

A novel modeling approach for job shop scheduling problem under uncertainty

Behnam Beheshti Pur^{*}, Siamak Noori and Reza Hosnavi Atashgah

Department of Industrial Engineering, Iran University of Science and Technology, Narmak, Tehran, Iran

CHRONICLE

Article history:
Received May 16, 2013
Received in revised format
16 August 2013
Accepted 28 September 2013
Available online
October 3 2013

Keywords:
*Stochastic job shop scheduling
problem (SJSSP)*
*Chance constrained programming
(CCP)*
Simple recourse
Utility theory
*Expected utility programming
(EUP)*

ABSTRACT

When aiming on improving efficiency and reducing cost in manufacturing environments, production scheduling can play an important role. Although a common workshop is full of uncertainties, when using mathematical programs researchers have mainly focused on deterministic problems. After briefly reviewing and discussing popular modeling approaches in the field of stochastic programming, this paper proposes a new approach based on utility theory for a certain range of problems and under some practical assumptions. Expected utility programming, as the proposed approach, will be compared with the other well-known methods and its meaningfulness and usefulness will be illustrated via a numerical examples and a real case.

© 2013 Growing Science Ltd. All rights reserved.

1. Introduction

Scheduling problem deals with the allocation of scarce resources to tasks over time and it is considered as a combinatorial optimization problem and may be handled as a decision-making process with the goal of optimizing one or more objectives (Lopez & Roubellat, 2008). A job shop scheduling problem (JSSP), as a branch of scheduling, is to schedule jobs when each job requires a sequence of operations on all different machines. Many researchers have considered this problem for deterministic cases and one may determine reliable solution for JSSP when all parameters are known in advance (Xhafa & Abraham, 2008; Brucker, 2007).

^{*}Corresponding author. Tel.: +989126794043
E-mail address: beheshti_pour@yahoo.com (B.Beheshtipour)

There are some certain information about resources and tasks, which define a scheduling problem. Many real-world problems are full of uncertainty, which makes it inevitable to devise complicated techniques where deterministic ones are considered as a basis. Although generally the task duration is uncertain, scholars have frequently suppressed that uncertainty when stating the problem (Baker & Trietsch, 2009). In the context of job shop scheduling, processing time of work pieces, the frequency of machine faults and the interval of orders often have a fluctuating nature and managers must make decisions without full knowledge of the coefficients. For instance, the exponential distribution may be an appropriate model for arrival processes, processing times and sometimes for waiting times (Baker & Trietsch, 2009).

Gua et al. (1984) employed an expected value model (EVM) to deal with stochastic job shop scheduling problem (SJSSP). EVM has been a popular modeling technique, which could be used as a benchmark for our approach (EVM substitutes uncertain parameters by their expected value, refer to Liu, 2007). Besides operational research methods, which are the primary focus of the present work, other approaches have been used. For example, Gu & Lu (2009) devised an online approximate policy for the preemptive stochastic online scheduling on two uniform machines and Gourgand et al. (2003) developed a recursive algorithm to compute the expected makespan based on a Markov chain. Liu (2007) has employed chance constrained programming (CCP) to model scheduling problems, namely machine and project scheduling. In addition, methods of recourse (two-stage and multi-stage optimization) may be applied to deal with uncertainty via mathematical frameworks (Birge & Louveaux, 1997).

Initially introduced by Charnes and Cooper (1959), CCP tries to maintain feasibility condition (e.g. not to exceed resource availability) by forcing each constraint to hold with a predetermined confidence level. For the objective function, CCP ensures that the provided solution will yield an acceptable value again with a predetermined confidence level. Here the primary objective is about feasibility of CCP solutions and their implications. For instance, if the problem remains feasible with the probability of 95 percent, what about the cases, which could occur with the probability of 5 percent? In fact, after realization of some uncertain parameters, the CCP solutions may not be feasible anymore. Furthermore, when deciding on confidence levels, what can the model tell the decision maker (DM) about the tradeoff between maintaining feasibility and having a better return from objective function? CCP is silent about these issues because it does not take into account the phenomenon of infeasibility occurrence (i.e. violation of stochastic constraints). With this background, expected utility programming (EUP), as the proposed approach, tries to model JSSP with respect to some practical assumptions, appropriately. Here, putting aside regular presuppositions, the following assumptions are different from some of the classical problem:

- 1) Processing times can be stochastic.
- 2) If the required resources (e. g. machine capacity) for processing of any job are not available at a requested time, the scheduled job should be delayed.
- 3) Job postponement will result in some measurable outcomes (e. g. shop floor disorder or extra effort for rescheduling).
- 4) If for some resource types, the required amounts at any time exceed the available supply, additional amounts cannot be acquired at the needed time or under any acceptable expenses.

The third assumption makes CCP not capable of adequately modeling the problem and the fourth assumption implies a situation, which contradicts the underlying assumptions of recourse methods. Although JSSP has frequently been recognized as a multiobjective problem (Qian et al., 2008; Lei, 2008; Huang, 2010) and the DM may wish to simultaneously satisfy several possibly conflicting goals (e. g. minimizing tardiness or makespan and maximizing preventive maintenance times), here the problem will be modeled as a single objective one to keep presentation of main contributions simple. Obviously one may simply apply the proposed approach for multiobjective cases.

Large-scale JSSPs have been recognized as extremely NP-hard problems for deterministic cases (Lenstra et al., 1977). Furthermore, incorporation of uncertainty into these mathematical programs will make them even more complicated. However, heuristic and metaheuristic algorithm can smooth the disadvantages of these programs. Here we employ a heuristic method based on genetic algorithm to solve our proposed model. The solution method will be discussed and detailed later in section 4.

The reminder of this paper is organized as follow: section 2 presents the problem definition along with its parameters and the deterministic JSSP. Briefly reviewing CCP and providing some required knowledge from utility theory, section 3 presents EUP and uses a numerical example to compare it with CCP and EVM. Section 4 contains a real case to illustrate usefulness and meaningfulness of proposed approach. Finally, section 5 summarizes the results and gives the concluding remarks.

2. Problem definition

The job shop scheduling deals with the allocation of n jobs ($1, 2, \dots, n$) to m different machines ($1, 2, \dots, m$). The jobs are subject to machine precedence constraints and the completion of job i consists of a sequence of operations: $O_{1i}, O_{2i}, \dots, O_{n_i i}$ (machine list). The precedence in the machine list indicates that job i should be processed first on machine 1, then machine 2, until the last machine, n_i . Here, the objective is to minimize the maximum completion time of the jobs and so the problem is to find a feasible schedule, which yields the minimum makespan. The followings are assumed in a deterministic JSSP:

- (1) There is no machine breakdown and no preemption of operations is allowed.
- (2) A machine can process only one job at a time.
- (3) A job is not allowed to visit the same machine twice.
- (4) Transportation times between different machines and setup times for different jobs are all neglected.
- (5) Among the operations on different machines, jobs can wait in an unlimited buffer.

2.1. Parameters

M	a large positive number
i, j	index of jobs
k, h	index of machines
l	index of states of the world
p_{ik}	the processing time of job i on machine k
l_{ik}	financial and non-financial returns from postponement of scheduled job i on machine k (specified by the DM)
S_i	the set which indicates precedence of machines ($O_{1i}, O_{2i}, \dots, O_{n_i i}$) for job i : (k, h) $\in S_i$ if machine k comes right before machine h in the machine list

2.2. Decision variable

x_{ik}	the starting time of job i on machine k
y_{ijk}	is 1 if job i is before job j on machine k , and 0 otherwise

2.3. Deterministic JSSP

When all processing times are known with certainty, the rational decision is to work with a schedule that never results in delays for scheduled jobs. This way, all returns from postponements may be disregarded. The deterministic problem can be formulated as:

$$\text{Min } Z \quad (1)$$

$$Z \geq x_{ik} + p_{ik} \quad \forall i, k \quad (2)$$

$$x_{ik} + p_{ik} \leq x_{ih} \quad \forall (k, h) \in S_i \quad (3)$$

$$x_{ik} + p_{ik} \leq x_{jk} + M(1 - y_{ijk}) \quad \forall i, j, k \quad (4)$$

$$x_{ik} + My_{ijk} \geq x_{jk} + p_{jk} \quad \forall i, j, k \quad (5)$$

$$x_{ik} \geq 0 \quad \forall i, k, y_{ijk} \in \{0,1\} \quad \forall i, j, k \quad (6)$$

The objective function given in Eq. (1) along with constraint (2) minimizes the makespan. Constraint (3) ensures that the processing sequence of each job's operations is in accordance with the machine list. Constraints (4, 5) guarantee that each machine will not process more than one job at a time. Constraint (6) forces decision variables to take their proper values.

3. EUP as an appropriate approach for modeling SJSSP

3.1. A brief review of CCP

Here, we review CCP, as a popular modeling approach, to justify the need for a new one. In addition, we use it as a benchmark to clarify some features of EUP as the proposed method. Consider the linear program above (1-6). Now, if the values of some parameters are not fixed, first we will need to clarify the meaning of minimization in the objective function. In addition, the feasibility of a decision, say, can be quite different from that of conventional linear programs. For this matter, CCP rewrites the problem as:

$$\text{Maximize } Z \quad (7)$$

$$\text{Pr}\{Z \geq x_{ik} + p_{ik}\} \geq \alpha \quad \forall i, k \quad (8)$$

$$\text{Pr}\{x_{ik} + p_{ik} \leq x_{ih}\} \geq \beta \quad \forall (k, h) \in S_i \quad (9)$$

$$\text{Pr}\{x_{ik} + p_{ik} \leq x_{jk} + M(1 - y_{ijk})\} \geq \beta \quad \forall i, j, k \quad (10)$$

$$\text{Pr}\{x_{ik} + My_{ijk} \geq x_{jk} + p_{jk}\} \geq \beta \quad \forall i, j, k \quad (11)$$

$$x_{ik} \geq 0 \quad \forall i, k, y_{ijk} \in \{0,1\} \quad \forall i, j, k \quad (12)$$

where α and β are predetermined confidence levels, and Z is the α -optimistic return. CCP tries to maintain feasibility of constraints with a high confidence level (e.g. $\beta = .95$) and assumes that violation of constraints can be disregarded due to the very low chance of occurrence. In some cases that consequences of constraint violations are not known or they are hard to estimate, CCP can be of good use. In constraint (9) the DM determines his/her acceptable confidence level for the value of objective function, which indicates his/her attitude toward risk. If $\alpha > .5$ the DM is risk averse, if $\alpha < .5$ he is risk seeker and $\alpha = .5$ models the neutral behavior.

3.2. Some required knowledge from utility theory

Here we examine the utility theory to understand, based on some rational axioms, what should be done about a decision in an uncertain environment. In the classical problems of utility theory, decision maker considers a situation that stochastic events determine the payoff to him.

Let X be a set of "prizes" and ℓ be a lottery with payoffs in X . Lottery ℓ is a function $p: X \rightarrow [0,1]$ such that $\sum_{x \in X} p(x) = 1$. $p(x)$ is the probability that the payoff is $x \in X$ and when $X = \{x_1, x_2, \dots, x_n\}$ for

some finite number n , we will write $p(x_i) = p_i$. The expected value of a lottery is the sum of the payoffs, where each payoff is weighted by the probability that the payoff occur (Gintis, 2009). If the lottery ℓ has payoffs x_1, x_2, \dots, x_n with probability of p_1, p_2, \dots, p_n , one can write $\ell = (x_1, p_1), (x_2, p_2), \dots, (x_n, p_n)$, with expected value of:

$$E(\ell) = \sum_{i=1}^n p_i x_i \quad (13)$$

Feller (1950) argued that Eq. (13) is significant because of the law of large numbers, which states that as the number of times a lottery is played goes to infinity, the average payoff converges to expected value of the lottery with probability 1. Assuming that individuals have consistent preferences over an appropriate set of lotteries, Von Neumann and Morgenstern (1944), Friedman and Savage (1948), Savage (1954), and Anscombe and Aumann (1963) showed that expected utility principle will be valid in stochastic environment.

The other issue from utility theory, which will be of good use for our argument, is utility functions. This is especially important because for practical cases, this is not generally true that one can repeat his/her choice for infinite times, and there are many cases that one can choose a lottery only one time. To model this condition, scholars have introduced various types of utility functions. The shape of utility function depends on the level of risk that DM is looking for, for example $\ln x$ and $\sqrt[3]{x}$ are two utility function which can model risk aversion and e^x can be used for a risk seeker DM. Therefore, the revised measure for assessing the value of a lottery which is more accurate can be written as below:

$$E(\ell) = \sum_{i=1}^n p_i u(x_i) \quad (14)$$

where $u(x_i)$ is the utility of payoff x_i for the DM.

3.3. The proposed approach

Consider some processing times in the deterministic model to be uncertain. Every scheduling decision is a lottery. In the other words, for a given decision we will have different payoffs based on realization of uncertain parameters. If after realization of uncertain parameters all constraints are satisfied, the payoff is the objective function stated in Eq. (1). Otherwise, the payoff is a combination of the obtained makespan and some function of the difficulties from postponement of some jobs. Here, we assume that the distributions of stochastic processing times to be discrete (if not, one can use discretization to convert continuous distributions, refer to Birge & Louveaux, 1997).

Abiding by Eq. (14), we construct the following model:

$$\text{Min } u(Z) + \sum_{i,j,k,h,l} \text{Pr}(l) \{ u(l_{ih})a_{ikh} + u(l_{jk})b_{ijkl} + u(l_{ik})c_{ijkl} \} \quad (15)$$

$$Z \geq x_{ik} + E[p_{ik}] \quad \forall i, k \quad (16)$$

$$x_{ik} + p_{ikl} - M a_{ikh} \leq x_{ih} \quad \forall (k, h) \in S_i, \quad l = 1, 2, \dots, L \quad (17)$$

$$x_{ik} + p_{ikl} - M b_{ijkl} \leq x_{jk} + M(1 - y_{ijk}) \quad \forall i, j, k, l \quad (18)$$

$$x_{ik} + M y_{ijk} + M c_{ijkl} \geq x_{jk} + p_{jkl} \quad \forall i, j, k, l \quad (19)$$

$$x_{ik} \geq 0 \quad \forall i, k, \quad y_{ijk}, a_{ikh}, b_{ijkl}, c_{ijkl} \in \{0, 1\} \quad \forall i, j, k, l \quad (20)$$

The objective function maximizes the utility of a given decision. For every states of the world, a scheduled job may or may not be delayed, if delayed its penalty will be incurred. Constraint (16) has an interpretation similar to constraint (2). Here, the only difference is that we have substituted p_{ik}

with its expected value, $E[p_{ik}]$, because doing so will have a negligible effect on the obtained makespan. Constraint (17) states that if a job is scheduled simultaneously on more than one machine, the penalty should be incurred. Constraints (18) and (19) have similar implications. Constraint (20) forces decision variables to take their proper values.

One may simply think that EUP works in the sense of an expected value operator. This is wrong because EUP brings the tradeoff between feasibility of the original problem and consequences of a given decision (makespan and penalty) in a single term and uses a set of constraints, which will hold almost surely.

3.4. Numerical example

Here we use a very simple but useful example to compare EUP with CCP and EVM.

Example1. Considering one job and two machines, we can rewrite the Deterministic JSSP as:

$$\min \quad x_{12} + p_{12} \quad (21)$$

$$x_{12} \geq x_{11} + p_{11} \quad (22)$$

$$x_{11}, x_{12} \geq 0 \quad (23)$$

where $p_{11}, p_{12} = 1, 3, 4$ with probability of $1/5, 3/5$ and $1/5$, respectively. When constraint (22) is violated, processing of job 1 on machine 2 should be delayed. Assume that the DM considers shop disorder unfavorable and sets $u(l_{12} = \text{disorder})$ is equal to 2. Considering $u(x) = x/2$ for makespan, one may rewrite (15-20) as:

$$\min \quad \frac{x_{12} + E[p_{12}]}{2} + \left\{ \frac{1}{5}(2a_{1121}) + \frac{3}{5}(2a_{1122}) + \frac{1}{5}(2a_{1123}) \right\} \quad (24)$$

$$x_{12} + Ma_{1121} \geq x_{11} + p_{111} \quad (25)$$

$$x_{12} + Ma_{1122} \geq x_{11} + p_{112} \quad (26)$$

$$x_{12} + Ma_{1123} \geq x_{11} + p_{113} \quad (27)$$

$$x_{11}, x_{12} \geq 0, \quad a_{1121}, a_{1122}, a_{1123} \in \{0,1\} \quad (28)$$

The above model is a linear one and the optimal solution is: $x_{11} = a_{1121} = a_{1122} = 0, a_{1123} = 1, x_{12} = 3$ with utility of 19/10.

Using EVM one can write:

$$\min \quad x_{12} + 2.2 \quad (29)$$

$$x_{12} \geq x_{11} + 2.2 \quad (30)$$

$$x_{11}, x_{12} \geq 0 \quad (31)$$

The optimal solution is $x_{11} = 0$ and $x_{12} = 2.2$. It is obvious that this solution is sub-optimal for the model stated by (24-28). In fact, this model cannot consider shop disorder caused by violation of the original constraint stated in Eq. (22). Comparing with the concept of safety stock (SS) in inventory planning, we may introduce safety time (ST) which, for this example, can be calculated as: $ST = 3 - 2.2 = .8$. Here, we schedule job 1 on machine 2 a little later, compared with the optimal solution from

EVM, to guard against unfavorable events in the implementation stage. This notion is quite similar to that of safety stock.

Using CCP one can write:

$$\min Z \tag{32}$$

$$\Pr\{x_{12} + p_{12} \leq Z\} \geq \alpha \tag{33}$$

$$\Pr\{x_{12} \geq x_{11} + p_{11}\} \geq \beta \tag{34}$$

$$x_{11}, x_{12} \geq 0 \tag{35}$$

With $\beta > .8$ we should have $x_{12} - x_{11} \geq 4$, with $2 < \beta \leq .8$ we should have $x_{12} - x_{11} \geq 3$ and with $0 < \beta \leq .2$ we should have $x_{12} - x_{11} \geq 1$. This means that the optimal solution for the first interval is $x_{11} = 0, x_{12} = 4$ and , for the second one is $x_{11} = 0, x_{12} = 3$ and and for the third one is $x_{11} = 0, x_{12} = 1$. Also α only has an effect on Z^* . Putting aside this fact that CCP cannot consider the consequences of job postponements, one can see that the quality of CCP solutions, compared with the rational approaches of utility theory, is very low. Besides, it is hard to repeat the same analysis performed above for large examples and virtually impossible for real cases.

4. A case study

Saipa Group is a reputable automobile manufacturer in Iran and has 10 subsidiaries and many more suppliers and contractors. The group has various manufacturing systems including job shops, flow shops, assembly lines and fully automated lines. The group produces nearly 20 different products. Some of the subsidiaries produce parts and components for low production rate products, such as molding dies, and use job shops to maintain flexibility. Table 1 summaries the production information for one of these shops which supplies parts for 4 different final products.

Table 1
Production information

Item	Job	Operations				
		1	2	3	4	5
Processing time	J_1	ξ_1	3	5	1	ξ_2
	J_2	1	4	ξ_1	2	3
	J_3	ξ_3	2	5	1	3
	J_4	1	3	1	ξ_4	ξ_1
	J_5	ξ_2	2	2	4	2
	J_6	ξ_5	3	4	ξ_2	1
	J_7	4	ξ_2	4	1	1
	J_8	ξ_2	4	ξ_3	2	ξ_2
	J_9	1	ξ_3	2	ξ_1	2
	J_{10}	2	3	ξ_2	1	1
	J_{11}	4	ξ_2	ξ_1	ξ_1	3
	J_{12}	ξ_4	ξ_1	ξ_3	ξ_1	ξ_1
Machine Sequence	J_1	M_1	M_2	M_3	M_4	M_5
	J_2	M_1	M_2	M_3	M_4	M_5
	J_3	M_1	M_4	M_2	M_3	M_5
	J_4	M_1	M_4	M_2	M_3	M_5
	J_5	M_2	M_1	M_3	M_4	M_5
	J_6	M_2	M_1	M_4	M_3	M_5
	J_7	M_1	M_5	M_3	M_4	M_2
	J_8	M_5	M_3	M_1	M_4	M_2
	J_9	M_4	M_2	M_3	M_5	M_1
	J_{10}	M_3	M_5	M_4	M_2	M_1
	J_{11}	M_3	M_1	M_5	M_2	M_4
	J_{12}	M_3	M_4	M_2	M_5	M_1

In Table 1, some jobs have uncertain processing times represented by $\xi_1, \xi_2, \xi_3, \xi_4, \xi_5$. Historical data shows that these parameters can be satisfactorily modeled by discrete distributions shown in table 2.

Table 2
Discrete distributions of processing times

ξ_1	1	2		
$\Pr(\xi_1)$.25	.5		
ξ_2	1	2	3	
$\Pr(\xi_2)$.25	.5	.25	
ξ_3	1	3	5	
$\Pr(\xi_3)$.3	.4	.3	
ξ_4	2	3	4	
$\Pr(\xi_4)$.55	.25	.2	
ξ_5	1	2	3	5
$\Pr(\xi_5)$.3	.4	.2	.1

The only scarce resource is machine capacity, which should be allocated appropriately to jobs in order to minimize makespan. Recently due to introduction of two new jobs (J_{11} and J_{12}) scheduling has become more important than before. A significant challenge is that some processing times are uncertain, and so when implementing a schedule, processing of some jobs on some machines may have overlaps which results in shop floor disorders. Production manager has announced that “it is better to have no schedule rather than a tight one”, which shows his/her attitude towards the problems arisen from coincidence of two scheduled jobs on a machine. Compared with the other jobs, J_{12} is considered to have more importance and it is undesired to encounter overlapping for this job. Therefore, the production manager decides to set $u(l_{12k}) = 5$ for $k = 1, 2, \dots, 5$ and for all other jobs he sets $u(l_{ik}) = 2$. Although it is possible to completely prevent overlapping by using a prolonged schedule, but it will result in an *expensive* makespan. Therefore, one should consider a tradeoff between having a better makespan and avoidance from overlapping. For this matter, one may use $u(Z) = Z$ to model the relative importance of makespan.

Using the above information, the problem can be written as:

$$\min \quad u(Z) + \sum_{i,j,k,h,l} \Pr(l) \{ u(l_{ih})a_{ikh} + u(l_{jk})b_{ijkl} + u(l_{ik})c_{ijkl} \} \quad (36)$$

$$Z \geq x_{ik} + E[p_{ik}] \quad \forall i, k \quad (37)$$

$$x_{ik} + p_{ikl} - M a_{ikh} \leq x_{ih} \quad \forall (k, h) \in S_i, \quad l = 1, 2, \dots, L \quad (38)$$

$$x_{ik} + p_{ikl} - M b_{ijkl} \leq x_{jk} + M(1 - y_{ijk}) \quad \forall i, j, k, l \quad (39)$$

$$x_{ik} + M y_{ijk} + M c_{ijkl} \geq x_{jk} + p_{jkl} \quad \forall i, j, k, l \quad (40)$$

$$x_{ik} \geq 0 \quad \forall i, k, \quad y_{ijk}, a_{ikh}, b_{ijkl}, c_{ijkl} \in \{0, 1\} \quad \forall i, j, k, l \quad (41)$$

4.1. Solution method

JSSP problem is regarded as NP-hard problem (Cheng et al., 1996). To solve the above model we use a heuristic approach, which incorporates genetic algorithm (GA) in its steps. Genetic algorithm is one of the modern metaheuristic algorithms and has been successfully applied to many NP-hard problems (Lin, 2008, Alves & Almeida, 2007, Li et al., 2009, Ghorbani & Rabbani, 2009). In addition, GA has been used to solve JSSP with satisfactory results compared to other available solution methods (Gonçalves et al., 2005, Pezzellaa et al., 2008, Qing-dao-er-ji & Wang, 2012, Yusof et al., 2011). GA uses the concept of evolutionary computation imitating the natural selection and biological

reproduction of animal species. It originates from Darwin's "survival of the fittest" concept, which means good parents produce better offspring.

First, we consider an initial sequence for the scheduling problem. Note that every schedule is feasible with regard to constraints (37-40) and it should be feasible only for constraints (41). To make all the variable binary we need to redefine x_{ik} . In our genetic algorithm, we consider a sequence and set $t_{111} = 1$ if job 1 is the first job in the sequence which visits machine 1 and set $t_{111} = 0$, otherwise. Similarly, we set $t_{ikr} = 1$ if job i is r th job in the sequence, which visits machine k and set $t_{ikr} = 0$, otherwise. Next, we compute x_{ik} as the sum of the "expected processing times" of the jobs coming before i in the sequence. Note that here we have 216 scenarios or states of the world: two states for ξ_1 , three states for each of ξ_2, ξ_3 and ξ_4 and four states for ξ_5 which results in a total of $2 \times 3 \times 3 \times 3 \times 4$ or 216 states of the world for our production system. Now we may write the following algorithm:

Algorithm 1

Step1: Candidate an initial population using t_{ikr} for a potential sequence of jobs and then compute x_{ik} (as the sum of "the expected processing times" of the jobs coming before i in the sequence).

Step 2: Based on 216 different scenarios postpone some jobs. For this matter set binary variables in a way that constraint (41) be satisfied and the best value for the objective function stated in Eq. (36) be achieved.

Step 3: Check the stopping criterion and if confirmed go to step 5, otherwise go to step 4.

Step 4: Generate the next population using crossover and mutation operators and return to step 2.

Step 5: Stop.

Obviously, we need some efficient methods or algorithms to find the optimal or near optimal decisions in step 2. For this matter we use the following algorithm:

Algorithm 2

Step1: Candidate an initial population of feasible solutions using genetic algorithm respect to constraints represented by Eq. (41).

Step 2: Simplify the objective function in (36) using proposed solution in step 1.

Step 3: Consider all of the constraints under 216 scenarios and compute value of the objective function.

Step 4: Check the stopping criterion and if confirmed go to step 6, otherwise go to step 5.

Step 5: Generate the next population using crossover and mutation operators and return to step 2.

Step 6: Stop.

Algorithms 1 and 2 solve the sequencing problem and now we can proceed to solve the scheduling problem. We should determine values of x_{ik} based on values of t_{ikr} . If we set values of x_{ik} equal to the sum of "the expected processing times" of the jobs coming before i in the sequence, the probability of job postponements will be about 50 percent. Now we employ the following algorithm to balance makespan and adverse consequences of postponements.

Algorithm 3

Step1: Candidate an initial population of feasible solutions for $y_{ijk}, a_{ikh}, b_{ijkl}, c_{ijkl}$.

Step 2: Simplify the objective function and constraints using the values of $y_{ijk}, a_{ikh}, b_{ijkl}, c_{ijkl}$.

Step 3: Consider the resulting linear program with the variable x_{ik} . Solve this problem subject to the sequence specified by values of t_{ikr} from algorithm 2.

Step 4: Check the stopping criterion and if confirmed go to step 6. Otherwise go to step 5.

Step 5: Generate the next population using crossover and mutation operators and return to step 2.

Step 6: Stop.

In all algorithms, the stopping criterion is based on improvement in objective function, i.e. if for two successive iterations the improvement fall below a predetermined value, the algorithm will be terminated. Here, we solve the problem in 3 steps: algorithms 1 and 2 solve the sequencing problem and then algorithm 3 finalizes the problem with scheduling information. The three algorithms were coded in MATLAB and the solution of the problem is given in Table 3:

Table 3
Problem Solution

Job	Machine				
	1	2	3	4	5
J_1	$M_1(0)$	$M_2(1.9)$	$M_3(3.2)$	$M_4(5.6)$	$M_5(7.2)$
J_2	$M_1(1.9)$	$M_2(3.5)$	$M_3(5.5)$	$M_4(7.4)$	$M_5(10.5)$
J_3	$M_1(7.1)$	$M_4(8.3)$	$M_2(9.5)$	$M_3(11.4)$	$M_5(12.8)$
J_4	$M_1(6.1)$	$M_4(7.7)$	$M_2(9.3)$	$M_3(14.2)$	$M_5(15.5)$
J_5	$M_2(3.9)$	$M_1(4.5)$	$M_3(6.7)$	$M_4(7.4)$	$M_5(8.5)$
J_6	$M_2(5.1)$	$M_1(6.7)$	$M_4(7.9)$	$M_3(8.4)$	$M_5(10.1)$
J_7	$M_1(6.5)$	$M_5(8.5)$	$M_3(12.2)$	$M_4(14.3)$	$M_2(15.1)$
J_8	$M_5(0)$	$M_3(4.5)$	$M_1(6.8)$	$M_4(9.3)$	$M_2(11.2)$
J_9	$M_4(0)$	$M_2(1)$	$M_3(1.35)$	$M_5(3.35)$	$M_1(3.7)$
J_{10}	$M_3(1)$	$M_5(1.4)$	$M_4(3.2)$	$M_2(4.1)$	$M_1(5.5)$
J_{11}	$M_3(4.5)$	$M_1(6.7)$	$M_5(8.5)$	$M_2(10.3)$	$M_4(12.4)$
J_{12}	$M_3(0)$	$M_4(1)$	$M_2(1.5)$	$M_5(1.9)$	$M_1(2.4)$

According to the above solution, total makespan is 15.5 minutes, which is the result of a balance between makespan and adverse consequences of postponements. More details about GA can be found in Gonçalves et al. (2005), Pezzellaa et al. (2008), Qing-dao-er-ji & Wang (2012), Yusof et al. (2011).

5. Conclusion

This paper proposed EUP as an appropriate modeling approach for SJSSP under certain practical assumptions. It was discussed that EUP could take into account consequences of infeasibility of decisions which might be revealed after realization of some unknown parameters. Utility theory formed the foundation of proposed approach and two numerical examples clarified some features of EUP compared to other popular modeling approaches like CCP and EVM. Finally, to show its practical meaningfulness, EUP was applied to a real case and the results were presented.

Acknowledgment

The authors would like to thank the anonymous referees for constructive comments on earlier version of this paper.

References

- Alves, M. J. & Almeida, M. (2007). MOTGA: A multiobjective Tchebycheff based genetic algorithm for the multidimensional knapsack problem. *Computers and Operations Research*, 34, 3458 – 3470.
- Anscombe, F. & Aumann, R. J. (1963). A definition of subjective probability. *Annals of Mathematical Statistics*, 34, 199–205.

- Baker, K. R. & Trietsch, D. (2009). *Principles of Sequencing and Scheduling*. John Wiley & Sons, Hoboken, New Jersey.
- Birge, J.R., & Louveaux, F. (1997). *Introduction to Stochastic Programming*. Springer, New York.
- Brucker, P. (2007). *Scheduling Algorithms*. Springer, Berlin.
- Charnes, A. & Cooper, W.W. (1959). Chance-constrained programming. *Management Science*, 6, 73–79.
- Cheng, R., Gen, M., & Tsujimura, Y. (1996). A tutorial survey of job-shop scheduling problems using genetic algorithms. *Computers & Industrial Engineering*, 30(4), 983–997.
- Feller, W. (1950). *An Introduction to Probability Theory and Its Applications*. Vol. 1. John Wiley & Sons, New York.
- Friedman, M., & Savage, L. J. (1948). The utility analysis of choices involving risk. *Journal of Political Economy*, 56, 279–304.
- Ghorbani, S. & Rabbani, M. (2009). A new multi-objective algorithm for a project selection problem. *Advances in Engineering Software*, 40, 9–14.
- Gintis, H. (2009). *Game Theory Evolving*, 2nd Ed. Princeton University Press, Princeton.
- Gonçalves, J. F., Mendes, J. J. M., & Resende, M. G. C. (2005). A hybrid genetic algorithm for the job shop scheduling problem. *European Journal of Operational Research*, 167 (1), 77–95.
- Gourgand, M., Grangeon, N., & Norre, S. (2003). A contribution to the stochastic flow shop scheduling problem. *European Journal of Operational Research*, 151(2), 415–33.
- Gu, M. & Lu, X. (2009). Preemptive stochastic online scheduling on two uniform machines. *Information Processing Letters*, 109, 369–75.
- Gua, J., Gub, M., Cao, C., & Gu, X. (2010). A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem. *Computers & Operations Research*, 37, 927–937.
- Huang, R. (2010). Multi-objective job-shop scheduling with lot-splitting production. *International Journal of Production Economics*, 124, 206–213.
- Lei, D. (2008). Pareto archive particle swarm optimization for multiobjective fuzzy job shop scheduling problems. *International Journal of Advanced Manufacturing Technology*, 37, 157–165.
- Lenstra, J. K., Rinnooy Kan, A. H. G., & Brucker, P. (1977). Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, 1, 343–362.
- Lin, F. (2008). Solving the knapsack problem with imprecise weight coefficients using genetic algorithms. *European Journal of Operational Research*, 185, 133–145.
- Li, Y.F., Xie, M., & Goh, T.N. (2009). A study of project selection and feature weighting for analogy based software cost estimation. *Journal of Systems and Software*, 82, 241–252.
- Liu, B. (2007). *Theory and Practice of Uncertain Programming*. 2nd ed. Tsinghua University, Beijing.
- Lopez, P. & Roubellat, F. (2008). *Production Scheduling*. John Wiley & Sons, Hoboken, New Jersey.
- Neumann, V., Morgenstern, J. & Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton University Press, Princeton.
- Pezzella, F., Morgantia, G. & Ciaschetti, G. (2008). A genetic algorithm for the flexible job-shop scheduling problem. *Computers & Operations Research*, 35(10), 3202–3212.
- Qian, B., Wang, L., Huang, D., & Wang, X. (2008). Scheduling multi-objective job shops using a memetic algorithm based on differential evolution. *International Journal Advanced Manufacturing Technology*, 35, 1014–1027.
- Qing-dao-er-ji, R. & Wang, Y. (2012). A new hybrid genetic algorithm for job shop scheduling problem. *Computers & Operations Research*, 39(10), 2291–2299.
- Savage, J. (1954). *The Foundations of Statistics*. John Wiley & Sons, New York.
- Khafa, F., & Abraham, A. (2008). *Metaheuristics for Scheduling in Industrial and Manufacturing Applications*. Springer, Berlin.

Yusof, R., Khalid, M., Teck Hui, G., Yusof, S. & Othman, M. (2011). Solving job shop scheduling problem using a hybrid parallel micro genetic algorithm. *Applied Soft Computing*, 11(8), 5782–5792.