i=1}^{n+1} \min_{x\in\mathcal{Y}} \sum_{j\leq n+1, j\neq i} f(x,\omega_j) \ \text{ a.s.}$$

Let x^* be a value x for which the minimum on the l.h.s. occurs, i.e.,

$$\sum_{i=1}^{n+1} \sum_{j\leq n+1, j\neq i} f(x^*,\omega_j) \ = \ \sum_{i=1}^{n+1} \min_{x\in\mathcal{Y}} \sum_{j\leq n+1, j\neq i} f(x,\omega_j) \ \text{ a.s.}$$

Integration over $\mathbb{P}(d\omega_1)\ldots\mathbb{P}(d\omega_{n+1})$ yields on the l.h.s.

$$(n+1)n \int f(x^*,\omega_j)\mathbb{P}(d\omega) = (n+1)n \cdot \mathbf{E}(f(x^*,\omega)),$$

and on the r.h.s.

$$\sum_{i=1}^{n+1} \int \min_{x\in\mathcal{Y}} \sum_{j=1}^{n} f(x,\omega_j)\mathbb{P}(d\omega_1)\ldots\mathbb{P}(d\omega_n) = (n+1)n \cdot G_n(\mathcal{Y}).$$

In total,

$$G_n(\mathcal{Y}) = \mathbf{E}(f(x^*,\omega)) \ \geq \ \min_{x\in\mathcal{Y}} \mathbf{E}(f(x,\omega)) = F^*(\mathcal{Y}),$$

which yields, together with (i), the assertion.

(iv) Since the minimum of linear functions is concave, the function mapping α to

$$\min_{x\in\mathcal{Y}} \sum_{i=1}^{n} \alpha_i f(x,\omega_i)$$

is (for fixed $\omega_1,\ldots,\omega_n$) a concave function, mapping the $(n-1)$ - dimensional simplex into $I\!R$. Because of the linearity and monotonicity of the integral operator, this entails the concavity of $G_n^{(\alpha)}(\mathcal{Y})$ as a function of α. Furthermore, $G_n^{(\alpha)}(\mathcal{Y})$ is symmetric in the variables $\alpha_1,\ldots,\alpha_n$. A symmetric, concave function on a simplex takes its maximum in the center of the simplex. Hence the assertion follows. □

In view of Theorem 4.1, it seems now promising to work rather with $G_n(\mathcal{Y})$ for some $n > 1$ as a lower bound than with $G_1(\mathcal{Y}) = L(\mathcal{Y})$. Of course, n must not be too large, otherwise the determination of

$$\min_{x\in\mathcal{Y}} \left(\frac{1}{n}\sum_{i=1}^{n} f(x,\omega_i)\right) \tag{7}$$

would be too time–consuming. So a compromise between the quality of the lower bound and the computational costs of its determination has to be found. This can be done experimentally, as it will be shown in section 6. After an appropriate value of n has been chosen, a lower bound estimate for $F(x)$, $x \in \mathcal{X}^s$, is determined as follows (cf. (5)): Draw independent sample values $\omega_{\nu i}$ for $\nu = 1, \ldots, N$ and $i = 1, \ldots, n$ according to $\mathbb{P}$ and compute

$$\frac{1}{N}\sum_{\nu=1}^{N} \min_{x\in\mathcal{X}^s} \left(\frac{1}{n}\sum_{i=1}^{n} f(x,\omega_{\nu i})\right). \tag{8}$$

This yields the required lower bound estimate, where, again, the variance of the estimate can be decreased by increasing the value of N.

5 Safe Jobs First

In sequencing, it is always of advantage to obtain *precedence constraints* for certain jobs ([7], ch. 4): If, for example, in the deterministic SMTTP, job 1 has a smaller processing time and an earlier due date than job 2, then there exists an optimal solution where job 1 is arranged before job 2 . So the precedence constraint $x^{-1}(1) < x^{-1}(2)$ to schedule job 1 before job 2 can be imposed on the solution x. In a complete enumeration approach, this reduces the number of permutations x to be investigated from $k!$ to $k!/2$. Also in a dynamic programming approach, the runtime can be reduced considerably by exploiting the precedence relation ([7], section 6.4). In our context of stochastic branch-and-bound, a precedence relation of this kind has the *additional* positive effect that certain branches of the tree can be truncated *a priori*, e.g. the branch $\mathcal{X}_{2,1}$, if $x^{-1}(1) < x^{-1}(2)$. This spares a considerable amount of computation time, since even if the branch $\mathcal{X}_{2,1}$ would never be partitioned again using the standard procedure, it would nevertheless have to be re–evaluated periodically. Therefore, we are very interested in obtaining a partial order of precedence relations also for the stochastic version of the SMTTP, which is not easy since stochastic processing times drastically complicate the situation.

Fortunately, however, for the SSMTTP, precedence relations of the mentioned type may be derived. They rely on the basic observation that, *ceteris paribus*, “safe” jobs, i.e., jobs with rather concentrated processing time distributions, are to be sequenced before “unsafe” jobs, i.e., jobs whose processing times are uncertain to a high degree. Although this principle is commonly used in everyday life, its validity is not easy to prove. So our formal results on the mentioned principle seem to be of interest in their own right.

Let us start with a definition of what we mean by a "more concentrated" random variable respectively distribution. It is convenient to use the well-known *Lorenz concentration curve* (see, e.g., [10], p. 48) as a concentration measure. In particular, for two random variables $Y_1 \geq 0$ and $Y_2 \geq 0$ with $\mathrm{E}(Y_1) = \mathrm{E}(Y_2)$, we shall consider Y_1 as more concentrated than Y_2 if the Lorenz curve $L_{Y_1}(p)$ of Y_1 is above the Lorenz curve $L_{Y_2}(p)$ of Y_2, i.e., if $L_{Y_1}(p) \geq L_{Y_2}(p)$. Since also random variables Y_1, Y_2 with different expectations have to be compared, and since $L_Y(p)$ is only defined for nonnegative Y, this definition has to be extended as follows:

Definition 5.1. $Y_1 \geq 0$ is called *more concentrated* than $Y_2 \geq 0$, if for some $c > 0$,

$$L_{Y_1 - E(Y_1) + c}(p) \geq L_{Y_2 - E(Y_2) + c}(p) \quad \text{for all}\, p. \tag{9}$$

Recall that

$$L_Y(p) = \frac{\int\limits_0^{F^{-1}(p)} x f(x) dx}{\int\limits_0^{\infty} x f(x) dx},$$

where f and F are the density function and the distribution function of Y, respectively.

The following lemma shows that it does not matter which constant c is used in Definition 5.1:

Lemma 5.1. If Y_1 is more concentrated than Y_2, then (9) holds for *all* $c > 0$.

Proof. Let (9) hold for a special $c > 0$. Consider $c' \neq c, c' > 0$. By elementary calculations, one verifies that

$$L_{Y+b}(p) = \frac{E(Y) L_Y(p) + bp}{E(Y) + b}. \tag{10}$$

Therefore,

$$\begin{aligned}
L_{Y_1 - E(Y_1) + c'}(p) &= L_{(Y_1 - E(Y_1) + c) + (c' - c)}(p) \\
&= (1/c') \cdot [c L_{Y_1 - E(Y_1) + c}(p) + (c' - c)p] \\
&\geq (1/c') \cdot [c L_{Y_2 - E(Y_2) + c}(p) + (c' - c)p] \\
&= L_{(Y_2 - E(Y_2) + c) + (c' - c)p}(p) = L_{Y_2 - E(Y_2) + c'}(p).
\end{aligned}$$

□

In particular, this implies that "more concentrated" induces a partial order on the set of distributions of nonnegative random variables.

Special Case:

An important special case is the following: Assume that the distributions of two random variables Y_1 and Y_2 have the same expectation and the same "shape", except that the distribution of Y_2 is "broader" than that of Y_1. To be more precise: Let

$$Y_1 = a + Z, \quad Y_2 = a + bZ$$

with $a > 0, b > 1$ and $\mathbf{E}(Z) = 0$. Then by short computation, it is easy to verify that Y_1 is more concentrated (in the sense of Definition 5.1) than Y_2.

Lemma 5.2. Let Y_1 and Y_2 be nonnegative random variables with Riemann-integrable densities and compact supports, and assume that $\mathrm{E}(Y_1) \leq \mathrm{E}(Y_2)$ and that Y_1 is more concentrated than Y_2. Then

$$\mathrm{E}(\psi(Y_1)) \leq \mathbf{E}(\psi(Y_2))$$

for each continuous nondecreasing convex function ψ.

Proof. We start with the following observation: For random variables X_1, X_2 with Riemann-integrable densities and compact supports, $L_{X_1}(p) \geq L_{X_2}(p)$ if and only if $\mathrm{E}_{X_1}(\varphi) \leq \mathrm{E}_{X_2}(\varphi)$ for each continuous convex function φ. This is a consequence of the fact that the distributions of X_1 and X_2 can be approximated uniformly (to each desired degree of accuracy) by distributions μ, ν of the following type: μ and ν are discrete distributions with point masses of size $1/n$ in points x_i and p_i $(i = 1, \ldots, n)$, respectively. For distributions μ, ν of the described type, it follows from [11], pp. 5 and $16 - 17$, that $L_\mu(p) \geq L_\nu(p)$ if and only if $\int \varphi d\mu \leq \int \varphi d\nu$ for each continuous convex function φ, where $L_\mu(p)$ denotes the Lorenz curve $L_X(p)$ for some X distributed according to μ. The observation is obtained by letting $n \to \infty$. Now assume that ψ satisfies the conditions of the Lemma. Because of the monotonicity of ψ,

$$\mathrm{E}(\psi(Y_1)) \leq \mathbf{E}(\psi(Y_1 - \mathrm{E}(Y_1) + \mathrm{E}(Y_2))). \tag{11}$$

Setting $c := \mathrm{E}(Y_2)$, Lemma 5.1 implies that

$$L_{Y_1 - E(Y_1) + E(Y_2)}(p) \geq L_{Y_2}(p)$$

for all p. Because ψ is continuous and convex, the observation at the beginning of the proof, applied to $X_1 := Y_1 - \mathbf{E}(Y_1) + \mathrm{E}(Y_2)$ and $X_2 := Y_2$, finally yields

$$\mathrm{E}(\psi(Y_1 - \mathbf{E}(Y_1) + \mathrm{E}(Y_2))) \leq \mathrm{E}(\psi(Y_2)).$$

Combination with (11) proves the assertion. $\square$

Now we are in the position to demonstrate our result:

Theorem 5.1. If job 1 has smaller or equal expected processing time, a more concentrated processing time distribution, an earlier or equal due date and higher or equal weight, then it can be assumed to precede job 2 in an optimal arrangement. More precisely: Suppose that for the processing times P_1, P_2 with Riemann-integrable densities and compact supports, $\mathrm{E}(P_1) \leq \mathrm{E}(P_2)$, and P_1 is more concentrated than P_2. Moreover, suppose $d_1 \leq d_2$ and $\gamma_1 \geq \gamma_2$. Then for two arrangements x and x' of the jobs $1, \ldots, k$ with

$$x(j_1) := 1,\ x(j_2) := 2,\ x'(j_1) := 2,\ x'(j_2) := 1,\ x(j) := x'(j)\ (j \neq 1, 2),$$

where $j_1 < j_2$, the relation $F(x) \leq F(x')$ holds.

Proof. First, we assume that the processing times of the jobs $3, 4, \ldots, k$ are deterministic, i. e., P_i is fixed for $i \geq 3$. Set

$$\begin{aligned}
A_j &:= P_{x(1)} + \ldots + P_{x(j)} && (1 \leq j \leq j_1 - 1),\\
B_j &:= P_{x(j_1+1)} + \ldots + P_{x(j)} && (j_1 + 1 \leq j \leq j_2 - 1),\\
C_j &:= P_{x(j_2+1)} + \ldots + P_{x(j)} && (j_2 + 1 \leq j \leq k).
\end{aligned}$$

The numbers A_j, B_j and C_j only depend on P_i with $i \neq 1, 2$, so they are fixed. Moreover, the analogously defined numbers A'_j, B'_j and C'_j for x' are identical to A_j, B_j and C_j, respectively.

We obtain

$$\begin{aligned}
f(x, \omega) = {} & \sum_{j=1}^{j_1-1} \gamma_{x(j)}[A_j - d_{x(j)}]^+ \\
& + \gamma_1[A_{j_1-1} + P_1 - d_1]^+ \\
& + \sum_{j=j_1+1}^{j_2-1} \gamma_{x(j)}[A_{j_1-1} + P_1 + B_j - d_{x(j)}]^+ \\
& + \gamma_2[A_{j_1-1} + P_1 + B_{j_2-1} + P_2 - d_2]^+ \\
& + \sum_{j=j_2+1}^{k} \gamma_{x(j)}[A_{j_1-1} + P_1 + B_{j_2-1} + P_2 - d_{x(j)} + C_j]^+.
\end{aligned}$$

Computation of $f(x', \omega)$ and comparison yields:

$$\begin{aligned}
f(x, \omega) - f(x', \omega) = {} & \gamma_1[A_{j_1-1} + P_1 - d_1]^+ - \gamma_2[A_{j_1-1} + P_2 - d_2]^+ \\
& + \sum_{j=j_1+1}^{j_2-1} \gamma_{x(j)}\{[A_{j_1-1} + P_1 + B_j - d_{x(j)}]^+ - [A_{j_1-1} + P_2 + B_j - d_{x(j)}]^+\} \\
& + \gamma_2[A_{j_1-1} + P_1 + B_{j_2-1} + P_2 - d_2]^+ - \gamma_1[A_{j_1-1} + P_1 + B_{j_2-1} + P_2 - d_1]^+.
\end{aligned}$$

The components of this expression are now estimated separately:

(a) Estimation of $\{\ldots\}$: For fixed j ($j_1 + 1 \leq j \leq j_2 - 1$), let

$$\psi(y) := [y + A_{j_1-1} + B_j - d_{x(j)}]^+.$$

By taking the expectation with respect to the joint distribution of P_1 and P_2, and by applying Lemma 5.2, we obtain

$$\mathrm{E}_{(P_1,P_2)}(\{\ldots\}) = \mathrm{E}_{(P_1,P_2)}(\psi(P_1) - \psi(P_2)) \leq 0.$$

(b) Estimation of the remaining terms: For abbreviation, set

$$a := A_{j_1-1}, \quad b := A_{j_1-1} + B_{j_2-1},$$

$$R := \gamma_1(a + P_1 - d_1)^+ - \gamma_2(a + P_2 - d_2)^+ + \gamma_2(b + P_1 + P_2 - d_2)^+ - \gamma_1(b + P_1 + P_2 - d_1)^+.$$

Because of $a \leq b$ and $P_1, P_2 \geq 0$, one immediately gets

$$R \leq (\gamma_1 - \gamma_2)\,[(a + P_1 - d_1)^+ - (b + P_1 + P_2 - d_1)^+] + \gamma_2 S \;\leq\; \gamma_2 S$$

with

$$S := (a + P_1 - d_1)^+ - (a + P_2 - d_2)^+ + (b + P_1 + P_2 - d_2)^+ - (b + P_1 + P_2 - d_1)^+,$$

since $\gamma_1 - \gamma_2 \geq 0$ and

$$(a + P_1 - d_1)^+ - (b + P_1 + P_2 - d_1)^+ \leq 0.$$

Because $a + P_1 - d \leq b + P_1 + P_2 - d$ for all d, $(a + P_1 - d)^+ - (b + P_1 + P_2 - d)^+$ is nondecreasing in d, both terms are zero, or both terms are positive (then the difference does not depend on d), or only the second term is positive (then the difference has slope 1 in d). Therefore

$$(a + P_1 - d_1)^+ - (b + P_1 + P_2 - d_1)^+ \leq (a + P_1 - d_2)^+ - (b + P_1 + P_2 - d_2)^+,$$

and hence

$$S \leq (a + P_1 - d_2)^+ - (a + P_2 - d_2)^+ + (b + P_1 + P_2 - d_2)^+ - (b + P_1 + P_2 - d_2)^+$$

$$= (a + P_1 - d_2)^+ - (a + P_2 - d_2)^+ = \overline{\psi}(P_1) - \overline{\psi}(P_2)$$

with $\overline{\psi}(y) := (y + a - d_2)^+$. Application of Lemma 5.2 yields $\mathrm{E}_{(P_1,P_2)}(S) \leq 0$ and therefore also

$$\mathrm{E}_{(P_1,P_2)}(R) \leq 0.$$

In total, we have shown $\mathrm{E}_{(P_1,P_2)}(f(x,\omega)) \leq \mathrm{E}_{(P_1,P_2)}(f(x',\omega))$. The assertion follows by taking expectation with respect to the remaining variables $P_3, P_4, \ldots, P_k$. $\square$

Corollary ("Safe jobs first"). Suppose that (for processing times with densities as in Theorem 5.1), $\mathrm{E}(P_1) = \mathrm{E}(P_2), d_1 = d_2, \gamma_1 = \gamma_2$, and P_1 is more concentrated than P_2. Then it is equally favorable or better to schedule job 1 before job 2 than vice versa.

It is difficult to estimate the performance gain for our algorithm achievable by exploiting Theorem 5.1. On the one hand, four conditions need to be simultaneously satisfied for a precedence between two jobs to be derived. In a simple probabilistic model assuming independence between the properties of the two considered jobs, this event has only a probability of 1/16. On the other hand, if k jobs are to be scheduled, there exist $k(k-1)/2$ pairs of jobs, so, except for very small k, it is likely that at least *one* precedence constraint is obtained. Nevertheless, we admit that, for the moment, a general assertion on the performance gain cannot be made. Even a computational comparison would be of little use, since the result would strongly depend on the chosen input model for the properties of the jobs, and, in absence of empirical data, it could always be questioned whether a specific input model is adequate for real applications. So we prefer to consider Theorem 5.1 rather as a theoretical result.

6 Experimental Results

Before reporting our experimental results, let us briefly mention some details of our C++ implementation of the stochastic branch-and-bound algorithm for the SSMTTP.

First, it is obvious that a realization of the whole branch-and-bound tree (cf. Fig. 1) as a data structure is not necessary. It suffices to maintain a list of the current leaves of this tree.

Secondly, for determining $\min_{x \in \mathcal{X}^s} f(x, \omega)$ (in order to obtain a lower bound estimate), two methods have been applied:

- Complete enumeration (CE), for relatively small k.
- Dynamic Programming (DP) according to the approach in Schrage and Baker [16].

Let us briefly outline the Schrage–Baker approach: If J is a subset of the set of jobs $\{1, \ldots, k\}$, $c(J)$ is the sum of processing times of jobs in J, and $g(j, s)$ is the cost contribution of job j, provided that job j is finished at time s, then the cost $f(J)$ of a minimum–cost sequence of jobs in J can be computed by the recursion

$$f(J) = \min \{ f(J \setminus \{j\}) + g(j, c(J)) \mid k \in J \}.$$

Schrage and Baker supply this formula with a simple procedure for enumerating the subsets of J in such a way that $J \setminus \{j\}$ is enumerated before J. Roughly speaking, the effect of this technique is a reduction of the computational effort from order $k! \cdot$ poly(k) (complete enumeration) to order $2^k \cdot$ poly(k). In the presence of precedence constraints, the algorithm is appropriately modified, resulting in a further reduction of the runtime (cf. the remarks at the beginning of section 5).

Unfortunately, there is no obvious way of combining the DP-algorithm by Schrage and Baker with the *inner sampling* approach described in section 4: The determination of (7) does not lend itself to the application of DP. Therefore, when using inner sampling, we combine it with complete enumeration.

Thirdly, lower bound estimations satisfying the convergence conditions of Theorem 2.1 could be performed in the following (straightforward) way: At iteration step r of the stochastic branch-and-bound algorithm, take, for each $\mathcal{X}^s \in \mathcal{P}_{r+1}$, an independent sample of size $N = r+c$ (c constant) and compute $\xi_r(\mathcal{X}^s)$ according to (8). We call this procedure *estimation without inheritance*. However, if $\mathcal{X}^s$ has not been partitioned in this iteration step, it already has an assigned lower bound estimate $\xi_{r-1}(\mathcal{X}^s)$, and it would possibly be a waste to omit the information contained in this (previous) estimate. So it seems better to compute $\xi_r(\mathcal{X}^s)$ as a *weighted average* of $\xi_{r-1}(\mathcal{X}^s)$ and the new estimate, where the weights have to be chosen proportional to the respective sample sizes used in each partial estimate. This procedure is called *estimation with inheritance at old nodes*. We also experimented with a third variant, where even for newly generated nodes $\mathcal{Y}_i^r$, the estimate $\xi_r(\mathcal{X}^s)$ is computed as a weighted average of a new estimate and the value $\xi_{r-1}(\mathcal{Y}^r)$ assigned to the predecessor node. Although this third variant, called *estimation with inheritance at all nodes*, does not satisfy the conditions of Theorem 2.1, it worked fairly well in our experiments.

Fourthly, the stochastic branch-and-bound algorithm uses the *upper bounds* only for the determination of an approximate solution. If the algorithm is to be terminated after a fixed number of, say, R iteration steps, it does not make sense to compute approximate solutions at each intermediate iteration step $r = 0, \ldots, R-1$. It suffices to defer the computation of the upper bounds and of the approximate solution to the last iteration step. Of course, the accuracy of the estimate $\eta_R(\mathcal{X}^s)$ has then to be chosen sufficiently high, otherwise the benefit of the previous iteration runs cannot be exploited. In our experiments, the upper bound estimates were determined according to the technique described at the end of section 3, with the exception that we solved the resulting deterministic SMTTPs *exactly* by CE or DP. We observed, however, that the lowest–upper–bound subsets $\mathcal{X}^s$ in the last iteration step were usually already singletons, so it seems to make little difference on the solution quality whether the deterministic SMTTPs for upper bound estimation are solved exactly or heuristically.

6.1 Example

The way we performed our experiments is illustrated using the following concrete example:

Problem 1

Job Nr.	1	2	3	4	5	6	7	8	9	10
Due dates	1100	1500	1600	500	700	1400	200	1200	1300	100
Processing times	100	100	100	U[200,400]	U[100,300]	100	200	100	100	100
Weights	90	50	40	20	10	60	50	80	70	100

Here $U[a,b]$ denotes the uniform distribution in the interval $[a,b]$.

Applied method: Stochastic branch-and-bound with

- Dynamic Programming (DP) according to [16],
- lower bound estimation with inheritance at all nodes,
- sample size $N = r$ in iteration step r,
- 150 iteration steps.

In this example, the optimal solution x can be directly determined (in a tedious, but straightforward way) by computing the convolutions of the involved processing time distributions and comparing the expected tardiness values for different permutations x. (Many permutations can be excluded as suboptimal relatively soon, so not all 10! possible permutations have to be evaluated explicitely.) The best solution turns out to be the permutation

$$x = (x(1), \ldots, x(10)) = (7, 4, 5, 10, 1, 8, 7, 6, 2, 3).$$

It should be emphasized that replacing the processing times by their expected values and solving the corresponding *deterministic* SMTTP leads to *another* solution, namely

$$x = (x(1), \ldots, x(10)) = (7, 4, 5, 10, 9, 8, 1, 6, 3, 2).$$

Our program was able to produce the optimal solution, using a runtime of 259.29 sec on a PC 486 DX 2/66 (including 8.14 sec for the computation of the upper bounds). In fact, the final shape of the branch-and-bound tree was already reached at iteration step $r = 46$ and remained stable during the 104 remaining iteration steps, such that using only a third of the invested computing time would have produced the same result.

Working with different examples of similar type, we varied the involved parameters: iteration number, sample sizes, bound estimation method etc. In particular, we were interested in obtaining answers to two special questions:

- Does *inner sampling* (see section 4) pay, and if it does, which (inner) sample number n is advantageous?

- Should one do estimation without inheritance, with inheritance at old nodes, or with inheritance at all nodes?

Before dealing with these questions, let us remark that we often encountered a situation where the optimal solution of the SSMTTP was not identical with that of the deterministic problem obtained by replacing each P_i by $\mathrm{E}(P_i)$. This seems to be a strong justification for our approach. Expecially for strongly varying job weights (priorities), the optimal solution of the deterministic problem may be a very poor solution of the corresponding SSMTTP.

6.2 The Effect of Inner Sampling

In order to judge the effect of the *inner sampling* modification described in section 4, one should keep in mind that (as mentioned before) inner sampling can only be combined with determining the minimum in (7) by *complete enumeration* (CE). Although the number of permutations to be enumerated gets smaller in nodes generated later in the branch-and-bound tree, where several jobs have already been fixed (cf. Fig. 1), the combinatorial explosion of the number of permutations makes the application of CE for k larger than about 12 infeasible. So we restricted the application of inner sampling to experiments with relatively small k. Since DP turned out to be consistently faster than CE (expect in the rather trivial case $k = 3$), a fair comparison of "inner sampling" vs. "no inner sampling" should apply DP in case of "no inner sampling", where it *can* be applied. As a general trend, our experiments showed that inner sampling is advantageous for "hard" scheduling problems, i.e., problems with due dates that are hard to satisfy and/or processing times with large variances. On the other hand, in "easy" scheduling problems with rather loose due dates and processing times with small variances, inner sampling turned out to be inferior to the DP-based approach for obtaining lower bounds. This is illustrated quantitatively by the following two problem instances:

Problem 2 ("hard" problem):

Job Nr.	1	2	3	4	5	6
Due dates	34.7	44.6	48.4	42.8	52.2	46.9
Processing times	U[16,17]	U[15,16]	U[20,24]	U[19,35]	U[1,2]	U[9,45]
Weights	6	1	10	10	3	13

To evaluate the results of our program with and without inner sampling, we performed, for each parameter constellation, 40 runs of the program, each with 150 seconds of CPU time.

The optimal solution of this problem instance is $(1, 6, 5, 3, 4, 2)$ with an expected tardiness of 793.7. We counted the number of "hits", i.e., of cases where the stochastic branch-and-bound approach found the exact optimum. The maximum of 150 sec CPU time has been chosen in such a way that the number of "hits" and the number of "nonhits" were of the same order of magnitude. Fig. 2 shows the results. Therein, the following abbreviations have been used:

DPs ... Dynamic Programming with sample size $r + s$ at iteration step r $(s = 1, 500)$

CEi ... Complete enumeration with inner sample size i $(i = 1, 2, 3, 10, 30, 50)$

In all program runs, lower bounds have been estimated by "inheritance at old nodes". It can be observed that inner sampling with an inner sample size of $10 - 30$ increases the number of hits by 30 – 40 %, compared to the variant without inner sampling.

Problem 3 ("easy" problem):

Job Nr.	1	2	3	4	5	6
Due dates	100	200	80	20	300	250
Processing times	U[50,55]	U[100,105]	U[20,25]	U[1,3]	U[150,160]	U[80,90]
Weights	1	2	4	4	6	5

For each parameter constellation, 20 runs of the program were performed, each with 20 seconds of CPU time. (Again, the CPU time budget has been tuned in such a way that "hits" and "nonhits" have occured in a comparable frequency. Compare with the "hard" Problem 2, where about 150 seconds of CPU time where required to obtain hits in about 50% of cases.) The results are presented in Fig. 3. Obviously, inner sampling is no longer advantageous in this problem instance.

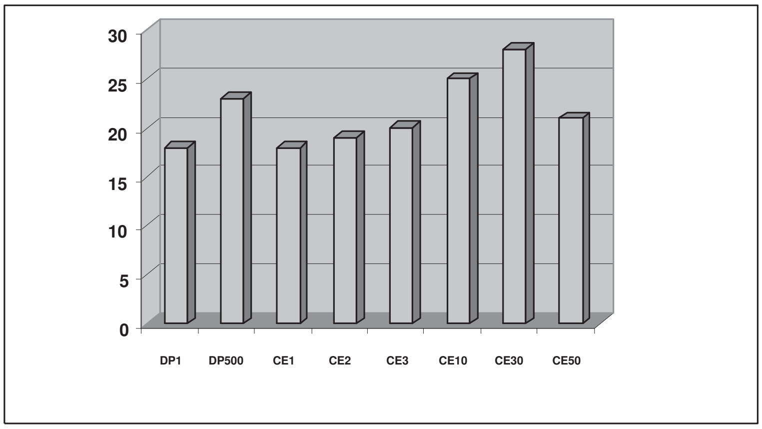

Fig. 2. Number of hits in problems instance 1 of section 6.2., CPU time of 150 sec per program execution.

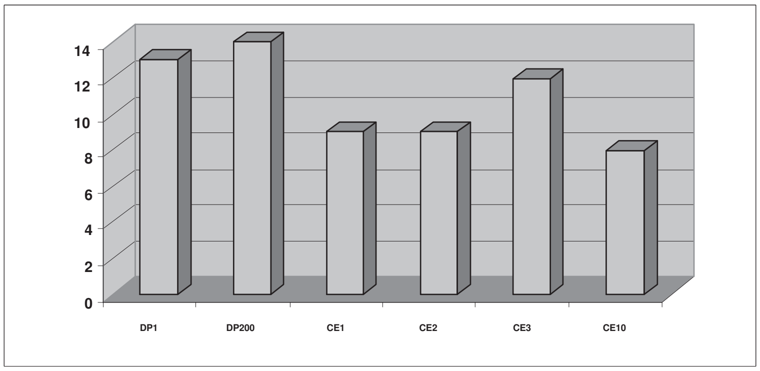

Fig. 3. Number of hits in problem instance 2 of section 6.2, CPU time 20 sec per program execution.

6.3 The Effect of Inheritance

Now let us study the influence of the three different ways of estimating the lower bound (described at the beginning of this section) on the solution quality. As expected, "estimation without inheritance" turned out as significantly and consistently worse than the two other variants. In our test instances, "estimation with inheritance at *all* nodes" turned to be slightly superior than "estimation with inheritance at *old* nodes". This is somewhat surprising, since the last-mentioned approach seems to be more sound from a statistical point of view: the lower-bound estimates remain always unbiased, while they get biased by the estimates at their predecessor nodes, if inheritance is applied at *all* nodes. Nevertheless, the advantage of exploiting a larger part of sampling information may possibly dominate the disadvantage of producing biased estimates. As a quantitative illustration, we report the results on Problem 2 (section 6.2). For each variant, 20 program runs were performed. The numbers of hits are shown in Table 1. As it can be seen, the difference between "inheritance at old nodes" and "inheritance at all nodes" is not significant in a statistical sense, whereas the respective differences to "estimation without inheritance" are highly significant.

7 Conclusion

In this article, we have presented a stochastic branch-and-bound technique for the solution of stochastic single-machine tardiness problems (SSMTTPs) with job weights.

Lower bound estimation approaches have been investigated in detail, both theoretically and experimentally. The theoretical results also hold for other applications of stochastic branch-and-bound to discrete stochastic optimization problems. Our computational experiments show that the algorithm works well for SSMTP instances with a small number of jobs; for a medium number of jobs, the approach can only be expected to yield approximate solutions. This is a consequence of the enormous complexity of the problem which is already NP-hard in its deterministic special case.

For a *large* number of jobs, our algorithm does not yield a solution within reasonable time limits, since the underlying deterministic SMTTP cannot be solved efficiently in view of its NP–hardness. Admittedly, this is a drawback of our approach. It might possibly be overcome by solving the deterministic subproblem only *heuristically*; in another application of stochastic branch–and–bound, this modification has been applied with good results (see [9]).

Furthermore, we derived a general precedence rule for the single-machine scheduling of jobs with uncertain processing times, essentially saying that *ceteris paribus* "safer" jobs are to be scheduled first. At present, it is not clear whether (in real applications) exploiting this rule improves the performance of our algorithm significantly. Maybe, our result can lead the way to the derivation of other precedence rules in the future. In any case, the result seems to be of interest in its own right.

inheritance	hits	non-hits
none	3	17
at old nodes	10	10
at all nodes	12	8

Table 1. Number of hits in Problem 2 (section 6.2), CPU time 150 sec per program execution.

References

[1] Birge, J.R., and Dempster, M.A.H, "Optimal match-up strategies in stochastic scheduling", *J. of Discr. Appl. Math.* 57 (1995), 105–120.

[2] Birge, J.R., and Dempster, M.A.H., "Stochastic programming approaches to stochastic scheduling", *J. of Global Optimization* 9 (1996), 417–451.

[3] Chang, Ch.-Sh., and Yao, D.Y., "Rearrangement, majorization and stochastic scheduling", *Mathematics of O.R.* 18 (1993), 658–684.

[4] Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G, and Shmoys, D.B., "Sequencing and Scheduling: Algorithms and Complexity", in: S.C. Graves, A.H.G. Rinnooy Kan and P.H. Zipkin (eds.), *Logistis of Production and Inventory*, Handbooks in Operations Research and Management, vol. 4, North–Holland (1993).

[5] Lin, Ch.-S., "Single machine stochastic scheduling: minimizing the number of tardy jobs", *Am. J. of Math. and Management Sci.* 14 (1994), 289–300.

[6] Du, J., and Leung, J.Y.T., "Minimizing total tardiness on one machine is NP-hard", *Mathematics of Operations Research* 15 (1990), 483–495.

[7] French, S., *Sequencing and Scheduling*, Horwood: Chichester (1982).

[8] Gutjahr, W.J., and Pflug, G.Ch., "Simulated annealing for noisy cost functions", *Journal of Global Optimization* 8 (1996), 1–13.

[9] Gutjahr, W.J., Strauss, C., and Wagner, E., "A stochastic branch–and–bound approach to activity crashing in project management", Technical Report, University of Vienna.

[10] Kendall, M. G. and Stuart, A., "Advanced Theory of Statistics". Griffin and Company, London (1958).

[11] Marshall, A., Olkin, I., *Inequalities: Theory of Majorization and its Applications*, New York: Academic Press (1979).

[12] Norkin, I., Ermoliev, Y.M., and Ruszczynski, A., "On optimal allocations of indivisibles under uncertainty", Technical Report WP-94-021, International Institute for Applied Systems Analysis, Laxenburg, Austria (1994).

[13] Panwalkar, S.S., Smith, M.L., and Koulamas, C.P., "A heuristic for the single machine tardiness problem", *EJOR* 70 (1993), 304 – 310.

[14] Pinedo, M., *Scheduling: Theory, Algorithms, and Systems*, Prentice Hall (1995).

[15] Rubinstein, R.Y., Shapiro, A., *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*, Wiley: New York (1993).

[16] Schrage, L.E., Baker, K.R., "Dynamic programming solutions of sequencing problems with precedence constraints", *Operations Research* 26 (1978), 444 – 449.