Possible optimality of solutions in a single machine scheduling problem with fuzzy parameters

Stefan Chanas
Inst. of Industrial Eng. and Man.
Wroclaw University of Technology
Smoluchowskiego 25, 50-370 Wroclaw,
POLAND
chanas@ioz.pwr.wroc.pl

Adam Kasperski
Inst. of Industrial Eng. and Man.
Wroclaw University of Technology
Smoluchowskiego 25, 50-370 Wroclaw,
POLAND
kaspersk@po.opole.pl

Abstract

A single machine scheduling problem with parameters given in the form of fuzzy numbers is considered. It is assumed that the optimal schedule in such a problem cannot be determined precisely (since the parameters are not known a priori). In the paper the concept of degree of possible optimality of a given schedule is introduced. An algorithm of calculating the degree of optimality in the general case is presented and then it is applied to two particular cases of the problem.

Keywords: Scheduling; Single machine; Fuzzy number; Degree of optimality

1 Basic notions on fuzzy numbers

Let us remind the most general definition of a fuzzy number:

Definition 1 *A fuzzy number $\tilde{A}$ is a normal*[1] *fuzzy set in the space of real numbers $\Re$ with the membership function $\mu_{\tilde{A}} : \Re \to [0,1]$, convex and upper semicontinuous on $\Re$.*

Definition 2 *The λ-cut of a fuzzy set $\tilde{A}$ is the set:*

$$A^{\lambda} = \{x|\mu_A(x) \geq \lambda\},\ \lambda \in (0,1]. \qquad (1)$$

[1] A fuzzy set $\tilde{A}$ is normal if and only if there exists $x \in \Re$ such that $\mu_{\tilde{A}}(x) = 1$.

It follows from the properties of the membership function of a fuzzy number $\tilde{A}$ that each its λ-cut A^{λ}, $\lambda \in (0,1]$, is a closed interval. We denote it by $A^{\lambda} = [\underline{a}(\lambda), \overline{a}(\lambda)]$.

Definition 3 *A fuzzy number $\tilde{A}$ is called interval number if its membership function is of the following form:*

$$\mu_{\tilde{A}}(x) = \begin{cases} 1 & for \quad x \in [\underline{a}, \overline{a}], \\ 0 & for \quad x \in (-\infty, \underline{a}) \cup (\overline{a}, \infty), \end{cases} \qquad (2)$$

where $-\infty < \underline{a} \leq \overline{a} < \infty$.

We denote the interval number as $\tilde{A} = [\underline{a}, \overline{a}]$. If $\tilde{A}$ is an interval number, then for each $\lambda \in (0,1]$ $\tilde{A}^{\lambda} = [\underline{a}, \overline{a}]$. Let us notice that a real number y is a special case of interval number with the membership function:

$$\mu_{\tilde{A}}(x) = \begin{cases} 1 & \text{for} \quad x = y, \\ 0 & \text{for} \quad x \neq y. \end{cases} \qquad (3)$$

2 The concept of possible optimality in a single machine scheduling problem

Consider a single machine scheduling problem S with the cost function $f(p)$. The value of the function $f(p)$ depends on the schedule p and on the set of real parameters $Z = \{z_1, ..., z_l\}$, such as processing times, due dates, release times etc. Thus, the cost function $f(p)$ can be also described as $f(p; z_1, ..., z_l)$. Assume now that all the parameters are given imprecisely by means of fuzzy numbers $\tilde{z}_1, ..., \tilde{z}_l$ with membership functions $\mu_{\tilde{z}_1}(x), ..., \mu_{\tilde{z}_l}(x)$. Such a problem is called a **fuzzy scheduling problem** S . In this case it is not obvious how to define the optimal schedule but it is natural to define the degree of optimality of given schedule p as the possibility degree that p is optimal. Assume that $\Gamma^S(p) \subseteq \Re^l$ is the set of all configurations of parameters $(z_1, ..., z_l) \in \Re^l$ for which p is optimal solution of the problem S, i.e. for each $(z_1, ..., z_l) \in \Gamma^S(p)$ we have

$$\forall_\pi \; f(p; z_1, ..., z_l) \leq f(\pi; z_1, ..., z_l).$$

The degree of optimality, $opt_S(p)$, of a given schedule p in fuzzy problem S is defined as follows:

$$opt_S(p) = \sup_{(z_1,...,z_l)\in\Gamma^S(p)} \min\{\mu_{\tilde{z}_1}(z_1), ..., \mu_{\tilde{z}_l}(z_l)\}.$$

If $\Gamma^S(p) = \emptyset$ then we assume that $opt_S(p) = 0$. Let us notice that $opt_S(p) \in [0, 1]$ for each schedule p.

The special case of the fuzzy problem S, in which all parameters are interval numbers $\tilde{z}_i = [\underline{z}_i, \overline{z}_i]$, $i = 1, ..., l$, is of great importance. The problem of calculating the value of $opt_S(p)$ for any schedule p, where all parameters are interval numbers is called the **interval problem** S and denoted by $i - S$. It follows from definition of the interval number (see Definition 3) that for each schedule p in the problem $i - S$ $opt_S(p) \in \{0, 1\}$. In each interval problem $opt_S(p) = 1$ if and only if there exists a configuration of parameters $(z_1, ..., z_l)$ such that $z_i \in [\underline{z}_i, \overline{z}_i]$ and $(z_1, ..., z_l) \in \Gamma^S(p)$. The interval scheduling problem is important since the following theorem holds:

Theorem 1 *The degree of optimality $opt_S(p)$ in fuzzy scheduling problem S is not less than given value of $\lambda \in (0, 1]$, i.e. $opt_S(p) \geq \lambda$, if and only if there exists a configuration of parameters $(z_1, ..., z_l) \in \Gamma^S(p)$ such that $z_i \in \tilde{z}_i^\lambda$ for $i = 1, ..., l$.*

Theorem 1 implies that in order to check whether $opt_S(p) \geq \lambda$ in the problem S with fuzzy parameters $Z = \{\tilde{z}_1, ..., \tilde{z}_l\}$ it is enough to solve the corresponding interval problem $i - S$ with the set of parameters $Z = \{\tilde{z}_1^\lambda, ..., \tilde{z}_l^\lambda\}$. We have used this property to construct Algorithm 1 that calculates the value of $opt_S(p)$ with a given computational accuracy ε.

Algorithm 1 Calculation of degree of optimality of p in fuzzy scheduling problem S

Require: Parameters of S, p, $\varepsilon \in (0, 1]$
Ensure: $opt_S(p)$

if p is infeasible **then**
  return 0 / * $opt_S(p) = 0$ * /
end if
if $opt_S(p) = 1$ in the problem $i - S$ for parameters $\tilde{z}_i^1$, $i = 1, ..., l$, **then**
  return 1 / * $opt_S(p) = 1$ * /
end if
$\lambda_1 \leftarrow 0.5$, $\lambda_2 \leftarrow 0$, $k \leftarrow 1$
while $|\lambda_1 - \lambda_2| \geq \varepsilon$ **do**
  $\lambda_2 \leftarrow \lambda_1$
  if $opt_S(p) = 1$ in the problem $i - S$ for parameters $\tilde{z}_i^{\lambda_1}$, $i = 1, ..., l$ **then**
    $\lambda_1 \leftarrow \lambda_1 + \frac{1}{2^{k+1}}$
  else
    $\lambda_1 \leftarrow \lambda_1 - \frac{1}{2^{k+1}}$
  end if
  $k \leftarrow k + 1$
end while
return λ_1 / * $opt_S(p) = \lambda_1$ * /

The computational complexity of Algorithm 1 is equal to $O(|\log\varepsilon| f(n))$, where $f(n)$ is the computational complexity of the corresponding interval problem $i - S$. It is assumed that $f(n)$ depends only on the number of jobs. The complexity of the problem $i - S$ depends on the problem S and the algorithm solving the interval problem must be constructed for every problem S separately. In the next sections we show how to solve sample interval scheduling problems being special cases of the interval problem $1|prec|f_{max}$. To specify the particular scheduling problems we apply the commonly used notation (see [1]).

3 Some new results on the conventional problem $1|prec|f_{max}$

Let us remind the formulation of the classical single machine scheduling problem $1|prec|f_{max}$. A set of jobs $J = \{1, ..., n\}$ to be processed on a single machine is given. It is assumed that preemption of jobs and idle machine times are not allowed. For each job $i \in J$ a positive, real processing time t_i is given. The machine can process only one job in a time and each job $i \in J$ must be processed by the time t_i. Thus each schedule can be represented by the sequence of jobs $p = (p(1), ..., p(n))$, where $p(i) \in J$ for $i = 1, ..., n$. Let us denote by p^i the set containing all jobs processed before i in the sequence p. The completion time $C_i(p)$ of job $i \in J$ in a given sequence p can be calculated as follows:

$$C_i(p) = \sum_{j \in p^i \cup \{i\}} t_j. \tag{4}$$

Assume that an acyclic and transitive precedence relation $prec \subset J \times J$ between jobs is given. A sequence p is **feasible** if and only if it fulfills the following condition:

$$(i,j) \in prec \Rightarrow i \in p^j. \tag{5}$$

Condition 5 means that if $(i,j) \in prec$ then the job j cannot be started before the job i is completed. Let us associate with each job $i \in J$ a cost function $f_i : \Re^+ \to \Re$, where $f_i(C_i(p))$ denotes the completion cost of i-th job in the sequence p. The problem consists in finding such a feasible sequence p for which the maximal completion cost, i.e. $\max_{i \in J}\{f_i(C_i(p))\}$, is minimal. If we assume that f_i, $i = 1, ..., n$, are non-decreasing functions then the problem can be easily solved by the known Lawlers's algorithm [3].

Let us introduce some additional notations on the considered problem [2].

Definition 4 *A job $k \in J$ is called* **critical** *in a sequence p if the following condition holds:*

$$f_k(C_k(p)) = \max_{i \in J}\{f_i(C_i(p))\}.$$

A job $k \in J$ is critical in p if its completion cost is the biggest among all jobs.

Definition 5 *A set $Succ_j(p)$ consists of these jobs $i \in p^j$, that can be moved after job j in p without violating the precedence relation prec.*

The following theorem holds [2]:

Theorem 2 *Suppose that all cost functions f_i, $i = 1, ..., n$, are strictly increasing and a sequence p is feasible. The sequence p is optimal if and only if there exists in p a critical job k such that for each $j \in Succ_k(p)$ the following condition holds:*

$$f_k(C_k(p)) \le f_j(C_k(p)).$$

We use Theorem 2 to construct efficient algorithms for several interval scheduling problems.

4 The problem $i - 1|prec|max\{w_i L_i\}$

The classical problem $1|prec|max\{w_i L_i\}$ is a special case of the problem $1|prec|f_{max}$ with the cost function of the following form:

$$f_1(p) = \max_{i \in J}\{w_i L_i(p)\},$$

where $L_i(p) = C_i(p) - d_i$ is the lateness of i-th job in schedule p. The set of parameters of the problem includes processing times $t_1, ..., t_n$, due dates $d_1, ..., d_n$ and weights $w_1, ..., w_n$. Consider now the corresponding interval problem $i - 1|prec|max\{w_i L_i\}$. We assume that processing times and due dates are interval numbers $[\underline{t}_i, \overline{t}_i]$ and $[\underline{d}_i, \overline{d}_i]$, respectively, $i = 1, ..., n$, and all weights are positive real numbers. In order to check whether $opt_S(p) = 1$ we must check if there exists configuration of parameters $t_i \in [\underline{t}_i, \overline{t}_i]$, $d_i \in [\underline{d}_i, \overline{d}_i]$, $i = 1, ..., n$, for which p is optimal. Since all cost functions $f_i(C_i(p)) = C_i(p) - d_i$ are strictly increasing it is possible to apply Theorem 2 to the considered problem.

Theorem 3 *The degree of optimality of a schedule p in the problem $i-1|prec|max\{w_i L_i\}$ is equal to 1 if and only if p is feasible and there exists $k \in J$ for which the following system of inequation is feasible:*

$$U_k(p) = \begin{cases} \underline{t}_i \le t_i \le \overline{t}_i & i = 1, ..., n, \\ \underline{d}_i \le d_i \le \overline{d}_i & i = 1, ..., n, \\ C_{p(i)} = t_{p(1)} + ... + t_{p(i)} & i = 1, ..., n, \\ w_k(C_k - d_k) \ge w_i(C_i - d_i) & i = 1, ..., n, \\ w_k(C_k - d_k) \le w_i(C_k - d_i) & i \in Succ_k(p). \end{cases}$$

Theorem 3 is a direct consequence of Theorem 2. In the first two lines of the system $U_k(p)$ we assure that all parameters belong to the proper intervals. In the third line we calculate the completion time of each job in p, i.e. if $p(i) = j$ then $C_{p(i)} = C_j(p)$.

In the last two lines we assure that job k is critical and check the sufficient and necessary condition from Theorem 2. It is clear that there exists a proper configuration of processing times and due dates for which p is optimal if and only if at least one system $U_k(p)$ is feasible.

Let us notice that for a given schedule (sequence of jobs) p all systems $U_k(p)$, $k = 1, ..., n$, are linear and their feasibility can be easily checked in polynomial time. In order to check whether $opt_S(p) = 1$ for a given, feasible schedule p it is enough to solve at most n system of linear inequations $U_k(p)$ for $k = 1, ..., n$ (see Algorithm 2).

Algorithm 2 Algorithm for the problem $i - 1|prec|max\{w_i L_i\}$

Require: $n, (w_i)_{i=1}^n, ([\underline{d}_i, \overline{d}_i])_{i=1}^n, ([\underline{t}_i, \overline{t}_i])_{i=1}^n, prec, p$
Ensure: $opt_S(p)$
for $k = 1$ **to** n **do**
  if The system $U^k(p)$ is feasible **then**
    return 1
  end if
end for
return 0 /* All $U^k(p)$ are infeasible */

Algorithm 2 together with Algorithm 1 allows to calculate effectively the degree of optimality of a given schedule p in the fuzzy problem $1|prec|max\{w_i L_i\}$.

5 The problem $i - 1|prec|max\{w_i T_i\}$

Consider the special case of the problem $1|prec|f_{max}$ in which the cost function takes the following form:

$$f_2(p) = \max_{i \in J}\{w_i T_i(p)\},$$

where $T_i(p) = \max\{0, C_i(p) - d_i\}$ is the tardiness of the i-th job in schedule p. The corresponding interval problem $i - 1|prec|max\{w_i T_i\}$ is defined similarly to the problem $i - 1|prec|max\{w_i L_i\}$ considered in the previous section so processing times and due dates are interval numbers and weights are positive real numbers. Both problems are closely connected since the following theorem holds:

Theorem 4 *If $f_2(p; \underline{t}_1, ..., \underline{t}_n, \overline{d}_1, ..., \overline{d}_n) > 0$ for a given, feasible sequence p, then the problem $i - 1|prec|max\{w_i T_i\}$ is equivalent to the problem $i - 1|prec|max\{w_i L_i\}$ with the same set of parameters.*

It is obvious that if $f_2(p; \underline{t}_1, ..., \underline{t}_n, \overline{d}_1, ..., \overline{d}_n) = 0$ then the schedule p is optimal for the configuration $(\underline{t}_1, ..., \underline{t}_n, \overline{d}_1, ..., \overline{d}_n)$ and $opt_S(p) = 1$. The algorithm for the considered problem is similar to Algorithm 2 and is presented as Algorithm 3.

Algorithm 3 Algorithm for the problem $i - 1|prec|max\{w_i T_i\}$

Require: $n, (w_i)_{i=1}^n, ([\underline{d}_i, \overline{d}_i])_{i=1}^n, ([\underline{t}_i, \overline{t}_i])_{i=1}^n, prec, p$
Ensure: $opt_S(p)$
if $f_2(p; \underline{t}_1, ..., \underline{t}_n, \overline{d}_1, ..., \overline{d}_n) = 0$ **then**
  return 1
end if
for $k = 1$ **to** n **do**
  if The system $U^k(p)$ is feasible **then**
    return 1
  end if
end for
return 0 /* All $U^k(p)$ are infeasible */

Algorithm 3 together with Algorithm 1 allows to calculate effectively the degree of optimality of a given schedule p in the fuzzy problem $1|prec|max\{w_i T_i\}$.

Acknowledgements

This work was supported by grant no. 7T11F02120 from the State Comitte for Scientific Research (Komitet Badań Naukowych).

References

[1] P. Brucker (1998). Scheduling Algorithms, Springer–Verlag, Berlin.

[2] S. Chanas, A. Kasperski (2001). Sensitivity analysis in the single machine sheduling problem with max-min criterion. Submitted to Journal of the Operational Research Society.

[3] E.L. Lawler (1973). Optimal sequencing of a single machine subject to precedence constraints, Management Science 19, 544–546.