Scheduling in stochastic bicriteria single machine systems with set-up times

H.M. Soroush

Department of Statistics and Operations Research,
Kuwait University,
P.O. Box 5969, Safat 13060, Kuwait
Fax: +965 2483 7332
E-mail: h.soroush@ku.edu.kw

Abstract: This paper deals with a single machine scheduling problem where job attributes are random variables, setup times are sequence-dependent, and a scheduler utilises a cost function to evaluate two criteria associated with a sequence. The objective is to determine the optimal sequence that minimises the scheduler’s expected cost. We show that problem scenarios wherein cost functions are linear, exponential, and fractional can be formulated as Quadratic Assignment Problems (QAPs). Also, special cases with sequence-independent setup times are shown to be solvable optimally in polynomial time. Our computational results on scenarios with sequence-dependent setup times demonstrate that good solutions can be approximated within reasonable amounts of time.

Keywords: scheduling; stochastic; bicriteria; single machine; set-up times; QAP; quadratic assignment problem.

Reference to this paper should be made as follows: Soroush, H.M. (2011) ‘Scheduling in stochastic bicriteria single machine systems with set-up times’, Int. J. Planning and Scheduling, Vol. 1, Nos. 1/2, pp.109–145.

Biographical notes: H.M. Soroush is an Associate Professor of Operations Research and Statistics at Kuwait University. He obtained his MS and PhD in Operations Research and Statistics from Rensselaer Polytechnic Institute, USA. He has published in Transportation Science, Networks, European Journal of Operational Research, Computers and Operations Research, European Journal of Industrial Engineering, International Journal of Production Economics, International Journal of Control, etc. He is a member of the Editorial Board of International Journal of Planning and Scheduling, and has also served as a Guest Editor on an EJOR feature issue. His research interests include modelling and optimisation of stochastic processes and networks, scheduling and transportation analysis.

1 Introduction

Scheduling is concerned with the allocation of scarce resources to perform a set of jobs (or tasks) over a period. In real-world scheduling systems, decision-makers require considering multiple criteria rather than a single criterion before arriving at some decisions. These criteria are often conflicting and no single schedule would
simultaneously optimise all criteria; thus, a schedule that is optimal for one criterion might be arbitrarily bad for other criteria (e.g., Hoogeveen, 2005; Sarin and Hariharan, 2000; Sarin and Prakash, 2004; T’Kindt and Billaut, 2001, 2002). Furthermore, in scheduling problems in both manufacturing and service environments, it is important to utilise resources efficiently. One of the ways to improve resource utilisation is to treat set-up times separately from processing times allowing operations to be performed simultaneously. This is, in particular, important in modern production systems such as just-in-time, optimised production technology, group technology, cellular manufacturing and time-based competition (e.g., Allahverdi et al., 2008; Allahverdi and Soroush, 2008; Ang et al., 2009; Angel-Bello et al., 2011). Scheduling problems are divided into the classes with sequence-independent and sequence-dependent set-up times. If the set-up time of a job depends solely on the job, regardless of its preceding job, it is called sequence-independent and is treated separately from the job’s processing time. However, if the set-up time depends on both the current job and its immediately preceding job, it is called sequence-dependent and is treated separately from the processing time. Another important issue in most real-life scheduling problems is the stochasticity of job attributes such as arrival times, processing times and set-up times since they are subject to random variability (e.g., Baker and Trietsch, 2009b; Soroush, 2010; Sotskov and Lai, 2012). It is important to incorporate variations of these attributes into scheduling decisions because schedulers encounter such deviations. The significance of research in stochastic scheduling is also emphasised by the interest in synchronous manufacturing or the theory of constraints, which recognises that variations in job attributes disrupt schedules (e.g., Umble and Srikanth, 1995).

Among the scheduling problems, single machine scheduling has received a great deal of attention. A majority of the literature on this problem is concerned with the deterministic and single criterion case in which set-up times are either sequence-independent or sequence-dependent (e.g., Allahverdi et al., 2008; Baker and Trietsch, 2009a; Conway et al., 2007; Soroush, 2011; Tasgetiren et al., 2009; Wang and Li, 2011). The corresponding research on the bicriteria case utilises linear cost functions for sequence evaluation (e.g., Angel et al., 2005; Dileepan and Sen, 1988; Eren and Guner, 2006; Hoogeveen, 2005; Schaller and Gupta, 2008; T’Kindt and Billaut, 2001, 2002), and can be divided into three classes based on their exact solution strategies. In the first class, an optimal sequence is obtained by minimising a linear composite objective function of the criteria (e.g., Hoogeveen and van de Velde, 2001; Mazdeh et al., 2011; Shabtay et al., 2010; Smith, 1956; Yedidsion et al., 2009). In the second class, an optimal sequence is derived by minimising a primary linear objective function with respect to (w.r.t.) one criterion subject to the constraint that a secondary linear objective function w.r.t. another criterion is attained for some specified value (e.g., Angel et al., 2005; Chen and Sheen, 2007; Erenay et al., 2010; Liu, 2010; Shabtay and Steiner, 2011; Wang and Wang, 2011). In the last class, owing to the presence of conflicting and equally important criteria, a set of Pareto-optimal, efficient, or non-dominated sequences is determined (e.g., Gawiejnowicz et al., 2006; Koksalan and Keha, 2003; Molaei et al., 2010; Nelson et al., 1986; Steiner and Stephenson, 2007; Tadei et al., 2002). In general, a Pareto-optimal sequence is such that it is not possible to find another sequence with a better value in at least one criterion without worsening the value of at least one other criterion.

A number of researchers have extended single machine scheduling with sequence-independent set-up times and single criterion to include stochastic job attributes.
(e.g., Baker and Trietsch, 2009b; Soroush and Alqallaf, 2009; Soroush, 2010; Sotskov and Lai, 2011; Wu and Zhou, 2008). There are also a few studies on the stochastic bicriteria case of the problem, which utilise linear cost functions for sequence evaluation allowing schedulers to minimise linear functions of the expected values of criteria. For example, Forst (1995) examines a stochastic bicriteria problem with the objective of minimising the expected total weighted tardiness plus the expected total weighted flow time. Soroush and Fredendall (1994) minimise the expected value of a linear function of job earliness and tardiness. Soroush (2006, 2007) considers the bicriteria problem of minimising the weighted sum of the expected value of the number of early and tardy jobs.

In this paper, we study a stochastic bicriteria single machine scheduling problem in which job attributes (e.g., processing times, set-up times and reliabilities) are random variables, set-up times and job reliabilities are sequence-dependent and a scheduler uses a cost (or disutility) function to evaluate two criteria associated with a sequence. The objective is to determine the optimal sequence that minimises the expected value of the scheduler’s cost function of both criteria. The cost function can represent the scheduler’s disutility characterising his or her risk-taking behaviour (risk averse, risk prone and risk neutral) in uncertain scheduling environments. We formulate scenarios of the problem with linear, exponential and fractional cost functions as QAPs that are solvable either exactly or approximately. The criteria considered here include the makespan, total absolute variations in completion times, total completion time, total waiting time and their weighted counterparts, and the total reliability of jobs in a sequence when a job’s reliability depends on either the job’s position or the job’s adjacent preceding job. We further show that the special cases with sequence-independent set-up times can be solved exactly in polynomial time. To the best of our knowledge, there exist no prior studies on stochastic bicriteria single machine scheduling with non-linear cost functions and either sequence-independent or sequence-dependent set-up times. (The use of non-linear cost functions in scheduling has been justified by Alidaee (1993), Baker and Scudder (1990), Soroush (2010) and Valente and Goncalves (2009)).

The organisation of the rest of this paper is as follows. In Section 2, the stochastic bicriteria single machine scheduling problem with set-up times is defined. In Sections 3–5, we formulate problem scenarios with linear, exponential and fractional cost functions as QAPs, and develop exact solution strategies when set-up times are either sequence-dependent or sequence-independent. Section 6 contains some computational results. Finally, a summary and some concluding remarks are given in Section 7.

2 Problem definition and formulation

Consider a set of jobs that are simultaneously available at time zero to be processed on a continuously available single machine in which no job pre-emption and idle time insertions are allowed. Job attributes (e.g., processing times, set-up times, reliabilities, or qualities) are independent random variables. A job sequence is evaluated using a cost (or disutility) function of two criteria associated with the sequence. The objective is to find the optimal sequence that minimises the expected value of the cost function of both criteria among all sequences.
We use the following notation to formulate the problem.

- $N = \{1, \ldots, n\}$: The set of $n$ jobs
- $\Psi$: The set of all $n!$ sequences
- $S = ([1], \ldots, [i], \ldots, [n]) \in \Psi$: A sequence where $[i], [i] = 1, \ldots, n$, is the job in the $i$th position, $i = 1, \ldots, n$, of $S$
- $p_{[i]}$: Random processing time of job $[i]$
- $b_{[i-1][i]}$: Random set-up time of job $[i]$, $[i] = 1, \ldots, n$, when scheduled immediately after job $[i-1] \neq [i] = 1, \ldots, n$, (i.e., the time to download job $[i-1]$ and upload job $[i]$) where $b_{[0][1]}$ is the random set-up time of job $[1], [1] = 1, \ldots, n$, when scheduled first in $S$
- $d_{[i]}$: Due date of job $[i]$
- $\gamma_{[i]}$: Weight or unit delay cost of job $[i]$
- $E[X]$: The expected value of random variable $X$
- $M_{X}(\cdot)$: Moment generating function (MGF) of $X$ evaluated at $\cdot$
- $\mu_{[i]} = E[p_{[i]}]$: Expected value of $p_{[i]}$
- $B_{[i-1][i]} = E[b_{[i-1][i]}]$: Expected value of $b_{[i-1][i]}$
- $t_{[i]} = \sum_{i=1}^{i} (p_{[i]} + b_{[i-1][i]})$: Completion time of job $[i]$
- $L_{[i]} = t_{[i]} - d_{[i]}$: Lateness of job $[i]$
- $C_{\text{max}} = \sum_{i=1}^{n} (p_{[i]} + b_{[i-1][i]})$: Makespan of jobs in $S$
- $TADC = \sum_{i=1}^{n} \sum_{j=i}^{n} |t_{j} - t_{i}| = \sum_{i=1}^{n} (i-1)(n-i+1)(p_{[i]} + b_{[i-1][i]})$: Total absolute variations in completion times of jobs in $S$
- $TWT = \sum_{i=1}^{n} \sum_{j=i}^{n} (p_{[i]} + b_{[i-1][i]}) = \sum_{i=1}^{n} (n-i)(p_{[i]} + b_{[i-1][i]})$: Total waiting time of jobs in $S$
- $TWC = \sum_{i=1}^{n} \gamma_{[i]} \sum_{j=i}^{n} (p_{[i]} + b_{[i-1][i]}) = \sum_{i=1}^{n} (p_{[i]} + b_{[i-1][i]}) \sum_{i=1}^{n} \gamma_{[i]}$: Total waiting cost (i.e., total weighted waiting time) of jobs in $S$ where $TWC = TWT$ if $\gamma_{[i]} = 1, \quad i = 1, \ldots, n$
- $TCT = \sum_{i=1}^{n} \sum_{j=i}^{n} (p_{[i]} + b_{[i-1][i]}) = \sum_{i=1}^{n} (n-i+1)(p_{[i]} + b_{[i-1][i]})$: Total completion time of jobs in $S$
- $TCC = \sum_{i=1}^{n} \gamma_{[i]} \sum_{j=i}^{n} (p_{[i]} + b_{[i-1][i]}) = \sum_{i=1}^{n} (p_{[i]} + b_{[i-1][i]}) \sum_{i=1}^{n} \gamma_{[i]}$: Total completion cost (i.e., total weighted completion time) of jobs in $S$ where $TCC = TCT$ if $\gamma_{[i]} = 1, \quad i = 1, \ldots, n$
Using the notation of Graham et al. (1979), our bicriteria scheduling problem can be formulated as a stochastic bicriteria problem in which three batches of jobs with unequal sizes and weights are to be processed on a machine (i.e., aircraft). The decreasing function \( \gamma_{[i]} = \tau, 0 < \tau < 1, i = 1, 2, 3 \), where \( \gamma_{[1]} > \gamma_{[2]} > \gamma_{[3]} \) represent, respectively, the stochastic attributes \( t_{ij} \) and \( C_{ij} \) of job \([i,j]\). As shown later, it is difficult to derive \( S^* \) for \( 1/b_{ij}-\text{sdst}/E[g(C_1, C_2)] \) involving TCC or TW when job weights (or unit delay costs) \( g_i = 1, \ldots, n \) are general. However, \( S^* \) can be obtained if \( \gamma_{[i]} = \tau, \tau > 1 (0 < \tau < 1), i = 1, \ldots, n \), i.e., if the weight for job \([i]\), \( [i] = 1, \ldots, n \), increases (decreases) non-linearly with the job position \( i \), \( i = 1, \ldots, n \). An example to justify the use of weight function \( \gamma_{[i]} = \tau, 0 < \tau < 1, i = 1, \ldots, n \), is in boarding the first-class passengers who have the highest priority but the smallest size, the batch of business class has less priority but a larger size, and the batch of economy class has the least priority but the largest size. This problem can be viewed as a scheduling problem in which three batches of jobs with unequal sizes and weights are to be processed on a machine (i.e., aircraft). The decreasing function \( \gamma_{[i]} = \tau, 0 < \tau < 1, i = 1, 2, 3 \), where \( \gamma_{[1]} > \gamma_{[2]} > \gamma_{[3]} \) represent, respectively, the...
weights for the first batch (first class), second batch (business class) and third batch (economy class) can be used to schedule the first-class passengers first followed by the business class and the economy class last. (This may be appropriate if, e.g., the environment outside the aircraft is inferior to that of inside; thus, the first, business and economy classes are boarded first, second and last, respectively.) On the other hand, one can use the increasing function $\gamma_i = \tau_i, \tau_{i'} > 1, i = 1, 2, 3$, where $\gamma_{[1]} < \gamma_{[2]} < \gamma_{[3]}$ are, respectively, the weights for the first batch (economy class), second batch (business class) and third batch (first class) to embark the economy class first followed by the business class and then the first class (This may be suitable if, e.g., the environment outside the aircraft is superior to that of inside.) Another example is in production environments involving the ABC inventory system. Here, the decreasing function $\gamma_i = \tau_i, 0 < \tau_{i'} < 1, i = 1, 2, 3$ can represent, respectively, the weights of the batch of ‘A’ items (with the highest weight but the smallest size) followed by the batch of ‘B’ items (with a lower weight but a larger size) and the batch of ‘C’ items (with the least weight but the largest size). On the other hand, the increasing function $\gamma_i = \tau_i, \tau_{i'} > 1, i = 1, 2, 3$, can be applied to signify the weights for the batch of ‘C’ items (with the least weight), the batch of ‘B’ items, and the batch of ‘A’ items (with the highest weight), respectively.

3 The scenario with linear cost function

Consider the linear cost function $g(C_1, C_2) = \alpha C_1 + (1 - \alpha) C_2$ where $0 < \alpha < 1$. We remark that $g(C_1, C_2)$ models risk-neutral schedulers (e.g., Keeney and Raiffa, 1976). The objective function $G_3(C_1, C_2)$ for $1//\text{E}\[\alpha C_1 + (1 - \alpha) C_2\]$ is defined as

$$G_3(C_1, C_2) = \alpha E[C_1] + (1 - \alpha) E[C_2].$$

Hence, using equations (1) and (2), $S^*$ minimises the weighted sum of the expected values of $C_1$ and $C_2$.

In what follows, we solve $1/b_{ij}$-$\text{dst}/\text{E}\[\alpha C_1 + (1 - \alpha) C_2\]$ and its special case $1/b_{ij}$-$\text{sist}/\text{E}\[\alpha TCT + (1 - \alpha) C_{\text{max}}\]$.

3.1 The $1/b_{ij}$-$\text{dst}/\text{E}\[\alpha C_1 + (1 - \alpha) C_2\]$ scheduling problem

3.1.1 $1/b_{ij}$-$\text{dst}/\text{E}\[\alpha C_1 + (1 - \alpha) C_{\text{max}}\]$ with $C_j = TCT, TWT, TCC, TWC$ and $TL$

Substituting $C_1 = TCT = \sum_{j=1}^{n} (n - i + 1)(p_{i}[j] + h_{[i-1][j]})$ and $C_{\text{max}} = \sum_{j=1}^{n} (p_{i}[j] + h_{[i-1][j]})$ into equation (2), the $G_3(TCT, C_{\text{max}})$ for $1/b_{ij}$-$\text{dst}/\text{E}\[\alpha TCT + (1 - \alpha) C_{\text{max}}\]$ is computed as

$$G_3(TCT, C_{\text{max}}) = \sum_{i=1}^{n} \{ \alpha(n - i)\mu_{[i]} + [1 + \alpha(n - i)]B_{[i-1][j]} \} + \sum_{i=1}^{n} \mu_{[i]},$$

where $\sum_{i=1}^{n} \mu_{[i]}$, a sequence-independent constant can be dropped from equation (3) without affecting $S^*$. Since $G_3(TCT, C_{\text{max}})$ depends on the $\mu_{[i]}$ of job $[i]$ and the $B_{[i-1][j]}$ of jobs $[i - 1]$ and $[i], i = 1, \ldots, n$, the problem of finding $S^*$ for $1/b_{ij}$-$\text{dst}/\text{E}\[\alpha TCT + (1 - \alpha) C_{\text{max}}\]$ can be equivalently formulated as a QAP. In general, QAP is defined as
Minimise \( \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_{ij} + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{\ell=1}^{n} u_{ijk\ell} x_{ij} x_{k\ell} \)

subject to

\( \sum_{j=1}^{n} x_{ij} = 1, \quad j = 1, \ldots, n, \quad \sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \ldots, n, \quad x_{ij} = 0, i = 1, \ldots, n, \quad j = 1, \ldots, n. \)

In context of \( 1/E[g(C_1, C_2)] \), \( x_{ij} = 1 \) if job \( i, i = 1, \ldots, n, \) is assigned to position \( j, j = 1, \ldots, n, \) in a sequence and \( x_{ij} = 0 \) otherwise, \( q_{ij} \) is the linear cost for assigning job \( i \) to position \( j, j = 1, \ldots, n, \) and \( u_{ijk\ell} \) is the interaction cost for assigning job \( i, i = 1, \ldots, n, \) to position \( j, j = 1, \ldots, n, \), and job \( k, k = 1, \ldots, n, \) to position \( \ell, \ell = 1, \ldots, n, \), \( i \neq k, j \neq \ell. \) Thus, using equation (3), \( S^* \) for \( 1/b_g\text{-sdst}/E[\alpha TCT + (1 - \alpha) C_{\max}] \) can be obtained by solving QAP with the objective function:

Minimise \( \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_{ij} + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{\ell=1}^{n} u_{ijk\ell} x_{ij} x_{k\ell} \)

where

\( q_{ij} = \begin{cases} \alpha(n-j)\mu_i + [1 + \alpha(n-j)]B_{ij}, & \text{if } j = 1, i = 1, \ldots, n; \\ \alpha(n-j)\mu_i, & \text{otherwise}; \end{cases} \)

and

\( u_{ijk\ell} = \begin{cases} [1 + \alpha(n-j)]B_{k\ell}, & i, j, k = 1, \ldots, n, k \neq i; \\ 0, & \text{otherwise} \end{cases} \)

where \( B_{ij} \) is the expected set-up time of job \( j \) when scheduled immediately after job \( i, i \neq j, i = 0, 1, \ldots, n, j = 1, \ldots, n, \) and \( B_0 \) is the expected set-up time of job \( j \) when scheduled first.

QAP is a well-known NP-hard problem and can be solved by one of the available exact or heuristic methods (e.g., Adams and Johnson, 1994; Adams et al., 2007; Huntley and Brown, 1991; James et al., 2009; Sherali and Rajagopal, 1986; Xia, 2010; Zhang et al., 2010). Majority of the exact methods are Branch-and-Bound (B&B) algorithms that differ essentially in the branching strategies and in the computation of lower bounds for the objective function to fathom partial solutions. The heuristic methods are, in general, lower bound approximation procedures.

Similarly, \( S^* \) for \( 1/b_g\text{-sdst}/E[\alpha TWT + (1 - \alpha) C_{\max}] \) is also the solution to QAP with objective function (4) where

\( q_{ij} = \begin{cases} \alpha(n-j)\mu_i + [1 + \alpha(n-j-1)]B_{ij}, & \text{if } j = 1, i = 1, \ldots, n; \\ \alpha(n-j)\mu_i, & \text{otherwise}; \end{cases} \)

and

\( u_{ijk\ell} = \begin{cases} [1 + \alpha(n-j-1)]B_{k\ell}, & i, j, k = 1, \ldots, n, k \neq i; \\ 0, & \text{otherwise} \end{cases} \)
Considering $\gamma[i] = \tau$, $0 < \tau \neq 1$, $i = 1, \ldots, n$, the $G_s(C_1, C_{max})$ for 1/$b_\gamma$-sdst, $\gamma[i] = \tau$, $0 < \tau \neq 1/E[\alpha C_1 + (1 - \alpha)TADC]$ where $C_1 = TCC$, TWC, using equation (2), are computed as

$$G_s(TCC, C_{max}) = \frac{1}{1 - \tau} \sum_{i=1}^{n} \left[ \alpha(\tau' - \tau^{*i}) \mu_{ij} + [\alpha(\tau' - \tau^{*i}) + (1 - \alpha)(1 - \tau)]B_{i-i(j)} \right] + (1 - \alpha) \sum_{i=1}^{n} \mu_{ij},$$

and

$$G_s(TWC, C_{max}) = \frac{1}{1 - \tau} \sum_{i=1}^{n} \left[ \alpha(\tau' - \tau^{*i}) \mu_{ij} + [\alpha(\tau' - \tau^{*i}) + (1 - \alpha)(1 - \tau)]B_{i-i(j)} \right] + (1 - \alpha) \sum_{i=1}^{n} \mu_{ij}. \tag{5}$$

Equivalently, the solution to QAP provides $S^*$ for (i) 1/$b_\gamma$-sdst, $\gamma[i] = \tau$, $0 < \tau \neq 1/E[\alpha TCC + (1 - \alpha)C_{max}]$ where, using equation (5),

$$q_{ij} = \begin{cases} \frac{1}{1 - \tau} \left[ \alpha(\tau' - \tau^{*i}) \mu_{ij} + [\alpha(\tau' - \tau^{*i}) + (1 - \alpha)(1 - \tau)]B_{i-j} \right], & \text{if } j = 1, i = 1, \ldots, n; \\ \frac{1}{1 - \tau} \left[ \alpha(\tau' - \tau^{*i}) \mu_{ij} \right], & \text{otherwise}; \end{cases}$$

and

$$u_{ijk} = \begin{cases} \frac{1}{1 - \tau} \left[ \alpha(\tau' - \tau^{*i}) \mu_{ij} + (1 - \alpha)(1 - \tau)B_{i-k} \right], & i, j, k = 1, \ldots, n, k \neq i; \\ 0, & \text{otherwise}; \end{cases}$$

and (ii) 1/$b_\gamma$-sdst, $\gamma[i] = \tau$, $0 < \tau \neq 1/E[\alpha TWC + (1 - \alpha)C_{max}]$ where, using equation (6),

$$q_{ij} = \begin{cases} \frac{1}{1 - \tau} \left[ \alpha(\tau' - \tau^{*i}) \mu_{ij} + [\alpha(\tau' - \tau^{*i}) + (1 - \alpha)(1 - \tau)]B_{i-j} \right], & \text{if } j = 1, i = 1, \ldots, n; \\ \frac{1}{1 - \tau} \left[ \alpha(\tau' - \tau^{*i}) \mu_{ij} \right], & \text{otherwise}; \end{cases}$$

and

$$u_{ijk} = \begin{cases} \frac{1}{1 - \tau} \left[ \alpha(\tau' - \tau^{*i}) \mu_{ij} + (1 - \alpha)(1 - \tau)B_{i-k} \right], & i, j, k = 1, \ldots, n, k \neq i; \\ 0, & \text{otherwise}. \end{cases}$$

Since, $G_s(TL, C_{max}) = G_s(TCT, C_{max}) - \alpha \sum_{i=1}^{n} d_{i}$, where $\sum_{i=1}^{n} d_{i}$ is a constant, using equation (2), $S^*$ for 1/$b_\gamma$-sdst/E[\alpha TCT + (1 - \alpha)C_{max}] is also optimal for 1/$b_\gamma$-sdst/E[\alpha TL + (1 - \alpha)C_{max}].
3.1.2 1/b_{ij}-sdst/E[\alpha C_1 + (1 - \alpha)TADC] with C_1 = TCT, TWT, C_{max}, TCC, TWC and TL

The objective in 1/b_{ij}-sdst/E[\alpha C_1 + (1 - \alpha)TADC] is to minimise the expected value of a weighted sum of C_1 and TADC where TADC, a measure of variation in completion times, was introduced by Kanet (1981).

For 1/b_{ij}-sdst/E[\alpha C_1 + (1 - \alpha)TADC] where C_1 = TCT, TWT, for 1/b_{ij}-sdst, \gamma_1 = \tau, 0 < \tau \neq 1/E[\alpha C_1 + (1 - \alpha)TADC] where C_1 = TCT, TWT, using equation (2), we obtain, respectively,

\[ G_i(TCT, TADC) = \sum_{j=1}^{n} (n-i+1)[i(1-\alpha) + 2\alpha - 1]\mu_{ij} + B_{\{i\{i\}}, \quad (7) \]

\[ G_i(TWT, TADC) = \sum_{j=1}^{n} (\alpha(n-i) + (1-\alpha)(i-1)(n-i+1))\mu_{ij} + B_{\{i\{i\}}, \quad (8) \]

\[ G_i(C_{max}, TADC) = \sum_{i=1}^{n} [(1-\alpha)(i-1)(n-i+1)(\mu_{ij} + B_{\{i\{i\}} + \alpha B_{\{i\{i\}} + \alpha \sum_{j=1}^{n} \mu_{ij}], \quad (9) \]

\[ G_i(TCC, TADC) = \sum_{i=1}^{n} (\alpha(\tau^j - \tau^{i-1})/(1-\tau) + (1-\alpha)(i-1)(n-i+1))\mu_{ij} + B_{\{i\{i\}}, \quad (10) \]

and

\[ G_i(TWC, TADC) = \sum_{i=1}^{n} (\alpha(\tau^j - \tau^{i-1})/(1-\tau) + (1-\alpha)(i-1)(n-i+1))\mu_{ij} + B_{\{i\{i\}}. \quad (11) \]

Similar to 1/b_{ij}-sdst/E[\alpha C_1 + (1 - \alpha)C_{max}] (see Section 3.1.1), solving QAP with objective function (4) yields S' for

i 1/b_{ij}-sdst/E[\alpha TCT + (1 - \alpha)TADC] where, using equation (7)
\[ q_{ij} = \begin{cases} (n-j+1)[j(1-\alpha) + 2\alpha - 1]\mu_{ij} + B_{\{k\}}, \quad \text{if } j = 1, i = 1,...,n; \\ (n-j+1)[j(1-\alpha) + 2\alpha - 1]\mu_{ij}, \quad \text{otherwise}; \end{cases} \]

and
\[ u_{ijk} = \begin{cases} (n-j+1)[j(1-\alpha) + 2\alpha - 1]B_{\{k\}}, \quad \text{if } j = 1, i = 1,...,n, k \neq i; \\ 0, \quad \text{otherwise.} \]

ii 1/b_{ij}-sdst/E[\alpha TWT + (1 - \alpha)TADC] where, using equation (8)
\[ q_{ij} = \begin{cases} (\alpha(n-j) + (1-\alpha)(j-1)(n-j+1))\mu_{ij} + B_{\{k\}}, \quad \text{if } j = 1, i = 1,...,n; \\ (\alpha(n-j) + (1-\alpha)(j-1)(n-j+1))\mu_{ij}, \quad \text{otherwise}; \end{cases} \]

and
\[
\begin{align*}
\forall i, j, k = 1, \ldots, n, k \neq i; & \\
0, & \text{otherwise.}
\end{align*}
\]

\[
\begin{align*}
qu(j-1)(n-j+1) & (\mu_j + B_{ij}), & \text{if } j = 1, i = 1, \ldots, n; \\
(1-\alpha)(j-1)(n-j+1) & \mu_i, & \text{otherwise;}
\end{align*}
\]

\[
\begin{align*}
u_{ijk} = \left\{ \begin{array}{ll}
[\alpha(n-j)+(1-\alpha)(j-1)(n-j+1)]B_{ik}, & i, j, k = 1, \ldots, n, k \neq i; \\
0, & \text{otherwise.}
\end{array} \right.
\end{align*}
\]

\[
\begin{align*}
\forall i, j, k = 1, \ldots, n, k \neq i; & \\
0, & \text{otherwise.}
\end{align*}
\]

\[
\begin{align*}
qu(j-1)(n-j+1) & (\mu_j + B_{ij}), & \text{if } j = 1, i = 1, \ldots, n; \\
(1-\alpha)(j-1)(n-j+1) & \mu_i, & \text{otherwise;}
\end{align*}
\]

\[
\begin{align*}
u_{ijk} = \left\{ \begin{array}{ll}
[\alpha+(1-\alpha)(j-1)(n-j+1)]B_{ik}, & i, j, k = 1, \ldots, n, k \neq i; \\
0, & \text{otherwise.}
\end{array} \right.
\end{align*}
\]

\[
\begin{align*}
qu(j-1)(n-j+1) & (\mu_j + B_{ij}), & \text{if } j = 1, i = 1, \ldots, n; \\
(1-\alpha)(j-1)(n-j+1) & \mu_i, & \text{otherwise;}
\end{align*}
\]

\[
\begin{align*}
u_{ijk} = \left\{ \begin{array}{ll}
[\alpha+(1-\alpha)(j-1)(n-j+1)]B_{ik}, & i, j, k = 1, \ldots, n, k \neq i; \\
0, & \text{otherwise.}
\end{array} \right.
\end{align*}
\]

\[
\begin{align*}
qu(j-1)(n-j+1) & (\mu_j + B_{ij}), & \text{if } j = 1, i = 1, \ldots, n; \\
(1-\alpha)(j-1)(n-j+1) & \mu_i, & \text{otherwise;}
\end{align*}
\]

\[
\begin{align*}
u_{ijk} = \left\{ \begin{array}{ll}
[\alpha+(1-\alpha)(j-1)(n-j+1)]B_{ik}, & i, j, k = 1, \ldots, n, k \neq i; \\
0, & \text{otherwise.}
\end{array} \right.
\end{align*}
\]

\[
\begin{align*}
qu(j-1)(n-j+1) & (\mu_j + B_{ij}), & \text{if } j = 1, i = 1, \ldots, n; \\
(1-\alpha)(j-1)(n-j+1) & \mu_i, & \text{otherwise;}
\end{align*}
\]

\[
\begin{align*}
u_{ijk} = \left\{ \begin{array}{ll}
[\alpha+(1-\alpha)(j-1)(n-j+1)]B_{ik}, & i, j, k = 1, \ldots, n, k \neq i; \\
0, & \text{otherwise.}
\end{array} \right.
\end{align*}
\]

For \(1/b_{ij}\)-sdst/E[\alpha C + (1-\alpha)TADC], \(S^*\) is the same as that for \(1/b_{ij}\)-sdst/E[\alpha C + (1-\alpha)TADC] because \(G_2(TL, TADC) = G_2(TCT, TADC) - \alpha \sum_{i=1}^{n} d_{ij} \).

3.1.3 \(1/b_{ij}\)-sdst/E[\alpha C + (1-\alpha)TCC] and \(1/b_{ij}\)-sdst/E[\alpha C + (1-\alpha)TWC]
with C1 = TCT, TWT and TL

For \(1/b_{ij}\)-sdst, \(\gamma_{ij} = \tau^*, 0 < \tau \neq 1/E[\alpha C + (1-\alpha)TCC] \), using equation (2), we have

\[
G_3(TCT, TCC) = \sum_{i=1}^{n} [\alpha(n-i+1) + (1-\alpha)(\tau^* - \tau')/\tau](\mu_j + B_{ij}).
\]

Accordingly, the solution to QAP with objective function (4) also provides \(S^*\) when

\[
\begin{align*}
qu(j-1)(n-j+1) & (\mu_j + B_{ij}), & \text{if } j = 1, i = 1, \ldots, n; \\
(1-\alpha)(j-1)(n-j+1) & \mu_i, & \text{otherwise;}
\end{align*}
\]

\[
\begin{align*}
u_{ijk} = \left\{ \begin{array}{ll}
[\alpha+(1-\alpha)(j-1)(n-j+1)]B_{ik}, & i, j, k = 1, \ldots, n, k \neq i; \\
0, & \text{otherwise.}
\end{array} \right.
\end{align*}
\]
Scheduling in stochastic bicriteria single machine systems

\[ u_{ijk} = \begin{cases} 
\frac{\alpha(n-j)+1}{\tau} (1-\alpha) (t^i - t^{i+1}) / (1-\tau) \alpha_k, & i,j,k = 1,\ldots,n, k \neq i; \\
0, & \text{otherwise.} 
\end{cases} \]

Similarly, solving QAP yields \( S^* \) for

i) \[ 1/b_{ij}-\text{stdt}, \gamma[i] = \tau, \quad 0 < \tau \neq 1/E[\alpha TWT + (1 - \alpha) TCC] \]

\[ q_i = \begin{cases} 
\frac{\alpha(n-j)+1}{\tau} (1-\alpha) (t^i - t^{i+1}) / (1-\tau) (\mu_i + B_j), & \text{if } j=1, i=1,\ldots,n; \\
\left(\frac{\alpha(n-j)+1}{\tau} (1-\alpha) (t^i - t^{i+1}) / (1-\tau) \mu_i, & \text{otherwise;}
\right.
\end{cases} \]

and

\[ u_{ijk} = \begin{cases} 
\frac{\alpha(n-j)+1}{\tau} (1-\alpha) (t^i - t^{i+1}) / (1-\tau) \alpha_k, & i,j,k = 1,\ldots,n, k \neq i; \\
0, & \text{otherwise.} 
\end{cases} \]

ii) \[ 1/b_{ij}-\text{stdt}, \gamma[i] = \tau, \quad 0 < \tau \neq 1/E[\alpha TCT + (1 - \alpha) TWC] \]

\[ q_i = \begin{cases} 
\frac{\alpha(n-j)+1}{\tau} (1-\alpha) (t^i - t^{i+1}) / (1-\tau) (\mu_i + B_j), & \text{if } j=1, i=1,\ldots,n; \\
\left(\frac{\alpha(n-j)+1}{\tau} (1-\alpha) (t^i - t^{i+1}) / (1-\tau) \mu_i, & \text{otherwise;}
\right.
\end{cases} \]

and

\[ u_{ijk} = \begin{cases} 
\frac{\alpha(n-j)+1}{\tau} (1-\alpha) (t^i - t^{i+1}) / (1-\tau) \alpha_k, & i,j,k = 1,\ldots,n, k \neq i; \\
0, & \text{otherwise.} 
\end{cases} \]

iii) \[ 1/b_{ij}-\text{stdt}, \gamma[i] = \tau, \quad 0 < \tau \neq 1/E[\alpha TWT + (1 - \alpha) TCC] \]

\[ q_i = \begin{cases} 
\frac{\alpha(n-j)+1}{\tau} (1-\alpha) (t^i - t^{i+1}) / (1-\tau) (\mu_i + B_j), & \text{if } j=1, i=1,\ldots,n; \\
\left(\frac{\alpha(n-j)+1}{\tau} (1-\alpha) (t^i - t^{i+1}) / (1-\tau) \mu_i, & \text{otherwise;}
\right.
\end{cases} \]

and

\[ u_{ijk} = \begin{cases} 
\frac{\alpha(n-j)+1}{\tau} (1-\alpha) (t^i - t^{i+1}) / (1-\tau) \alpha_k, & i,j,k = 1,\ldots,n, k \neq i; \\
0, & \text{otherwise.} 
\end{cases} \]

Since \( G_0(TL, TCC) = G_0(TCT, TCC) - \alpha \sum_{i=1}^n d[i], \) and \( G_0(TL, TWC) = G_0(TCT, -\alpha \sum_{i=1}^n d[i], \) \( S^* \) for \( 1/b_{ij}-\text{stdt}, \gamma[i] = \tau, \quad 0 < \tau \neq 1/E[\alpha TCT + (1 - \alpha) TCC] \) is optimal for \( 1/b_{ij}-\text{stdt}, \gamma[i] = \tau, \quad 0 < \tau \neq 1/E[\alpha TL + (1 - \alpha) TCC] \), and \( S^* \) for \( 1/b_{ij}-\text{stdt}, \gamma[i] = \tau, \quad 0 < \tau \neq 1/E[\alpha TCT + (1 - \alpha) TWC] \) is optimal for \( 1/b_{ij}-\text{stdt}, \gamma[i] = \tau, \quad 0 < \tau \neq 1/E[\alpha TL + (1 - \alpha) TWC] \).

3.2 Special case: the \( 1/b_{ij}-\text{stdt}/E[\alpha C_J + (1 - \alpha) C_J] \) scheduling problem

For the sequence-independent set-up time case where \( p_i \) reflects both the processing time and the set-up time of job \( i \), the \( 1/b_{ij}-\text{stdt}/E[\alpha C_J + (1 - \alpha) C_J] \) problem of Section 3.1 reduces to \( 1/b_{ij}-\text{stdt}/E[\alpha C_J + (1 - \alpha) C_J] \) by removing the terms explicitly dealing with \( b[y_{i-1}] \) or \( B_{[y_{j-1]}}, i = 1, \ldots, n \) (i.e., \( b_i \) (or \( B_i \), \( i \neq j, i = 0, 1, \ldots, n, j = 1, \ldots, n). \)
Using the results of Section 3.1.1, for 1/bγ-sist/E[αC₁ + (1 – α)Cₘ₃], we obtain 
G₅(C₁, Cₘ₃) = αE[C₁] + (1 – α)Σₙₙμ₁[j]. Since, Σₙₙμ₁[j] is a constant, using equation
(1), S* for this problem is also optimal for the stochastic single criterion problem
1/bγ-sist/E[C₁]. Hence, S* can be derived, in O(nlogn) time, by the Shortest Expected
Processing Time (SEPT) rule: (μ₁[1] ≤ ... ≤ μ₁[n]) when C₁ = TCT, TWT and TL, and by the
Shortest Weighted Expected Processing Time (SWEPT) rule: (μ₁[1]/γ₁[1] ≤ ... ≤ μ₁[n]/γ₁[n])
when C₁ = TCC, TWC (e.g., Baker and Trietsch, 2009a).

Using the results of Section 3.1.2, we can find S* for 1/bγ-sist/E[αC₁ +
(1 – α)TADC] where C₁ = TCT, TWT, Cₘ₃, TCC, TWC and TL. For 1/bγ-sist
/E[αC₁ + (1 – α)TADC], removing the term Bᵢ[1], from equation (7) gives
G₅(TCT, TADC) = Σₙₙμ₁[j] [(n + i + 1)(i + 1) + 2α – 1]μ₁[j]. This is the scalar product of vectors
[(n + i + 1)(i + 1) + 2α – 1]μ₁[j] and μ₁[j] (i = 1, ..., n). On the basis of Hardy et al. (1967),
the product can be minimised, in O(nlogn) time, by matching μ₁[j] in the opposite order of
[(n + i + 1)(i + 1) + 2α – 1], i = 1, ..., n, i.e., the job with the longest μ₁[j] is assigned to the position with
the smallest (n + i + 1)(i + 1) + 2α – 1], the job with the second longest μ₁[j] is assigned to the position with the second
smallest (n + i + 1)(i + 1) + 2α – 1], etc. The resultant ordering of μ₁[j] is S* for 1/bγ-sist/E
[αC₁ + (1 – α)TADC]. For 1/bγ-sist/E[αTWT + (1 – α)TADC] and 1/bγ-sist/E
[αTL + (1 – α)TADC], using equation (2), G₅(TWT, TADC) = G₅(TCT, TADC) –
αΣₙₙμ₁[j] and G₅(TL, TADC) = G₅(TCT, TADC) – αΣₙₙμ₁[j]. Hence, S* for 1/bγ-sist/E
[αC₁ + (1 – α)TADC] is also optimal for 1/bγ-sist/E[αTWT + (1 – α)TADC] and
1/bγ-sist/E[αTL + (1 – α)TADC]. For 1/bγ-sist/E[αCₘ₃ + (1 – α)TADC], equation (9)
reduces to G₅(Cₘ₃, TADC) = Σₙₙμ₁[j] [μ₁[j] (n + i + 1)]μ₁[j] indicating that S* for
1/bγ-sist/E[αCₘ₃ + (1 – α)TADC] is also optimal for the stochastic single criterion
problem 1/bγ-sist/E[TADC]. Thus, S* is obtained by matching μ₁[j] in the opposite order of
(i + 1)(n + i + 1), i = 1, ..., n (see Hardy et al., 1967).

Similarly, using equations (10) and (11), G₅(C₁, TAD) for 1/bγ-sist,
γ₁[1] = τ[1], 0 < τ[1] < 1/E[αC₁ + (1 – α)TAD] where C₁ = TCC, TWC are, respectively,
G₅(TCC, TAD) = Σₙₙμ₁[j] [ατ[1](1 – τ[1]) + (1 – α)(i + 1)]μ₁[j] = ατ[1] – 1

and
G₅(TWC, TAD) = Σₙₙμ₁[j] [ατ[1](1 – τ[1]) + (1 – α)(i + 1)]μ₁[j] = ατ[1] – 1

From equation (12) since Σₙₙμ₁[j] [ατ[1](1 – τ[1]) + (1 – α)(i + 1)]μ₁[j] is the product of vectors
[ατ[1](1 – τ[1]) + (1 – α)(i + 1)]μ₁[j] and μ₁[j] (i = 1, ..., n), then S* for 1/γ₁[1] = τ[1],
τ[1] ≠ 1/E[αTCC + (1 – α)TAD] can be obtained, in O(nlogn) time, by matching μ₁[j] in
the opposite order of [ατ[1](1 – τ[1]) + (1 – α)(i + 1)]μ₁[j], i = 1, ..., n. Likewise, using
equation (13), S* for 1/γ₁[1] = τ[1], τ[1] ≠ 1/E[αTWT + (1 – α)TAD] is found, in O(nlogn)
time, by matching μ₁[j] in the opposite order of [ατ[1](1 – τ[1]) + (1 – α)(i + 1)]μ₁[j],
i = 1, ..., n.

According to the results of Section 3.1.3, for 1/bγ-sist/E[αTWT(1 – α)TWC], using
equation (2), we have
\[ G_2(TWT, TWC) = \sum_{i=1}^{n} \sum_{k=1}^{\gamma_i} \lambda_{ik}, \] (14)

where \( \lambda_{ik} = \alpha + (1 - \alpha) \gamma_{ik} \). Hence, using equations (1) and (14), \( S^* \) can be derived, in \( O(n \log n) \) time, by the SWEPT rule: \( (\mu_1/\lambda_{11}) \leq \ldots \leq \mu_n/\lambda_{n,n} \). For \( 1/b_{ij}\)-sdist/E \( \{\alpha C_1 + (1 - \alpha) C_2 \} \) and \( 1/b_{ij}\)-sdist/E \( \{\alpha C_1 + (1 - \alpha) C_2 \} \), where \( C_1 = TCT, TWT \) and TL, using equation (2), \( G_2(TWT, TCC) = G_2(TWT, TWC) + (1 - \alpha) \sum_{i=1}^{n} \gamma_i \mu_i \), \( G_2(TCT, TCC) = G_2(TWT, TCC) + \alpha \sum_{i=1}^{n} \mu_i \), \( G_2(TCT, TWC) = G_2(TWT, TWC) + \alpha \sum_{i=1}^{n} \mu_i d_i \), and \( G_2(TL, TCC) = G_2(TWT, TCC) + (1 - \alpha) \sum_{i=1}^{n} \gamma_i \mu_i + \alpha \sum_{i=1}^{n} \mu_i - d_i \). Thus, \( S^* \) for \( 1/b_{ij}\)-sdist/E \( \{\alpha TWT + (1 - \alpha) TWC \} \) is also optimal for the other five corresponding problems.

4 The scenario with exponential cost function

Consider the non-decreasing exponential cost function \( g(C_1, C_2) = \text{sgn}(\alpha) \exp [\alpha(C_1 + C_2)] \) where \( \text{sgn}(\alpha) = 1 \) if \( \alpha > 0 \), and \( \text{sgn}(\alpha) = -1 \) if \( \alpha < 0 \). We remark that when \( \alpha > 0 \), \( g(C_1, C_2) \) is convex and the scheduler is constant risk averse, and when \( \alpha < 0 \), \( g(C_1, C_2) \) is concave and the scheduler is constant risk prone (e.g., Keeney and Raiffa, 1976).

The objective function \( G_3(C_1, C_2) \) for \( 1/E[\text{sgn}(\alpha) \exp [\alpha(C_1 + C_2)]] \) is defined as

\[ G_3(C_1, C_2) = E[\text{sgn}(\alpha) \exp [\alpha(C_1 + C_2)]] = \text{sgn}(\alpha) M_{C_1+C_2}(\alpha). \] (15)

Hence, using equations (1) and (15), \( S^* \) for \( 1/E[\text{sgn}(\alpha) \exp [\alpha(C_1 + C_2)]] \) optimises the MGFs of \( C_1 \) and \( C_2 \).

In what follows, we solve \( 1/b_{ij}\)-sdist/E \( \{\text{sgn}(\alpha) \exp [\alpha(C_1 + C_2)] \} \) and \( 1/b_{ij}\)-sdist/E \( \{\text{sgn}(\alpha) \exp [\alpha(C_1 + C_2)] \} \).

4.1 The \( 1/b_{ij}\)-sdist/E \( \{\text{sgn}(\alpha) \exp [\alpha(C_1 + C_2)] \} \) scheduling problem

4.1.1 \( 1/b_{ij}\)-sdist/E \( \{\text{sgn}(\alpha) \exp [\alpha(TAD + C_{\text{max}})] \} \) with \( C_i = TAD, TCT, TWT, TCC \) and TL

For \( 1/b_{ij}\)-sdist/E \( \{\text{sgn}(\alpha) \exp [\alpha(TAD + C_{\text{max}})] \} \), using equation (15), we have

\[ G_3(TAD, C_{\text{max}}) = \text{sgn}(\alpha) M_{\sum_{i=1}^{n} \alpha[i+(i-1)(n-i)] M_{\sum_{i=1}^{n} \alpha[i+(i-1)(n-i)]}}(\alpha). \] (16)

Using equations (1) and (16), then

\[
S^* = \begin{cases} 
\arg \min_{S_{\alpha \in \mathcal{W}}} \left\{ \prod_{i=1}^{n} M_{\alpha[i+(i-1)(n-i)]} M_{\alpha[i+(i-1)(n-i)]} \right\}, & \text{if } \alpha > 0, \\
\arg \max_{S_{\alpha \in \mathcal{W}}} \left\{ \prod_{i=1}^{n} M_{\alpha[i+(i-1)(n-i)]} M_{\alpha[i+(i-1)(n-i)]} \right\}, & \text{if } \alpha < 0.
\end{cases}
\]
Since $M_{p_\alpha} [\alpha \{ i + (i-1)(n-i) \}] > 1$ and $M_{b_{i-1}} [\alpha \{ i + (i-1)(n-i) \}] > 1$ for $\alpha > 0$, $S^*$ can be equivalently determined as

$$S^* = \arg \min_{S \in \Psi} \left\{ \sum_{i=1}^{n} \ln \left[ M_{p_\alpha} [\alpha \{ i + (i-1)(n-i) \}] M_{b_{i-1}} [\alpha \{ i + (i-1)(n-i) \}] \right] \right\}. \tag{17}$$

Similarly, since $0 < M_{p_\alpha} [\alpha \{ i + (i-1)(n-i) \}] < 1$ and $0 < M_{b_{i-1}} [\alpha \{ i + (i-1)(n-i) \}] < 1$ for $\alpha < 0$, $S^*$ is such that

$$S^* = \arg \min_{S \in \Psi} \left\{ \prod_{i=1}^{n} M_{p_\alpha} [\alpha \{ i + (i-1)(n-i) \}] M_{b_{i-1}} [\alpha \{ i + (i-1)(n-i) \}] \right\}^{-1},$$

or, equivalently,

$$S^* = \arg \min_{S \in \Psi} \left\{ \sum_{i=1}^{n} \ln \left[ M_{p_\alpha} [\alpha \{ i + (i-1)(n-i) \}] M_{b_{i-1}} [\alpha \{ i + (i-1)(n-i) \}] \right] \right\}^{-1}. \tag{18}$$

Combining equations (17) and (18), we have

$$S^* = \arg \min_{S \in \Psi} \left\{ \sum_{i=1}^{n} \ln \left[ M_{p_\alpha} [\alpha \{ i + (i-1)(n-i) \}] M_{b_{i-1}} [\alpha \{ i + (i-1)(n-i) \}] \right] \right\}^{\text{sgn}(\alpha)},$$

or

$$S^* = \arg \min_{S \in \Psi} \left\{ \text{sgn}(\alpha) \sum_{i=1}^{n} \ln M_{p_\alpha} [\alpha \{ i + (i-1)(n-i) \}] + \ln M_{b_{i-1}} [\alpha \{ i + (i-1)(n-i) \}] \right\}. \tag{19}$$

Since equation (19) depends on the MGFs of $p_{i|j}$ and $b_{i-1|j}$, $S^*$ for $1/b_{j-1|\text{sdst}}E[\text{sgn}(\alpha)\exp(\alpha(TADC + C_{\text{max}}))]$ can be equivalently formulated as QAP with objective function (4) where

$$q_{ij} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{p_\alpha} [\alpha \{ j + (j-1)(n-j) \}] + \ln M_{b_{i-1}} [\alpha \{ j + (j-1)(n-j) \}]; & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{p_\alpha} [\alpha \{ j + (j-1)(n-j) \}]; & \text{otherwise};
\end{cases}$$

and

$$u_{ijk} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{p_\alpha} [\alpha \{ j + (j-1)(n-j) \}]; & i, j, k = 1, \ldots, n; k \neq i; \\
0, & \text{otherwise}.
\end{cases}$$

Similarly, the solution to QAP produces $S^*$ for

$$q'_{ij} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{p_\alpha} [\alpha \{ j + (j-1)(n-j) \}] + \ln M_{b_{i-1}} [\alpha \{ j + (j-1)(n-j) \}]; & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{p_\alpha} [\alpha \{ j + (j-1)(n-j) \}]; & \text{otherwise};
\end{cases}$$

and

$$u'_{ijk} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{p_\alpha} [\alpha \{ j + (j-1)(n-j) \}]; & i, j, k = 1, \ldots, n; k \neq i; \\
0, & \text{otherwise}.
\end{cases}$$
Scheduling in stochastic bicriteria single machine systems

\[ u_{ij} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{n_k}[\alpha(n-j+2)]; & i, j, k = 1, \ldots, n; k \neq i; \\
0, & \text{otherwise.} 
\end{cases} \]

ii \[ 1/b_{ij} \text{-stdt/E}[\text{sgn}(\alpha)\exp[\alpha(TWT + C_{\text{max}})]] \] when
\[ q_{ij} = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{n_k}[\alpha(n-j+1)] + \ln M_{n_k}[\alpha(n-j+1)] \right]; & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{n_k}[\alpha(n-j+1)]; & \text{otherwise}; 
\end{cases} \]

and
\[ u_{ij} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{n_k}[\alpha(n-j+1)]; & i, j, k = 1, \ldots, n; k \neq i; \\
0, & \text{otherwise.} 
\end{cases} \]

iii \[ 1/b_{ij} \text{-stdt, } \gamma(t) = \tau, 0 < \tau \neq 1/E[\text{sgn}(\alpha)\exp[\alpha(TCC + C_{\text{max}})]] \] when
\[ q_{ij} = \begin{cases} \text{sgn}(\alpha) \left[ \ln M_{n_k}[\alpha(1+\tau' - \tau'')/(1-\tau)] \right]; & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{n_k}[\alpha(1+\tau' - \tau'')/(1-\tau)]; & \text{otherwise}; 
\end{cases} \]

and
\[ u_{ij} = \begin{cases} \text{sgn}(\alpha) \ln M_{n_k}[\alpha(1+\tau' - \tau'')/(1-\tau)]; & i, j, k = 1, \ldots, n; k \neq i; \\
0, & \text{otherwise.} 
\end{cases} \]

iv \[ 1/b_{ij} \text{-stdt, } \gamma(t) = \tau, 0 < \tau \neq 1/E[\text{sgn}(\alpha)\exp[\alpha(TWC + C_{\text{max}})]] \] when
\[ q_{ij} = \begin{cases} \text{sgn}(\alpha) \left[ \ln M_{n_k}[\alpha(1+\tau' - \tau'')/(1-\tau)] \right]; & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{n_k}[\alpha(1+\tau' - \tau'')/(1-\tau)]; & \text{otherwise}; 
\end{cases} \]

and
\[ u_{ij} = \begin{cases} \text{sgn}(\alpha) \ln M_{n_k}[\alpha(1+\tau' - \tau'')/(1-\tau)]; & i, j, k = 1, \ldots, n; k \neq i; \\
0, & \text{otherwise.} 
\end{cases} \]

Since \( G_3(TL, C_{\text{max}}) = G_3(TCT, C_{\text{max}}) \) \( \exp(-\alpha \sum_{i=1}^{n} d_i) \) where \( \exp(-\alpha \sum_{i=1}^{n} d_i) \) is a constant, \( S^* \) is for \( 1/b_{ij} \text{-stdt/E}[\text{sgn}(\alpha)\exp[\alpha(TCT + C_{\text{max}})]] \) is optimal for \( 1/b_{ij} \text{-stdt/E}[\text{sgn}(\alpha)\exp[\alpha(TL + C_{\text{max}})]] \).

4.1.2 \( 1/b_{ij} \text{-stdt/E}[\text{sgn}(\alpha)\exp[\alpha(C_1 + TADC)]] \) with \( C_1 = TCT, TWT, TCC, TWC \) and TL

For \( 1/b_{ij} \text{-stdt/E}[\text{sgn}(\alpha)\exp[\alpha(TCT + TADC)]] \), using equation (15), we have
Using arguments similar to those of Section 4.1.1, we can show that solving QAP with objective function (4) provides $S^*$ where

$$q_0 = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{\rho_0} [\alpha j (n-j+1)] + \ln M_{b_0} [\alpha j (n-j+1)] \right]; & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{\rho_0} [\alpha j (n-j+1)]; & \text{otherwise};
\end{cases}$$

and

$$u_{ijk1} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{b_0} [\alpha j (n-j+1)]; & i, j, k = 1, \ldots, n; k \neq i; \\
0, & \text{otherwise}.
\end{cases}$$

Analogously, the solution to QAP yields $S^*$ for

i. $1/b_{pd}-\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha(TWT + TADC))]$ when

$$q_0 = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{\rho} [\alpha j (n-j+1)-1] + \ln M_{b_0} [\alpha j (n-j+1)-1] \right]; & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{\rho} [\alpha j (n-j+1)-1]; & \text{otherwise};
\end{cases}$$

and

$$u_{ijk1} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{b_0} [\alpha j (n-j+1)-1]; & i, j, k = 1, \ldots, n; k \neq i; \\
0, & \text{otherwise}.
\end{cases}$$

ii. $1/b_{pd}-\text{sdst}, \gamma_{|\tau|} = \tau', 0 < \tau \neq 1/E[\text{sgn}(\alpha)\exp(\alpha(TCC + TADC))]$ when

$$q_0 = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{\rho} [\alpha j (n-j+1)-1] (1-\tau)+ (j-1)(n-j+1)] + \ln M_{b_0} [\alpha j (n-j+1)-1] (1-\tau)+ (j-1)(n-j+1)] \right]; & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{\rho} [\alpha j (n-j+1)-1] (1-\tau)+ (j-1)(n-j+1)]; & \text{otherwise};
\end{cases}$$

and

$$u_{ijk1} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{b_0} [\alpha j (n-j+1)-1] (1-\tau)+ (j-1)(n-j+1)] ; & i, j, k = 1, \ldots, n; k \neq i; \\
0, & \text{otherwise}.
\end{cases}$$

iii. $1/b_{pd}-\text{sdst}, \gamma_{|\tau|} = \tau', 0 < \tau \neq 1/E[\text{sgn}(\alpha)\exp(\alpha(TWC + TADC))]$ when

$$q_0 = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{\rho} [\alpha j (n-j+1)-1] (1-\tau)+ (j-1)(n-j+1)] + \ln M_{b_0} [\alpha j (n-j+1)-1] (1-\tau)+ (j-1)(n-j+1)] \right]; & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{\rho} [\alpha j (n-j+1)-1] (1-\tau)+ (j-1)(n-j+1)] ; & \text{otherwise};
\end{cases}$$

and
Scheduling in stochastic bicriteria single machine systems

\[ u_{ij} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{b_{ik}} \left[ \alpha (\tau^{+1} - \tau^{-1}) / (1 - \tau) + (j - 1)(n - j + 1) \right]; & i, j, k = 1, \ldots, n; k \neq i; \\
0, & \text{otherwise}.
\end{cases} \]

Since \( G_0(TL, TADC) = G_0(TCT, TADC) \exp(-\alpha \sum_{i=1}^{n} d_i) \), \( S^* \) for \( 1/b_{ij}-\text{sdst}/E[\text{sgn}(\alpha)\exp[\alpha(TCT + TADC)]] \) is optimal for \( 1/b_{ij}-\text{sdst}/E[\text{sgn}(\alpha)\exp[\alpha(TL + TADC)]] \).

4.1.3 \( 1/b_{ij}-\text{sdst}/E[\text{sgn}(\alpha)\exp[\alpha(C_1 + TCC)]] \) and \( 1/b_{ij}-\text{sdst}/E[\text{sgn}(\alpha)\exp[\alpha(C_1 + TWC)]] \) with \( C_1 = TCT, TWT \) and \( TL \)

Similar to Sections 4.1.1 and 4.1.2, the solution to QAP with objective function (4) produces \( S^* \) for

\[ i 1/b_{ij}-\text{sdst}, \gamma_{[i]} = \tau, 0 < \tau \neq 1/E[\text{sgn}(\alpha)\exp[\alpha(TCT + TCC)]] \] when

\[ q_{ij} = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{b_{ik}} \left[ \alpha (n - j + 1 + (\tau' - \tau^{-1}) / (1 - \tau) \right] \right]; & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{b_{ik}} \left[ \alpha (n - j + 1 + (\tau' - \tau^{-1}) / (1 - \tau) \right]; & \text{otherwise};
\end{cases} \]

and

\[ u_{ij} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{b_{ik}} \left[ \alpha (n - j + 1 + (\tau' - \tau^{-1}) / (1 - \tau) \right]; & i, j, k = 1, \ldots, n; k \neq i; \\
0, & \text{otherwise}.
\end{cases} \]

\[ ii 1/b_{ij}-\text{sdst}, \gamma_{[i]} = \tau, 0 < \tau \neq 1/E[\text{sgn}(\alpha)\exp[\alpha(TWT + TCC)]] \] when

\[ q_{ij} = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{b_{ik}} \left[ \alpha (n - j + 1 + (\tau' - \tau^{-1}) / (1 - \tau) \right] \right]; & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{b_{ik}} \left[ \alpha (n - j + 1 + (\tau' - \tau^{-1}) / (1 - \tau) \right]; & \text{otherwise};
\end{cases} \]

and

\[ u_{ij} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{b_{ik}} \left[ \alpha (n - j + 1 + (\tau' - \tau^{-1}) / (1 - \tau) \right]; & i, j, k = 1, \ldots, n; k \neq i; \\
0, & \text{otherwise}.
\end{cases} \]

\[ iii 1/b_{ij}-\text{sdst}, \gamma_{[i]} = \tau, 0 < \tau \neq 1/E[\text{sgn}(\alpha)\exp[\alpha(TCT + TWC)]] \] when

\[ q_{ij} = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{b_{ik}} \left[ \alpha (n - j + 1 + (\tau' - \tau^{-1}) / (1 - \tau) \right] \right]; & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{b_{ik}} \left[ \alpha (n - j + 1 + (\tau' - \tau^{-1}) / (1 - \tau) \right]; & \text{otherwise};
\end{cases} \]

and

\[ u_{ij} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{b_{ik}} \left[ \alpha (n - j + 1 + (\tau' - \tau^{-1}) / (1 - \tau) \right]; & i, j, k = 1, \ldots, n; k \neq i; \\
0, & \text{otherwise}.
\end{cases} \]
iv \( 1/b_{ij}-\text{sdst}, \gamma_{[\tau]} = \tau', 0 < \tau' \neq 1/E[\text{sgn}(\alpha)\exp(\alpha(TWT + TWC))] \) when
\[
q_{ji} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{\rho_i} \left[ \alpha(n - j + (\tau_i^{ij} - \tau_j^{ii})/(1 - \tau)) \right], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{\rho_i} \left[ \alpha(n - j + (\tau_i^{ij} - \tau_j^{ii})/(1 - \tau)) \right], & \text{otherwise};
\end{cases}
\]
and
\[
u_{ji} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{\rho_i} \left[ \alpha(n - j + (\tau_i^{ij} - \tau_j^{ii})/(1 - \tau)) \right], & i, j, k = 1, \ldots, n; k \neq i; \\
0, & \text{otherwise}.
\end{cases}
\]
The \( S^* \) for \( 1/b_{ij}-\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha(TCT + TCC))] \) is optimal for \( 1/b_{ij}-\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha(TL + TCC))] \), and \( S^* \) for \( 1/b_{ij}-\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha(TCT + TWC))] \) is optimal for \( 1/b_{ij}-\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha(TL + TWC))] \) since \( G_3(TL, TCC) = G_3(TCT, TCC) \exp(\alpha(TCT + TCC)) \) is optimal for \( 1/b_{ij}-\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha(TL + TWC))] \).

4.2 Special case: the \( 1/b_{ij}-\text{sist}/E[\text{sgn}(\alpha)\exp(\alpha(C_1 + C_2))] \) scheduling problem

For \( 1/b_{ij}-\text{ sist}/E[\text{sgn}(\alpha)\exp(\alpha(C_1 + C_2))] \), \( S^* \) can be obtained by deleting the terms involving \( b_{i-j}[\alpha] \) (or \( M_{\rho_i-j}[\alpha] \)), \( i = 1, \ldots, n \) (i.e., \( b_y \) (or \( M_{\rho_i-y}[\alpha] \)) \( i \neq j, i = 0, 1, \ldots, n, j = 1, \ldots, n \)) from the QAP formulation for \( 1/b_{ij}-\text{ sist}/E[\text{sgn}(\alpha)\exp(\alpha(C_1 + C_2))] \) (see Section 4.1). This results in a linear Assignment Problem (AP) formulation for \( 1/b_{ij}-\text{ sist}/E[\text{sgn}(\alpha)\exp(\alpha(C_1 + C_2))] \) with \( d_{ij} \) being the cost of assigning job \( i, i = 1, \ldots, n \), to position \( j, j = 1, \ldots, n \). Thus, the solution to AP, found in \( O(n^3) \) time (e.g., Papadimitriou and Steiglitz, 1982), yields \( S^* \) for \( 1/b_{ij}-\text{ sist}/E[\text{sgn}(\alpha)\exp(\alpha(C_1 + C_2))] \). In particular, based on the results of Section 4.1.1, solving AP provides \( S^* \) for

i \( 1/b_{ij}-\text{ sist}/E[\text{sgn}(\alpha)\exp(\alpha(TADC + C_{\max})] \) when
\[
q_{ji} = \text{sgn}(\alpha) \ln M_{\rho_i} \left[ \alpha((j-1)(n-j) + j) \right], \ i = 1, \ldots, n, j = 1, \ldots, n.
\]

ii \( 1/b_{ij}-\text{ sist}/E[\text{sgn}(\alpha)\exp(\alpha(TCT + C_{\max})] \) when
\[
q_{ji} = \text{sgn}(\alpha) \ln M_{\rho_i} \left[ \alpha(n - j + 2) \right], \ i = 1, \ldots, n, j = 1, \ldots, n.
\]

iii \( 1/b_{ij}-\text{ sist}/E[\text{sgn}(\alpha)\exp(\alpha(TWT + C_{\max})] \) when
\[
q_{ji} = \text{sgn}(\alpha) \ln M_{\rho_i} \left[ \alpha(n - j + 1) \right], \ i = 1, \ldots, n, j = 1, \ldots, n.
\]

iv \( 1/b_{ij}-\text{ sist}, \gamma_{[\tau]} = \tau', 0 < \tau' \neq 1/E[\text{sgn}(\alpha)\exp(\alpha(TCC + C_{\max})] \) when
\[
q_{ji} = \text{sgn}(\alpha) \ln M_{\rho_i} \left[ \alpha(1 + (\tau_i^{ij} - \tau_j^{ii})/(1 - \tau)) \right], \ i = 1, \ldots, n, j = 1, \ldots, n.
\]

v \( 1/b_{ij}-\text{ sist}, \gamma_{[\tau]} = \tau', 0 < \tau' \neq 1/E[\text{sgn}(\alpha)\exp(\alpha(TWC + C_{\max})] \) when
\[
q_{ji} = \text{sgn}(\alpha) \ln M_{\rho_i} \left[ \alpha(1 + (\tau_i^{ij} - \tau_j^{ii})/(1 - \tau)) \right], \ i = 1, \ldots, n, j = 1, \ldots, n.
\]
Since \( G_s(TL, C_{\max}) = G_{\alpha}(TCT, C_{\max}) \exp \left( -\alpha \sum_{i=1}^n d_i \right) \), \( S^* \) for \( 1/\beta,\gamma \)-sist/E[\( \text{sgn}(\alpha) \exp \left( \alpha(TCT + C_{\max}) \right) \)] is optimal for \( 1/\beta,\gamma \)-sist/E[\( \text{sgn}(\alpha) \exp \left( \alpha(TL + C_{\max}) \right) \)].

Similarly, based on the results of Section 4.2.2, the solution to AP produces \( S^* \) for

1. \( 1/\beta,\gamma \)-sist/E[\( \text{sgn}(\alpha) \exp \left( \alpha(TCT + \text{TADC}) \right) \)] when

\[
q_{ij} = \text{sgn}(\alpha) \ln M_{p_i} \left[ \alpha j(n-j+1) \right], \quad i = 1, \ldots, n, j = 1, \ldots, n.
\]

2. \( 1/\beta,\gamma \)-sist/E[\( \text{sgn}(\alpha) \exp \left( \alpha(TWT + \text{TADC}) \right) \)] when

\[
q_{ij} = \text{sgn}(\alpha) \ln M_{p_i} \left[ \alpha j(n-j+1) - 1 \right], \quad i = 1, \ldots, n, j = 1, \ldots, n.
\]

3. \( 1/\beta,\gamma \)-sist/E[\( \text{sgn}(\alpha) \exp \left( \alpha(TCC + \text{TADC}) \right) \)] when

\[
q_{ij} = \text{sgn}(\alpha) \ln M_{p_i} \left[ \alpha (j(n-j+1) - 1) \right], \quad i = 1, \ldots, n, j = 1, \ldots, n.
\]

4. \( 1/\beta,\gamma \)-sist/E[\( \text{sgn}(\alpha) \exp \left( \alpha(C_{\max}) \right) \)] when

\[
q_{ij} = \text{sgn}(\alpha) \ln M_{p_i} \left[ \alpha ((\tau - \tau^{*+1}) / (1-\tau)) + (j-1)(n-j-1) \right], \quad i = 1, \ldots, n, j = 1, \ldots, n.
\]

The \( S^* \) for \( 1/\beta,\gamma \)-sist/E[\( \text{sgn}(\alpha) \exp \left( \alpha(TCT + \text{TADC}) \right) \)] is optimal for \( 1/\beta,\gamma \)-sist/E[\( \text{sgn}(\alpha) \exp \left( \alpha(TL + \text{TADC}) \right) \)] since \( G_s(TL, TADC) = G_{\alpha}(TCT, TADC) \exp \left( -\alpha \sum_{i=1}^n d_i \right) \).

Finally, utilising the results of Section 4.1.3, solving AP yields \( S^* \) for

1. \( 1/\beta,\gamma \)-sist, \( \gamma_{[0]} = \tau, \quad 0 < \tau \neq 1 \)/E[\( \text{sgn}(\alpha) \exp \left( \alpha(TCT + \text{TCC}) \right) \)] when

\[
q_{ij} = \text{sgn}(\alpha) \ln M_{p_i} \left[ \alpha(n-j+1 + (\tau - \tau^{*+1}) / (1-\tau)) \right], \quad i = 1, \ldots, n, j = 1, \ldots, n.
\]

2. \( 1/\beta,\gamma \)-sist, \( \gamma_{[0]} = \tau, \quad 0 < \tau \neq 1 \)/E[\( \text{sgn}(\alpha) \exp \left( \alpha(TWT + \text{TCC}) \right) \)] when

\[
q_{ij} = \text{sgn}(\alpha) \ln M_{p_i} \left[ \alpha(n-j+1 + (\tau - \tau^{*+1}) / (1-\tau)) \right], \quad i = 1, \ldots, n, j = 1, \ldots, n.
\]

3. \( 1/\beta,\gamma \)-sist, \( \gamma_{[0]} = \tau, \quad 0 < \tau \neq 1 \)/E[\( \text{sgn}(\alpha) \exp \left( \alpha(TCC + \text{TCC}) \right) \)] when

\[
q_{ij} = \text{sgn}(\alpha) \ln M_{p_i} \left[ \alpha(n-j+1 + (\tau - \tau^{*+1}) / (1-\tau)) \right], \quad i = 1, \ldots, n, j = 1, \ldots, n.
\]

4. \( 1/\beta,\gamma \)-sist, \( \gamma_{[0]} = \tau, \quad 0 < \tau \neq 1 \)/E[\( \text{sgn}(\alpha) \exp \left( \alpha(TWT + \text{TCC}) \right) \)] when

\[
q_{ij} = \text{sgn}(\alpha) \ln M_{p_i} \left[ \alpha(n-j+1 + (\tau - \tau^{*+1}) / (1-\tau)) \right], \quad i = 1, \ldots, n, j = 1, \ldots, n.
\]

Since \( G_s(TL, TCC) = G_{\alpha}(TCT, TCC) \exp \left( -\alpha \sum_{i=1}^n d_i \right) \) and \( G_s(TL, TWC) = G_{\alpha}(TCT, TWC) \exp \left( -\alpha \sum_{i=1}^n d_i \right) \), \( S^* \) for \( 1/\beta,\gamma \)-sist/E[\( \text{sgn}(\alpha) \exp \left( \alpha(TCT + TCC) \right) \)] is optimal for \( 1/\beta,\gamma \)-sist/E[\( \text{sgn}(\alpha) \exp \left( \alpha(TL + TCC) \right) \)], and \( S^* \) for \( 1/\beta,\gamma \)-sist/E[\( \text{sgn}(\alpha) \exp \left( \alpha(TCT + TWC) \right) \)] is optimal for \( 1/\beta,\gamma \)-sist/E[\( \text{sgn}(\alpha) \exp \left( \alpha(TL + TWC) \right) \)].

5. **The scenario with fractional cost function**

Consider the cost function \( g(C_1, C_2) = \text{sgn}(\alpha) \exp(\alpha C_1) C_2 \) where \( \text{sgn}(\alpha) = 1 \) if \( \alpha > 0 \), and \( \text{sgn}(\alpha) = -1 \) if \( \alpha < 0 \). We remark that when \( \alpha > 0 \), \( g(C_1, C_2) \) is convex, non-decreasing
in \( C_1 \), non-increasing in \( C_2 \), and models risk-averse schedulers and when \( \alpha < 0 \), \( g(C_1, C_2) \) is concave, non-decreasing in both \( C_1 \) and \( C_2 \), and models risk-prone schedulers.

The use of \( g(C_1, C_2) = \text{sgn}(\alpha)\exp(\alpha C_1)/C_2 \) with \( \alpha > 0 \) can be justified in scheduling environments where the minimisation of \( C_1 \) and the maximisation of \( C_2 \) is required. For example, a scheduler may need to find the optimal sequence that simultaneously minimises \( \text{TCT} \) and maximises the total reliability of jobs where the latter is defined as the product of the random reliabilities of jobs in a sequence. We define the reliability of a job as the processing quality of the finished job. Two types of reliability models are considered here. First, the random reliability \( r_{ij} \leq r_{ij} \leq h_{ij} \) of job \([i], [i] = 1, \ldots, n \), where \( 0 < e_{ij} < h_{ij} \leq 1 \) are known constants, depends solely on the job position \( i, i = 1, \ldots, n \).

In this job-position-dependent model, the total random reliability of jobs in \( S \) is defined as \( \text{JPDR} = \prod_{i=1}^{n} r_{ij} \). Second, the random reliability \( e_{i-1,j} \leq r_{i-1,j} \leq h_{i-1,j} \) of job \([i], [i] = 1, \ldots, n \), where \( 0 < e_{i-1,j} < h_{i-1,j} \leq 1 \) are known constants, depends on its immediately preceding job \([i-1]\). Here, \( r_{01} \) is the reliability of job \([1]\), \([1] = 1, \ldots, n \), when scheduled first. In this adjacent-job-dependent model, the total random reliability of jobs in \( S \) is defined as \( \text{AJDR} = \prod_{i=1}^{n} r_{i-1,j} \).

The use of \( g(C_1, C_2) = \text{sgn}(\alpha)\exp(\alpha C_1)/C_2 \) with \( \alpha < 0 \) is appropriate in scheduling systems where the joint minimisation of \( C_1 \) and \( C_2 \) is required. For instance, a scheduler may need to find the optimal sequence that simultaneously minimises \( \text{TADC} \) and the total probability of job failure where the latter is defined as the product of the probabilities of failure of jobs in a sequence. Similar to the case with \( g(C_1, C_2) = \text{sgn}(\alpha)\exp(\alpha C_1)/C_2 \), \( \alpha > 0 \), we can define two types of failure models. In the job-position-dependent model, the probability of failure of job \([i], [i] = 1, \ldots, n \), is also denoted by \( r_{ij} \) where \( e_{ij} \leq r_{ij} \leq h_{ij} \). \( 0 < e_{ij} < h_{ij} \leq 1 \), \( i = 1, \ldots, n \), and the total probability of failure of jobs in \( S \) is \( \text{JPDR} = \prod_{i=1}^{n} r_{ij} \). In the adjacent-job-dependent model, the probability of failure of job \([i], [i] = 1, \ldots, n \), which immediately follows job \([i-1]\), is also denoted by \( r_{i-1,j} \), where \( e_{i-1,j} \leq r_{i-1,j} \leq h_{i-1,j} \). \( 0 < e_{i-1,j} \leq h_{i-1,j} \leq 1 \). \( r_{01} \) is the probability of failure of job \([1]\), \([1] = 1, \ldots, n \), when scheduled first, and the total probability of failure is \( \text{AJDR} = \prod_{i=1}^{n} r_{i-1,j} \).

The objective function \( G_x(C_1, C_2) \) for \( 1/E[\text{sgn}(\alpha)\exp(\alpha C_1)/C_2] \) is defined as

\[
G_x(C_1, C_2) = \text{sgn}(\alpha)E\left[\exp(\alpha C_1)/C_2\right] 
\]

where \( \text{sgn}(\alpha) = 1 \) if \( \alpha > 0 \), \( \text{sgn}(\alpha) = -1 \) if \( \alpha < 0 \), \( C_1 \) is defined as a function of \( p_i \), \( i = 1, \ldots, n \), and \( b_{ij}, i \neq j, i = 0, 1, \ldots, n, j = 1, \ldots, n \), and \( C_2 \) is either JPDR or AJDR. Since \( C_1 \) is independent of \( C_2 \), using equation (20), we have

\[
G_x(C_1, C_2) = \text{sgn}(\alpha)M_{C_1}(\alpha)E[1/C_2]. 
\]

Hence, using equations (1) and (21), \( S^* \) for \( 1/E[\text{sgn}(\alpha)\exp(\alpha C_1)/C_2] \) minimises a function of the MGF of \( C_1 \) and the expected value of \( 1/C_2 \).

In what follows, we derive \( S^* \) for \( 1/b_{ij}\text{std}(\text{sgn}(\alpha)\exp(\alpha C_1)/C_2) \) and its special case \( 1/b_{ij}\text{std}(\text{sgn}(\alpha)\exp(\alpha C_1)/C_2) \) where \( C_2 = \text{JPDR}, \text{AJDR} \).
5.1 The 1/E[\text{sgn}(\alpha)\exp(\alpha C_i)/\text{JPDR}] scheduling problem

In 1/E[\text{sgn}(\alpha)\exp(\alpha C_i)/\text{JPDR}] where \text{JPDR} = \prod_{i=1}^{n} e_{ij}/r_{ij}, \ e_{ij} \leq r_{ij} \leq h_{ij}, \ 0 < e_{ij} < h_{ij} \leq 1, we can define the reliability \ r_{ij} \ of job \ [i], \ [i] = 1, \ldots, n, appearing in position \ i, \ i = 1, \ldots, n, \ of \ S \ in two different ways. First, \ r_{ij} \ is general in the sense that there are no relations among \ r_{ij} \ of job \ [i], \ [i] = 1, \ldots, n, \ when \ assigned \ to \ different \ positions \ i, \ i = 1, \ldots, n \ (i.e., \ reliabilities \ are \ position-independent). Second, \ r_{ij} \ is a function of \ the \ job \ position \ i, \ i = 1, \ldots, n, \ defined \ as \ r_{ij} = \beta^{-1} r_{ij} \ where \ \beta > 0 \ is \ a \ constant \ reliability \ index \ and \ 0 < r_{ij} \leq 1 \ is \ the \ random \ reliability \ of \ job \ [i], \ [i] = 1, \ldots, n, \ when \ scheduled \ first \ (i.e., \ reliabilities \ are \ position-dependent). Here, \ r_{ij} \ is \ a \ decreasing \ function \ of \ i \ if \ \beta > 1, \ and \ an \ increasing \ function \ of \ i \ if \ \beta < 1 \ (r_{ij} = r_{ij} \ if \ \beta = 1). \ The \ function \ r_{ij} = \beta^{-1} r_{ij}, \ 0 < \beta < 1, \ i = 1, \ldots, n, \ is \ appropriate \ in \ scheduling \ systems \ where \ the \ reliability \ of \ a \ job \ deteriorates \ with \ the \ job \ position \ in \ a \ sequence. Thus, \ job \ [i] \ has \ the \ highest \ reliability \ of \ r_{ij} \ when \ scheduled \ first, \ and \ the \ lowest \ reliability \ of \ r_{ij} \ when \ scheduled \ last. For \ example, \ when \ drilling \ holes \ into \ different \ alloys, \ the \ tool \ wears \ out \ with \ each \ alloy; \ thus, \ the \ reliability \ of \ a \ hole \ is \ higher \ if \ the \ alloy \ is \ scheduled \ earlier. \ The \ use \ of \ r_{ij} = \beta^{-1} r_{ij}, \ \beta > 1, \ i = 1, \ldots, n, \ can \ be \ justified \ in \ scheduling \ environments \ where \ the \ reliability \ of \ a \ job \ is \ higher \ when \ scheduled \ later \ owing \ to \ the \ learning \ phenomenon. In \ this \ case, \ a \ job \ [i] \ has \ the \ lowest \ reliability \ of \ r_{ij} \ when \ scheduled \ first, \ and \ the \ highest \ reliability \ of \ r_{ij} \ when \ scheduled \ last. We \ solve \ here \ 1/b_{ij}-\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha C_i)/\text{JPDR}] \ and \ 1/b_{ij}-\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha C_i)/\text{JPDR}] \ where, \ using \ equation \ (21), \ G_d(C_1, \text{JPDR}) \ is \ given \ by

\[G_d(C_1, \text{JPDR}) = \text{sgn}(\alpha) M_{\text{C}_1}(\alpha) E\left[1/\prod_{i=1}^{n} r_{ij}\right]. \quad (22)\]

5.1.1 1/b_{ij}-\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha C_i)/\text{JPDR}] with \ C_i = C_{\text{max}}, TADC, TCT, TWT, TCC, TWC and TL

The \ G_d(C_{\text{max}}, \text{JPDR}) \ for \ 1/b_{ij}-\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha C_{\text{max}})/\text{JPDR}], \ using \ equation \ (22), \ is \ computed \ as

\[G_d(C_{\text{max}}, \text{JPDR}) = \text{sgn}(\alpha) M_{\sum_{i=1}^{n} (r_{ij} + h_{ij})} (\alpha) E[1/r_{ij}]
\quad = \text{sgn}(\alpha) \prod_{i=1}^{n} M_{r_{ij}} (\alpha) M_{h_{ij}} (\alpha) E[1/r_{ij}].\]

Using arguments similar to those of Section 4.1.1, we can show that \ S^* \ for \ 1/b_{ij}-\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha C_{\text{max}})/\text{JPDR}] \ is \ such \ that

\[S^* = \arg \min_{S \in \Psi} \left\{ \sum_{i=1}^{n} \text{sgn}(\alpha) \left[ \ln M_{r_{ij}} (\alpha) + \ln M_{h_{ij}} (\alpha) + \ln E[1/r_{ij}] \right] \right\}.\]

Therefore, \ S^* \ can be equivalently determined by solving QAP with objective function (4) where
\[ q_j = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{\mu}(\alpha) + \ln M_{\nu}(\alpha) + \ln E[1/r_j] \right], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \left[ \ln M_{\mu}(\alpha) + \ln E[1/r_j] \right], & \text{otherwise}; 
\end{cases} \]

and

\[ u_{ijk} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{\nu}(\alpha), & i, j, k = 1, \ldots, n, k \neq i; \\
0, & \text{otherwise}. 
\end{cases} \]

where \( r_j \) is the random reliability of job \( i \) when assigned to position \( j \) of a sequence.

For \( 1/b_{ij}'-\text{dst} \), \( r_{ij} = \beta^{-1} r_{ij}/E[\text{sgn}(\alpha)\exp(\alpha CT_{\text{TADC}})/\text{JPDR}] \) since \( \ln E[1/r_{ij}] = (1 - j) \ln B + \ln E[1/r_i] \), \( S^* \) is the solution to the QAP where \( u_{ijk} \) are defined above, and

\[ q_j = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{\mu}(\alpha) + \ln M_{\nu}(\alpha) + (1 - j) \ln \beta + \ln E[1/r_j] \right], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \left[ \ln M_{\mu}(\alpha) + (1 - j) \ln \beta + \ln E[1/r_j] \right], & \text{otherwise}; 
\end{cases} \]

Likewise, solving the QAP provides \( S^* \) for

\[ i \quad 1/b_{ij}'-\text{dst}/E[\text{sgn}(\alpha)\exp(\alpha CT_{\text{TADC}})/\text{JPDR}] \]

when

\[ q_j = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{\mu}(\alpha) + \ln M_{\nu}(\alpha) + \ln E[1/r_j] \right], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \left[ \ln M_{\mu}(\alpha) + \ln E[1/r_j] \right], & \text{otherwise}; 
\end{cases} \]

and

\[ u_{ijk} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{\nu}(\alpha), & i, j, k = 1, \ldots, n, k \neq i; \\
0, & \text{otherwise}; 
\end{cases} \]

and for \( 1/b_{ij}'-\text{dst} \), \( r_{ij} = \beta^{-1} r_{ij}/E[\text{sgn}(\alpha)\exp(\alpha CT_{\text{TADC}})/\text{JPDR}] \) when \( u_{ijk} \) are defined above, and

\[ q_j = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{\mu}(\alpha) + \ln M_{\nu}(\alpha) + (1 - j) \ln \beta + \ln E[1/r_j] \right], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \left[ \ln M_{\mu}(\alpha) + (1 - j) \ln \beta + \ln E[1/r_j] \right], & \text{otherwise}. 
\end{cases} \]

\[ i \quad 1/b_{ij}'-\text{dst}/E[\text{sgn}(\alpha)\exp(\alpha CT_{\text{TCT}})/\text{JPDR}] \]

when

\[ q_j = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{\mu}(\alpha) + \ln M_{\nu}(\alpha) + \ln E[1/r_j] \right], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \left[ \ln M_{\mu}(\alpha) + \ln E[1/r_j] \right], & \text{otherwise}; 
\end{cases} \]

and

\[ u_{ijk} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{\nu}(\alpha), & i, j, k = 1, \ldots, n, k \neq i; \\
0, & \text{otherwise}; 
\end{cases} \]
and for $1/b_{ij}$-sdst, $r_{ij} = \beta^{-1} r_{ij}/E[\text{sgn}(\alpha)\exp(\alpha T_{CT})/\text{JPDR}]$ when $u_{ijkj+1}$, $i, j, k = 1, \ldots, n$, $k \neq i$ are defined above, and

$$q_y = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{h_i} (\alpha(n-j+1))^+ \ln M_{h_i} [\alpha(n-j+1)]^+ \right], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \left[ \ln M_{h_i} (\alpha(n-j+1)) + (1-j) \ln \beta + \ln E[1/r_i] \right], & \text{otherwise.}
\end{cases}$$

and for $1/b_{ij}-\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha T_{WT})/\text{JPDR}]$ when

$$q_y = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{h_i} [\alpha(n-j)]^+ + \ln M_{h_i} [\alpha(n-j)]^+ \right], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \left[ \ln M_{h_i} [\alpha(n-j)] + \ln E[1/r_i] \right], & \text{otherwise;}
\end{cases}$$

and

$$u_{ijkj+1} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{h_i} [\alpha(n-j)], & i, j, k = 1, \ldots, n, k \neq i; \\
0, & \text{otherwise;}
\end{cases}$$

and for $1/b_{ij}-\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha T_{CC})/\text{JPDR}]$ when $u_{ijkj+1}$, $i, j, k = 1, \ldots, n$, $k \neq i$ are defined above, and

$$q_y = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{h_i} [\alpha(n-j)] + \ln M_{h_i} [\alpha(n-j)] \right] / (1-j) \ln \beta + \ln E[1/r_i], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \left[ \ln M_{h_i} [\alpha(n-j)] + \ln E[1/r_i] \right], & \text{otherwise.}
\end{cases}$$

and

$$u_{ijkj+1} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{h_i} [\alpha(n-j)] / (1-j), & i, j, k = 1, \ldots, n, k \neq i; \\
0, & \text{otherwise;}
\end{cases}$$

and for $1/b_{ij}$-sdst, $\gamma_{[ij]} = \tau^i, 0 < \tau \neq 1/E[\text{sgn}(\alpha)\exp(\alpha T_{CT})/\text{JPDR}]$ when $u_{ijkj+1}$, $i, j, k = 1, \ldots, n, k \neq i$ are defined above, and

$$q_y = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{h_i} [\alpha(\tau^{-1})/(1-\tau)] + \ln M_{h_i} [\alpha(\tau^{-1})/(1-\tau)] \right], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \left[ \ln M_{h_i} [\alpha(\tau^{-1})/(1-\tau)] + \ln E[1/r_i] \right], & \text{otherwise;}
\end{cases}$$

and

$$u_{ijkj+1} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{h_i} [\alpha(\tau^{-1})/(1-\tau)], & i, j, k = 1, \ldots, n, k \neq i; \\
0, & \text{otherwise;}
\end{cases}$$

and for $1/b_{ij}$-sdst, $r_{ij} = \beta^{-1} r_{ij}/E[\text{sgn}(\alpha)\exp(\alpha T_{CC})/\text{JPDR}]$ when $u_{ijkj+1}$, $i, j, k = 1, \ldots, n, k \neq i$ are defined above, and

$$q_y = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{h_i} [\alpha(\tau^{-1})/(1-\tau)] \right] / \ln E[1/r_i], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \left[ \ln M_{h_i} [\alpha(\tau^{-1})/(1-\tau)] + (1-j) \ln \beta + \ln E[1/r_i] \right], & \text{otherwise.}
\end{cases}$$
Since \( u_{ij+kj} = \beta \), \( 0 < \tau \neq 1/E[\text{sgn}(\alpha)\exp(\alpha TWC)/\text{JPDR}] \) when

\[
q_j = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{\alpha}[\alpha(\tau^{i+1} - \tau^{i})]/(1-\tau) \right] + \ln E[1/r_i], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \left[ \ln M_{\alpha}[\alpha(\tau^{i+1} - \tau^{i})]/(1-\tau) \right] + \ln E[1/r_i], & \text{otherwise}; 
\end{cases}
\]

and

\[
u_{ij+kj} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{\alpha}[\alpha(\tau^{i+1} - \tau^{i})]/(1-\tau), & i, j, k = 1, \ldots, n, k \neq i; \\
0, & \text{otherwise}; 
\end{cases}
\]

and for \( 1/b_j-sdst, r_{ij} = \beta^{-1}r_{ij}, \gamma_{ij} = \tau, 0 < \tau \neq 1/E[\text{sgn}(\alpha)\exp(\alpha TWC)/\text{JPDR}] \) when

\[
u_{ij+kj} = \begin{cases} 
\text{sgn}(\alpha) \ln M_{\alpha}[\alpha(\tau^{i+1} - \tau^{i})]/(1-\tau), & i, j, k = 1, \ldots, n, k \neq i; \\
0, & \text{otherwise}; 
\end{cases}
\]

and for \( 1/b_j-sdst, r_{ij} = \beta^{-1}r_{ij}, \gamma_{ij} = \tau, 0 < \tau \neq 1/E[\text{sgn}(\alpha)\exp(\alpha TWC)/\text{JPDR}] \) when

\[
q_j = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{\alpha}[\alpha(\tau^{i+1} - \tau^{i})]/(1-\tau) \right] + \ln E[1/r_i], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \left[ \ln M_{\alpha}[\alpha(\tau^{i+1} - \tau^{i})]/(1-\tau) \right] + \ln E[1/r_i], & \text{otherwise}. 
\end{cases}
\]

Since \( G(\beta TL, \text{JPDR}) = G(\beta TCT, \text{JPDR}) \exp \left( -\alpha \sum_{i=0}^{n} d_{ii} \right) \), \( S^* \) for \( 1/b_j-sdst/E[\text{sgn}(\alpha)\exp(\alpha TCT)/\text{JPDR}] \) is optimal for \( 1/b_j-sdst/E[\text{sgn}(\alpha)\exp(\alpha TL)/\text{JPDR}] \).

### 5.1.2 Special case: \( 1/b_j-sdst/E[\text{sgn}(\alpha)\exp(\alpha C_j)/\text{JPDR}] \) with \( C_j = \text{TADC}, \text{TCT}, \text{TWT}, \text{TCC}, \text{TWC} \) and \( \text{TL} \)

Discarding the terms containing \( b_{[i \rightarrow j]} \) (or \( M_{\alpha(i \rightarrow j)} \)), \( i = 1, \ldots, n \) (i.e., \( b_j \) (or \( M_j \)), \( i \neq j \), \( i = 0, 1, \ldots, n \), \( j = 1, \ldots, n \), from the QAP formulation for \( 1/b_j-sdst/E[\text{sgn}(\alpha)\exp(\alpha C_j)/\text{JPDR}] \) (see Section 5.1.1), an AP formulation is obtained for \( 1/b_j-sdst/E[\text{sgn}(\alpha)\exp(\alpha C_j)/\text{JPDR}] \). Hence, the solution to AP yields \( S^* \) for

i) \( 1/b_j-sdst/E[\text{sgn}(\alpha)\exp(\alpha \text{TADC})/\text{JPDR}] \) when

\[
q_j = \text{sgn}(\alpha) \left[ \ln M_{\alpha}[\alpha(j-1)(n-j+1)] + \ln E[1/r_i] \right], \ i = 1, \ldots, n, j = 1, \ldots, n; 
\]

and for \( 1/b_j-sdst, r_{ij} = \beta^{-1}r_{ij} \) when

\[
q_j = \text{sgn}(\alpha) \left[ \ln M_{\alpha}[\alpha(j-1)(n-j+1)] + (1-j) \ln \beta + \ln E[1/r_i] \right], 
\]

\[
i = 1, \ldots, n, j = 1, \ldots, n. 
\]

ii) \( 1/b_j-sdst/E[\text{sgn}(\alpha)\exp(\alpha \text{TCT})/\text{JPDR}] \) when

\[
q_j = \text{sgn}(\alpha) \left[ \ln M_{\alpha}[\alpha(n-j+1)] + \ln E[1/r_i] \right], \ i = 1, \ldots, n, j = 1, \ldots, n; 
\]

and for \( 1/b_j-sdst, r_{ij} = \beta^{-1}r_{ij} \) when

\[
q_j = \text{sgn}(\alpha) \left[ \ln M_{\alpha}[\alpha(n-j+1)] + (1-j) \ln \beta + \ln E[1/r_i] \right], \ i = 1, \ldots, n, j = 1, \ldots, n. 
\]
Using argument similar to those of Section 5.1.1, we can show that

\[ q_i = \text{sgn}(\alpha) [\ln M_{n_i} (\alpha (n - j)) + \ln E[1/r_i]], \quad i = 1, \ldots, n, j = 1, \ldots, n; \]

and for 1/bij-sist/E[\text{sgn}(\alpha)\exp(\alpha TWT)/\text{JPDR}] when

\[ q_i = \text{sgn}(\alpha) [\ln M_{n_i} (\alpha (n - j)) + (1 - j) \ln \beta + \ln E[1/r_i]], \quad i = 1, \ldots, n, j = 1, \ldots, n. \]

The \( S^* \) for 1/bij-sist/E[\text{sgn}(\alpha)\exp(\alpha TWT)/\text{JPDR}] is optimal for 1/bij-sist/E[\text{sgn}(\alpha)\exp(\alpha TL)/\text{JPDR}] since \( G_{S}(\text{TL}, \text{JPDR}) = G_{S}(\text{TCT}, \text{JPDR}) \exp \left[ -\alpha \sum_{i=1}^{n} d_i \right] \).

### 5.2 The 1/E[\text{sgn}(\alpha)\exp(\alpha C_{ij})/\text{AJDR}] scheduling problem

In 1/E[\text{sgn}(\alpha)\exp(\alpha C_{ij})/\text{AJDR}] where AJDR = \prod_{i=1}^{n} e_{ij}, e_{ij} \leq r_{ij} \leq h_{ij}, and 0 < e_{ij} < h_{ij} \leq 1, using equation (21), we have

\[ G_{S}(C_{ij}, \text{AJDR}) = \text{sgn}(\alpha) M_{M_{C_{ij}}}(\alpha) E \left[ 1/\prod_{i=1}^{n} e_{ij} \right]. \tag{23} \]

In what follows, we solve 1/bij-sdst/E[\text{sgn}(\alpha)\exp(\alpha C_{ij})/\text{AJDR}] and 1/bij-sist/E[\text{sgn}(\alpha)\exp(\alpha C_{ij})/\text{AJDR}]

#### 5.2.1 1/bij-sdst/E[\text{sgn}(\alpha)\exp(\alpha C_{ij})/\text{AJDR}] with \( C_i = C_{\text{max}}, \text{TADC}, \text{TCT}, \text{TWT}, \text{TCC}, \text{TWC and TL} \)

The \( G_{S}(C_{\text{max}}, \text{JPDR}) \) for 1/bij-sdst/E[\text{sgn}(\alpha)\exp(\alpha C_{\text{max}})/\text{AJDR}], using equation (23), is

\[ G_{S}(C_{\text{max}}, \text{JPDR}) = \text{sgn}(\alpha) \prod_{i=1}^{n} M_{M_{C_{\text{max}}}}(\alpha) E \left[ 1/\prod_{i=1}^{n} e_{ij} \right] \]

Using argument similar to those of Section 5.1.1, we can show that

\[ S^* = \arg \min_{S_{\alpha}} \left\{ \text{sgn}(\alpha) \sum_{i=1}^{n} \left[ \ln M_{M_{C_{ij}}}(\alpha) + \ln M_{M_{C_{\text{max}}}}(\alpha) + \ln E[1/r_{ij}] \right] \right\}. \]
Equivalently, $S^*$ can be obtained by solving QAP with objective function (4) where

$$q_j = \begin{cases} \text{sgn}(\alpha) \left[ \ln M_{h_i} (\alpha) + \ln M_{h_j} (\alpha) + \ln E[1/r_{a_k}] \right], & j = 1, i = 1, \ldots, n; \\ \text{sgn}(\alpha) \ln M_{h_i} (\alpha), & \text{otherwise}; \end{cases}$$

and

$$u_{ij} = \begin{cases} \text{sgn}(\alpha) \left[ \ln M_{h_i} (\alpha) + \ln E[1/r_{a_k}] \right], & i, j, k = 1, \ldots, n, k \neq i; \\ 0, & \text{otherwise}. \end{cases}$$

where $r_{a_k}$ is the random reliability of job $k$ when scheduled immediately after job $i$, $k \neq i$, $i = 0, 1, \ldots, n, k = 1, \ldots, n$.

Likewise, the solution to QAP yields $S^*$ for

i 1/bjs-dstd/E[sgn($\alpha$)exp($\alpha$TADC)/AJDR] when

$$q_j = \begin{cases} \text{sgn}(\alpha) \left[ \ln M_{h_i} (\alpha) + \ln M_{h_j} (\alpha) + \ln E[1/r_{a_k}] \right], & j = 1, i = 1, \ldots, n; \\ \text{sgn}(\alpha) \ln M_{h_i} (\alpha), & \text{otherwise}; \end{cases}$$

and

$$u_{ij} = \begin{cases} \text{sgn}(\alpha) \left[ \ln M_{h_i} (\alpha) + \ln E[1/r_{a_k}] \right], & i, j, k = 1, \ldots, n, k \neq i; \\ 0, & \text{otherwise}. \end{cases}$$

ii 1/bjs-dstd/E[sgn($\alpha$)exp($\alpha$TCT)/AJDR] when

$$q_j = \begin{cases} \text{sgn}(\alpha) \left[ \ln M_{h_i} (\alpha) + \ln M_{h_j} (\alpha) + \ln E[1/r_{a_k}] \right], & j = 1, i = 1, \ldots, n; \\ \text{sgn}(\alpha) \ln M_{h_i} (\alpha), & \text{otherwise}; \end{cases}$$

and

$$u_{ij} = \begin{cases} \text{sgn}(\alpha) \left[ \ln M_{h_i} (\alpha) + \ln E[1/r_{a_k}] \right], & i, j, k = 1, \ldots, n, k \neq i; \\ 0, & \text{otherwise}. \end{cases}$$

iii 1/bjs-dstd/E[sgn($\alpha$)exp($\alpha$TWT)/AJDR] when

$$q_j = \begin{cases} \text{sgn}(\alpha) \left[ \ln M_{h_i} (\alpha) + \ln M_{h_j} (\alpha) + \ln E[1/r_{a_k}] \right], & j = 1, i = 1, \ldots, n; \\ \text{sgn}(\alpha) \ln M_{h_i} (\alpha), & \text{otherwise}; \end{cases}$$

and

$$u_{ij} = \begin{cases} \text{sgn}(\alpha) \left[ \ln M_{h_i} (\alpha) + \ln E[1/r_{a_k}] \right], & i, j, k = 1, \ldots, n, k \neq i; \\ 0, & \text{otherwise}. \end{cases}$$
iv $1/b_{ij}-sdist, \gamma_{ij} = \tau, 0 < \tau \neq 1/E[\text{sgn}(\alpha)\exp(\alpha TCC)/AJDR]$ when

$$q_b = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{\mu_b} [\alpha(\tau' - \tau^{*i})/(1-\tau)] + \ln M_{b_0} [\alpha(\tau' - \tau^{*i})/(1-\tau)] \right], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{\mu_b} [\alpha(\tau' - \tau^{*i})/(1-\tau)], & \text{otherwise}; 
\end{cases}$$

and

$$u_{ijk} = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{\mu_b} [\alpha(\tau' - \tau^{*i})/(1-\tau)] + \ln E[1/r_{ij}] \right], & i, j, k = 1, \ldots, n, k \neq i; \\
0, & \text{otherwise}. 
\end{cases}$$

v $1/b_{ij}-sdist, \gamma_{ij} = \tau, 0 < \tau \neq 1/E[\text{sgn}(\alpha)\exp(\alpha TWC)/AJDR]$ when

$$q_b = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{\mu_b} [\alpha(\tau' - \tau^{*i})/(1-\tau)] + \ln M_{b_0} [\alpha(\tau' - \tau^{*i})/(1-\tau)] + \ln E[1/r_{ij}] \right], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{\mu_b} [\alpha(\tau' - \tau^{*i})/(1-\tau)], & \text{otherwise}; 
\end{cases}$$

and

$$u_{ijk} = \begin{cases} 
\text{sgn}(\alpha) \left[ \ln M_{\mu_b} [\alpha(\tau' - \tau^{*i})/(1-\tau)] + \ln E[1/r_{ij}] \right], & i, j, k = 1, \ldots, n, k \neq i; \\
0, & \text{otherwise}. 
\end{cases}$$

Since $G_{ij}(TL, AJDR) = G_{ij}(TCT, AJDR) \exp(-\alpha \sum_{j=1}^{n} d_{ij})$, $S^*$ for $1/b_{ij}-sdist/E[\text{sgn}(\alpha)\exp(\alpha TCT)/AJDR]$ is optimal for $1/b_{ij}-sdist/E[\text{sgn}(\alpha)\exp(\alpha TWC)/AJDR]$. Here, QAP has the objective function (4) with the interaction costs $u_{ijk} = \text{sgn}(\alpha) \ln E[1/r_{ij}]$, $i, j, k = 1, \ldots, n, k \neq i$. Thus, solution to QAP produces $S^*$ for

i $1/b_{ij}-sdist/E[\text{sgn}(\alpha)\exp(\alpha C_{max})/JPDR]$ when

$$q_b = \begin{cases} 
\text{sgn}(\alpha) [\ln M_{\mu} (\alpha) + \ln E[1/r_{ij}]], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{\mu} (\alpha), & \text{otherwise}; 
\end{cases}$$

$(r_{ij})$ is the random reliability of job $j$ when scheduled first)

ii $1/b_{ij}-sdist/E[\text{sgn}(\alpha)\exp(\alpha TADC)/JPDR]$ when

$$q_b = \begin{cases} 
\text{sgn}(\alpha) [\ln M_{\mu} (\alpha(j-1)(n-j+1)) + \ln E[1/r_{ij}]], & j = 1, i = 1, \ldots, n; \\
\text{sgn}(\alpha) \ln M_{\mu} (\alpha(j-1)(n-j+1)), & \text{otherwise}. 
\end{cases}$$
Finally, since $G_{\delta}(TL, AJDR) = G_{\delta}(TCT, AJDR) \exp\left(-\alpha \sum_{i=1}^{n} d_{ij}\right)$, $S^*$ for $1/bij-sdst/E[\text{sgn}(\alpha)\exp(\alpha TCT)/AJDR]$ is optimal for $1/bij-sdst/E[\text{sgn}(\alpha)\exp(\alpha TL)/AJDR]$.

### 6 Computational results

We carry out some computational experiments on the exact and heuristic solution methods for the QAP formulations of a number of $1/bij-sdst/E[\text{sgn}(\alpha)\exp(\alpha TCT)/AJDR]$ problems. For comparison purposes, we consider $1/bij-sdst/E[\alpha TCT + (1 - \alpha)TADC]$ where $\alpha$ is randomly generated from $U(0, 1)$, $1/bij-sdst/E[\text{sgn}(\alpha)\exp(\alpha C_{\text{max}} + TWT)]$, $1/bij-sdst/sgn(\alpha)\exp(\alpha TADC)/JPDR$, and $1/bij-sdst/E[\exp(\alpha TCT)/AJDR]$ where $\alpha \neq 0$ for the latter three problems are sampled from $U(-0.5, 0.5)$. Thus, cost functions for schedulers with risk-taking behaviour including risk neutral, risk averse and risk prone are generated. For $1/bij-sdst/E[\alpha TCT + (1 - \alpha)TADC]$, $p_{ij}$, for $i = 0, 1, \ldots, n$, $j = 1, \ldots, n$, are sampled from $U(0, 1)$ and $U(0.0, 0.1)$, respectively. For $1/bij-sdst/E[\text{sgn}(\alpha)\exp(\alpha C_{\text{max}} + TWT)]$, $1/bij-sdst/sgn(\alpha)\exp(\alpha TADC)/JPDR$ and $1/bij-sdst/E[\exp(\alpha TCT)/AJDR]$, the probability distributions for $p_{ij}$, $i = 0, 1, \ldots, n$, $j = 1, \ldots, n$, are assumed uniform where the lower and upper bounds for $p_{ij}$ are, respectively, sampled from $U(1, 5)$ and $U(15, 20)$, for $b_{ij}$ from $U(0.5, 3.5)$ and $U(7, 10)$, and for $r_{ij}$ from $U(0, 0.4)$ and $U(0.6, 1)$.

The experiments are run on a computer with Intel core 2 due, 2.24 GHz processor and with 1.99 GB RAM. For each problem size and type, 10 generated instances are solved exactly using Cplex, and heuristically using our modified version of the Integer Linear Programming (ILP) formulation of Zhang et al. (2010). This ILP, which is based on the well-known linearisation formulation of Adams and Johnson (1994), is given as
min \sum_{j \in S} c_{ij} y_j + \sum_{1 \leq i \leq n} \sum_{1 \leq j \leq n} \tilde{u}_{ijk} y_{ijk}\]

where \(c_{ij} = q_{ijk} + u_{ijk}\) and \(\tilde{u}_{ijk} = \min \{u_{ijk}, u_{ij}\}\)

subject to

\[\sum_{j \in S} y_{ijk} = x_{ij}, \ i, j, k = 1, \ldots, n, i \neq k; \ \sum_{j \in S} y_{ijk} \leq x_{ij}, \ i, k, \ell = 1, \ldots, n, i \neq k;\]

\[\sum_{j = 1}^{n} x_{ij} = 1, \ i = 1, \ldots, n; \ \sum_{j = 1}^{n} x_{ij} = 1, \ i = 1, \ldots, n;\]

\[x_{ij} = 0, 1, \ i, j = 1, \ldots, n; \ y_{ijk} = 0, 1, i, j, k, \ell = 1, \ldots, n, i \neq k.\]

Table 1 displays the computational results on \(1/b_j\cdot\text{sdst}/E[\alpha\text{TCT}] + (1 - \alpha\text{TADC})\). The first column of the table shows the problem size, the second and third columns report the average CPU times (in seconds) for the exact and heuristic methods, respectively, the fourth column gives the number of heuristic solutions for the ten instances of each size problem that turned out to be optimal, and the last column displays the average percentage of gaps between the objective values of the optimal and heuristic solutions. The percentage of gap is defined by \(100 \times \frac{\text{heuristic objective value} - \text{optimal objective value}}{\text{optimal objective value}}\%\). As the table shows, the heuristic found the optimal sequences for at least 60% of problem instances with up to 11 jobs. As expected, the heuristic’s CPU times are much lower than those of the exact method. We did not obtain exact solutions for problems with more than 11 jobs owing to their excessive time requirements, which surpassed our time limit of 6000 CPU seconds; however, such problems were solved heuristically with attractive CPU times.

Table 2 shows the performance of the exact and heuristic methods for \(1/b_j\cdot\text{sdst}/E[\alpha\text{Cmax} + \text{TWT}]\). In comparison with \(1/b_j\cdot\text{sdst}/E[\alpha\text{TCT}] + (1 - \alpha\text{TADC})\), the heuristic here finds the optimal sequences for more instances (at least 80%) of the problems with up to 11 jobs with lower percentages of gap. In addition, the CPU times of Tables 1 and 2 indicate that \(1/b_j\cdot\text{sdst}/E[\alpha\text{TCT}] + (1 - \alpha\text{TADC})\) is harder to solve than \(1/b_j\cdot\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha\text{Cmax} + \text{TWT})]\) both exactly and heuristically. As the problem size increases, \(1/b_j\cdot\text{sdst}/E[\alpha\text{TCT}] + (1 - \alpha\text{TADC})\) becomes more difficult to solve than \(1/b_j\cdot\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha\text{Cmax} + \text{TWT})]\). This points out that the nature of cost function and the two criteria affect the performance of the solution methods for QAP and thus the problem difficulty.

Table 3 displays the computational results on \(1/b_j\cdot\text{sdst}/\text{sgn}(\alpha)\exp(\alpha\text{TADC})/\text{JPRD}\). The performance of the heuristic (in terms of number of optimal solutions and percentages of gap) for problems with up to 11 jobs is better than that for \(1/b_j\cdot\text{sdst}/E[\alpha\text{TCT}] + (1 - \alpha\text{TADC})\), but worse than that for \(1/b_j\cdot\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha\text{Cmax} + \text{TWT})]\). In addition, \(1/b_j\cdot\text{sdst}/\text{sgn}(\alpha)\exp(\alpha\text{TADC})/\text{JPRD}\) is harder to solve than \(1/b_j\cdot\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha\text{Cmax} + \text{TWT})]\) but easier than \(1/b_j\cdot\text{sdst}/E[\alpha\text{TCT}] + (1 - \alpha\text{TADC})\).

Finally, Table 4 shows the results on \(1/b_j\cdot\text{sdst}/E[\exp(\alpha\text{TCT})]/\text{AJDR}\). Here, the heuristic’s performance is the worst when compared with those for \(1/b_j\cdot\text{sdst}/E[\alpha\text{TCT}] + (1 - \alpha\text{TADC})\), \(1/b_j\cdot\text{sdst}/E[\text{sgn}(\alpha)\exp(\alpha\text{Cmax} + \text{TWT})]\), and \(1/b_j\cdot\text{sdst}/\text{sgn}(\alpha)\exp(\alpha\text{TADC})/\text{JPRD}\) (see Tables 1–3). This indicates that \(1/b_j\cdot\text{sdst}/E\)
H.M. Soroush

[exp(\(\alpha\)TCT)/AJDR] is more difficult to solve than the other three problems owing to the non-regularity of both set-up times and AJDR. The comparison again demonstrates that the nature of cost function and the criteria affect the problem difficulty.

Table 1  
Computation results on 1/bij-sdst/E[\(\alpha\)TCT + (1 – \(\alpha\))TADC]

<table>
<thead>
<tr>
<th>Prob. size</th>
<th>CPU time (sec)</th>
<th>No. of heuristic solutions opt.</th>
<th>Average perc. gap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exact</td>
<td>Heuristic</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.125</td>
<td>0.064</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>0.149</td>
<td>0.126</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>0.416</td>
<td>0.167</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>0.789</td>
<td>0.315</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>39.672</td>
<td>1.204</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>531.209</td>
<td>2.863</td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td>4,582.358</td>
<td>18.291</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>*</td>
<td>49.347</td>
<td>na</td>
</tr>
<tr>
<td>13</td>
<td>*</td>
<td>251.935</td>
<td>na</td>
</tr>
<tr>
<td>14</td>
<td>*</td>
<td>529.782</td>
<td>na</td>
</tr>
<tr>
<td>15</td>
<td>*</td>
<td>802.439</td>
<td>na</td>
</tr>
<tr>
<td>16</td>
<td>*</td>
<td>1,476.613</td>
<td>na</td>
</tr>
<tr>
<td>17</td>
<td>*</td>
<td>2,565.164</td>
<td>na</td>
</tr>
<tr>
<td>18</td>
<td>*</td>
<td>3,943.902</td>
<td>na</td>
</tr>
</tbody>
</table>

*The time exceeded the time limit of 6000 CPU seconds.

Table 2  
Computation results on 1/bij-sdst/E[sgn(\(\alpha\))exp(\(\alpha\)Cmax + TWT)]

<table>
<thead>
<tr>
<th>Prob. size</th>
<th>CPU time (sec)</th>
<th>No. of heuristic solutions opt.</th>
<th>Average perc. gap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exact</td>
<td>Heuristic</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.098</td>
<td>0.052</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>0.121</td>
<td>0.095</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>0.327</td>
<td>0.121</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>0.618</td>
<td>0.184</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>33.562</td>
<td>0.753</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>401.255</td>
<td>1.842</td>
<td>8</td>
</tr>
<tr>
<td>11</td>
<td>2,843.138</td>
<td>15.651</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>*</td>
<td>38.589</td>
<td>na</td>
</tr>
<tr>
<td>13</td>
<td>*</td>
<td>221.466</td>
<td>na</td>
</tr>
<tr>
<td>14</td>
<td>*</td>
<td>465.312</td>
<td>na</td>
</tr>
<tr>
<td>15</td>
<td>*</td>
<td>698.510</td>
<td>na</td>
</tr>
<tr>
<td>16</td>
<td>*</td>
<td>1,101.856</td>
<td>na</td>
</tr>
<tr>
<td>17</td>
<td>*</td>
<td>1,845.104</td>
<td>na</td>
</tr>
<tr>
<td>18</td>
<td>*</td>
<td>2,989.743</td>
<td>na</td>
</tr>
<tr>
<td>19</td>
<td>*</td>
<td>5,741.825</td>
<td>na</td>
</tr>
</tbody>
</table>

*The time exceeded the time limit of 6000 CPU seconds.
Table 3  Computational results on $1/b_\gamma$-sdst/sgn(\(\alpha\))$E[\exp(\alpha T_{ADC})]/JPDR$

<table>
<thead>
<tr>
<th>Prob. size</th>
<th>CPU time (sec)</th>
<th>No. of heuristic solutions opt.</th>
<th>Average perc. gap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exact</td>
<td>Heuristic</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.108</td>
<td>0.061</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>0.135</td>
<td>0.115</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>0.387</td>
<td>0.143</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>0.692</td>
<td>0.225</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>36.462</td>
<td>1.038</td>
<td>7</td>
</tr>
<tr>
<td>10</td>
<td>453.305</td>
<td>2.132</td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td>3,489.532</td>
<td>17.824</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>*</td>
<td>43.081</td>
<td>na</td>
</tr>
<tr>
<td>13</td>
<td>*</td>
<td>233.127</td>
<td>na</td>
</tr>
<tr>
<td>14</td>
<td>*</td>
<td>497.059</td>
<td>na</td>
</tr>
<tr>
<td>15</td>
<td>*</td>
<td>712.045</td>
<td>na</td>
</tr>
<tr>
<td>16</td>
<td>*</td>
<td>1,325.413</td>
<td>na</td>
</tr>
<tr>
<td>17</td>
<td>*</td>
<td>2,037.032</td>
<td>na</td>
</tr>
<tr>
<td>18</td>
<td>*</td>
<td>3,435.120</td>
<td>na</td>
</tr>
</tbody>
</table>

*The time exceeded the time limit of 6000 CPU seconds.

Table 4  Computational results on $1/b_\gamma$-sdst/$E[\exp(\alpha T_{CT})]/AJDR$

<table>
<thead>
<tr>
<th>Prob. size</th>
<th>CPU time (sec)</th>
<th>No. of heuristic solutions opt.</th>
<th>Average perc. gap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Exact</td>
<td>Heuristic</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.142</td>
<td>0.115</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>0.160</td>
<td>0.186</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>0.586</td>
<td>0.269</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>0.905</td>
<td>0.395</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>59.542</td>
<td>1.572</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>679.108</td>
<td>4.903</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>5418.046</td>
<td>22.340</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>*</td>
<td>65.517</td>
<td>na</td>
</tr>
<tr>
<td>13</td>
<td>*</td>
<td>318.526</td>
<td>na</td>
</tr>
<tr>
<td>14</td>
<td>*</td>
<td>601.321</td>
<td>na</td>
</tr>
<tr>
<td>15</td>
<td>*</td>
<td>915.238</td>
<td>na</td>
</tr>
<tr>
<td>16</td>
<td>*</td>
<td>1,624.374</td>
<td>na</td>
</tr>
<tr>
<td>17</td>
<td>*</td>
<td>2,907.653</td>
<td>na</td>
</tr>
<tr>
<td>18</td>
<td>*</td>
<td>4,502.059</td>
<td>na</td>
</tr>
</tbody>
</table>

*The time exceeded the time limit of 6000 CPU seconds.
Summary and conclusions

We have studied a stochastic bicriteria single machine scheduling problem in which job attributes are random variables, set-up times are sequence-dependent, and a scheduler utilises a cost (or disutility) function, characterising his or her risk-taking behaviour (risk averse, risk prone and risk neutral), to evaluate two performance criteria associated with a sequence. The objective is to derive the optimal sequence that minimises the expected value of the scheduler’s cost with respect to both criteria. The problem is NP-hard to solve; however, we have formulated three scenarios in which cost functions are linear, exponential and fractional as QAPs that can be solved either exactly or approximately by the existing specialised B&B methods. The criteria include the makespan, total absolute variations in completion times, total completion time, total waiting time and their weighted counterparts, and the total reliability of jobs in a sequence when a job’s reliability depends on either the job’s position or the job’s immediately preceding job. Some cases in which the criteria are non-linear functions of job attributes have been also examined. As summarised in Table 5, most of the special cases with sequence-independent set-up times are solvable exactly in polynomial time. The proposed models incorporate the scheduler’s risk-taking behaviour, the stochasticity of job attributes and the underlying criteria; thus, such factors affect the optimal scheduling decisions. Our computational experiments on the problem scenarios with sequence-dependent set-up times demonstrate that the heuristic performs well in producing either optimal or good sequences. We have been able to solve problems with up to 11 jobs exactly, and up to 30 jobs heuristically within our time limit of 6000 CPU seconds. Given the NP-hardness of the problem, the reported CPU times for the heuristic method are attractive, and the amounts of gap between the objective values of exact and heuristic solutions are relatively small. The computational results also indicate that the nature of cost function and the two criteria affect the performance of solution methods for QAP and thus the problem difficulty. Furthermore, the problem studied here is general in the sense that its special cases reduce to some new stochastic and deterministic single machine models. Future research should focus on the inclusion of other criteria, the consideration of different non-linear cost functions, the extension to the cases with three or more criteria, the development of meta- and hyper-heuristics such as simulated annealing, tabu search and ant colony for the general problem, and the investigation of other stochastic bicriteria scheduling problems with non-linear cost functions.

Table 5
The complexity and the solution methods for $1/E[\alpha C_1 + (1-\alpha)C_{\text{max}}]$ with linear, exponential and fractional cost functions

<table>
<thead>
<tr>
<th>Problem</th>
<th>Complexity</th>
<th>Solution method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1/b_{\gamma}\text{-sist}/E[\alpha C_1 + (1-\alpha)C_{\text{max}}]$ where $C_1 = \text{TCT}, \text{TWT}, \text{TL}$</td>
<td>$O(n \log n)$</td>
<td>SEPT</td>
</tr>
<tr>
<td>$1/b_{\gamma}\text{-sist}, \gamma_0 = \tau, 0 &lt; \tau \neq 1/E[\alpha C_1 + (1-\alpha)C_{\text{max}}]$ where $C_1 = \text{TCT}, \text{TWC}$</td>
<td>$O(n \log n)$</td>
<td>SEPT</td>
</tr>
<tr>
<td>$1/b_{\gamma}\text{-sdst}/E[\alpha C_1 + (1-\alpha)C_{\text{max}}]$ where $C_1 = \text{TCT}, \text{TWT}, \text{TL}$</td>
<td>NP-hard</td>
<td>B&amp;B</td>
</tr>
<tr>
<td>$1/b_{\gamma}\text{-sdst}, \gamma_0 = \tau, 0 &lt; \tau \neq 1/E[\alpha C_1 + (1-\alpha)C_{\text{max}}]$ where $C_1 = \text{TCT}, \text{TWC}$</td>
<td>NP-hard</td>
<td>B&amp;B</td>
</tr>
<tr>
<td>$1/b_{\gamma}\text{-sist}/E[\alpha C_1 + (1-\alpha)\text{TADC}]$ where $C_1 = C_{\text{max}}, \text{TCT}, \text{TWT}, \text{TL}$</td>
<td>$O(n \log n)$</td>
<td>Matching</td>
</tr>
<tr>
<td>$1/b_{\gamma}\text{-sist}, \gamma_0 = \tau, 0 &lt; \tau \neq 1/E[\alpha C_1 + (1-\alpha)\text{TADC}]$ where $C_1 = \text{TCT}, \text{TWC}$</td>
<td>$O(n \log n)$</td>
<td>Matching</td>
</tr>
</tbody>
</table>
The complexity and the solution methods for $1/E[\max(C_1, C_2)]$ with linear, exponential and fractional cost functions (continued)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Complexity</th>
<th>Solution method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1/b_{p_{sdst}}/E[\alpha C_1 + (1-\alpha)TADC]$ where $C_1 = C_{\max}$, TCT, TWT, TL</td>
<td>NP-hard</td>
<td>B&amp;B</td>
</tr>
<tr>
<td>$1/b_{p_{sdst}}$, $\gamma_0 = \tau$, $0 &lt; \tau \neq 1/E[\alpha C_1 + (1-\alpha)TADC]$ where $C_1 = TCC$, TWC</td>
<td>NP-hard</td>
<td>B&amp;B</td>
</tr>
<tr>
<td>$1/b_{p_{sists}}$, $\gamma_0 = \tau$, $0 &lt; \tau \neq 1/E[\alpha C_1 + (1-\alpha)TCC]$ where $C_1 = TCT$, TWT, TL</td>
<td>$O(n\log n)$</td>
<td>SWEP</td>
</tr>
<tr>
<td>$1/b_{p_{sists}}$, $\gamma_0 = \tau$, $0 &lt; \tau \neq 1/E[\alpha C_1 + (1-\alpha)TWC]$ where $C_1 = TCT$, TWT, TL</td>
<td>$O(n\log n)$</td>
<td>SWEP</td>
</tr>
<tr>
<td>$1/b_{p_{sdst}}$, $\gamma_0 = \tau$, $0 &lt; \tau \neq 1/E[\alpha C_1 + (1-\alpha)TCC]$ where $C_1 = TCT$, TWT, TL</td>
<td>NP-hard</td>
<td>B&amp;B</td>
</tr>
<tr>
<td>$1/b_{p_{sdst}}$, $\gamma_0 = \tau$, $0 &lt; \tau \neq 1/E[\alpha C_1 + (1-\alpha)TWC]$ where $C_1 = TCT$, TWT, TL</td>
<td>$O(n)$</td>
<td>AP</td>
</tr>
<tr>
<td>$1/b_{p_{sists}}/E[\text{sgn}(\alpha)\exp(\alpha C_1 + C_{\max})]$ where $C_1 = TADC$, TCT, TWT, TL</td>
<td>NP-hard</td>
<td>B&amp;B</td>
</tr>
<tr>
<td>$1/b_{p_{sists}}$, $\gamma_0 = \tau$, $0 &lt; \tau \neq 1/E[\text{sgn}(\alpha)\exp(\alpha C_1 + C_{\max})]$ where $C_1 = TCC$, TWC</td>
<td>$O(n)$</td>
<td>AP</td>
</tr>
<tr>
<td>$1/b_{p_{sists}}/E[\text{sgn}(\alpha)\exp(\alpha C_1 + \text{TADC})]$ where $C_1 = TCT$, TWT, TL</td>
<td>$O(n)$</td>
<td>AP</td>
</tr>
<tr>
<td>$1/b_{p_{sists}}$, $\gamma_0 = \tau$, $0 &lt; \tau \neq 1/E[\text{sgn}(\alpha)\exp(\alpha C_1 + \text{TADC})]$ where $C_1 = TCC$, TWC</td>
<td>$O(n)$</td>
<td>AP</td>
</tr>
<tr>
<td>$1/b_{p_{sdst}}/E[\text{sgn}(\alpha)\exp(\alpha C_1 + \text{TADC})]$ where $C_1 = TCT$, TWT, TL</td>
<td>NP-hard</td>
<td>B&amp;B</td>
</tr>
<tr>
<td>$1/b_{p_{sists}}$, $\gamma_0 = \tau$, $0 &lt; \tau \neq 1/E[\text{sgn}(\alpha)\exp(\alpha C_1 + \text{TADC})]$ where $C_1 = TCC$, TWC</td>
<td>$O(n)$</td>
<td>AP</td>
</tr>
<tr>
<td>$1/b_{p_{sdst}}$, $\gamma_0 = \tau$, $0 &lt; \tau \neq 1/E[\text{sgn}(\alpha)\exp(\alpha C_1 + \text{TCC})]$ where $C_1 = TCT$, TWT, TL</td>
<td>NP-hard</td>
<td>B&amp;B</td>
</tr>
<tr>
<td>$1/b_{p_{sdst}}$, $\gamma_0 = \tau$, $0 &lt; \tau \neq 1/E[\text{sgn}(\alpha)\exp(\alpha C_1 + \text{TWC})]$ where $C_1 = TCT$, TWT, TL</td>
<td>NP-hard</td>
<td>B&amp;B</td>
</tr>
<tr>
<td>$1/b_{p_{sdst}}/E[\text{sgn}(\alpha)\exp(\alpha C_1)/\text{JPDR}]$ where $C_1 = C_{\max}$, TADC, TCT, TWT, TL</td>
<td>$O(n\log n)$</td>
<td>AP</td>
</tr>
<tr>
<td>$1/b_{p_{sists}}$, $\gamma_0 = \beta &gt; 0/E[\text{sgn}(\alpha)\exp(\alpha C_1)/\text{JPDR}]$ where $C_1 = C_{\max}$, TADC, TCT, TWT, TL</td>
<td>$O(n\log n)$</td>
<td>AP</td>
</tr>
<tr>
<td>$1/b_{p_{sdst}}$, $\gamma_0 = \tau$, $0 &lt; \tau \neq 1/E[\text{sgn}(\alpha)\exp(\alpha C_1)/\text{JPDR}]$ where $C_1 = TCC$, TWC</td>
<td>$O(n\log n)$</td>
<td>AP</td>
</tr>
<tr>
<td>$1/b_{p_{sdst}}$, $\gamma_0 = \tau$, $0 &lt; \tau \neq 1/E[\text{sgn}(\alpha)\exp(\alpha C_1)/\text{JPDR}]$ where $C_1 = TCC$, TWC</td>
<td>$O(n\log n)$</td>
<td>AP</td>
</tr>
<tr>
<td>$1/b_{p_{sdst}}$, $\gamma_0 = \tau$, $0 &lt; \tau \neq 1/E[\text{sgn}(\alpha)\exp(\alpha C_1)/\text{JPDR}]$ where $C_1 = TCC$, TWC</td>
<td>$O(n\log n)$</td>
<td>AP</td>
</tr>
</tbody>
</table>
Table 5  The complexity and the solution methods for $1/E[\gamma(C_1, C_2)]$ with linear, exponential and fractional cost functions (continued)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Complexity</th>
<th>Solution method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1/b_{ij}$-sist $E[\sgn(\alpha)\exp(\alpha C_1)/AJDR]$ where $C_1 = C_{\text{max}}, TADC, TCT, TWT, TL$</td>
<td>NP-hard</td>
<td>B&amp;B</td>
</tr>
<tr>
<td>$1/b_{ij}$-sist, $\gamma_i = t, 0 &lt; t \neq 1/E[\sgn(\alpha)\exp(\alpha C_1)/AJDR]$ where $C_1 = TCC, TWC$</td>
<td>NP-hard</td>
<td>B&amp;B</td>
</tr>
<tr>
<td>$1/b_{ij}$-xstd $E[\sgn(\alpha)\exp(\alpha C_1)/AJDR]$ where $C_1 = C_{\text{max}}, TADC, TCT, TWT, TL$</td>
<td>NP-hard</td>
<td>B&amp;B</td>
</tr>
<tr>
<td>$1/b_{ij}$-xstd, $\gamma_i = t, 0 &lt; t \neq 1/E[\sgn(\alpha)\exp(\alpha C_1)/AJDR]$ where $C_1 = TCC, TWC$</td>
<td>NP-hard</td>
<td>B&amp;B</td>
</tr>
</tbody>
</table>

Acknowledgements

The author thanks the referees for their valuable comments in improving the paper, and the Office of the Vice President for Research, Kuwait University, for supporting this research under Grant No. SS06/07.

References


