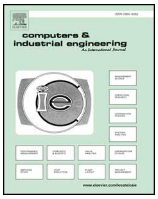
Computers & Industrial Engineering 130 (2019) 317–326

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie



A hybrid multi-objective EDA for robust resource constraint project scheduling with uncertainty

Jing Tian[a,*], Xinchang Hao[b], Mitsuo Gen[c]

[a] State Key Laboratory of Air Traffic Management System and Technology, Nanjing, China
[b] Changzhou Institute of Technology, Changzhou, China
[c] Fuzzy Logic Systems Institute and Tokyo University of Science, Tokyo, Japan

ARTICLE INFO

Keywords:
Estimation distribution of algorithm
Genetic algorithm
Multi-objective
Markov network
Robust scheduling
Resource constraint scheduling problem

ABSTRACT

This paper presents a multi-phased algorithm hybrid genetic algorithm and multi-objective Markov network based Estimation of Distribution Algorithm (robust hGMEDA), to solve the robust scheduling problem for resource constrained scheduling problem (RCSP) with time uncertainty. Firstly, for modelling, two kinds of robust measures on time-based-robust and capacity-based-robust are introduced to evaluate the robustness of scheduling solutions. Secondly, for solving methodology, within the multi stage architecture based on sequential coevolutionary paradigm, genetic algorithm (GA) is used to find feasible solution for sequencing sub-problem, and multi-objective Markov network based Estimation of Distribution Algorithm (MMEDA) is adopted to model the interrelation for resource allocation and calculate the Pareto set with the scenario based approach. Next, the alternative solutions are checked by the chance constraints by using scenario-based simulation. Moreover, one problem-specific local search with considering both makespan and robustness is designed to improve the solution quality. The implementation results provide practical support that experiment results based on a benchmark "Project Scheduling Problems Library" (PSPLIB) and comparisons demonstrate that our approach is highly effective and tolerant of uncertainty.

1. Introduction

Resource constrained scheduling problems (RCSPs) have attracted an ever growing attention, encountered within a very wide variety of applications in industries, which provides an optimal project schedule, which not only satisfies the precedence constraints but also make the resource allocation feasible (Herroelen & Leus, 2005). However, in real-world problems, project parameters such as activity durations and resource requirements are seldom precisely known. This uncertainty can originate from a great number of potential sources, and make the problems much more complicated to solve but close to real-world circumstance (Zhang, Xu, Liu, & Gen, 2017). During the project, the activities may take longer time than expected, resources may become breakdown or unavailable, due dates may change and rush order may come. All these uncertainties would disrupt the original schedule and incur high costs because of resource idleness, high inventory, and missing deadlines or due dates. As a result, dealing with uncertainty in a project scheduling environment becomes one critical problem.

For deterministic RCSP, we have not only to consider the makespan with precedence relations, but also the resource constraints should be well satisfied. When some kind of uncertainty involved into the problems, the robust has to be considered at the same time. In general, RCSP with uncertainty can be viewed as three objectives: time-based, resource-based and robust-based. Dealing with multi-objective optimization problems, one simple way is, we take these multiple objectives as one with weighting and normalization methods. However, it lies in the proper selection of the weights or utility functions to characterize the decision maker's preferences. In practice, it can be very difficult to precisely and accurately select these weights, even for someone familiar with the problem domain (Shakya, Santana, & Lozano, 2011). Decision makers often prefer a set of good solutions considering the multiple objectives, which make the results much more reasonable and easy to make trade-off decision.

Recently, there are growing interests in the application of estimation of distribution algorithms (EDAs) (Larranaga & Lozano, 2002), which are a class of population-based optimization algorithms. The key idea of EDA is to build an explicit probabilistic model for the distribution of promising solutions found so far, and use the constructed model to guide further search behavior. Furthermore, probabilistic graphical models (PGMs) (Larrañaga, Karshenas, Bielza, & Santana,

[*] Corresponding author at: No. 8, Yongzhi Road, Qinhuai District, Nanjing City, Jiangsu Province, China.
E-mail addresses: jingtiancn@hotmail.com (J. Tian), haoxc@outlook.com (X. Hao), gen@flsi.or.jp (M. Gen).

https://doi.org/10.1016/j.cie.2019.02.039

Available online 23 February 2019
0360-8352/

2012), which are used to represent the interaction behavior among the discrete variables, have been proved to improve the searching ability of EDAs.

In our previous proposal multi-objective Markov network based estimation of distribution algorithm (MMEDA) (Tian, Hao, & Murata, 2016), as one kind of probabilistic graphical models based EDA for solving multi-objective optimization problems, one hybridized fitness assignment function is proposed, and Markov network based EDA acting as constraint handling is integrated.

In this paper, in order to improve the applicability and flexibility of MMEDA, we propose a robust scheduling method based on hybridization of EDA and GA, dealing with resource constrained scheduling problems with uncertainty of activity duration time.

The paper is structured as follows. In Section 2, as one typical application of RCSP, problem of multi-mode resource-constrained project scheduling with uncertainty and the robustness model are illustrated. Section 3 describes the method proposed with multi-phased architecture, including the brief explanation of our previous work MMEDA to make it easy to understand. In Section 4, it provides experimental comparisons of the proposed robust scheduling algorithm on benchmark problems. Finally, Section 5 summarizes the work and the future research direction.

2. Robust optimization of scheduling problem of RCSP

In this section, we mainly discuss the problem of RCSP under uncertainty, which is treated as our application of this study, including one typical application of RCSP, the formulation of robustness measures, and the mathematical model of the problem.

2.1. Multi-mode resource constrained project scheduling problem (MRCPSP)

In MRCPSP, as one typical RCSP, each activity j, $j = 1, \ldots, N$ has to be executed in one of modes M_j. It is assumed that all activities are non-preemptable and decision on the mode of each activity is not changeable until the activity is completed.

In Fig. 1, it shows an illustrative example of project scheduling problem. The network consists of 9 activities, including two dummy activities S and T. For each activity, one mode must be selected from multiple mode candidates. Each mode requires different resource and durations, one illustrate example for determinist problem is listed in Table 1. For the uncertainty problem, the duration of each activity is not a number, but follows some probability distribution or belongs to an interval with lower and upper bound.

MRCPSP can be solved with two decision making processes:

(1) Activity assignment sequencing (*a-seq*): determine the sequence for executing all activities so that the precedence relationships are satisfied.
(2) Mode selection (*m-select*): determine the mode of activity that can be processed among the multiple mode candidates.

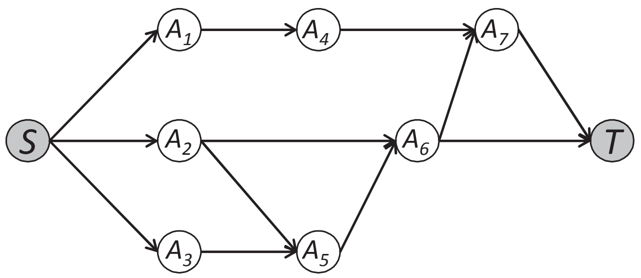


Fig. 1. An illustrative example of MRCPSP.

Table 1
Resource requirement and duration.

Activity	Mode 1			Mode 2		
	$r_{j,k1,m1}$ [a]	$r_{j,k2,m1}$	$d_{j,1}$ [b]	$r_{j,k1,m2}$	$r_{j,k2,m2}$	$d_{j,2}$
1	2	5	2	4	2	1
2	3	5	3	1	2	6
3	1	2	1	3	1	1
4	2	5	2	3	3	3
5	2	4	1	1	3	3
6	3	3	2	5	2	1
7	2	3	3	1	2	5

[a] $r_{j,k,m}$: resource k occupied by the activity j in mode m.
[b] $d_{j,m}$: duration of activity j performed in mode m.

For deterministic MRCPSP, usually, makespan is taken as one optimization criterion. In previous literatures, there are several other kinds of criteria (Brucker, Drexl, Möhring, Neumann, & Pesch, 1999), such as cost minimization (CM) and net present value (NPV). In this paper, we propose some robust measures for uncertainty problems.

2.2. Robust project scheduling measures

In job shop environment, the robustness is often defined as the difference between expected value (e.g., makespan) and realized one (Jensen, 2001). In RCSP, except the duration of project, the resource usage is also need to take into consider. For example, the deviation of the starting time of each activity in the realized schedule and the expected one is to be minimized, or minimize the resource flow network for the problems with unrestricted resource availability (Herroelen & Leus, 2004).

We proposed two kinds of robust measures for RCSP: time-based-robust and resource-based-robust.

(a) Time-based-robust measure (TRM): In order to measure the robustness on the time, one popular way is slack-based. Two types of slack are widely used in the scheduling literature: total slack and free slack. Total slack is the difference between the earliest start time and latest start time of an activity, while free slack is the amount of time that an activity can be delayed without delaying the start of the next activity.

In Fig. 2, it shows one example of total slack time in project scheduling environment. There are 5 activities, and the yellow area is the slack time period for activity A2 while the red area is for activity A3. In previous studies, most of them thought the slack time for A2 and A3 are equal, because the lengths of time period are same. However, from the view of resource allocation in RCSP, A3 requires more amount of resource than A2, in other words, if A3 delayed, more resources should be

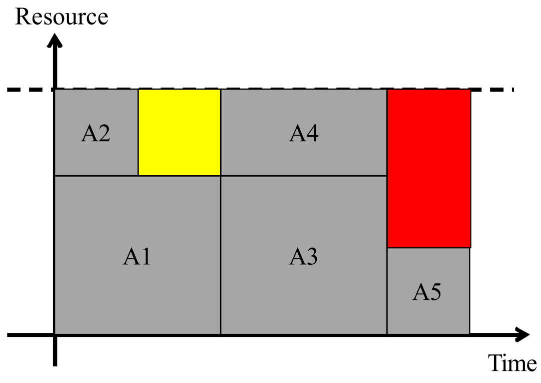


Fig. 2. An illustrative example of slack time.

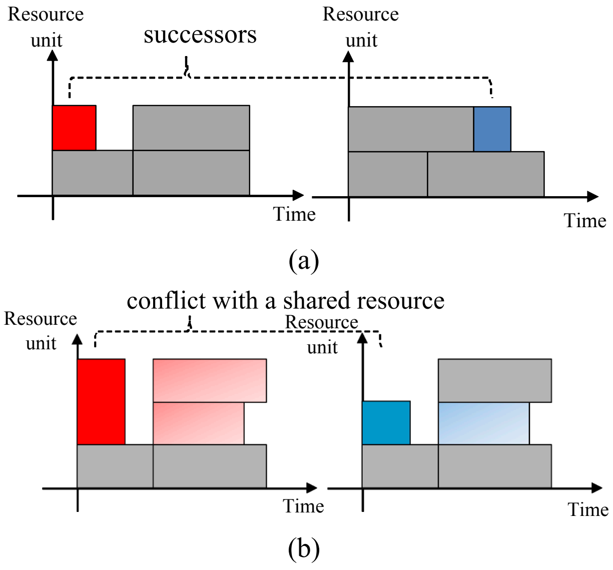


Fig. 3. Amount of successors and conflict resource.

held by it and impact to the system is larger than delay of A2. Meanwhile, from the view of successors, activity A2 has more successors than A3, and the impact of delay of A2 is larger than A3.

As a result, conventional slack-based approach only focus on the length of slack time period to evaluate the robustness of one schedule, but ignoring considering the affect by different amount of resource. As shown in Fig. 3, it shows two typical conditions of slack time. In Fig. 3(a), two activities with red color and blue color have the same time periods of slack time, but for red one, it has more successors than blue one, in other words, if the red one delayed, more operations will be affected. So the amount of successors should be taken into consider together with slack time.

In Fig. 3(b), the red one and blue one have the same slack time, but red one requires more resources than the blue one, if red one delayed, more resources are required and hold by it. In other words, the higher amount of conflict with a shared resource, the bigger impact to the schedule system. As a result, the required resource of each operation has to be well considered also.

In this study, for RCSP, we proposed one new slack-based robust measure which includes amount of successors and resource requirement of activity. These together show the ability to absorb the uncertainty, while keeping the expected makespan.

$$TRM: \quad \sum_{j=1}^{N} s_j NSucc_j \sum_{k=1}^{K} r_{jk} \tag{1}$$

where s_j is the total slack time, $NSucc_j$ represents the number of immediate successors of activity j, and r_{jk} is the resource requirements for activity j.

(b) Capacity-based-robust measure (CRM): For RCSP, one of the key issues is how to allocate the resources, so that the robust measure for resource capacity need to be well studied. From the previous literatures, the uncertainty of duration time is modelled as following the normal distribution, which has been proved effective.

For RCSPs, the budget management on manpower is one critical issue to be considered for decision makers. In this study, we consider another kind of uncertainty of time-adjusted resource capacity: Time-adjusted resource capacity represents the total resource capacity which enforced by time. Take one project for example, we will employ some skilled workers and the total working time of workers could be known in advance (for example, we employ one skilled worker with 8 h per day and 5 days per week), which is the capacity of time-adjusted resource.

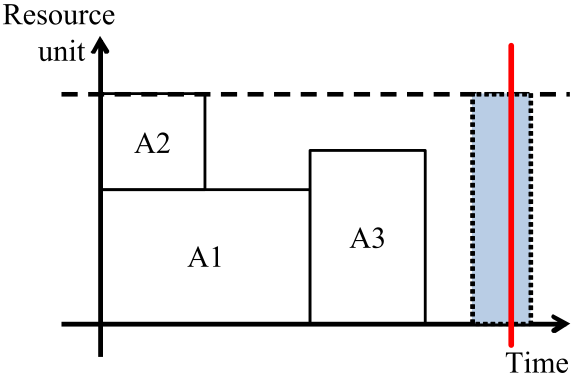


Fig. 4. Resource capacity exceeding with uncertainty.

In real world, there will be some uncertainties in time-adjusted resource capacity. For example, the total working time has a standard level for each worker, which is the original capacity. But sometimes one worker can work overtime. For health of workers or budget management, usually a company will have policy for overtime, which could be viewed as recommendation level, a goal we try to achieve as much as possible.

For example, in Fig. 4, there is one scheduling which containing 3 activities (A1, A2 and A3) and the resources are working time with capacity. The red line represents the total working time (including standard working time and overtime) recommendation level for this kind skilled-workers. Because the duration of each activity is uncertainty, so the total working time is also uncertainty but follows normal distribution. One target of the schedule is trying to satisfy the recommendation level (red line) under the uncertainty environment.

In order to deal with uncertainty of time-adjusted resource capacity, one way is to take it as objective as maximizing the probability of realized total working hour does not exceed the recommendation level, another way is to make it as one chance/soft constraint, which is adopted in our study.

The reasons we taking it as chance constraint are: (a) in real-life problem, one company always have the standard working hour and policy for overtime, so that we can easily get one recommendation level/goal reasonably; (b) making it as chance constraint with one threshold can provide some feasible solutions based on project manager's perspective, which has great significance on budget management; (c) from the view of problem modelling, both objective on time-based-robust and chance constraint on capacity-based-robust are considered, making our model more generic and becoming more easy to calculate.

Here we propose one capacity-based-robust for uncertainty of time-adjusted resource capacity:

$$prob_{\xi\in\Xi}\left(\max_{j=1,\ldots,N}\left(c_j^{\xi}\right) \leqslant G_k\right) \geqslant \text{threshold} \tag{2}$$

where G_k is the recommendation level of time-adjusted resource for resource k.

2.3. Mathematical model

For resource constrained project scheduling problem with duration uncertainty, the objective is to minimize the expected makespan and time-robust measure with the chance constraint of resource-robust:

– *Indices*

i, j	index of activities, $i, j = 1, \ldots, N$
m	index of modes of activity j, $m = 1, \ldots, M_j$
k	index of renewable resources, k, $k = 1, \ldots, K$

Ξ	scenario which corresponds to an assignment of reasonable duration on activity, $\xi \in \Xi$

– Parameters

N	number of activities
M_j	number of modes of activity j, $j = 1, \ldots, N$
K	number of renewable resources
N_k	number of available units of renewable resource k, $k = 1, \ldots, K$
G_k	the recommendation level of resource k
d_{jm}^{ξ}	duration of activity j performed in mode m under the scenario ξ, $j = 1, \ldots, N$, $\xi \in \Xi$
s_{jm}^{ξ}	starting time of activity j under the scenario ξ, $\xi \in \Xi$
c_{jm}^{ξ}	completion time of activity j under the scenario ξ, $\xi \in \Xi$
r_{jkm}	amount of the resource k occupied by the activity j in mode m
p_j	set of all predecessors of activity j
A_j	set of available modes for activity j
$NSucc_j$	number of immediate successors of activity j
s_j	slack time of activity j

– Decision Variable

$$x_{jmt} = \begin{cases} 1 \text{ if activity } j \text{ is performed in mode } m \text{ at time } t; \\ \qquad\qquad 0 \text{ otherwise.} \end{cases} \tag{3}$$

– Objective

$$\min E(C_{\max}) = \frac{\sum_{\xi\in\Xi} \max_{j=1,\ldots,N}(c_j^{\xi})}{|\Xi|} \tag{4}$$

$$\max \sum_{j=1}^{N} s_j NSucc_j \sum_{k=1}^{K} \sum_{m=1}^{M_j} r_{jkm} \tag{5}$$

– Subject to

$$\begin{aligned} &\textstyle\sum_{i=1}^{M_i} \sum_{t=s_i^{\xi}}^{c_i^{\xi}} t \cdot x_{imt} \leqslant \sum_{j=1}^{M_j} \sum_{t'=s_i^{\xi}}^{c_i^{\xi}} (t' - d_{jm}) \cdot x_{jmt'}, \\ &\qquad j = 1, \ldots, N; \quad i \in p_j; \quad \xi \in \Xi \end{aligned} \tag{6}$$

$$\sum_{m=1}^{M_j} \sum_{t=s_j^{\xi}}^{c_j^{\xi}} x_{jmt} = 1, \quad j = 1, \ldots., N; \quad \xi \in \Xi \tag{7}$$

$$\begin{gathered} prob_{\xi\in\Xi}(\max_{j=1,\ldots,N}(c_j^{\xi}) \leqslant G_k) \geqslant \text{threshold} \\ j = 1, \ldots., N; \quad k = 1, \ldots, K; \quad \xi \in \Xi; \quad t = 1. ..horizon \end{gathered} \tag{8}$$

$$\begin{gathered} x_{jmt} \in \{0, \ 1\}, \\ j = 1, \ldots., N; \quad m = 1. ..M_j; \quad t = 1. ..\text{horizon} \end{gathered} \tag{9}$$

$$s_j^{\xi} \geqslant 0, \ c_j^{\xi} \geqslant 0, \quad j = 1, \ldots., N; \quad \xi \in \Xi \tag{10}$$

Inequality (6) presents the precedence constraints of activity. Eq. (7) states that one mode must be selected from a set of available modes for each activity. Inequality (8) takes into account the chance constraint of the resource robustness. Eqs. (9) and (10) represent the nonnegative restrictions.

3. Multi-phased robust scheduling method based on GA and EDA (robust hGMEDA)

When we try to solve one RCSP with uncertainty, there are several points have to be concerned. Therefore, it is a very difficult and complex combinational optimization problem to produce one robust schedule for RCSP under uncertainty. How to handle them together or separately in an effective manner is one critical problem to solve. In this study, a two-phased scheduling method of stochastic optimization combined hGMEDA with scenario based simulation (robust hGMEDA) is developed.

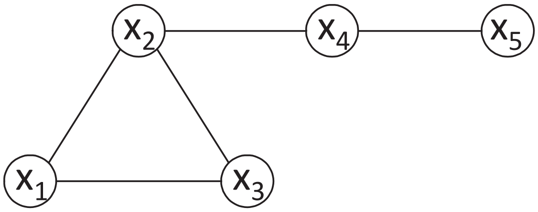


Fig. 5. An illustrative Markov network structure.

3.1. hGMEDA

The detail of our previous work hGMEDA could be found in paper (Tian, Hao, & Murata, 2014), which include combined fitness assignment functions, network structure of Markov network, estimation, sampling and evolving procedure. In this subsection, we will make a brief explanation.

(a) Structure learning

The model of Markov network (MN) consists of graph structure G and parameter Ψ. Fig. 5 shows the example of an MN structure with five random variables. For example, variable X_2 has three neighbors $N_2 = \{X_1, X_3, X_4\}$.

The conditional probability (Shakya, 2006) of a node X_i, can be calculated with given its neighboring states N_i:

$$p(x_i | x - \{x_i\}) = p(x_i | N_i) \tag{11}$$

In order to avoid high complexity, we use mutual information (MI) to estimate the structure (Chen, Anantha, & Lin, 2008), which can be easily adopted in low computation costs. In Eq. (12), we can calculate MI between two random variables X_i and Y_j.

$$\begin{aligned} \text{MI}(X_i; Y_j) &= \textstyle\sum_{x_i\in X_i} \sum_{y_j\in Y_j} \text{p}(x_i, y_j|D) \\ &\quad \times \log\left(\frac{p(x_i, y_j \mid D)}{p(x_i \mid D)p(y_j \mid D)}\right) \end{aligned} \tag{12}$$

where $p(x_i|D)$ and $p(y_j|D)$ are marginal probability of random variables X_i and Y_j over the promising solutions D, respectively, $p(x_i, y_j|D)$ is the joint probability, and the sum is overall possible combinations of random variables X_i and Y_j.

If the MI value of two variables is higher than a threshold, we treat them as neighbors and create an edge between them, which means that they have strong relationship in Markov network.

The value of threshold could be given as one fixed number or we can update the value by the information of MI values we already got, which is shown in Eq. (13).

$$\begin{aligned} &threshold = \alpha \times avg(MI) \\ &= \alpha \times \frac{2 \times \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} MI(X_i; X_j)}{N \times (N-1)} \end{aligned} \tag{13}$$

where the parameter α is used to control the complexity of structure.

If we take α as a high value, so that Markov network has fewer edges and requires less computation time. Otherwise, smaller value α can generate more edges but cost longer time to construct the structure. As a result, optimality and calculation time, partially can be controlled by parameter α.

(b) Sampling

After the structure of Markov network has been estimated, the new candidate solutions will be sampled from the estimated model. Markov network, different from Bayesian network (BN) (Hao, Lin, Chen, & Murata, 2012), does not satisfy the ancestral ordering. An extended Gibbs sampler, as one kind of Markov chain Monte Carlo methods, is selected as the sampling method.

```
Gibbs sampling for Markov network based EDA
begin
    for $i$ := 1 to $popSize$ do
        Step 1    Randomly generate a solution $x = (x_1, x_2,\dots, x_n)$ according to variable $X$;
        for $j$ := 1 to $n$ do
            Step 2    Choose a variable $x_j$ from each solution;
            Step 3    Using the promising data set $D$, estimate the conditional probability $p(x_j|N_j)$ for each value $x_j$ of the variable $X_j$ as Gibbs probability;
            Step 4    Sample conditional probability distribution $p(x_j|N_j)$ to new $x_j$;
        end
    end
end
```

Fig. 6. Gibbs sampling for Markov network based EDA.

The pseudo code of Gibbs sampling is given in Fig. 6.

In order to make the convergence smooth, the conditional probability $p(x_j|N_j)$ is estimated for each decision variable X_j by Gibbs probability with temperature control:

$$p(x_j|N_j) = \frac{e^{p(x_j,N_j)/T}}{\sum_{x_j' \in D(X_j)} e^{p(x_j',N_j)/T}} \tag{14}$$

$$T \sim \beta \times \frac{1}{gen} \tag{15}$$

where $p(x_j, N_j)$ states the joint probability of a variable $X_j = x_j$ and the given value of neighbors N_j. T is the temperature function, to control the trade-off between exploration and exploitation.

(c) Fitness assignment mechanism

Inspired by the idea of point system of decathlon (Westera, 2006), we design a novel fitness assignment function to hybrid different sampling strategies:

$$Central\,Fitness(X) = \frac{1}{1/(p(X)+1) + q(X)} \tag{16}$$

$$D - Fitness(X) = \sum_{i=1}^{m} (N - Ranking_{Obj_i}(X) + 1)^{\omega_i} + (N - Ranking_{CentralFitness}(X) + 1)^{\omega_{m+1}} \tag{17}$$

where $p(X)$ is the number of individuals which are dominated by individual X, $q(X)$ is the number of individuals which dominate X; N is the population size, m is the number of objectives, $Ranking_{Obji}(X)$ is the ranking of individual X based on ith objective, and $Ranking_{CentralFitness}(X)$ denotes the ranking based on the CentralFitness value, ω_i is the exponential parameter.

(d) Evolving process

In Fig. 7, it shows the evolving process of empowered MMEDA by GA (hGMEDA), with the cooperative sequential manner.

Instead of EDA employment during the whole evolving process, GA is adopted in stage 1, because (a) the sequencing problem is not so complex, compared with resource allocation problem. The searching speed of GA is better than Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and EDA; (b) GA can provide more "random" solutions and higher diversity of solutions for next stage, compared with other meta-heuristic algorithms.

In Fig. 8, it shows the problem coding of GA representation to decide activity sequence. We use random key (RK) to represent the priority value for each activity. Based on the vector λ of priority values attributed to each activity and the precedence relation, the activity with higher priority value will be executed before the smaller one.

The initial representation is made by binary coding. The next step is transfer to real number for priority values. The merit of taking the manner of binary coding is easy to perform crossover and mutation, to avoid illegal solutions.

In stage 2, Markov network based EDA is adopted to find and model the interrelation of resource allocation for which activities seizing the same resource. In the application of resource constraint project scheduling problems (RCPSP), each node in Markov network represents the decision variable X_j of mode selection of activity j. In Markov network, one edge between two variables denotes two activities have strong interrelation on resource seizing (due to resource capacity), which is calculated by mutual information.

3.2. Robust hGMEDA

The strategy of robust hGMEDA is shown in Fig. 9.

(a) Phase-1: solve the deterministic problem by hGMEDA

In the first phase, we try to solve the uncertainty RCSP as the deterministic one, taking the duration as the averaged value for each activity. Meanwhile, do not consider any chance constraints. In other words, we take the problem as multi-objective of makespan minimizing and time based robustness maximizing.

In phase-1, we take hGMEDA to calculate candidate solutions. There are some non-Pareto solutions in the archive, and the reason is, it's used for learning the structure and sampling new candidate solutions with diversity. More importantly, it can provide more alternative solutions for next phase in robust scheduling problems.

(b) Phase-2: solve the uncertainty problem by scenario based simulation

In this phase, some scenarios are generated in step 1. As shown in Fig. 10, for each activity, based on its probability model of duration time, sampling N conditions of possible time. Then pick one condition of duration time for each activity, and join them together to generate one scenario.

In step 2, firstly, by scenario based simulation, we evaluate each candidate solution whether to satisfy chance constraint, and reject one

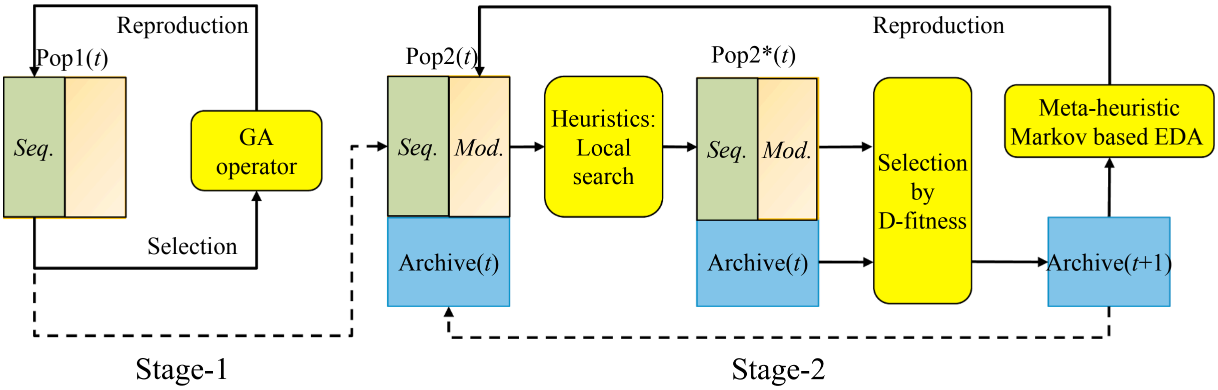


Fig. 7. The evolving process of hGMEDA.

which fails to satisfy. Next, based on the robustness measure, finally the robust schedule is selected.

The flowchart of simulation is shown in Fig. 11.

(c) Problem-specific Local Search

Variable neighborhood search (VNS) is one popular way to do a possibly randomized local search (Hansen & Mladenović, 2005). After solutions are sampled by Markov network in MMEDA, a problem-specific local search is proposed to improve the solution quality (Michiels, Aarts, & Korst, 2007).

The makespan cannot be reduced any further while keeping the critical paths. Our local search is attempting to make a new schedule with shorter makespan by breaking the existing critical path. Different to job shop problem, in project scheduling, it is possible to have several different critical paths on different resources. Here we randomly select only one critical path among all the critical paths, to reduce the computation workload. The target is minimizing makespan while maximizing robustness measure on free slack. So that our local search is based on critical path with considering free slack.

Usually, variable neighborhood search could only increase one objective and cannot guarantee others, especially when the objectives are very complex. Different to the conventional local search, for our problem, the solutions before and after the local search are both kept.

Fig. 12 shows the pseudo code of local search by moving activity for reducing makespan. The purpose of sequence changing is to move the activity from the current position to another feasible position. Since new schedule is obtained by deleting one activity and moving it to another position, it is obvious that the new makespan is no larger than original ones. For project scheduling problem, a new feasible position should satisfy all kinds of resource and without any precedence constraint violations.

When we decide moving one activity, sometimes we can find more than one feasible time interval for it. Next, we have to calculate each time interval:

$$Time\_Interval = \sum_{j \in TI_i} s_i \times NSucc_j \times r_{jkm} \tag{18}$$

where TI_i represent the set of all the activities taking the time interval i as their slack time period.

Finally, we select some promising solutions based on multi-objective, and update the archive. All solutions kept in archive belong to Pareto front set.

4. Experiment

To examine the practical viability and efficiency of our proposed algorithm, we designed a numerical study to compare with efficient algorithms from previous studies.

It is extremely difficult to get real world problem data for MRCPSP with uncertainty. Fortunately, there has been a significant amount of research conducted on the project scheduling problem and one popular benchmark problem data set PSPLIB (Kolisch & Sprecher, 1997) can be used to compare different methods.

In the benchmark PSPLIB, the duration of each activity is integer. In order to create them as the problem with duration uncertainty, we revised the duration of activity in each mode as the normal probability $N(\mu, \sigma^2)$, where μ is the original duration time, and we set σ of one activity as $\mu_i/\text{avg}(\mu)$.

In our experiments, we take other two algorithms to demonstrate: Improving the Strength Pareto Evolutionary Algorithm (SPEA2) (Zitzler, Laumanns, & Thiele, 2001) and Fast Multiobjective Hybrid Evolutionary Algorithm (MOHEA) (Zhang, Wang, Wang, Xiao, & Gen, 2017). For SPEA2, it's a very famous and popular algorithm among

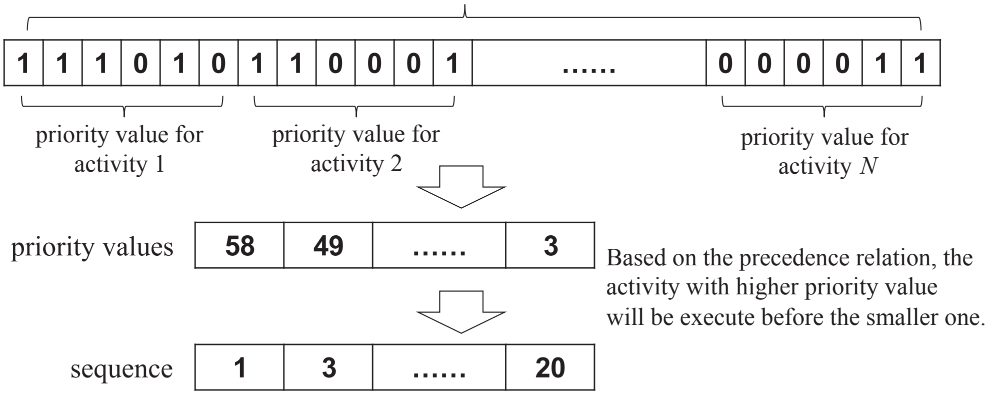

1 1 1 0 1 0 1 1 0 0 0 1 ...... 0 0 0 0 1 1
priority value for activity 1
priority value for activity 2
priority value for activity N
priority values
58
49
......
3
Based on the precedence relation, the activity with higher priority value will be execute before the smaller one.
sequence
1
3
......
20

Fig. 8. GA representation.

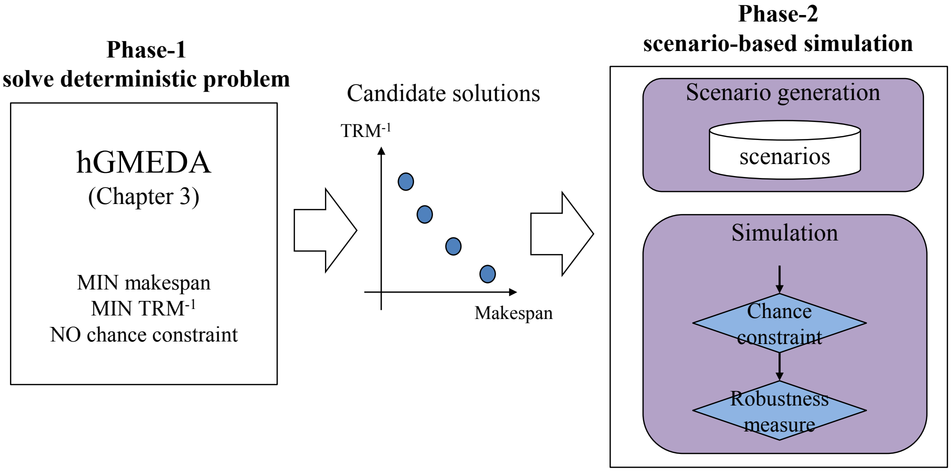


Fig. 9. Strategy of robust hGMEDA.

MOEAs, which is often taken as compared method for demonstration in numbers of previous literatures. For MOHEA, it's one multi-objective optimization method proposed recently, based on novel Pareto dominating and dominated relationship-based fitness function (PDDR-FF). In sum, we select one classic MOEA and one new-developed MOEA for comparison.

We design several group experiments to demonstrate:

(1) To make parameter tuning by sensitive analysis to support our parameter setting;
(2) To demonstrate the optimality of our proposal, which contains (a) the optimality of expected makespan based on different algorithms are compared, (b) the variance of makespan of our schedule compared with deterministic ones to evaluate time based robustness, and (c) the capacity based robustness by comparing the percentage of satisfaction of chance constraint;
(3) To compare with MOHEA on multi-objective optimization measures on (a) coverage and (b) spacing (Zhang, Gen, & Jo, 2014). (MOHEA has already been proved to be better on convergence index and the distributed index than Non-dominated Sorting Genetic Algorithm II (NSGA-II) and SPEA2. As a result, in this paper, we do not list the comparison with NSGA-II or SPEA2 on multi-objective optimization measures.)

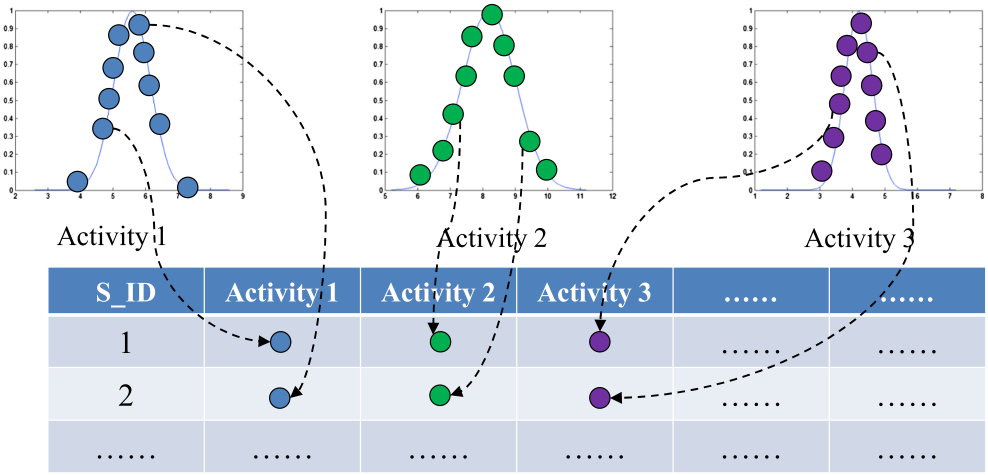
All algorithms were implemented using JAVA under the Eclipse environment and conducted on Intel Core i3 with 4G memory. Data were collected from 30 test runs for each algorithm. In order to make the same environment and fairly compare the performance of these algorithms, the strategies of related algorithms and their respective parameters are presented in Table 2.

4.1. Parameter tuning and sensitive analysis

In the domain of meta-heuristic algorithms, how to make parameter tuning becomes a critical task (Grefenstette, 1986). For example, there are crossover probability P_c and mutation probability P_m in GA, and different applications require different probabilities to reach optimal solutions. Unfortunately, there is no special rule to guide how to set up those appropriate parameters, which is a state-of-the-art problem to researchers.

In our algorithm robust hGMEDA, there are many parameters to adjust: EDA related, Markov-network-related, multi-objective-related, robust-optimization-related. Due to the paper limitation, we cannot give exhaustive parameters tuning. From previous literatures, a great number of experiments have performed on GA, but there is few studies conducted on parameters of EDA, especially of Markov network. Thereafter, we select one typical parameter of Markov network to illustrate, the parameter α in Eq. (13), which is to control the threshold of MI.

In the previous section, we have already discussed that he parameter α is used to control the complexity of structure. We take five different values of α ($\alpha = 0.5$, $\alpha = 0.8$, $\alpha = 1.0$, $\alpha = 1.2$, $\alpha = 1.5$) to



S_ID	Activity 1	Activity 2	Activity 3	……	……
1	●	●	●	……	……
2	●	●	●	……	……
……	……	……	……	……	……

Fig. 10. Scenario generation.

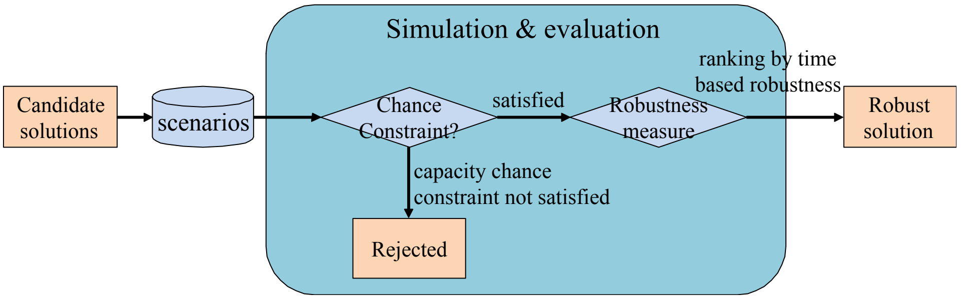


Fig. 11. Flowchart of simulation.

represent five different conditions: very low threshold, low threshold, medium threshold, high threshold, and very high threshold.

For multi-objective optimization problems, the solutions are Pareto set and very difficult to be compared. In order to make the comparison result much more easier to understand, we adopt the coverage (Zitzler & Thiele, 1999) to evaluate the optimality.

Let S_i be a solution set for each algorithm. Coverage $C(S_1, S_2)$ is defined as the percentage of the individuals in solution S_2 which are dominated by S_1.

$$C(S_1, S_2) = \frac{|\{s_2^{\beta} \in S_2;\ \exists\, s_1^{\alpha} \in S_1:\ s_1^{\alpha} \geqslant s_2^{\beta}\}|}{|S_2|} \tag{19}$$

In Eq. (19), if $C(S_1, S_2) = 0$ means that no individual in S_2 is dominated by S_1. If the value $C(S_1, S_2)$ equals to 1 represents that all individuals in Pareto set S_2 are dominated by some individuals in Pareto set S_1. The larger value of $C(S_1, S_2)$ is, the better S_1 is for coverage.

Usually, if we take α as a low value, the threshold value would be lower and Markov network would has more edges, which means more information are kept during the evolving process. So we assume the parameter $\alpha = 0.5$ would lead to better optimality, we take other solutions with different α values to compare with $\alpha = 0.5$ on performance of coverage. From Fig. 13, it shows the comparison results of coverage $(S^{*}_{\alpha=0.5}, S_{\alpha=})$, for different parameter α. (There are totally 1000 generations. We compare the results for every 50 generations.)

We can find that, during the first 50–100 generations, the results are very close, possible reason could be the random generated individuals. Then, the optimality of smaller α value becomes better till 350–400 generations. Next, the gap becomes smaller. The conclusion is smaller α value will reach better solutions in usual case.

From Table 3, we can know the calculation time with different α values with the same 1000 generations. With the α values increase, the calculation time also increases. By considering the results in Fig. 13 together, we select α value as 0.8, to realize the balance between searching optimality and time.

Table 2
The parameters of algorithms compared.

	Generation	Pop_Size	Operator	Parameter

SPEA2	1000	100	Crossover(P_c)
			Mutation(P_m)
			Tournament(k)
				P_c = 0.80
				P_m = 0.20
				elimRate = 0.1
				k = 2

MOHEA	1000	100	Crossover(P_c)
			Mutation(P_m)
			Tournament(k)
				Elite_pop_size = 100
				P_c = 0.4
				P_m = 0.8

Robust hGMEDA	1000	100	Tournament(k) Gibbs sampling Local search	*promisingRate* = 0.7 *elimRate* = 0.1 k = 2 P_c = 0.80 P_m = 0.20 ω_i = 3.0 α = 0.8 β = 0.95

4.2. Expected makespan

In this experiment, we evaluate the optimality of the schedule on makespan minimization. The schedule made by deterministic methods of hGMEDA and SPEA2 only consider bi-objective, and do not consider the chance constraint and without scenario based simulation. In this experiment, the expected makespan generated by robust hGMEDA and other two deterministic ones are compared.

The expected makespan represents the optimality of makespan minimization, which is calculated by Eq. (20):

$$E(C_{\max}) = \frac{\sum_{\xi\in\Xi} \max_{j=1,\dots,N}(c_j^{\xi})}{|\Xi|} \tag{20}$$

```
Local search by moving activity
begin
  Step L1   For a given solution $S$, identify a critical path $P$.
  Step L2   Set $q$ as the first activity in path $P$.
  repeat
      Step L3   Delete $q$ from Gantt chart.
      Step L4   Searching assignable time intervals for $q$.
      Step L5   If there is no assignment time interval, set $q$ as the next activity in
                path $P$. Otherwise, calculate each assignable time, and inset $q$ into
                the highest time interval.
  until    ($q$ is the last activity in path $P$ and no assignment time interval found, take
           $S$ as local optimal)
end
```

Fig. 12. Local search by moving activity.

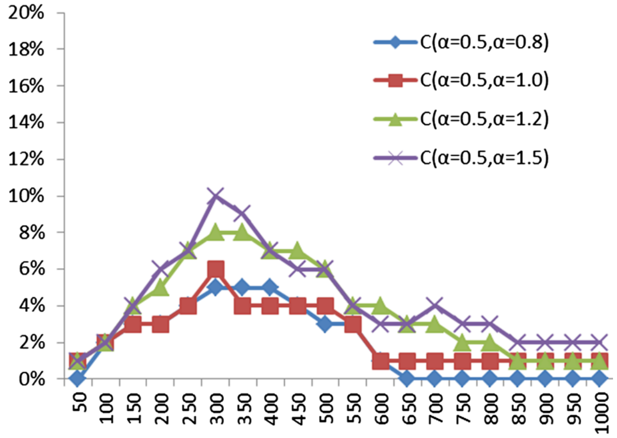


Fig. 13. Comparison on solution optimality with different α values.

Table 3
Comparison on calculation time with different α values.

Value of α	0.5	0.8	1.0	1.2	1.5
Calculation time (s)	185.4	176.1	171.4	165.9	162.2

Table 4
Comparison on expected makespan.

Problem	Mean value of makespan [time] (30 trials)			Improvement	
	Robust hGMEDA	hGMEDA	SPEA2	Decreased (vs hGMEDA)	Decreased (vs SPEA2)
#n041_1	29.9	29.1	28.7	−2.75%	−4.18%
#n042_1	35.1	34.2	34.6	−2.63%	−1.45%
#n043_1	39.3	37.9	38.3	−3.69%	−2.61%
#n044_1	32.2	31.3	30.7	−2.88%	−4.89%
#n045_1	42.7	41.1	40.7	−3.89%	−4.91%
Avg.	35.84	34.72	34.6	−3.17%	−3.61%

In the Table 4, there are the results of comparison on makespan of robust hGMEDA, deterministic hGMEDA and SPEA2. Robust hGMEDA achieved expected makespan about 3.71%, 3.61% larger than deterministic hGMEDA and SPEA2 respectively.

4.3. Variance of makespan

The research goal of this study is to produce one robust schedule for uncertainty project scheduling problem. In this experiment, we demonstrate the time based robustness of our proposal compared with deterministic ones.

In this experiment, three solutions are compared. First one is the single robust solution by our proposal robust hGMEDA; second ones are, under the deterministic manner with the duration taking the averaged value μ, some Pareto solutions are generated with two objectives as makespan and time based robustness, but ignoring the chance constraint of capacity based robustness. One robust solution is selected only based on the duration time choosing its averaged value.

In order to fairly compare the solutions given by different method: Firstly, 30 scenarios are randomly generated. Then, we apply randomly generated 30 scenarios to the solutions given by each method, and calculate the difference between averaged makespan and actual makespan with Eq. (21), and we evaluate the mean result with 30 trials.

$$\text{Variance} = \frac{\sum_{\xi\in\Xi}|MK^{\xi} - MK^{*}|}{|\Xi|} \tag{21}$$

Table 5
Comparison on variance of makespan.

Problem	Mean value of variance [time] (30 trials)			Decrease of variance	
	Robust hGMEDA	hGMEDA	SPEA2	Decreased (vs hGMEDA)	Decreased (vs SPEA2)
#n041_1	2.1	2.4	2.5	14.29%	19.05%
#n042_1	2.5	2.7	2.9	8.00%	16.00%
#n043_1	3.2	3.5	3.4	9.37%	6.25%
#n044_1	2.7	2.9	2.9	7.41%	7.41%
#n045_1	3.8	4.1	4.3	7.89%	13.16%
Avg.	2.86	3.12	3.2	9.39%	12.37%

where Ξ is the amount of the sampled scenarios, MK^{ξ} and MK^{*} are actual and averaged makespan under the scenario ξ.

Variance represents the tolerant ability for uncertainty of each solution. The smaller difference is, the higher time based robustness is.

In Table 5, it shows the results of robustness comparisons. On average, our method improved time based robustness about 9.39% and 12.37%, compared with the approach for deterministic scheduling methods based on deterministic hGMEDA and SPEA2 under the same condition of duration uncertainty respectively.

4.4. Percentage of satisfying chance constraint

In this experiment, we compare the capacity based robustness by using the percentage of satisfaction of chance constraint. For the solutions given by each method, based on the 30 scenarios, we check how many percentage of solutions satisfying the chance constraint of capacity based robustness measure (CRM).

The higher percentage is, the higher robustness on capacity based robustness. The results are shown in Table 6. Our proposal robust hGMEDA can increase the percentage of satisfaction of chance constraint about 11.5% and 10.2% for deterministic hGMEDA and SPEA2 averagely.

4.5. Coverage

In Eq. (19), the definition of "coverage" has already been given. In this experiment, the optimality of Pareto set are demonstrated by comparing with MOHEA.

In Table 7, it shows the comparison on coverage of robust hGMEDA and MOHEA. Mean value represents optimality of solutions in Pareto set among 30 trails. Robust hGMEDA outperforms MOHEA with benchmark problems. The percentage of improvements is very hard to evaluate two coverage measures. In order to make it simple to understand, we use coverage(robust hGMEDA, MOHEA) minus coverage (MOHEA, robust hGMEDA) to represent the improvements.

Table 6
Comparison on percentage of satisfying chance constraint.

Problem	Mean value of percentage (30 trials)			Improvement	
	Robust hGMEDA	hGMEDA	SPEA2	Increased (vs hGMEDA)	Increased (vs SPEA2)
#n041_1	85.4%	74.5%	75.1%	10.9%	10.3%
#n042_1	86.2%	77.3%	76.8%	8.9%	9.4%
#n043_1	85.8%	72.1%	73.9%	13.7%	11.9%
#n044_1	83.7%	70.1%	72.7%	13.6%	11.0%
#n045_1	82.1%	71.7%	73.5%	10.4%	8.6%
Avg.	84.6%	73.1%	74.4%	11.5%	10.2%

Table 7
Comparison on coverage of robust hGMEDA and MOHEA.

Problem	Mean value (30 trials)		
	A_1*	A_2*	Improved (A_1-A_2)
#n041_1	0.323	0.302	0.021
#n042_1	0.367	0.344	0.023
#n043_1	0.218	0.209	0.009
#n044_1	0.387	0.346	0.041
#n045_1	0.225	0.204	0.021
Avg.	0.304	0.281	0.023

* A_1 = C(robust hGMEDA, MOHEA); A_2 = C(MOHEA, robust hGMEDA).

Table 8
Comparison on spacing of robust hGMEDA and MOHEA.

Problem	Mean value (30 trials)		
	A_3*	A_4*	Improved (A_3-A_4)
#n041_1	37.26	38.04	0.78
#n042_1	35.31	36.12	0.81
#n043_1	31.97	32.27	0.3
#n044_1	35.24	36.15	0.91
#n045_1	29.83	30.58	0.75
Avg.	33.92	34.63	0.71

* A_3 = SP(robust hGMEDA); A_4 = SP(MOHEA).

4.6. Spacing

SP(*S*), usually used to represent the distribution performance, which is the standard deviation value of the nearest distances between any two individuals in the solution *S* (Schott, 1995). Smaller *SP*(*S*) means that solution *S* is in better diversity.

$$\bar{d} = \frac{\sum_{i=1}^{|S|} d_i}{|S|} \tag{22}$$

$$SP(S) = \sqrt{\frac{1}{|S|-1}\sum_{i=1}^{|S|}(d_i-\bar{d})^2} \tag{23}$$

where d_i is the nearest distance of individual i in solution set S.

Similar to evaluation of coverage, for spacing, it's still difficult to explain the improvement based the index of spacing. In the experiment results, we use the difference between SP(robust hGMEDA) and SP (MOHEA) to represent the improvement. From Table 8, it shows that robust hGMEDA has smaller *SP* than MOHEA, which demonstrates that our proposal is better on distribution performance. There could be some reasons: (1) With the combined sampling strategies, for both the edge region and the central region, our proposal can keep the solution with diversity. (2) One simple mechanism to preserve the diversity evenly through dynamic adjustment on D-Fitness is adopted, ensuring to achieve satisfactory dispersion performance. (3) Robust hGMEDA employs GA in the first stage, which can provide more "random" solutions.

5. Conclusion

This paper descries a robust scheduling method based on hGMEDA, dealing with scheduling problems with uncertainty of activity completion time durations. Firstly, two kinds of robust measures on time-based-robust and capacity-based-robust are introduced to evaluate the robustness of scheduling solutions, and we formulate the robust scheduling problem as two objectives of minimizing makespan and maximizing time based robustness under a chance constraint of satisfying the threshold of capacity based robustness. Thereafter, by using scenario-based simulation, a stochastic robust multi-objective optimization method named robust hGMEDA is proposed. In the first phase, with the averaged duration, the problem is solved as the deterministic multi-objective scheduling problem without considering duration uncertainty and chance constraints, and some candidate solutions are collected by using hGMEDA. In the second phase, the alternative solutions are checked by the chance constraints of capacity-based-robust measure and then, time based robustness measure is evaluated by using scenario-based simulation.

In our future work, more experiments will be conducted to determine the accuracy of the proposed robust hGMEDA in response to variations among the parameters. And decision support system for project manager could also be another research direction in future, by analyzing the results from the solutions.

References

Brucker, P., Drexl, A., Möhring, R., Neumann, K., & Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research, 112–1*, 3–41.

Chen, X., Anantha, G., & Lin, X. (2008). Improving Bayesian network structure learning with mutual information-based node ordering in the K2 algorithm. *IEEE Transactions on Knowledge and Data Engineering, 20–5*, 628–640.

Grefenstette, J. J. (1986). Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, 16*(1), 122–128.

Hansen, P., & Mladenović, N. (2005). Variable neighborhood search. In E. K. Burke, & G. Kendall (Eds.). *Search methodologies* (pp. 211–238). US: Springer.

Hao, X., Lin, H. W., Chen, X., & Murata, T. (2012). Cooperative Bayesian optimization algorithm: A novel approach to multiple resources scheduling problem. *IEEJ Transactions on Electronics, Information and Systems, 132–12*, 2007–2018.

Herroelen, W., & Leus, R. (2004). The construction of stable project baseline schedules. *European Journal of Operational Research, 156*, 550–565.

Herroelen, W., & Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials. *Project Management and Scheduling, 165*, 289–306.

Jensen, M. T. (2001). *Robust and flexible scheduling with evolutionary computation.* Ph.D. thesisDepartment of Computer Science, University of Aarhus.

Kolisch, R., & Sprecher, A. (1997). PSPLIB – A project scheduling problem library: OR Software – ORSEP operations research software exchange program. *European Journal of Operational Research, 96–1*, 205–216.

Larrañaga, P., Karshenas, H., Bielza, C., & Santana, R. (2012). A review on probabilistic graphical models in evolutionary computation. *Journal of Heuristics, 18–5*, 795–819.

Larranaga, P., & Lozano, J. A. (2002). *Estimation of distribution algorithms: A new tool for evolu-tionary computation.* Springer.

Michiels, W., Aarts, E., & Korst, J. (2007). *Theoretical aspects of local search.* Springer Science & Business Media.

Schott, J. R. (1995). Fault tolerant design using single and multicriteria genetic algorithm optimization. *Massachusetts Institute of Technology*.

Shakya, S. K. (2006). *DEUM: A framework for an estimation of distribution algorithm based on markov random fields.* The Robert Gordon University.

Shakya, S., Santana, R., & Lozano, J. A. (2011). A Markovianity based optimisation algorithm. *Genetic Programming and Evolvable Machines, 13–2*, 159–195.

Tian, J., Hao, X., & Murata, T. (2014). Markov network based EDA and robust project scheduling with multi-mode resource constraint and rework uncertainty. *The Third Asian Conference on Information Systems, 370–377*.

Tian, J., Hao, X., & Murata, T. (2016). Markov network based multi-objective EDA and its application for resource constrained project scheduling. *IEEJ Transactions on Electronics, Information and Systems, 136–3*, 290–298.

Westera, W. (2006). Decathlon, towards a balanced and sustainable performance assessment method. *New Studies in Athletics, IAAF, 21–1*, 39–51.

Zhang, W., Gen, M., & Jo, J. (2014). Hybrid sampling strategy-based multiobjective evolutionary algorithm for process planning and scheduling problem. *Journal of Intelligent Manufacturing, 25*(5), 881–897.

Zhang, W., Wang, Y., Wang, C., Xiao, L., & Gen, M. (2017). Fast multiobjective hybrid evolutionary algorithm based on mixed sampling strategy. *International conference on management science and engineering management* (pp. 106–116). Cham: Springer.

Zhang, W., Xu, W., Liu, G., & Gen, M. (2017). An effective hybrid evolutionary algorithm for stochastic multiobjective assembly line balancing problem. *Journal of Intelligent Manufacturing, 28*(3), 783–790.

Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm.

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation, 3*(4), 257–271.