Waseda University Doctoral Dissertation

Study on Multi-objective Estimation of Distribution Algorithm for Efficient and Robust Scheduling of Resource Constrained Scheduling Problem

Jing TIAN

Graduate School of Information, Production and Systems
Waseda University

September 2017

Table of Contents

List of figures

List of tables

Chapter 1
Introduction

1.1 Resource constrained scheduling problem (RCSP)

Today, the rapid changes coming from technology and social culture have forced manufacturing to achieve the improvement on the responsiveness and reactiveness of the manufacturing systems to increase the competition ability for one company. Production scheduling for manufacturing systems, is very important in production management and planning, which becomes increasingly impact on the productivity and profitability in a globally competitive market [1]. Meanwhile, most of scheduling problems are very famous as the complicated combinatorial optimization problems and NP hard, especially for the problems including the constraints with both precedence relations and resource capacity.

For manufacturing system, there are many typical problems and applications: Flow Shop Scheduling Problem (FSSP) [2], [Flexible] Job Shop Scheduling Problem (JSP/FJSP) [3][4], scheduling for Cellular Manufacturing System (CM) [5], and Project Scheduling Problem (PSP) [6], etc.

Scheduling problem could be view as one complex multi-dimensional discrete optimization procedure, which includes operations sequencing and resource allocation. Usually, the scheduling problems are described as some jobs have to be executed on various kinds of different resources. The final outputs are choosing the best sequence and amount of resource for each job, based on some kind of scheduling criteria.

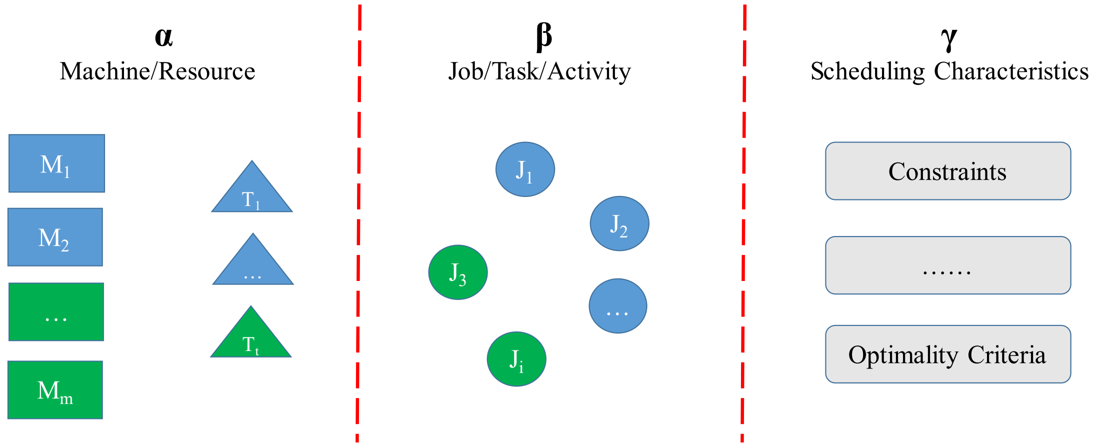


Fig. 1.1 Three-filed problem classification $\alpha|\beta|\gamma$ for scheduling problem

In general, scheduling problems can be modelled as 3-filed problem classification $\alpha|\beta|\gamma$, which represents machine environment, job characteristics and scheduling characteristics respectively [7]. As shown in Fig. 1.1, in left side, α is machine environment. For example, the machine configuration, the characteristics of tools, workers or other types of resources are described here. β means job, task or activity in different production systems. Each one has its own information, such as operation time, resource requirement, release date, due data, and some precedence relations among them, which could be represented as linear, tree or network structure. In the right side, γ is the scheduling characteristics, typically, it includes some specific constraints for types of manufacturing systems and objective functions of scheduling. Most of the objectives depend on the completion time, typically, the tardiness or makespan minimization.

For overviewing the scheduling problems, generally we can divided them into three groups hieratically in Fig. 1.2: Master Production Schedule (MPS), Resource Constrained Scheduling Problem (RCSP) and Flow Shop Scheduling Problem (FSSP). For MPS, it belongs to planning level without considering of capacity. The decision makers always consider the requirements from the sales and try to decide the total output. For RCSP, it belongs to medium level of operational level with capacity planning. From the view of budget management, the schedule by RCSP should decide the suitable required amount of capacity for each different kinds of resources. For FSSP,

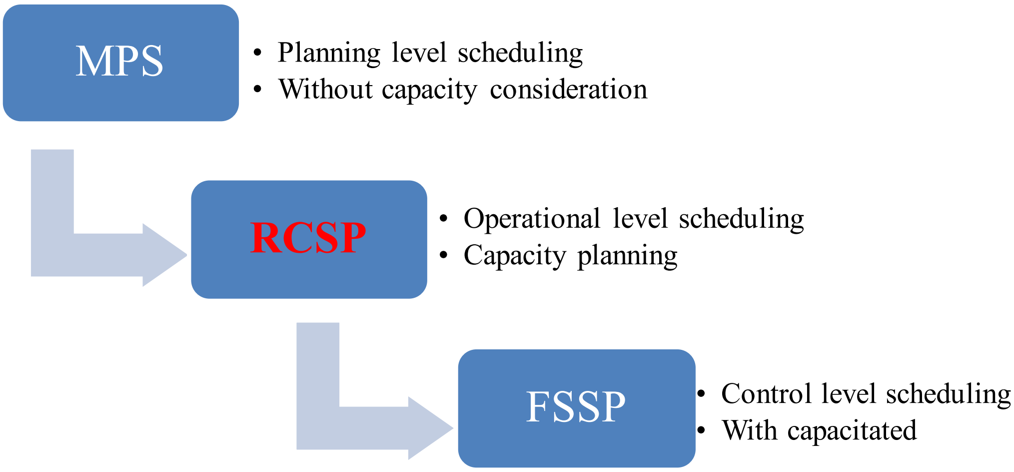


Fig. 1.2 Hierarchical structure of scheduling

it's one kind of control level scheduling with capacitated resources given. Under the limitation of resource, the scheduler tries to optimize the makespan or to achieve other schedule criteria.

Therefore, RCSP is an important and realistic scheduling problem in manufacturing scheduling to make feasible operational schedule which not only minimizes the makespan but also makes resource allocation load balancing with satisfying resource constraints [8]. It provides the connection from high-level planning to low-level controlling in manufacturing system, which is the reason we taking RCSP as our research's main target application.

For RCSP, many researches have been conducted in recent years. The methods can be divided into three parts: exact methods, heuristic methods and meta-heuristic methods. Here we briefly introduce them.

a) Exact methods: Johnson firstly developed an exact method by using branch-and-bound algorithms to solve RCSP [9]. After that, more and more researchers proposed approaches based on B&B. The main contributions of these methods are mainly depend on the searching technology on tree structure, for example, by using dominance rules [15], [16], lower bounds [12], [13], and immediate selection [14], [15].

The advantage and disadvantage of B&B methods are both apparent. B&B methods can provide optimal solutions, but the calculation times are very long for big-size problems. Though some technique on tree searching improve the solving ability, B&B still can not be a suitable and practical way to solve complex large scale problems.

b) Heuristic methods: Kelley proposed the first heuristic methods to solve the RCSP by using the priority-rule [16]. Except priority-based heuristics, there are some other researches, such as truncated B&B [17], integer programming based heuristics [18], local constraint based analysis [19], disjunctive arc concept [20] and so on.

Compared with exact methods based on B&B, heuristic methods can not provide the optimal solutions, but they can solve large problems in acceptable times. Moreover, some good heuristic methods can provide the initial solutions for meta-heuristic methods. However, designing a good heuristics method is one very difficult job, and the current approaches are very problem specific methods. It becomes one popular potential research direction on extending them to more general scheduling problems.

c) Meta-heuristic methods: For RCSP, the meta-heuristic methods mainly belong to Evolutionary Algorithms (EAs) we have discussed, such as Genetic Algorithm (GA) [21], Simulated Annealing (SA) [22], Tabu Search (TS) [23], Ant Colony Optimization (ACO) [24], Particle Swarm Optimization (PSO) [25], Estimation of Distribution Algorithm (EDA) [26], and Differential Evolution (DE) [27].

Compared with B&B methods and heuristic methods, meta-heuristic methods have better performance on calculation efficiency, which can generate optimal solutions with shorter times, especially for large-size and complex problems. Meanwhile, due to the generic evolving procedure, the adoption of meta-heuristic methods are much easier than other methods. In other words, they have more widely availability and flexibility to solve the scheduling problems based on meta-heuristic methods.

Estimation of Distribution Algorithms (EDAs), as a class of population-to-population meta-heuristic optimization algorithms, provide higher optimality than conventional Evolutionary Algorithms (EAs). In EDA, the core issue is the probability model estimating by promising data. Instead of crossover in GA, through sampling candidate solutions with the distribution of probability model, EDA can lead to further search in a convincing way. And, conventional EDA could be enhanced by probabilistic graphical models (PGMs) [28], which could be used for modelling the interaction relationship among the variables. Various experiments have illustrated that PGMs can improve the searching ability of EDAs [29]. As one kind of PGMs, Markov network (MN) was adopted to enhance the conventional EDA (MEDA) [30], by which the network structure is to model the interrelation among variables with the assumption of neighborhood relation, not in parenthood.

However, there are very seldom research of the current PGMs based EDA considering the application to solve the more complicated real-world problem with multi-objective or under uncertainty environment in RCSP.

Firstly, PGMs based EDA can provide more convincing solutions but very time-consuming. For solving the multi-objective problems based on Markov network based EDA, although the key issue for the multi-objective evolutionary algorithms is fitness assignment mechanism, it would become low performance or even impractical to take MEDA as searching engine independently with multi-island model. Therefore, the optimality and computing efficiency of these methods are insufficient and need to be improved, which is one changeling job to combine fitness assignment functions within the evolutionary procedure of EDA.

Secondly, in order to deal with uncertainties, a robust schedule is needed to against some disruptions occurred during the schedule executing. No matter to use stochastic optimization or chance constraint programming, in order to make the final solution of schedule more convincing, usually the enough number of scenarios are required to be sampled for evaluation. In other words, it's also a very time-consuming task to solve the uncertainty problems. PGMs based EDA can provide more convincing solutions but require longer time due to the structure learning and sampling, no matter by which kind of graphic model. Seldom researches conducted on the uncertainty resource constrained scheduling problems. As a result, it is another motivation for us to make the enhancement for PGMs based EDA.

1.1.1 Multi-objective optimization of RCSP

In RCSP, it contains two groups of decisions: how to sequence the tasks to avoid the precedence constraints and how to allocate the resources to each task.

In Fig. 1.3, it shows one feasible solutions containing sequencing and resource allocation. For example, there are three types of resources r_1, r_2, r_3, and 7 tasks (from t_1 to t_7). This is one simple example, but for the complex problems, some tasks require several types of resources simultaneously to complete. In that case, the constraints of resource allocation become more complicated to solve.

As one scheduling problem with resource constrained, the considering of resource utilization, makespan or budget management are always taken into account. All these criteria have to be well

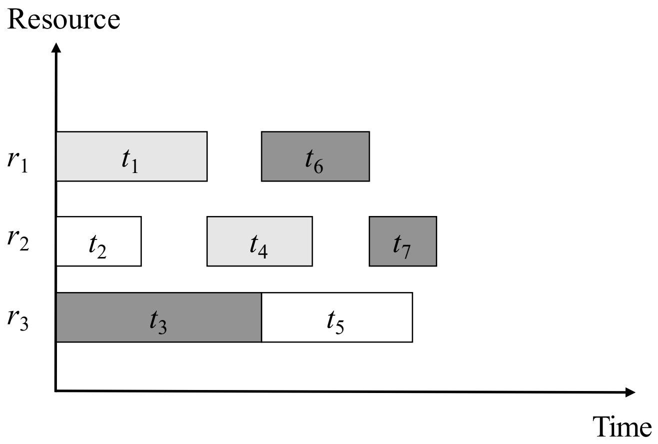


Fig. 1.3 One illustrative solution of RCSP

organized and simultaneously optimized to improve the competitiveness. Therefore, a great number of RCSPs are multi-objective optimization problems naturally.

For RCSPs, the objectives are classified as three groups in general: Regular objectives, Resource Leveling (RL), and Net Present Value (NPV) [31].

The regular objective for RCSP is mainly depend on the completion times, such as the most popular one that minimizing makespan. Besides, there are several other targets, for example, minimizing the delay for due date of project, or min-max the completion time of each sub-project. All these objectives are time-based criteria, which is similar to other types of production scheduling problems.

Second types are resource leveling problems [32]. In this domain, we try to minimize or maximize the variation of resource usage. All these problems are stated as follows:

$$\min \sum_{k \in K} \{c_k \times f(r_k(SS))\} \tag{1.1}$$

where *SS* represents solution of schedules, r_k and c_k are the amount of consumptions and unit cost for resource *k* respectively.

Here we list three typical types of objective functions:

$$f(r_k(SS)) = \max_{t=1,2...} r_k(SS) \tag{1.2}$$

$$f(r_k(SS)) = \sum_{t=1,2...} \left| r_k(SS) - G_k \right| \tag{1.3}$$

$$f(r_k(SS)) = \sum_{t=1,2...} \left| r_k(SS) - \overline{r_k(SS)} \right| \tag{1.4}$$

where G_k represents the goal value of resource usage.

In equation (1.2), it belongs to resource investment problem with minimizing the total consumption of resources. In equation (1.3), it calculates the deviation between actual usage of resource k for schedule SS and a goal value G_k. In equation (1.4), it means the variation with averaged utilization for resource k.

The third type NPV is depend on the concepts of cash flows:

$$\max \sum_{j \in J} (\beta_j \times c_j) \tag{1.5}$$

where β_j is the discount rate for activity j, and c_j is the cash flow, which could be positive (benefit achieved) or negative (cost incurred).

Therefore, only considering single objective is not suitable way to handle the RCSPs, a great number of RCSPs are multiply objectives optimization problems naturally. One simple way is that, we take these multiple objectives as one with weighting and normalization methods. However, in real-world problem, it is very impractical to set one suitable weights for each objective [33]. Firstly, even for problem experts, it is hard for them to decide the weighting value to characterize their own preferences. Secondly, different decision makers have different preferences, so that one single optimal solution maybe not the best answer for other project managers. Thirdly, a set of good solutions are always better than one single solution, because it provides more chances to select, making the results much more reasonable and easy to make trade-off decision.

1.1.2 Robust scheduling of RCSP

In real-world problems of RCSP, parameters such as activity durations of completion time can not be known exactly in advanced. For example, the duration of each activity is not a deterministic value. Other possible conditions for the project scheduling are, the resources may breakdown (for machine) or unavailable (for manpower), due dates may change and rush order may come [34].

These uncertainties could disrupt the original schedule and incur high costs by resource idleness, high inventory, and missing deadlines. Meanwhile, the uncertainties involved make the problems complex to address. Therefore, dealing with uncertainty in a scheduling environment becomes another critical problem in RCSP, which has significant impacts on productivity, customer satisfaction and profitability.

Based on different level of uncertainty, there will be different manner of schedule to cover them. As shown in Fig. 1.4, it describes three different kinds of schedules to manage different uncertainty levels.

From left to right, the level of uncertainty becomes higher. For low uncertainty or without uncertainty, we usually take all the parameters statically and make one optimal schedule with deterministic manner.

However, with the uncertainty increases, the deterministic schedule cannot afford any more. Then the proactive schedules are required, which are also called baseline schedules or robust schedules. For dealing with medium uncertainty, a satisfied schedule is made with considering uncertain conditions to avoid their effect on the schedule and to make its performance to be more

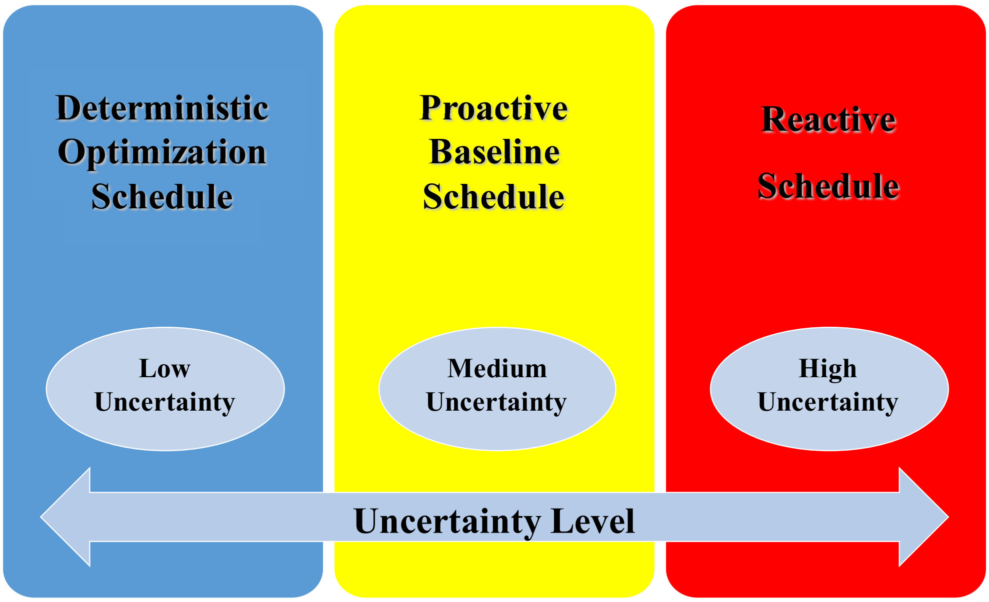


Fig. 1.4 Three types of schedule for different uncertainty level

predictable. Usually, one schedule is preferred that minimizing variance of performance from the expected or averaged one.

When the uncertainty is high or acted as some unexpected disruptions, including the emergency jobs, machine breakdown, or manpower unavailable [35], the baseline schedules cannot protect so well against the disruptions we discussed above. One revised or re-optimized schedule is generated, madding by some rules, policies or optimization approaches, to update a baseline schedule dealing with some disruptions.

Meanwhile, there is another type of hybrid scheduling manner to deal with uncertainty is called predictive-reactive scheduling, which could be viewed as the integration of proactive schedule and reactive schedule. It has three steps usually. In step 1, one predictive schedule is produced as one proactive baseline considering the uncertainty of disruptions. In step 2, after some disruptions occurred, if the predictive schedule can well absorb, the schedule is executed continue. In step 3, if the initial schedule cannot be executed any more, one reactive schedule is generated then.

1.2 Objective of research

As mentioned above, the optimality and computing efficiency of conventional methods are insufficient and need to be improved and also study of robust scheduling for RCSP has not been studied enough. In this study, we make two major contributions to solve RCSP based on EDA, developing efficient multi-objective scheduling method and robust scheduling method. In Fig. 1.5, it shows the bird view of proposed contents.

(1) We enhance MEDA for multi-objective optimization to solve RCSP as multi-objective scheduling problems and propose multi-objective Markov network based EDA (MMEDA) to find Pareto optimal solution set by introducing new fitness assignment functions. Two-stage

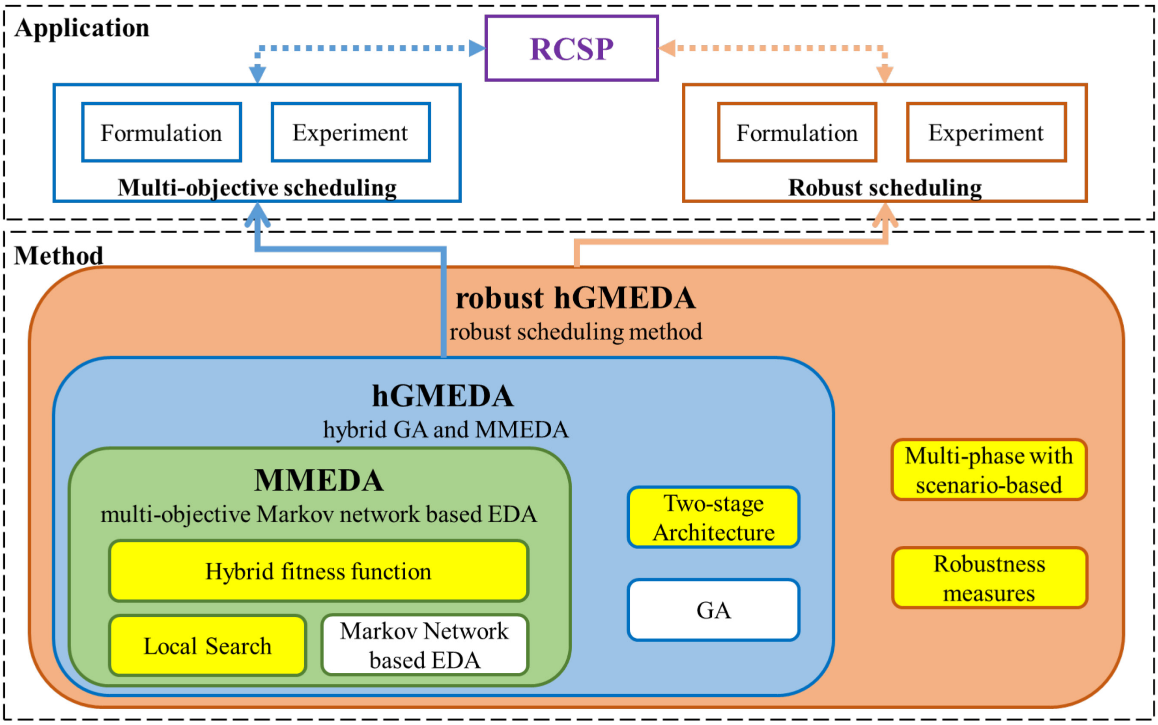


Fig. 1.5 Bird view of proposed contents

architecture of hybridizing GA and MMEDA (hGMEDA) is also proposed to improve the calculation efficiency of MMEDA.

(a) Multi-objective Markov network based EDA (MMEDA): Firstly, fitness assignment functions are developed to achieve diversity in distribution and low calculation cost simultaneously. Two kinds of simple but effective fitness assignment functions are proposed to cover both edge region and central region, that guarantee the solutions have better diversity in the Pareto set. Thereafter, inspired by the idea of point system of decathlon, we design a novel function to combine different functions. It can realize not only normalization of differences of scale size, but also normalize differences of increasing rate of scale with adjustable exponential parameter.

Secondly, in order to increase the searching performance, one PGM based EDA, Markov network based EDA is applied in this study, in which the network structure is very suitable way to model and solve the resource allocation problem in RCSP.

Thirdly, in order to improve the quality of each candidate solution, a problem-specific local search based on Variable Neighborhood Search (VNS) [36] is developed. For bi-objective problems, two types of local search are proposed, for time-based objective and resource-based objective.

(b) Two-stage hybrid GA and MMEDA (hGMEDA): Both algorithm of PGM based EDAs and MOEAs are very time consuming, in order to increase the calculation efficiency of proposed MMEDA, the algorithm hybrid GA and MMEDA (hGMEDA) is developed to solve resource capacitated scheduling problems. Inspired by the cooperative co-evolutionary, in hGMEDA, a two-stage architecture based on sequential co-evolutionary paradigm is proposed.

In the first stage, GA is employed to find feasible solution for sequencing sub-problem without resource capacitated, because GA can provide more "random" solutions and higher diversity of solutions. In the second stage, based on the partial solutions given by stage-1, MMEDA is adopted to model the interrelation for resource allocation and calculate the Pareto optimal solution set.

(2) In order to deal with these uncertainties, a multi-phase robust scheduling method based on hGMEDA is proposed for robust scheduling. Two measures of time-based robustness and capacity-based robustness are introduced and a robust multi-objective optimization method by using scenario-based simulation is also proposed.

(a) Robust scheduling method based on hGMEDA (robust hGMEDA)

Based on the algorithm of hGMEDA we proposed, a robust scheduling method based on hGMEDA was developed, to increase applicability and flexibility of EDA for more widely applications.

Firstly, two kinds of robust measures on time-based-robust and capacity-based-robust are well defined to evaluate the solutions, and we treat them as chance constraint and objective to make the problem more practical to solve.

Secondly, a multi-phase scheduling method of stochastic optimization combined hGMEDA with scenario based simulation is proposed. In the first phase, with the averaged duration, the problem is solved as the deterministic multi-objective manner and some solutions are collected by using hGMEDA. In the second phase, all the alternative solutions are checked by potential chance constraints, some unsatisfied solutions are cleaned out. In the third phase, the remaining solutions are evaluated with robustness measures, by using the scenario-based simulation, finally the one with the highest robustness is selected.

Thirdly, one problem-specific local search with considering both makespan and robustness under uncertainty environment is designed to increase the solution quality.

1.3 Organization of dissertation

The chapters in this dissertation are structured as follows (shown in Fig. 1.6). In Chapter 2, we give a literature review of meta-heuristic algorithms for solving RCSP, especially focus on EDA and PGM based EDA. Then some conventional multi-objective evolutionary algorithms and robust optimization approaches are presented briefly. Chapter 3 describes our proposal MMEDA and hGMEDA to enhance Markov network based EDA, with multi-objective optimization and GA hybridization. Next, in order to confirm the effectiveness of our methods, some experiments are performed on benchmark problems with comparisons with two famous MOEAs in Chapter 4. In Chapter 5, a multi-phase robust scheduling method based on hGMEDA (robust hGMEDA) is presented. Chapter 6 presents the application of robust scheduling problems for RCPSP with duration uncertainty, which acting as the case study for evaluating the robustness performance of

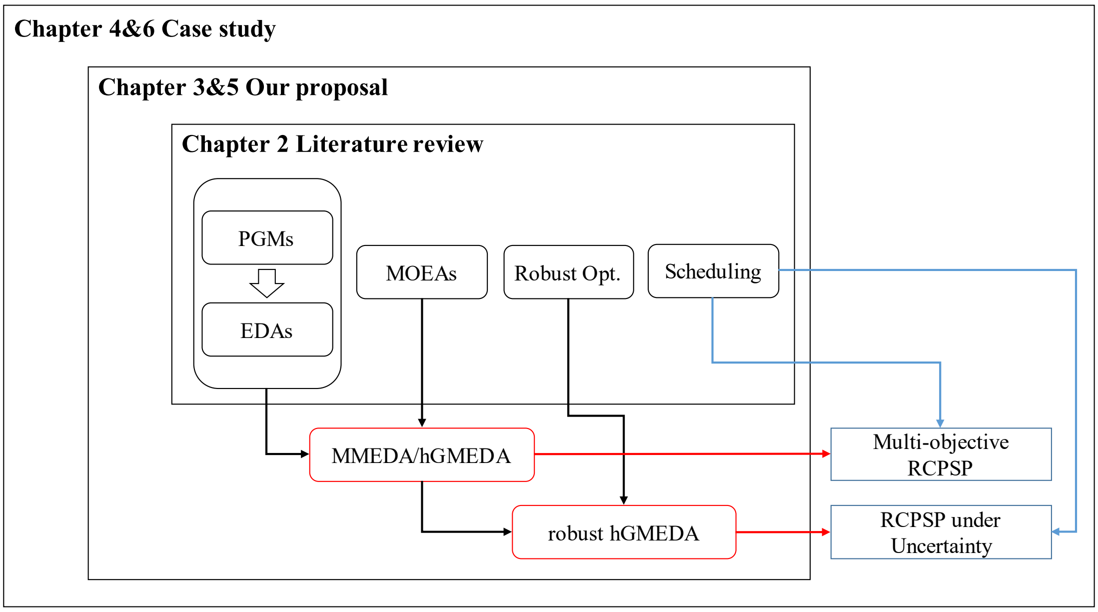


Fig. 1.6 Organization of dissertation

our proposal. In Chapter 7, we conclude the thesis by reviewing results of our approaches and contributions. The potential topics for further research are also discussed.

Chapter 2
Literature Review

2.1 Estimation of distribution algorithm

In the research domain of combinatorial optimization, a significant amount of algorithms were developed. Evolutionary Algorithm (EA), based on the operations of selection and mutation, is one kind of population-to-population meta-heuristic optimization algorithms [37]. EAs almost can perform good enough solutions to all kinds of research field due to its problem-independent. The general processes of EAs are:

a) Generate the initial population;

b) Evaluate each individual with some criteria;

c) Regenerate the population, and go to b) until termination.

In the field of Evolutionary Algorithms (EAs) or Evolutionary Computation (EC), there are several kinds of meta-heuristic algorithms proposed to solve the practical applications, such as GA [38], SA [39], TS [40], and PSO [41].

As one typical EA, GAs [21] are perhaps the most popular and well-known algorithms. Various of EAs mainly differ from the scheme of regeneration. In GA, next generation of population is generated based on some better solutions coming from the last generation, with the genetic operation including crossover and mutation.

In Fig. 2.1, it shows how a generic GA works.

Generic Genetic Algorithm

begin

Initialization:

Step 1 Set $t = 0$;

Step 2 Initialize the first generation population *pop*(0) randomly;

Step 3 Evaluate each individual in *pop*(0);

while *terminating criteria is not met* **do**

Step 4 Select *pop*(t) from *pop*(t-1) based on some criteria;

Step 5 Perform GA operations on the population selected;

Step 6 Evaluate the individual after performing the operations;

Step 7 Set $t = t+1$;

end

end

Fig. 2.1 Pseudo-code for generic genetic algorithm

However, for the conventional GAs, largely depend on the manner of crossover, mutation and the corresponding parameters. How to make parameter tuning becomes a critical task [42]. Different problems require different crossover probability. 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. Furthermore, for some complex problems, operators of crossover and mutation cannot ensure to get an optimal solution and how to deeply utilize the current promising data towards the final optimal solution is always one critical issue in the population-based optimization algorithms. Then, one probability model based algorithm without crossover was developed, called Estimation of Distribution Algorithm (EDA) [43], trying to overcome the drawback of conventional EAs.

Compared with the conventional methods, the key point of EDA is using the probabilistic model to describe the distribution of value selection for each decision variable. The probability is extracted from some promising date coming from all candidate solutions. From lots of previous literatures, EDAs can achieve better optimality on some benchmark problems, especially when the decision variables are dependent due to a high level of interaction [44].

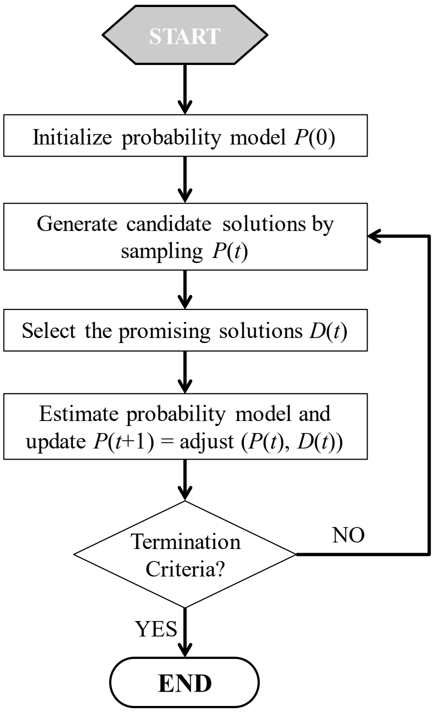


Fig. 2.2 Flowchart of generic EDA

Flowchart of generic EDA is shown in Fig. 2.2, which is used as the basic searching engine in this study.

2.1.1 Generic EDA

Same with other EAs, EDAs are also population-based approaches. The core issue of EDAs is the probability model involving. Through estimating the distribution and sampling candidate solutions, leading to further search until the termination achieved.

In the recent years, many literatures on the algorithm of EDAs have been proposed. It is impractical to give an exhaustive list of all developed EDAs. A common way to categorize EDAs is according to the variable dependency types and probabilistic models to model interdependence relationship between variables.

In Table 2.1, it lists some typical and representative EDAs.

Table 2.1 Representative EDA algorithms

Variable Dependency	Probability Model	Algorithm	
Univariate/ Independent	Marginal Distribution	PBIL	Population-based Incremental Learning
		cGA	Compact Genetic Algorithm
		UMDA	Univariate Marginal Distribution Algorithm
Bivariate/ Bi-dependent	Forest	BMDA	Bivariate Marginal Distribution Algorithm
	Copula Functions	CEDA	Copula-based EDA
	Marginal Histograms	HEDA	Histogram-based EDA
	Tree	COMIT	Combining Optimizers with Mutual Information Trees
	Chain	MIMIC	Mutual Information Maximizing Input Clustering
Multivariat/ Multi-dependent	Factor Graph	FDA	Factorized Distribution Algorithm
	Bayesian Network	BOA	Bayesian Optimization Algorithm
	Markov Network	MEDA	Markov Network based EDA
	Dependency Network	EDNA	Estimation of Dependency Network Algorithm
	Hierarchical Dependency Tree	LTGA	Linkage Tree GA
	Marginal Product Model	EcGA	Extended cGA

In the domain of EDAs, univariate EDAs were firstly developed for independent or univariate, such as PBIL [45], UMDA [46] and cGA [47]. This kind of EDAs make assumes that the joint probability for each variable is calculated as the marginal probability and each variable is independent from others. These algorithms, ignoring feature dependencies, are the simplest and fastest EDAs but still suit for some particular problems with high cardinality, meanwhile they are suitable to make theoretical analysis of EDA behavior [48].

To extend the univariate EDAs, bivariate models are evolved into the EDAs, which represent the pairwise dependencies between variables. Several kinds of probability model to address the dependence: forest and tree structure are adopted in BMDA [49] and COMIT [50], while MIMIC [51] uses the probability models of chain structure. The bivariate models are applicable to more widely problems. Compared with univariate EDAs, bivariate ones need longer calculation time.

2.1.2 Graphical models and PGM based EDA

The most popular and effective approaches are now multivariate EDAs, where the dependencies between variables are multi-dependent, and some probability graphical models (PGMs) evolved. FDA [52] is the first algorithm of multivariate EDA. Later more literatures on multivariate EDAs are published. Two important probability graphical models are adopted to enhance the EDAs: Bayesian networks and Markov networks. Based on Bayesian networks, EBNA [53] and BOA [54] are proposed and use BIC metric and BDe metric to learn the network structure respectively. Based on Markov networks, some approaches as Markov network based EDA [55] and DEUM [56] are proposed.

For multivariate EDAs or PGMs based EDA, there are different types of PGMs proposed, including Bayesian network (BN) [57], Markov network (MN) [58], dependency networks (DN) [59], chain graphs (CP) [60] and so on.

Based on conventional EDAs, PGMs can bring out high ability to solve problems widely, but need higher memory requirement and cost longer computation time. Especially for a complex problem without prior knowledge, it requires very long time to learn the structure. Consequently they are usually applied to applications where the network structure is known by experts [61].

2.1.3 Markov network based EDA (MEDA)

This subsection mainly discusses Markov network empowered EDA, which part related to our study.

a) Markov network

In Markov network, it consists of graph structure G and parameter Ψ. An example of Markov network containing five variables is shown in Fig. 2.3. In graph structure, each node represents one decision or stochastic variable and the edge represents relationship existing among these nodes based on its undirected structure. In Fig. 2.3, variable X_1 has two neighbors of variable X_2 and X_3. And the variable X_2 has three neighbors of variable X_1, X_3 and X_4.

A solution $x = (x_1, \ldots, x_n)$ containing the values of variable X, which are generated by calculating the joint distribution of decision variable $X = (X_1, \ldots, X_n)$. $D(X_i) = \{x_i^1, \ldots, x_i^{ni}\}$ represent the domain of X_i.

For each node, the conditional probability is calculated by its neighbors (which nodes have edge connecting to it). The equation of conditional probability is:

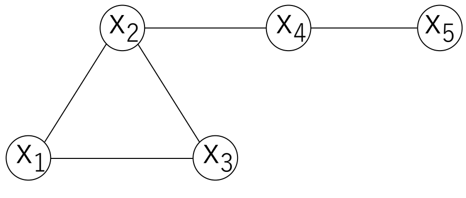


Fig. 2.3 An example of Markov network structure

$$p(x_i \mid x - \{x_i\}) = p(x_i \mid N_i) \tag{2.1}$$

where N_i is the set of neighborhood of node X_i.

In Table 2.2, it shows an example of the parameter table for factor $\{X_1, X_2, X_3\}$. It is difficult to establish the Markov properties, so that the concept of Markov random fields is adopted, which is factorized based on the cliques in network structure. The probability is calculated by

$$p(\mathrm{x}) = \frac{1}{Z}\prod_{i=1}^{m}\psi_i(c_i) \tag{2.2}$$

where m is the total amount of cliques, $\psi_i(c_i)$ is a potential function on each clique, and Z is a normalizing constant.

b) Structure learning

PGMs based on EDAs are extension to conventional EDAs. The basic evolutionary procedures are kept in remains, but there are two main differences: learning or estimating graphic structure, and using the structure to sample new candidate solutions [62].

In conventional EDA, estimation is performed and represented by a probability matrix, and each position in the matrix denotes a certain meaning with probability. However, in Markov network based EDA, this probability is represented by both probability matrix and a structure of network. The network structure represents the relationship among different variables, while

Table 2.2 Parameter table for factor $\{X_1, X_2, X_3\}$ in Fig. 2.3

X_1	1	1	1	1	1	1	2	2	2	2	2	2
X_2	1	2	1	2	1	2	1	2	1	2	1	2
X_3	1	2	3	1	2	3	1	2	3	1	2	3
$\psi(X_1,X_2,X_3)$	$\psi_{123,1}$	$\psi_{123,2}$	$\psi_{123,3}$	$\psi_{123,4}$	$\psi_{123,5}$	$\psi_{123,6}$	$\psi_{123,7}$	$\psi_{123,8}$	$\psi_{123,9}$	$\psi_{123,10}$	$\psi_{123,11}$	$\psi_{123,12}$

estimating the probability denotes how important the relationship based on the connection of network.

One way to construct the network structure is made by domain experts. However, it is hard to find experts and very time consuming. We can also perform conditional independence test to decide. Here we use mutual information (MI) [63] to estimate the structure, which can be easily adopted in low computation costs and avoid high complexity. In equation (2.3), we can calculate MI between two random variables X_i and Y_j.

$$\begin{aligned}\mathrm{MI}(X_i;Y_j)=&\sum_{x_i\in X_i}\sum_{y_j\in Y_j}\mathrm{p}(x_i,y_j\mid D)\\&\times\log\left(\frac{p(x_i,y_j\mid D)}{p(x_i\mid D)p(y_j\mid D)}\right)\end{aligned} \tag{2.3}$$

where $p(x_i|D)$ and $p(y_j|D)$ are probability of variables $X_i = x_i$ and $Y_j = y_j$ based on the promising solutions set D, $p(x_i, y_j|D)$ is the joint probability of $X_i = x_i$ and $Y_j = y_j$.

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 equation (2.4).

$$\begin{aligned}threshold&=\alpha\times avg(MI)\\&=\alpha\times\frac{2\times\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}MI(X_i;X_j)}{N\times(N-1)}\end{aligned} \tag{2.4}$$

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.

c) Sampling

New candidate solutions have to be sampled, after the structure of Markov network and parameters of probability model have been learned. Markov network is different from the ancestral ordering in Bayesian network (BN) [64]. As a result, in order to sample new solutions, one Gibbs sampler is proposed, which is one kind of Monte Carlo methods for Markov chain, to act as the sampling method. The pseudo code of Gibbs sampling is given in Fig. 2.4.

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

$$p\left(x_j \mid N_j\right) = \frac{e^{p(x_j, N_j)/T}}{\sum_{x_j' \in D(X_j)} e^{p(x_j', N_j)/T}} \tag{2.5}$$

$$T \sim \beta \times \frac{1}{gen} \tag{2.6}$$

where $p(x_j, N_j)$ represents the joint probability of a variable $X_j = x_j$ and its neighbors N_j. T is the temperature function, determined by cooling rate parameter β. Higher value of β makes the update

Gibbs sampling for Markov network based EDA		
begin		
for i := 1 to *popSize* **do**		
Step 1	Randomly generate a solution $x = (x_1, x_2, \ldots, x_n)$ 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. 2.4 Gibbs sampling for Markov network based EDA

mainly depend on old promising solutions, while smaller value represents that the present promising date affect the results a lot.

2.2 Multi-objective evolutionary algorithm (MOEA)

2.2.1 Overview of MOEA

It is always a challenging job to researchers that how to provide good solutions to the problems with multiply objectives. Evolutionary algorithms (EAs) have two important characterizes: multi-directional and population-based, which make them as suitable approaches to solve the multi-objective problems. This kind of population to population approach can search for good solutions in different regions of the searching space simultaneously, which makes it possible to find a set of good solutions, even for the non-convex or discrete problems [65].

For solving multi-objective optimization problems (MOOPs), one simple way is we can take these multiple objectives as one. For example, by using weighted average and transforming the problem as one single combined objective to optimize. Second way is that, we can propose goal programming for example, and give each objective a goal to achieve, and convert the multiply objectives to the deviation from the goal value, and try to minimize or maximize the total deviations. But all these methods still belong to single-objective methods, and can only provide one single solution [66].

However, it is not a suitable way to transform as single objective problem. Firstly, it’s still a difficult job for problem experts to decide the weighting value to characterize their own preferences, some technology such as AHP or ANP should be involved to increase the complexity of problems. Secondly, different decision makers have different preferences, so that one weighting value is not fit for other project managers. Thirdly, a set of good solutions provide more

Generic Multi-objective Evolutionary Algorithms
begin
Initialization:
Step 1 Initialize population $P(0)$;
Step 2 Evaluate objective value;
Step 3 Ranking based on Pareto Dominance;
while *terminating criteria is not met* **do**
Step 4 Select $P(t)$ from $P(t\text{-}1)$ based on Pareto Dominance;
Step 5 Do recombination and mutation to $P(t+1)$;
Step 6 Evaluate each individual in $P(t+1)$;
Step 7 Ranking $P(t+1)$ union $P(t)$, based on Pareto Dominance;
Step 8 Set $t = t+1$;
end
end

Fig. 2.5 Pseudo-code of generic MOEAs

chances to select, and easy to make trade-off among different objectives. All these reasons are the fundamental motivation of designing MOEAs.

In Fig. 2.5, it shows the Pseudo code for generic MOEAs.

For MOEAs, the key issues to make different algorithms are operators, fitness assignment mechanism, and schematic of selection and update. Based on these aspects, we list some typical MOEAs in Table 2.3. Most of them takes the GA as the optimization algorithm. However, we can use other stochastic search and optimization approaches, such as TS [67], SA [68], PSO [69] and other evolutionary algorithms [70], however, how to apply a suitable meta-heuristic solver to different types of scheduling problems is another critical problem and need to be well designed.

2.2.2 Typical MOEAs

In this subsection, we focus on several MOEAs, which related to our research. Through the explanations of three typical MOEAs, we try to make the brief understanding of evolution process of MOEAs.

Table 2.3 List of typical MOEAs and fitness

MOEAs		Fitness
VEGA	Vector Evaluated GA (1985)	Value of a single objective
WBGA	Weighted Average of Normalized Objectives (1992)	Niching, predefined weights
MOGA	Multiple Objective GA (1993)	Fonseca and Fleming's Pareto ranking
NPGA	Niched Pareto GA (1994)	Tournament
RWGA	Random Weighted Genetic Algorithm (1995)	Weighted average of normalized objectives
PAES	Pareto Archived ES (1999)	Single grid
PESA	Pareto Envelope-based Selection Algorithm (2000)	Pareto ranking
µGA	Micro GA-MOEA (2001)	Pareto ranking
NSGA	Nondominated Sorting GA (1994)	Nondominated sorting
NSGA-II	Nondominated Sorting GA-II (2002)	Nondominated sorting, crowding
SPEA	Strength Pareto Evolutionary Algorithm (1999)	Strength value
SPEA2	Strength Pareto Evolutionary Algorithm 2 (2001)	Strength value, raw fitness, density

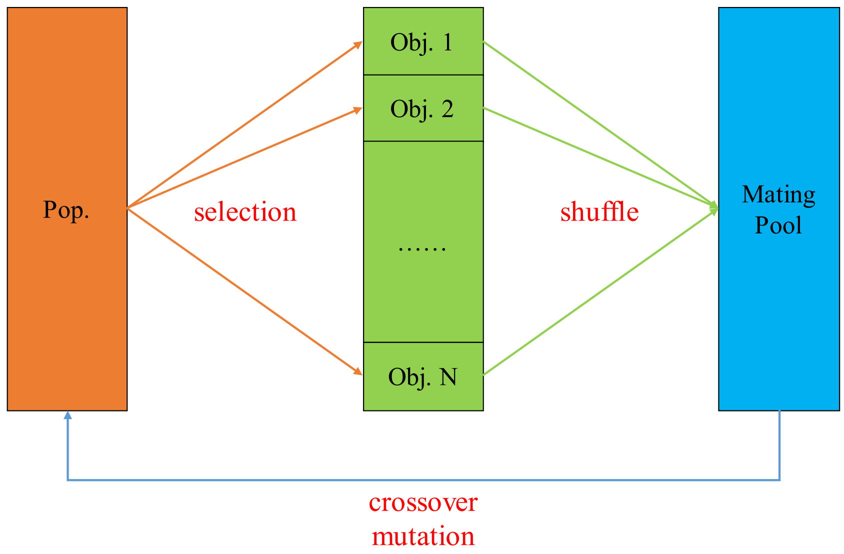


Fig. 2.6 The evolving process of VEGA

a) Vector evaluated genetic algorithm (VEGA)

Schaffer proposed first MOEA based on simple GA named vector evaluated genetic algorithm (VEGA) with vector-valued fitness measures.

In the algorithm of VEGA, k equal sized subpopulations are generated by dividing the population randomly, and each solution is evaluated only by one corresponding objective function. The major disadvantage of VEGA is poor diversity, because it attempts to find solution which is outstanding in one objective. And usually, the solutions in the central area of Pareto front are more important, because they achieve the balance among multiply objectives.

In Fig. 2.6, it shows the evolving process of VEGA. The merit of VEGA is the low complexity and towards the edge region of searching space.

b) Nondominated sorting genetic algorithm-II (NSGA-II)

The key issues of NSGA-II are nondominated ranking and crowding distance calculation. In Fig. 2.7, it shows the evolving process of NSGA-II. All individuals in population have been sorted into different ranking by non-dominated sorting. Next step is updating the new population. Starting from ranking 1, until the number of one ranking is more than the amount of left size of

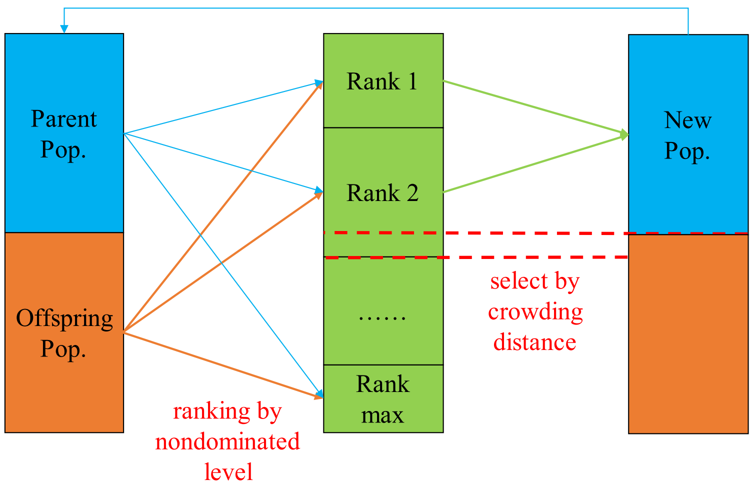


Fig. 2.7 The evolving process of NSGA-II

population. Then we calculate the crowding distance, to decide which ones update into the next generation.

c) Strength Pareto evolutionary algorithm 2 (SPEA2)

In SPEA2, three key issues have to be illustrated.

Strength(*i*): the number of individuals that individual *i* dominates.

Raw_fitness(*i*): sum of the strengths of individual *i*'s dominators.

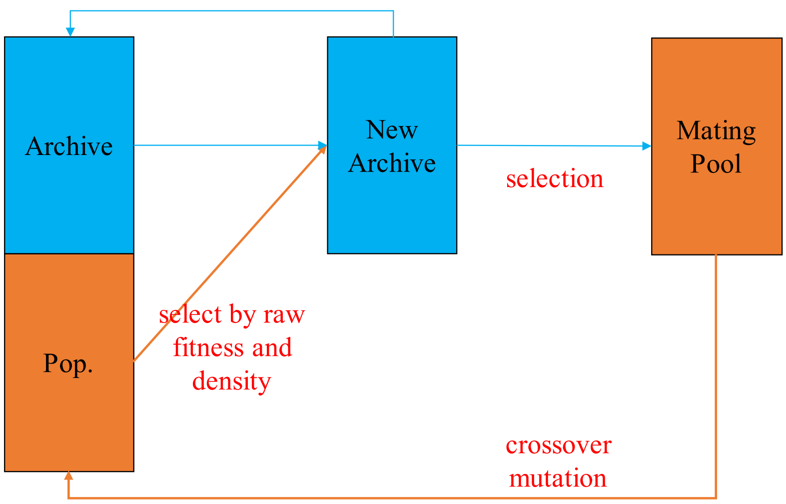


Fig. 2.8 The evolving process of SPEA2

Density(i): the k-th shortest distance of individual i to all other individuals.

From the Fig. 2.8 we can know that, an enhanced archive is added into the evolving process. The archive is updated by raw fitness and density value of each individual.

The SPEA2 and NSGA-II has been proved as two of the most outstanding MOEAs [71]. As a result, they are very convincing methods to compare, in order to demonstrate a newly designed MOEA.

2.3 Robust optimization and robust scheduling

2.3.1 Robust optimization

In order to deal with uncertainty, Robust Optimization (RO) was proposed to solve the optimization problems with some kind robustness measures [72]. At the beginning, the most famous issue of RO is worst case [73] and maximin model [74]. With the manner of worst case, it is very easy to solve but extremely conservative. The decision only focus on the worst case, to minimize the expected cost. Ignoring other conditions may occurred in future, will lost lots of information of uncertainty, so that cannot afford the uncertainty sufficiently.

Based on worst case, minimax regret approach is proposed to minimize the worst-case regret [75]. For a particular scenario, "regret" measures the difference between the averaged/expected value and the actual value gotten with that scenario [76]. The target of minimax regret is to execute as closely as possible to the optimal one. Similar with worst case, the advantage of minimax regret is independent of the probability, but still cannot estimate the expected outcomes.

Another more general and applicable scheme is made by chance constraint. For example, there is one uncertain linear constraint in equation (2.7):

$$\tilde{a}x \le \tilde{b} \tag{2.7}$$

where x are vector of decision variables, a and b are parameters with uncertainty.

We can convert the constraint as:

$$\text{Prob}(\tilde{a}x > \tilde{b}) \leq \beta \tag{2.8}$$

where β is the confidence level of constraint.

After transforming, we can obtain feasible solutions that satisfy the chance constraint, and the solutions are not as conservative as worst case.

2.3.2 Robust scheduling

To deal with medium uncertainty, a proactive or called robust scheduling is required. In recent years, various of researches on robust scheduling have been developed [77].

a) Resource-redundancy based

When there is some uncertainty on resource itself, for example, machines have the probability to breakdown, it is reasonable to prepare extra resource standby, which is called resource-redundancy [78]. The ability of fault tolerance can guarantee the overall system failure can be avoided, but the cost is very high [79]. As a result, pure resource redundancy is rather unrealistic in real-world problems.

b) Time-redundancy based

Compared with resource-redundancy, time-redundancy is a much more practical way for resource constrained scheduling problems.

One popular way is to inset additional idle time or buffer time to absorb the possible disruptions, which may come from dynamic job arrival [80] and machine breakdown [81].

Second way is slack-based approach [82]. Here we focus on typical one: total slack time. Total slack time is the difference between the possible earliest starting time of one activity and its possible latest starting time. Existing of slack time can be viewed as temporal protection for small disruptions. Actually this kind of protection not only for the certain activity, but also for every activities starting before the slack time.

c) Robust machine scheduling

In this domain, for example, in problem of FJSP, the criteria always related to makespan minimization.

Take the paper by Leon [83] for example, they defined the robustness measure of schedule is the difference between the expected makespan and actual makespan with the following equation.

$$R(S) = \alpha \times M(S) + (1-\alpha) \times (M(S) - M_0(S)) \tag{2.9}$$

where α is a weighting value between 0.0 and 1.0, $M(S)$ and $M_0(S)$ are actual makespan of schedule S and the pre-schedule makespan under deterministic manner.

Except the makespan, some papers consider the total flow time as the objective [84], and try to minimize the averaged difference between the flow time calculated by all operations choosing the shortest process time and the total flow time calculated with each scenario.

Another one belonging to robust machine scheduling is worst-case based. As explained in previous subsection, it contains minimax and minimax regret [85]. The output solutions made by worst-case are too conservative in most cases.

d) Robust project scheduling

In project environment, starting time of each activity is very important, because it related to the resource prepare. If the starting time delayed, the inventory cost will be very high. So lots of papers use the difference on starting time of activity as the objective [86].

Here we list one common objectives for generating stable robust schedules for project scheduling problems.

$$Minimize \sum_{j \in Activities} c_j \times [E(s_j) - s_j(S)] \tag{2.10}$$

where $E(s_j)$ and $s_j(S)$ represent the actual starting time of activity j, for one scenario and baseline S.

We have to mention that, four types of robust scheduling techniques cannot cover all kinds of robustness, and furthermore, some improvements based on existing measures have to be proposed, especially for RCSP with complex configurations.

Chapter 3
Multi-objective Scheduling Method based on MMEDA for RCSP

The previous chapters studied the literature on the optimization technique based on EDAs. This chapter gives a detailed description of our proposals, multi-objective Markov network based EDA (MMEDA) and hybridized GA and MMEDA (hGMEDA). In order to illustrate our approaches clearly, firstly we have to make introduction of problem of multi-objective RCSP. Secondly, some key components of multi-objective optimization related are presented, and the algorithm of MMEDA is developed. Thirdly, inspired by the cooperative co-evolutionary, hGMEDA is developed to improve calculation efficiency of MMEDA, in which a two-stage architecture based on sequential co-evolutionary paradigm is proposed.

3.1 Problem formulation of multi-objective RCSP

In chapter 1, we have introduced that RCSPs are naturally and always multi-objective problems. In this subsection, we mainly discuss the problem description of generic multi-objective RCSP, which is the target application of our proposed scheduling methods.

In RCSP, there are two main topics have to be illustrated, precedence relations and resource constraints.

(a) Precedence relations: In RCSP, it provides more complicated precedence relations than flow shop or job shop manufacturing systems. Take JSP for example, the tasks are classified into jobs and operations. In each job, there are several operations with precedence relations. But there has no special constraints among jobs. In other words, the precedence relations are given inside each job locally, not globally. Secondly, the relations between each operations are linear in FJSP. However, in RCSP, the precedence relations could be more complex. For example, in project scheduling problem, the precedence relationships among operations are in network structure.

(b) Renewable and nonrenewable resources: Usually, in the practical and complicated real-world problems, such as large building construction project, there are very various kinds of resources. Often, the utilization of resources is one of the key issue for decision maker in the budget management of one company. The renewable resources (k = 1,2,…,K^p) are available with amount of a_k^p in each time period of the whole project. For example, the availability of the machines or work force is one typical renewable resource. The non-renewable resources (k = 1,2,…,$K^ς$) are finite resources, not depend on time. For example, the total budget for one power plant project is one kind of non-renewable resource, each activity will cost some money and the amount would not renew during the whole scheduling.

Usually, makespan is taken as one optimization criterion, besides, there are several other kinds of criteria, such as net present value or cost minimization. For RCSP with manpower involved, there are so many workers with different skill levels and professions, which could be viewed as different types of resources. In the viewpoint of manufacturing planning, human resource management and profit optimization, the project manager attempts to make full use of each worker employed in this project. If the decision maker can find some resources have lower workload, they could decrease the amount of this resource. The knowledge could be used for employing workers and purchasing equipment in future.

As a result, in this study, we try to enhance the load balancing together with minimizing makespan with equations (3.1) and (3.2).

$$\min \ \left\{\max_{j=1,\ldots,N}(c_j)\right\} \tag{3.1}$$

$$\max \ \frac{1}{K} \times \sum_{k=1}^{K} \left\{ 1 - \left| UR_k - \frac{\sum_{k=1}^{K} UR_k}{K} \right| / \frac{\sum_{k=1}^{K} UR_k}{K} \right\} \tag{3.2}$$

$$UR_k = \max_{t=1,\ldots,horizon} \sum_{j=1}^{N} r_{jk} \sum_{q=t}^{t+d_j-1} x_{jq} \tag{3.3}$$

where c_j is the completion time of operation j, K is the total amount of resources, r_{jk} is the usage of resource k for operation j, and x_{jq} is the decision variable that whether operation j executed at time q.

The typical application of multi-objective RCSP: multi-mode resource constrained project scheduling problem (MRCPSP) is taken as the application in our study.

Compared with conventional RCSP, the most difference is the multi-mode configuration. In MRCPSP, similar to one operation could be performed on several candidate machines in FJSP, one activity j is processed in one of the M_j possible modes, in which defines different requirements and completion time [22]. Thus, when activity j processed in a mode m_j, it will have a duration time of d_{jm} with the requirement of r_{jm}^{r} units of the renewable resource r and r_{jm}^{n} units of the non-renewable resource n. It has assumption that all activities can not change its mode during project executing and non-preemptive.

Another reason to take MRCPSP as our application is, there are always manpower involved in the project and the budget management is very important issue compared with job shop scheduling problems or flow shop problems.

In FJSP, the flexibility is that we can use another machine to complete one operation. Similarly, in RCPSP with multi-mode configuration, we can utilize different types and different amount of resources to perform the same activity, which increase the calculation complexity a lot.

In Fig. 3.1, it shows an illustrative example of project scheduling problem, in which consists of 9 nodes, and each node represents one activity (including two dummy activities as activity S

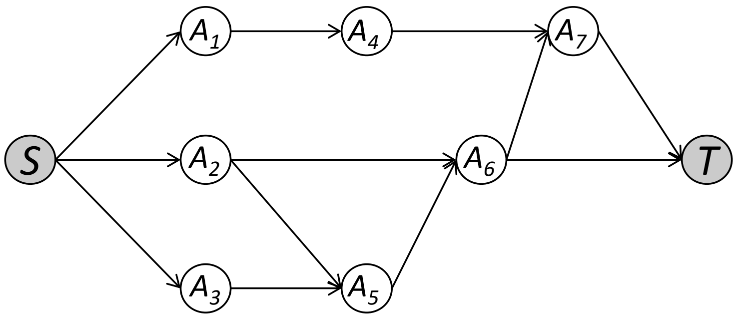


Fig. 3.1 One example of project network in RCPSP

and T). Take activity 5 (A_5) for example, the antecessors are A_2 and A_3, which means we could perform A_5 unless the A_2 and A_3 are completed. But for activity A_5 and A_1, there are no special precedence relations, without considering the limitation of resource capacity, it is possible to execute these two activates simultaneously.

For each activity, one mode must be selected from multiple mode candidates. Each mode requires different resource and durations, which are listed in Table 3.1.

MRCPSP can be solved with two decision making processes:

a) Activity sequencing (*a-seq*): to decide the sequence of activities that satisfying the constraint of precedence relationships.

Table 3.1 One example of two-mode project scheduling problem

Activity	Mode 1			Mode 2		
	$r_{j,k1,m1}$	$r_{j,k2,m1}$	d_{j1}	$r_{j,k1,m2}$	$r_{j,k2,m2}$	d_{j2}
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

$r_{j,k,m}$: requirements of activity j, which is for resource k with mode m

d_{jm}: the duration time of activity j with mode m

b) Mode selection (*m-select*): to decide the mode for each activity from the candidate modes.

Here we mainly discuss the manner of multi-mode involved and its effect to bi-objective of makespan and load balance for one schedule.

In Fig. 3.2, the activity *j* has three modes could be selected shown in Fig. 3.2(a). In Fig. 3.2(b), is shows we could use mode 2 instead of mode 1 to perform activity *j*, to decrease the makespan with higher resource usage. In Fig. 3.2(c), it is possible to use mode 3 to replace mode 1 with a different type of resource, to realize the load balance.

From the above illustrative example, in MRCPSP, the mode selection is a very important way to make resource allocation to realize makespan minimization and load balancing.

The mathematical model of MRCPSP with bi-objectives is illustrated as following:

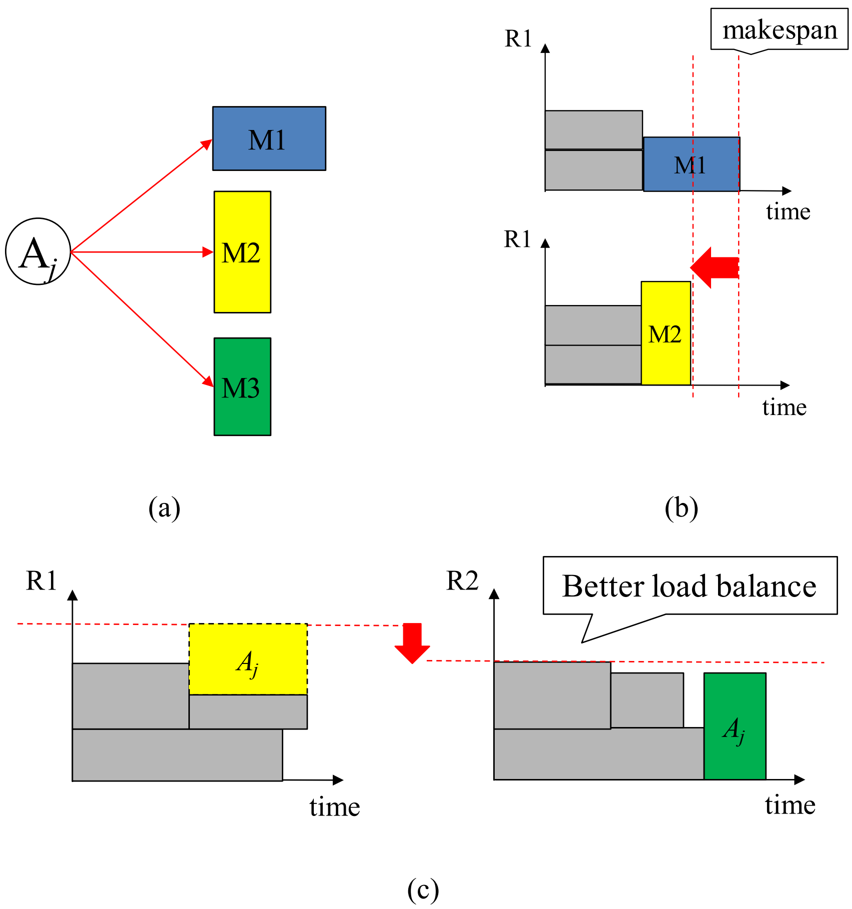


Fig. 3.2 Mode selection for makespan and load balance

- *Index:*

i activity index, $i = 1,\ldots, N$

m mode index, $m = 1,\ldots, M_j$

k resources index, $k = 1,\ldots, K$

- *Parameter:*

N total amount of activities

M_j total amount of modes of activity j

K total amount of resources

N_k capacity of resource k

d_{jm} duration time of activity j with mode m

s_j starting time of activity j

c_j completion time of activity j

r_{jkm} usage of resource k for activity j selecting mode m

p_j predecessors set of activity j

- *Decision Variables*:

$$x_{jmt} = \begin{cases} 1 & \text{activity } j \text{ is executed at time } t \text{ with mode } m; \\ 0 & \text{otherwise.} \end{cases}$$

- *Objectives*:

$$\min \ \left\{ \max_{j=1,\ldots,N} (c_j) \right\} \tag{3.4}$$

$$\max \ \frac{1}{K} \times \sum\nolimits_{k=1}^{K} \left\{ 1 - \left| UR_k - \frac{\sum\nolimits_{k=1}^{K} UR_k}{K} \right| / \frac{\sum\nolimits_{k=1}^{K} UR_k}{K} \right\} \tag{3.5}$$

- *Subject to*:

$$\sum\nolimits_{i=1}^{M_i} \sum\nolimits_{t=s_i}^{c_i} t \cdot x_{imt} \le \sum\nolimits_{j=1}^{M_j} \sum\nolimits_{t'=s_j}^{c_j} (t' - d_{jm}) \cdot x_{jmt'}, \quad j = 1,\ldots,N; i \in p_j \tag{3.6}$$

$$\sum_{m=1}^{M_j} \sum_{t=s_j}^{c_j} x_{jmt} = 1, \ \ j = 1,\ldots.,N \tag{3.7}$$

$$\sum_{j=1}^{N} \sum_{m=1}^{M_j} r_{jkm} \sum_{q=t}^{t+d_{jm}-1} x_{jmq} \le N_k, \quad j = 1,\ldots.,N; k = 1,\ldots,K; t = 1\ldots\text{horizon} \tag{3.8}$$

$$UR_k = \max_{t=1,\ldots,horizon} \sum_{j=1}^{N} \sum_{m=1}^{M_j} r_{jkm} \sum_{q=t}^{t+d_{jm}-1} x_{jmq} \tag{3.9}$$

$$\begin{aligned} & x_{jmt} \in \{0,\ 1\}, \\ & j = 1,\ldots.,N; m = 1\ldots M_j; t = 1\ldots\text{horizon} \end{aligned} \tag{3.10}$$

$$s_j \geq 0,\ c_j \geq 0, j = 1,\ldots.,N \tag{3.11}$$

Inequality (3.6) presents the constraints of precedence relation among activities. Equation (3.7) guarantees that one activity has to choose one mode to perform. Inequality (3.8) shows the capacity constraint of resources. Equation (3.9) calculates averaged utilization rate of each resource. Equation (3.10) and (3.11) represent the nonnegative restrictions.

3.2 Multi-objective Markov network based EDA (MMEDA)

In order to solve the multi-objective RCSPs, one scheduling method based on multi-objective Markov network based EDA (MMEDA) is developed.

Three key issues for the multi-objective evolutionary algorithms are meta-heuristic combinatorial solver, fitness assignment mechanism of Pareto optimization and local search, which are discussed in next subsections respectively.

3.2.1 Markov network based EDA (MEDA)

In chapter 2, we have reviewed the conventional EDAs and PGMs related. For the most conventional EDA, the relationships among variables are interdepended. However, it will lose some information during the process. In order to make the solution more convincing, some structures are added to model the relationships. In recent decades, one of the most popular way is Bayesian network (BN) based EDA [64]. As one typical probabilistic graphical model, Bayesian network could model two variables in cause-effect relationship. However, not all the problems belong to parenthood relationship, and some problems are difficult to address which kind of relationship.

Different from cause-effect relationship in Bayesian network, the relationship between two nodes in Markov network is neighborhood. In other words, Bayesian network could be viewed as one special case of Markov network, because undirected graph is two-way directed while directed graph is only one-way directed. Therefore, Markov network can be used to model the relationship among variables for widely applications.

In RCSP, for the decisions of machine assignment or resource allocation, it is very difficult to find which variable's decision would affects others, but these variables obviously have some kind of relationship due to seizing the same resource. Because of resource constraints existing, we can model the interaction among variables, and find the knowledge of this kind of relationship would lead to more convincing solutions.

Markov network based EDA, takes the assumption that the relation among decision variables are in neighborhood, not parenthood. That's the very fundamental reason of choosing Markov network as the graphic model for resource constrained scheduling problems and take it as the meta-heuristic searching engine.

Furthermore, based on conventional Markov network based EDA, we proposed the one enhanced EDA with mutation operation to avoid trapping into local optimal.

In EDAs, the solutions with better objective values are taken as promising data, by which we estimate the marginal probability distribution:

$$P_t^*(X = x) = \frac{N(X = x) + {}^{1}\!/_{|X|}}{N(D) + {}^{1}\!/_{|X|}} \tag{3.12}$$

where $N(X = x)$ represents the number of solutions with variable X choosing the value x in promising set, and $N(D)$ denotes the total number of solutions in set D.

The probability model is learned from the current promising data, but it may make the transition probability unstable, so we calculate the probability by:

$$P_t(X=x)=(1-\lambda)\times P_{t-1}(X=x)+\lambda\times P_t^*(X=x) \tag{3.13}$$

where λ is the learning rate for the current generation, specially, the distribution is completely learned from the current one if $\lambda = 1$.

For diversity, after learning the probability, a mutation operation is adopted with mutation probability p_m:

$$P_t(X=x)=\min(P_t(X=x)+\theta,1-\varepsilon) \tag{3.14}$$

where θ denotes the mutation shift value, ε is a very small positive number to keep the value of probability always smaller than 100%.

For activity sequencing, we employ the conventional EDA with the assumption that all the decision variables are independent. From the previous literatures, they demonstrated that for sequencing problems, the knowledge can be extracted is very few, so that conventional EDA is better than Markov network based EDA on calculation speed.

We adopt the probability model $P_{seq}(t)$ which is used to estimate the marginal probability that activity's priority or degree of importance in sequence in generation t. The priority is represented as the probability of the activity j scheduled before or at lth position in the activities sequence. This kind of probability mode can increase the stability of updating and is widely used in recent years [87]. The probability matrix is:

$$\mathrm{P}_{seq}(t)=\begin{bmatrix} p_{11} & \cdots\cdots & p_{1J} \\ \cdots\cdots & p_{lj} & \cdots\cdots \\ p_{J1} & \cdots\cdots & p_{JJ} \end{bmatrix} \tag{3.15}$$

where p_{lj} in matrix means the priority value of activity j for the position l in activities sequence. Initially we set each value as $1/J$.

Furthermore, the initial population of activity sequence is repeatedly applying the following steps to generate a feasible sequence: selecting next activity randomly from the set of activities whose predecessors have already been picked up.

For resource allocation, from previous literatures, finding the interrelation among decision variables would lead to more convincing solutions, and that's the reason we adopt the MEDA.

In the MRCPSP, this step is to decide the activity mode. The decisions of the mode selection for each activity have interdependence relation due to seizing the same resources. Because of some activities seizing the same resource, under the resource capacity constraints, if one mode is selected for activity j and there is high possibility that other strong related activities will select the mode with different resource requirement. Finding this kind of interrelation among different activities' mode selection will lead to convincing solutions.

Markov network based EDA is adopted to find and model the interrelation of resource allocation for which activities seizing the same resource. In Markov network, it consists of both structure of Markov network and probability parameters. Similar to the structure shown in Fig. 2.3, 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, which is calculated by mutual information. After the structure of Markov network is generated, how to select the mode for activity j is determined by the states of the nodes connecting to it or called its neighbors.

Estimation and sampling methods for Markov network based EDA have been illustrated in previous chapter. The parameters are represented by probability matrix of marginal probability.

$$\mathrm{P}_{\mathrm{mod}}(t) = \begin{bmatrix} p_{11} & \cdots\cdots & p_{1J} \\ \cdots\cdots & p_{ij} & \cdots\cdots \\ p_{J1} & \cdots\cdots & p_{JJ} \end{bmatrix} \tag{3.16}$$

where p_{ij} represents the marginal probability of activity j choosing mode i.

The conditional probability and mutual information are both calculated based on marginal probability in matrix $P_{mod}(t)$.

3.2.2 Fitness assignment function

In the research domain of multi-objective optimization, the most important issue is fitness assignment mechanism. It has been illustrated that fitness functions scheme is the main difference between various MOEAs [36].

In the algorithm of VEGA [89], k equal sized subpopulations are generated by dividing the population randomly, and the solution is evaluated only by one objective function.

$$EdgeFitness_i(X) = obj\_value_i(X) \tag{3.17}$$

As shown in Fig. 2.6 and Fig. 3.3, the major disadvantage of VEGA is poor diversity, because it attempt to find solutions outstanding only in one objective. In other words, it prefers the edge area than central area of Pareto front (the red circle area). For diversity performance, in order to cover the central region (the green circle area), a new fitness assignment function is proposed:

$$CentralFitness(X) = \frac{1}{1/(p(X)+1)+q(X)} \tag{3.18}$$

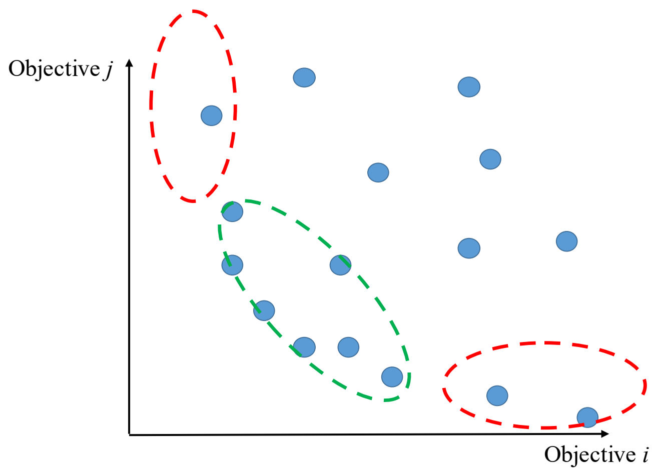


Fig. 3.3 Edge region and central region of Pareto front (Minimization problem)

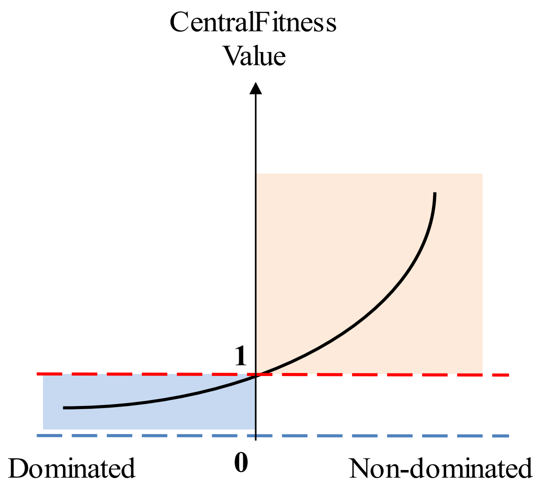


Fig. 3.4 CentralFitness value

where $p(X)$ and $q(X)$ are the number of individuals which are dominated by and dominate individual X.

For equation (3.18), as shown in Fig. 3.4, the fitness values of dominated ones are always smaller than 1 but larger than 0. The fitness value of individual is 1, if it is not dominated by any others and does not dominate others. With the fitness value larger than 1, the individual is non-dominated one, and the more individuals it dominates, the larger fitness value is. As a result, the fitness value is larger or equal than 1 denotes the individual is non-dominated, by which we can separate dominated and non-dominated ones.

In general cases, the nodes in central region can dominate more nodes than nodes in edge region, so that the CentralFitness prefers the central region. Meanwhile, similar to the sampling strategy from VEGA, the time complexity of calculating CentralFitness is very small.

One way to handle two fitness assignment functions is multi-island parallel optimization, which means that for each sub-population, we use one optimizer to train each. However, it will cost a lot of calculation time, especially for the PGM based EDA.

As a result, in order to keep the enough information given from each fitness while decrease the number of meta-heuristics optimizers, in this study, we use one exponential function to combine two sampling strategies.

More specifically, for a problem with *m* objectives, it is naturally to divide the population into *m*+1 subpopulations, and each part adopts one sampling strategy. However, simple partition like VEGA is not an appropriate way. Firstly, we need extra structure of subpopulations, and how to handle the solutions from different subpopulations is a difficult task. Secondly, because the population size is with limited, if there are so many solutions of Pareto front belong to one certain region, the size of this subpopulation is not enough. In other words, we may lose information of promising data or Pareto set due to fixed sized subpopulations. Thirdly, if there are so few Pareto optimal solutions coming from one part, in order to increase the searching performance, we have to make greater effort for this part by allowing more individuals belonging to this part into mating pool.

In the point system of decathlon in sports, the scoring is computed by the performance on each event by athletes and the event-dependent parameters listed in scoring table [90]. It can combine all the events results and finally give one score fairly with equation (3.19).

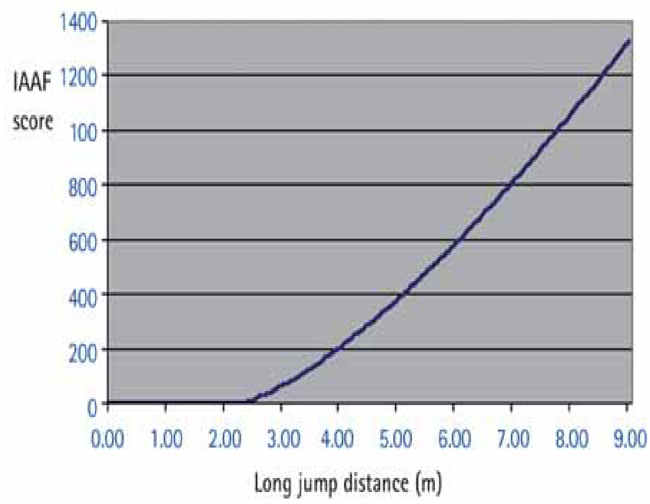


Fig. 3.5 Long jump scoring in decathlon system

$$Score = A \times (P - B)^C \tag{3.19}$$

where A is normalization parameter, P is performance, B is threshold, and C is exponential parameter to determine the performances are rated through a slightly progressive curve.

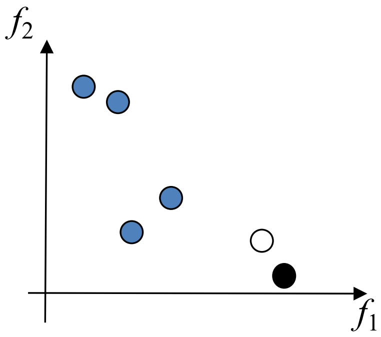
Especially, in order to distinguish the difference between improvement at low performance levels and high levels, the exponential parameter was proposed to determine the performances are rated through a slightly progressive curve. In Fig. 3.5, it shows one example of long jump scoring. For example, one player wants to improve his own performance from 4.00m to 5.00m, while another one tries to increase from 7.00m to 8.00m. The increments are same as 1.00 meter, but the difficulties are totally different, from 7m to 8m is much more difficult than from 4m to 5m. So that the scoring increments given to them should be also different.

For the traditional normalization methods, they try to normalize the differences of scale size. For our proposed normalization method, similar to idea of decathlon scoring system, we not only to normalize the differences of scale size, but also to normalize the differences of increasing rate of scale.

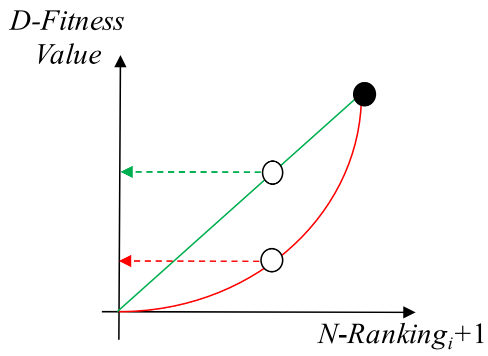
As a result, inspired by the idea of point system of decathlon, we design a novel fitness assignment function to combine different sampling strategies:

$$\begin{aligned} D\text{-}Fitness(X) = &\textstyle\sum_{i=1}^{m} (N - Ranking_{EdgeFitness_i}(X) + 1)^{\omega_i} \\ &+ (N - Ranking_{CentralFitness}(X) + 1)^{\omega_{m+1}} \end{aligned} \tag{3.20}$$

where m is the total number of objectives, N is the population size, $Ranking_{EdgeFitnessi}(X)$ 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 parameters with two purposes: expanding the difference of fitness values based on ranking number, and controlling the contribution to realize dynamic adjustment.



(a)



(b)

Fig. 3.6 Linear and exponential parameters

As shown in Fig. 3.6, it shows one illustrated example of fitness value by linear and exponential parameters. In Fig. 3.4(a), there are two nodes of black and white color, obviously the black one is preferred due to it's outstanding in objective f_2 (for minimal problem), while the white one is normal good. In Fig. 3.4(b), if we use the conventional way of linear function to combine them, the difference between fitness values of black one and while one is smaller than the exponential one.

Therefore, based on fitness assignment function of D-Fitness, we can realize that:

a) A solution is good, if and only if this solution is outstanding by one sampling strategy, which is highly controlled by exponential parameters ω_i;

b) We take the ranking by each sampling strategy to evaluate the final fitness value, which can overcome the different scale problem on original objective values or raw fitness values;

c) In order to keep the searching flexibility and diversity, the fitness value could be easily changed by parameter ω_i, which means that we can dynamically decide the contribution of different sampling strategies.

3.2.3 Local search

After solutions are sampled by Markov network and probability model of EDA, a problem-specific local search is proposed to improve the quality for each candidate solution [91]. Variable neighborhood search (VNS) is one popular way to do a possibly randomized local search [92]. In this study, we adopt the scheme of VNS, including two types of local search for makespan and load balancing.

If the critical path is kept, the makespan cannot be shorten. As a result, we try to make a new schedule with smaller makespan by breaking the existing critical path. Different to JSP or FJSP, in project scheduling problem, it has a high probability that there existing several different critical paths on different resources. Here we randomly select only one critical path among all the critical paths, to reduce the computation cost.

In MRCPSP, we decide both activity sequence and mode selection. In the local search, we can also change the mode for several activities. As a result, we have two types of local search with different target, one is for makespan by moving activity and second one is for load balancing by changing modes.

Local search of sequence changing for makespan

begin

- Step L1-1 Identify a critical path P for a given solution S;
- Step L1-2 Set q as the first activity in path P;
- **Repeat**
  - Step L1-3 Delete q from Gantt chart;
  - Step L1-4 Searching assignable time intervals for q;
  - Step L1-5 If there is no assignment time interval, set q as the next activity in path P. Otherwise, insert q into the earliest assignable time interval with probability 50%, else insert it randomly;
- **until** (q is the last activity in path P and no assignment time interval found, take S as local optimal;)

end

Fig. 3.7 Local search of sequence changing for makespan

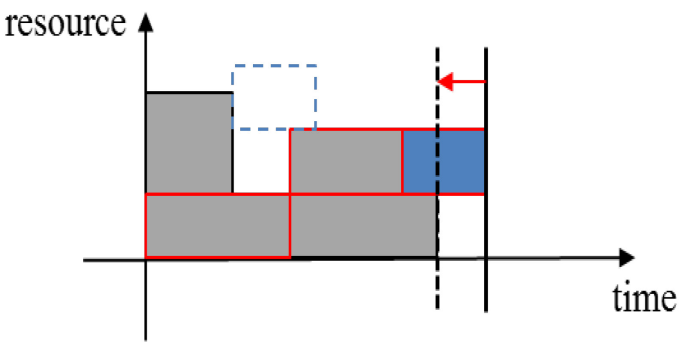


Fig. 3.8 Example of sequence changing

a) Local search for makespan: Fig. 3.7 shows the pseudo code of local search of sequence changing to reduce makespan. The purpose of sequence changing is to move one activity to another assignable position based on existing position of all other activities. For project scheduling problem, a new feasible position should satisfy all kinds of resource and without any precedence constraint violations. Since new schedule is obtained by deleting one activity and moving it to another position, it is obviously that the new makespan is no larger than original ones.

In Fig. 3.8, it shows one example of sequence changing on critical path to decrease makespan.

b) Local search for balancing: Fig. 3.9 shows the pseudo code of local search of mode changing for balancing. In this local search progress, ignoring the makespan, we only focus on

---

Local search of mode changing for balancing

---

begin

    Step L2-1   For a given solution *S*;

    Repeat

        Step L2-2   Randomly select *k* activities and choose the activity *q* with the highest load resource requirement among them;

        Step L2-3   Delete *q* from Gantt chart;

        Step L2-4   Select another mode of *q* based on load of resource;

        Step L2-5   Insert *q* into the schedule with the new mode selection;

    until   Number of iterations is achieved;

end

Fig. 3.9 Local search of mode changing for balancing

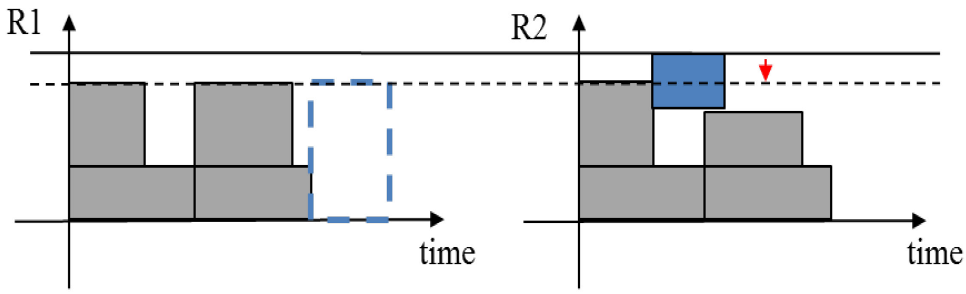


Fig. 3.10 Example of mode changing

resource allocation. In our application, better balancing requires smaller difference of utilization of various resources. Changing the sequence of activity is useless for resource balancing, while the mode changeover is required.

In Fig. 3.10, it shows one example of mode changing to improve load balance.

Local search for makespan can be only conducted on the activities in the critical paths. However, in local search for balancing, every activity could affect the load balancing by changing its mode. One simple way is exhaustive approach, and we check every activity, but the computation cost is very high and becomes impractical for large problems. It is reasonably to check certain number of activities. In order to avoid too greedy searching, we take *k*-tournament strategy. For a given solution, *k* activities are randomly picked up, and the one which requires the highest load resource is selected. We change the mode of that activity to a new mode. Two kinds of mode can be changeover to enhance the load balancing. One is the mode with less requirements on the same resources, another one is the mode without requirement the same resources. In this study, if two kinds of modes existing at the same time, we choose the mode with different type resources to changeover.

3.2.4 Algorithm of MMEDA

In this subsection, two main topics are discussed. First one is how to integrate the new fitness assignment function D-Fitness into the evolutionary process of Markov Network based EDA. Second one is the evolving process of multi-objective optimization.

In Fig. 3.11, it shows the flow chart of Markov Network based EDA. Different with the conventional EDA, there is one more network structure of Markov network involved. During the evolution process, the structure need to be learned and by using that to sample the new candidate solutions.

For multi-objective Markov Network based EDA, we could have two types of manner to integrate fitness assignment mechanism. One way is taking the fitness assignment functions inside the evaluation of EDA. The output solutions by EDA have already ranking by multi-objective functions, in the later evolving process, only update for Pareto set is needed. Another way is taking EDA as only searching engine, after the solutions given by EDA, we use the functions to rank, and then update.

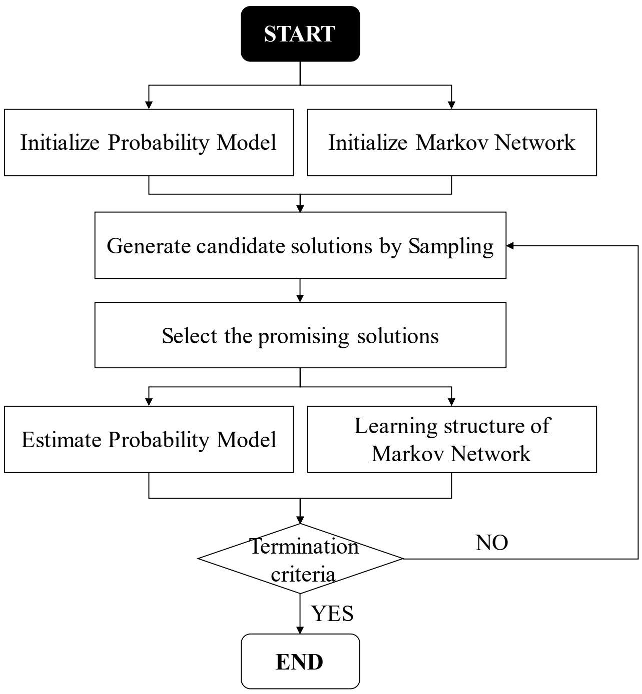


Fig. 3.11 Flow chart of Markov Network based EDA

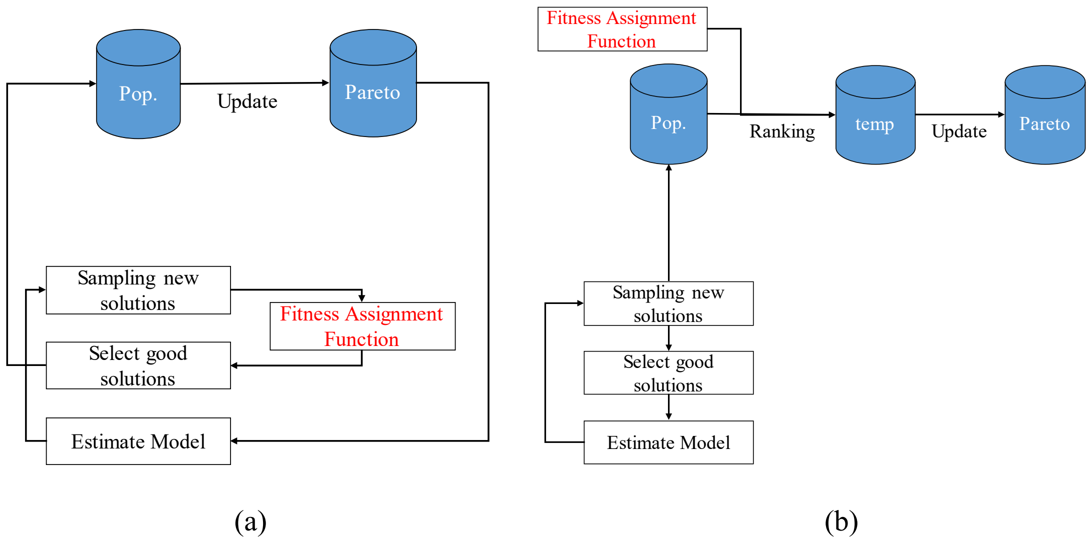


Fig. 3.12 Integrating fitness assignment functions with Markov network based EDA

Fig. 3.12(a) represents the outline of the first way, and Fig. 3.12(b) represents independent way. The second way similar with the idea of VEGA, based on the searching ability of GA, by ranking the population from GA on each objective value, some candidate Pareto solutions are generated. In (a), we only have to evaluate one time to decide which are good solutions. However, in (b), we have two times of evaluation: selecting which are good solutions and ranking which are candidate Pareto solutions. As a result, in our proposal MMEDA, we take the manner (a), aiming to decrease the calculation time.

Fig. 3.13 shows the evolving process of multi-objective optimization of MMEDA. Here we focus on the updating searching space, others will be discussed in next section with the application of RCPSP.

With the fitness values by D-Fitness, we sort all individuals. For the elitist sampling strategy, the best individuals in $P'(t)$ are updated into new archive $A(t+1)$ by replacing worst individuals in archive $A(t)$. In this study, we do not simply select the best individuals from the joint set of $P'(t)$ and $A(t)$, but select the best Q individuals (in this research, we take $Q = 0.3*|A(t)|$) from $P'(t)$, and replace the worst Q individuals in $A(t)$. On the one hand, it makes the convergence smooth. If we update the new archive by the best from joint set $P'(t)$ and $A(t)$ directly, sometimes the

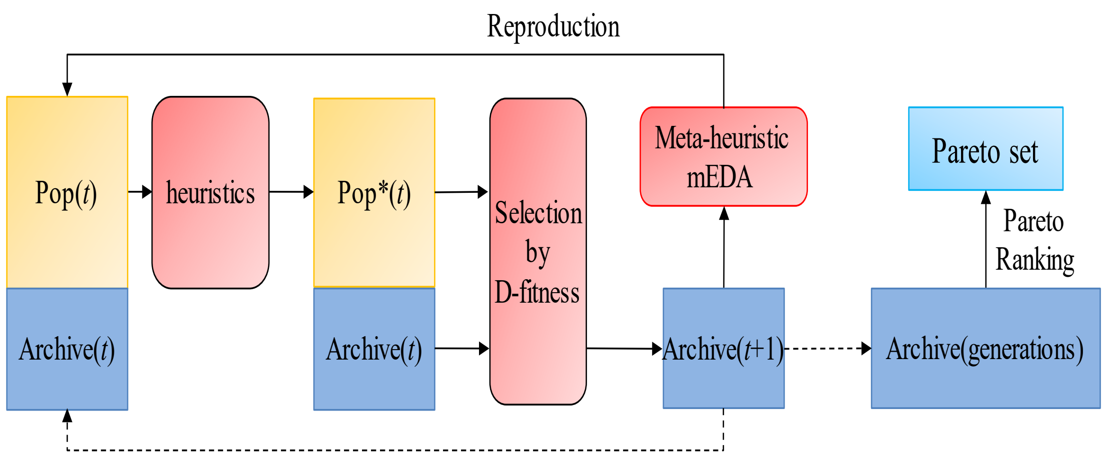


Fig. 3.13 The evolving process of multi-objective optimization

individuals in archive change so much and so does the probability model. On the other hand, it can avoid premature of EDA. When all the individuals in *P'*(*t*) is worse than *A*(*t*), in order to keep the diversity, we still take some individuals of *P'*(*t*) into *A*(*t*+1).

Furthermore, we provide a scheme of dynamic adjustment based on D-Fitness by tracking the candidate solutions. In each generation, if we find too few individuals (for bi-objective problems, the percentage of threshold is set as 0.15) belonging to one part are selected into the archive, the parameter ω_i corresponding to that part will increase in next generation. By which, we can increase the opportunity to be selected into archive for the individuals belonging to the "weak" division. Through increasing the chance of non-dominated solutions appearance in weak area, and distribution performance could be enhanced. The dynamic adjustment of D-Fitness is inspired by the point system of decathlon, in where the scoring becomes higher when the improvement on the performance is difficult. Similarity, for D-Fitness, the value becomes higher when the improvement on that sampling strategy becomes difficult. As we said, the normalization of D-Fitness not only for the differences of scale size, but also for the differences of increasing rate of scale.

As shown in Fig. 3.14, MMEDA firstly generates the solutions randomly and probability models are initialized. Good solutions are selected by combined fitness assignment function D-

Multi-objective Markov network based EDA	
begin	
***Initialization*:**	
Step 1	Initialize Markov network and probability model $P(0)$ of EDA;
Step 2	Initialize the population $Pop(0)$ randomly;
Step 3	Find promising set $D(0)$ by fitness assignment function and update $Archive(0)$;
***Optimization*:**	
while *terminating criteria not achieved* **do**	
Step 4	Estimate the structure of Markov Network based on $Archive(t\text{-}1)$;
Step 5	Estimate Markov conditional probability $p(x_{ik}\|N_{ik})$ for each variable X_i, and sample candidates solutions by Gibbs sampling;
Step 6	Update probability model $P(t)$ of EDA, perform mutation operation, and sample solutions based on $P(t)$;
Step 7	Perform a problem-specific local search;
Step 8	Calculate fitness value by fitness assignment function and update $Archive(t)$ with the best solutions;
end	
end	

Fig. 3.14 Pseudo code of algorithm of MMEDA

Fitness from the population. The structure of Markov network and its corresponding parameters are estimated by the promising date. Next, the conditional probabilities are learned. The new candidate solutions are sampled by the Gibbs sampling method based on Markov network structure and the probability parameters. Then, for each solution, two kinds of local search are applied to improve the quality. Finally, the new solutions with high fitness values are updated into the archive. The iteration will not stop unless the termination criteria are achieved.

3.3 Two-stage architecture hybrid GA and MMEDA (hGMEDA)

Furthermore, in order to improve the calculation efficiency of proposed MMEDA, two-stage architecture of hybridizing GA and MMEDA (hGMEDA) is developed.

The decision processes can be divided into two parts in RCSP: sequencing and resource allocation. We have already generated two parts by its own probability model and combine them together into one evolving process [93]. However, similar to some meta-heuristic method, with the searching space dimension increased, the searching performance would decrease a lot due to the curse of dimensionality. One popular way to overcome the disadvantages caused by high dimension complex problems is the cooperative co-evolutionary paradigm [94]. In co-evolutionary algorithm, the basic idea is to split the solution containing all of decision variables into many subcomponents. Each subcomponent is represented by a corresponding model and each model evolves sequentially or concurrently. The fitness function is evaluated by combining all the subcomponents together.

For RCSPs, the decision variables are naturally divided into two groups, so that we can take them as subcomponents. The sub-problem of sequencing is much easier than resource allocation, no matter on number of decision variables or the dependence relationships among them. So that the sequencing problem can be solved by GA with short time. The resource allocation problem requires Markov network leading to more convincing solutions. As a result, inspired by the idea of cooperative co-evolutionary, we proposed a two-stage architecture hybrid GA and MMEDA (hGMEDA) for solving RCSP.

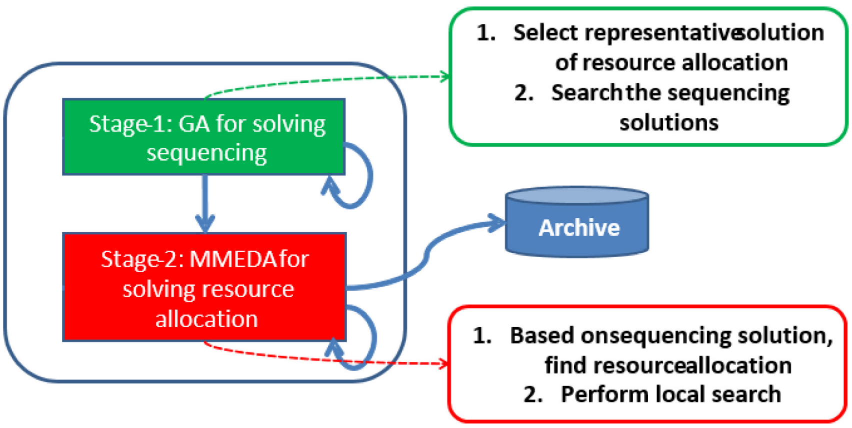


Fig. 3.15 Outline of two-stage algorithm hybrid GA and MMEDA

In Fig. 3.15, it shows the outline of two-stage algorithm hybrid GA and MMEDA. In stage-1, we solve the sequencing problem. Based on the selected representative resource allocation solutions, we search the sequencing solutions by GA. In stage-2, based on the results of stage-1, MMEDA are adopted and to find the relationship among variables leading to convincing solutions. All promising solutions are kept in the archive, and the iteration will go on until the predefined termination criteria are met.

3.3.1 Cooperative co-evolutionary

Cooperative co-evolutionary paradigm was developed to overcome the disadvantages caused by high dimension complex problems. As shown in Fig. 3.16, in cooperative co-evolution, first step is to split all decision variables into many subcomponents, which is called species. Each subcomponent is represented by a corresponding model in the manner of sequentially or concurrently. Different species can have different probability model. The fitness value is

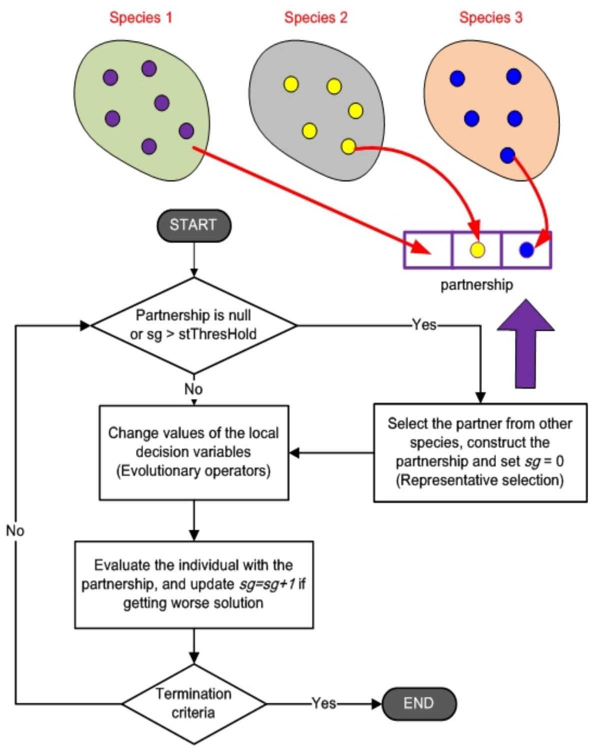


Fig. 3.16 Outline of cooperative co-evolutionary paradigm

evaluated by combining the representative solutions (or called partnership) from other species [95].

This approach can significantly reduce the complexity on exploiting the search space, so that it can decrease the calculation time a lot and increase the searching performance [1]. However, the cooperative co-evolutionary still has its own disadvantage. For complex problems, it is hard to divide the solution into small sub-problems without considering the characteristics of the problem. Fortunately, for our target RCSP, it consists of multiple sub-problems: sequencing and resource allocation. That's the reason why we propose two-stage architecture in cooperative co-evolutionary manner.

3.3.2 Stage-1 GA

In Fig. 3.17, it shows the pseudo code of GA for solving sequencing.

In this stage, the target is to find some candidate solutions of sequencing without resource capacitated, which will be used for next stage.

Inspired by the idea of cooperative co-evolutionary paradigm with sequential evolving process, in first three steps, we firstly try to find the representative solution of resource allocation (create partnership in co-evolutionary). In step 1-1, we randomly generate solutions of sequencing and resource allocation initially. In step 1-2, we evaluate the resource allocation solutions generated in step 1-1, by combining every solutions of sequencing. We evaluate the candidate solutions with the objective of makespan minimization. In step 1-3, in order to avoid too greedy search, we select top solutions and randomly select one of them to act as representative solution of resource allocation.

Next steps are GA-based sequencing searching processes. Based on the representative solution of resource allocation given from step 1-3, with the same problem setting and objective, we search the optimal solutions of sequencing based on GA optimization.

Stage-1: GA for sequencing

Problem setting

Objective: Minimize makespan

begin

__Initialization__:

Step 1-1 Randomly generate solutions of sequencing and resource allocation;

Step 1-2 Evaluate each resource allocation solution, by combining every solutions of sequencing, on the objective of minimizing makespan;

Step 1-3 Randomly selected one of the top $T1$ resource allocation solutions with highest averaged fitness values;

GA:

while *terminating criteria not achieved* **do**

Step 1-4 Based on the selected resource allocation solution, search the sequencing solutions;

Step 1-5 Perform GA operations;

Step 1-6 Evaluate candidate solutions with the objective;

end

Step 1-7 Top $T2$ sequencing solutions are selected;

end

Fig. 3.17 GA for sequencing

Finally, in step 1-7, similar to step 1-3, in order to avoid too greedy search, we select the top $T2$ sequencing solutions for stage 2.

Instead of EDA employment in MMEDA, here GA is adopted, because a) the sequencing problem is not so complex, compared with resource allocation problem. The searching speed of GA is better than PSO, 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. 3.18, 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 execute before the smaller one.

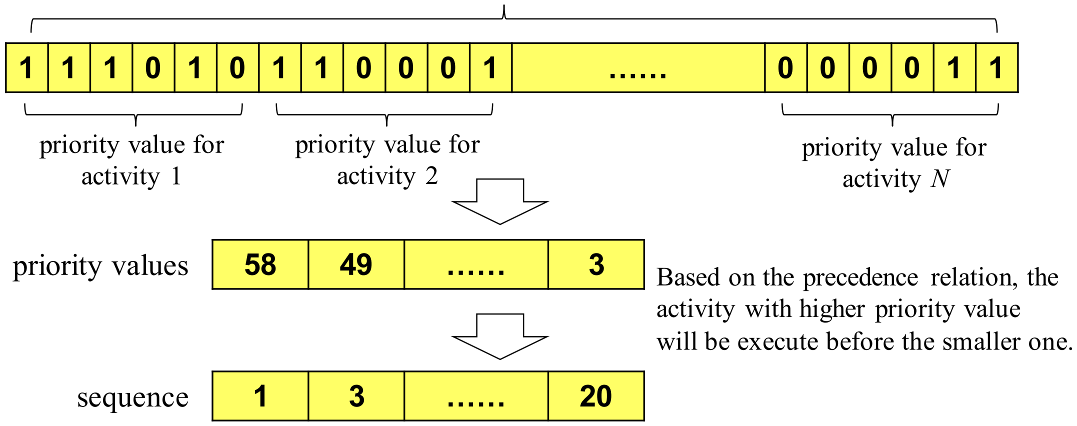


Fig. 3.18 GA representation of activity sequence

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.

3.3.3 Stage-2 MMEDA

The pseudo code of MMEDA in stage 2 for solving resource allocation is listed in Fig. 3.19

In this stage, based on the solutions given by the stage-1, firstly we initially generate some solutions of resource allocation randomly, and evaluate by combining sub-solution given by stage-1, based on bi-objective with D-fitness.

In next steps, we apply the MMEDA to search the resource allocation solutions with multi-objectives. With the help of MMEDA, we can get some candidate solutions. Then the problem-specific local search and evaluation are performed, some promising data are generated and updated into the archive.

The illustrated procedure of two stages are illustrated in Fig. 3.20.

Stage-2: MMEDA for resource allocation

Problem setting

Objective: Minimize makespan
Maximize load balancing

begin

Initialization:

Step 2-1 Randomly select one of the top *T*2 sequencing solutions;

Step 2-2 Randomly generate solutions of resource allocation;

Step 2-3 Evaluate resource allocation by combining with the selected sequencing solution, based on bi-objectives with D-Fitness;

MMEDA Optimization:

while *terminating criteria not achieved* **do**

Step 2-4 Do problem-specific local search;

Step 2-5 Find top *M* solutions as promising set to make Markov network structure and parameters learning;

Step 2-6 Sampling candidate solutions by Gibbs sampler;

Step 2-7 Evaluate the candidate solutions, on bi-objectives by D-fitness;

end

Step 2-8 Update the archive with the new promising solutions;

end

Fig. 3.19 MMEDA for resource allocation

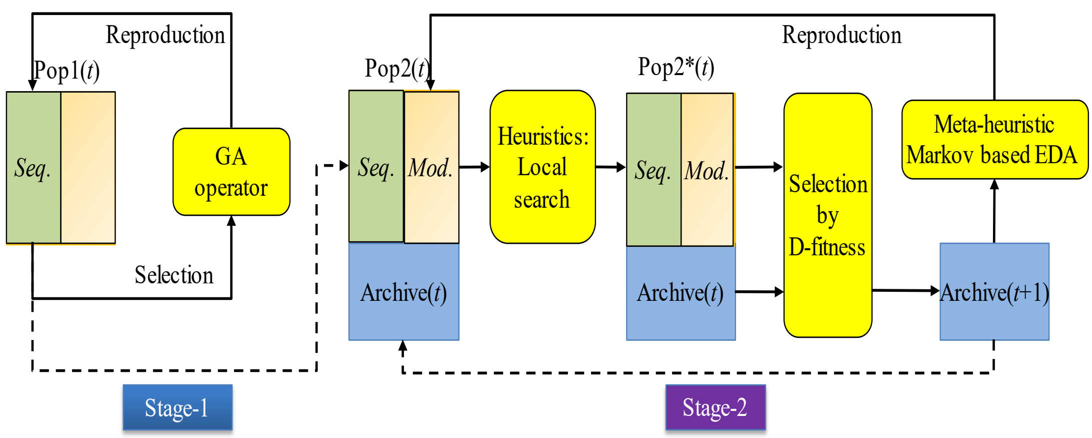


Fig. 3.20 The evolving process of two-stage hGMEDA

3.3.4 Algorithm of hGMEDA

The main difference between MMEDA and hGMEDA is the two-stage solving procedure. In other words, hGMEDA is one approach to enhance MMEDA to improve the calculation efficiency, due to cooperative co-evolutionary manner and GA replacing EDA to solve sequencing problem. The pseudo code of algorithm of hGMEDA is shown in Fig. 3.21.

Two-stage hybridizing GA and MMEDA

begin

    ***Initialization*:**

        Step 1 Initialize the population *Pop*1(0) for sequencing randomly;

        Step 2 Initialize the population *Pop*2(0) for resource allocation;

        Step 3 Initialize Markov network and probability model *P*2(0) for MMEDA;

        Step 4 Initialize *Archive*(0);

    ***Stage-1*:**

    while *terminating criteria not achieved* **do**

        Step 5-1 Generate representative solutions *Pop*2(*t*) from mode selection every *X* generations;

        Step 5-2 Combined with *Pop*2(*t*), and evaluate the fitness values;

        Step 5-3 Perform crossover and mutation to generate new solutions *Pop*1(*t*);

    end

    ***Stage-2*:**

    while *terminating criteria not achieved* **do**

        Step 6-1 Perform a problem-specific local search;

        Step 6-2 Calculate fitness value by fitness assignment function combing with solution gotten from stage-1, and update *Archive*(*t*);

        Step 6-3 Update probability model *P2*(*t*) of MMEDA;

        Step 6-4 Estimate the structure of Markov Network;

        Step 6-5 Sample solutions based on *P2*(*t*) and Markov network structure;

    end

end

Fig. 3.21 Pseudo code of algorithm of hGMEDA

Generally, to combine two stage of GA for sequencing and MMEDA for resource allocation, one cooperative co-evolutionary paradigm with sequential evolving process is adopted. After MMEDA, some complete candidate solutions are generated.

Chapter 4
Experimental Evaluation on Resource Constrained Project Scheduling

4.1 Introduction

In this chapter, the typical application of multi-objective RCSP: multi-mode resource constrained project scheduling problem (MRCPSP) is taken as the study case and used for illustrative the performance of our proposed algorithm MMEDA and hGMEDA, with the comparisons on the optimality and the distribution performance.

It is extremely hard to find real data for application of MRCPSP. Fortunately, there has been a significant amount of research conducted on the project scheduling problem and one popular benchmark problem data set PSPLIB [96] can be used to compare different methods. In PSPLIB, the duration of each activity is integer, which is a well design for conventional optimization problems. However, for multi-objective problems with Pareto set, the possible values of makespan in front set are very few. In order to make the results more convincing to compare, we randomly add 0.0 ~ 0.9 to duration of each activity, which will not break the structure of benchmarks.

Here we make an explanation of the benchmark problem briefly. Take the benchmark problem #n041_1 for instance, in Fig. 4.1, it is one problem in multi-mode data sets of PSPLIB. There are totally 22 activities (including 2 dummy activities) and 2 kinds of renewable resources. Three modes could be chosen for each activity, and for each mode, one corresponding duration and resource requirements are assigned. The network structure of project scheduling problem is

```
************************************************************************
file with basedata            : me41_.bas
initial value random generator: 26075
************************************************************************
projects                      :  1
jobs (incl. supersource/sink ):  22
horizon                       :  180
RESOURCES
  - renewable                 :  2   R
  - nonrenewable              :  0   N
  - doubly constrained        :  0   D
************************************************************************
PROJECT INFORMATION:
pronr.  #jobs rel.date duedate tardcost  MPM-Time
    1     20      0       23        8       23
************************************************************************
PRECEDENCE RELATIONS:
jobnr.    #modes  #successors   successors
   1        1          3           2   3   4
   2        3          3           6   7  14
   3        3          1          20
   4        3          3           5  10  13
   5        3          3           8   9  16
   6        3          3          12  13  17
   7        3          1           9
   8        3          2          11  15
   9        3          3          11  17  21
  10        3          2          16  20
  11        3          1          18
  12        3          2          15  18
  13        3          2          15  19
  14        3          3          16  17  21
  15        3          1          21
  16        3          1          19
  17        3          1          18
  18        3          2          19  20
  19        3          1          22
  20        3          1          22
  21        3          1          22
  22        1          0
************************************************************************
```

(a)

```
************************************************************************
REQUESTS/DURATIONS:
jobnr. mode duration  R 1  R 2
------------------------------------------------------------------------
  1      1     0       0    0
  2      1     2       4    0
         2     4       0    7
         3     7       0    3
  3      1     5       0    9
         2     5       5    0
         3     8       4    0
  4      1     3       4    0
         2     8       2    0
         3    10       0    2
  5      1     6       0    2
         2    10       1    0
         3    10       0    1
  6      1     2       0    3
         2     7       0    2
         3     9       6    0
  7      1     2       5    0
         2     6       0    9
         3    10       0    8
  8      1     4       0    9
         2     5       0    7
         3     7       0    6
  9      1     1       0    2
         2     5       9    0
         3     8       0    1
 10      1     3       0   10
         2     6       7    0
         3    10       0    9
```

(b)

Fig. 4.1 The text file of problem #n041_1 in date set PSPLIB (part)

represented as the successors of each activity. The network structure of problem #n041_1 is shown in Fig. 4.2.

The other benchmark problems adopted in our experiments all belong to multi-mode data sets, but with different structure of project network, different duration and resource requirements.

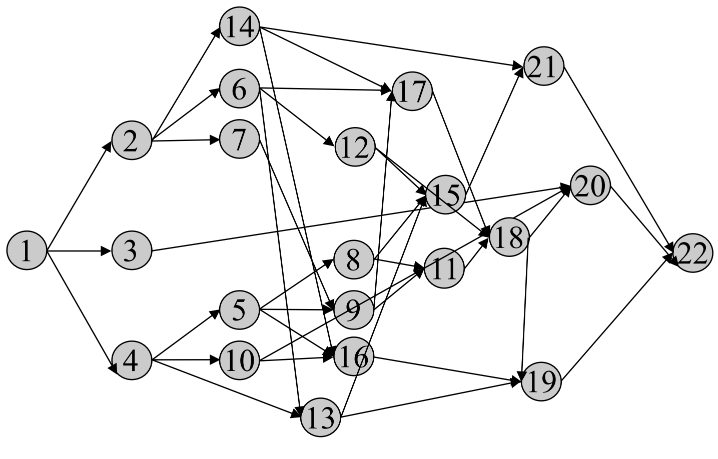


Fig. 4.2 Project network of benchmark problem #n041_1

4.2 Experiment and discussion

To demonstrate the efficiency performance of our proposal, some numerical experiments are conducted to compare hGMEDA and MMEDA with other popular methods. A significant amount of MOEAs have been proposed to solve the multi-objective problems. Typically, Schaffer proposed first MOEA based on simple GA named vector evaluated genetic algorithm (VEGA) with vector-valued fitness measures. Deb introduced an algorithm called non-dominated sorting genetic algorithm II (NSGA-II) with the use of crowding distance mechanism and Pareto ranking method. Zitzler developed strength Pareto evolutionary algorithm II (SPEA2) by a novel raw fitness assignment function and density mechanism. All these methods have been proved very effective and applicable to different kinds of applications, and can be acted as possible comparing methods to evaluated new design approach.

Comparing with single VEGA is an intuitive way, but the diversity of VEGA has been proved very poor because of the selection bias. To make the comparison results more convincing, in our experiments, NSGA-II and SPEA2 are selected. To make the comparisons fairly (EDAs have been proved having better efficacy than GAs), we use the sampling strategies and update mechanism of two algorithms, and hybrid with conventional EDA as the searching engine for optimization.

All algorithms were implemented by JAVA language and conducted on Intel Core i3 with 4G memory. For each algorithm and each benchmark problem, we evaluate the mean result with 30 trials. To make the same environment and fairly comparisons, the major parameters of methods are listed in Table 4.1.

In this study, we adopt coverage [98] and generational distance [99] to evaluate the optimality, and spacing [100] to evaluate the distribution performance, which are very popular performance measures for MOEAs.

Table 4.1 The parameters of compared algorithms

	NSGA-II, SPEA2	MMEDA, hGMEDA
Generations	1000	1000
Population	100	100
Operators	Sampling	Tournament(k) Gibbs-Sampling Local search
Parameters	*promisingRate* = 0.7	*promisingRate* = 0.7 $\alpha = 1.5, \beta = 0.5$ *elimRate* = 0.1 $k = 2$

4.2.1 Coverage

To evaluate the optimality of Pareto solutions, comparisons on coverage are illustrated. 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.

$$C(S_1, S_2) = \frac{\left|\{s_2^\beta \in S_2 ; \exists s_1^\alpha \in S_1 : s_1^\alpha \geq s_2^\beta\}\right|}{\left|S_2\right|} \tag{4.1}$$

In equation (4.1), 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.

In Table 4.2, it shows the comparison on coverage of NSGA-II, SPEA2, MMEDA and hGMEDA on the results of mean value with 30 runs of three algorithms. Mean value represents optimality of solutions in Pareto set, and hGMEDA, MMEDA outperforms NSGA-II and SPEA2 with three benchmark problems. Compared with NSGA-II, SPEA2 and MMEDA, hGMEDA can improve about 17%, 22% and 4.16% on average respectively.

Table 4.2 Comparison on coverage of NSGA-II, SPEA2, MMEDA and hGMEDA

Problem	Mean Value (30 trials)			Mean Value (30 trials)			Mean Value (30 trials)		
	$A_1 =$ C(hGMEDA, NSGA-II)	$A_2 =$ C(NSGA-II, hGMEDA)	Improved (A1-A2)	$A_3 =$ C(hGMEDA, SPEA2)	$A_4 =$ C(SPEA2, hGMEDA)	Improved (A3-A4)	$A_5 =$ C(hGMEDA, MMEDA)	$A_6 =$ C(MMEDA, hGMEDA)	Improved (A5-A6)
#n041_1	0.487	0.398	8.90%	0.556	0.304	25.20%	0.292	0.231	6.10%
#n042_1	0.624	0.328	29.60%	0.614	0.308	30.60%	0.228	0.168	6.00%
#n043_1	0.554	0.401	15.30%	0.592	0.325	26.70%	0.284	0.189	-2.40%
#n044_1	0.541	0.336	20.50%	0.584	0.379	20.50%	0.165	0.198	8.60%
#n045_1	0.472	0.365	10.70%	0.481	0.387	9.40%	0.271	0.246	2.50%
Average	0.536	0.366	17.00%	0.565	0.341	22.48%	0.248	0.206	4.16%

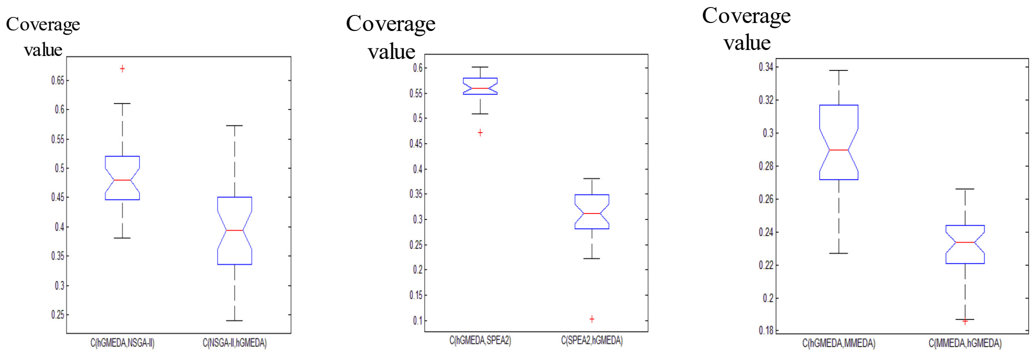


Fig. 4.3 Boxplot of coverage by hGMEDA, NSGA-II, SPEA2 and MMEDA on #n041_1

In Fig. 4.3, it shows the boxplot of coverage by hGMEDA with other three methods on problem #n041_1. The boxplot figure shows the mean value, largest value, smallest value and first and third quartiles (to represent standard deviations). From the results of boxplot, compared with NSGA-II and SPEA2, hGMEDA have better averaged value and smaller deviation. For MMEDA, the deviation of hGMEDA is larger, probability coming from GA involved.

4.2.2 Generational distance

$GD(S_i)$ represents an averaged minimum distance of the solutions in S_i from reference Pareto set PF^*, which comes from the Pareto set gotten from all the algorithms. The smaller GD of S_i represents better optimality with considering of approaching PF^*.

$$GD(S_i) = \frac{\sum_{\alpha=1}^{|S_i|} (\min(d_{s_i^\alpha, s_*^\beta}), \forall s_*^\beta \in PF^*)}{|S_i|} \tag{4.2}$$

From the result in Table 4.3, it indicates that MMEDA and hGMEDA has smaller GD values than NSGA-II and SPEA2. Our proposals outperform other two algorithms, with the improvement of 7.59% and 10.28% on average of five benchmark problems. hGMEDA adopts Markov network to solve the constraint problems by representing the relationship among activities for mode selection. With the knowledge getting from Markov network and the strong convergence

Table 4.3 Comparison on generational distance of NSGA-II, SPEA2, MMEDA and hGMEDA

Problem	Mean Value [capacity • time] (30 trials)				Improvement of hGMEDA		
	GD(hGMEDA)	*GD*(NSGA-II)	*GD*(SPEA2)	*GD*(MMEDA)	Improved (with NSGA-II)	Improved (with SPEA2)	Improved (with MMEDA)
#n041_1	75.14	82.39	82.98	76.75	8.80%	9.45%	2.10%
#n042_1	92.48	96.51	98.34	94.04	4.18%	5.96%	1.66%
#n043_1	76.39	80.07	85.53	76.01	4.60%	10.69%	-0.50%
#n044_1	82.34	92.16	95.63	88.97	10.66%	13.90%	7.45%
#n045_1	69.33	76.81	78.25	72.47	9.74%	11.40%	4.33%
Avg.	79.13	85.58	88.14	81.64	7.59%	10.28%	3.01%

performance of simple fitness assignment function, our proposal outperforms other two algorithms with more convincing solutions.

In Fig. 4.4, it shows the boxplot of generational distance by hGMEDA with other three methods on benchmark problem #n041_1.

4.2.3 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*. Smaller *SP*(*S*) means that solution *S* is in better diversity.

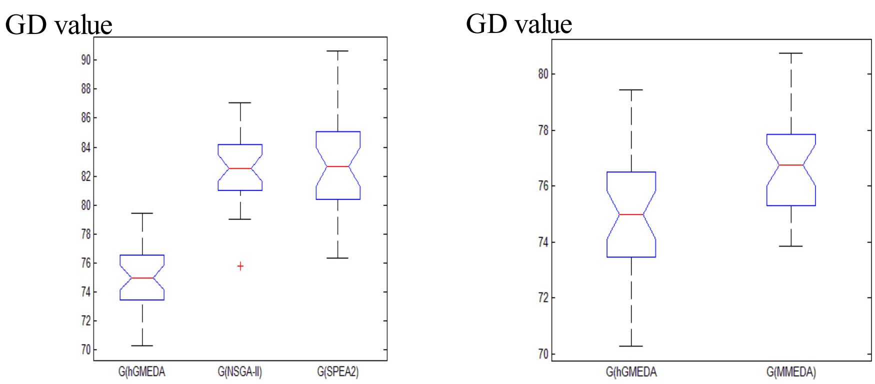


Fig. 4.4 Boxplot of generational distance by hGMEDA, NSGA-II, SPEA2 and MMEDA on #n041_1

Table 4.4 Comparison on spacing of NSGA-II, SPEA2, MMEDA and hGMEDA

Problem	Mean Value [capacity・time] (30 trials)				Improvement		
	SP(hGMEDA)	*SP*(NSGA-II)	*SP*(SPEA2)	*SP*(MMEDA)	Improved (with NSGA-II)	Improved (with SPEA2)	Improved (with MMEDA)
#n041_1	34.78	37.98	37.21	36.75	8.43%	6.53%	5.36%
#n042_1	30.27	37.11	36.54	33.74	18.43%	17.16%	10.28%
#n043_1	29.75	35.47	34.18	31.82	16.13%	12.96%	6.51%
#n044_1	27.91	30.17	29.98	28.76	7.49%	6.90%	2.96%
#n045_1	31.12	34.78	33.29	32.94	10.52%	6.52%	5.53%
Average	30.76	35.10	34.24	32.80	12.20%	10.01%	6.13%

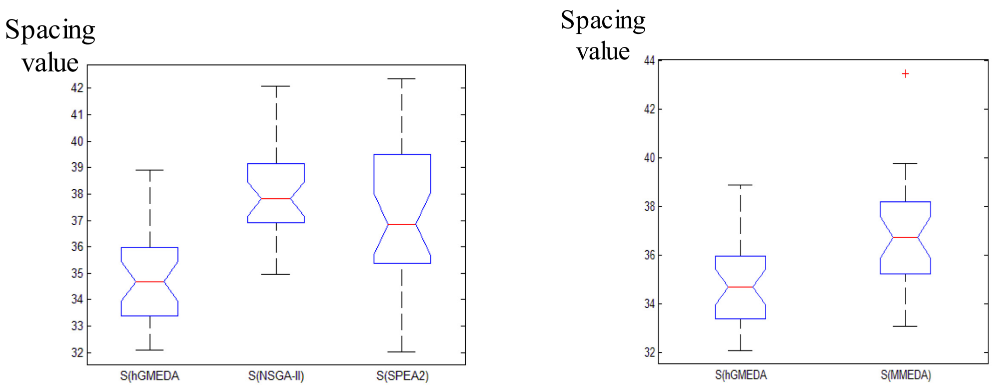
$$\overline{d} = \frac{\sum_{i=1}^{|S|} d_i}{|S|} \tag{4.3}$$

$$SP(S) = \sqrt{\frac{1}{|S|-1}\sum_{i=1}^{|S|}(d_i - \overline{d})^2} \tag{4.4}$$

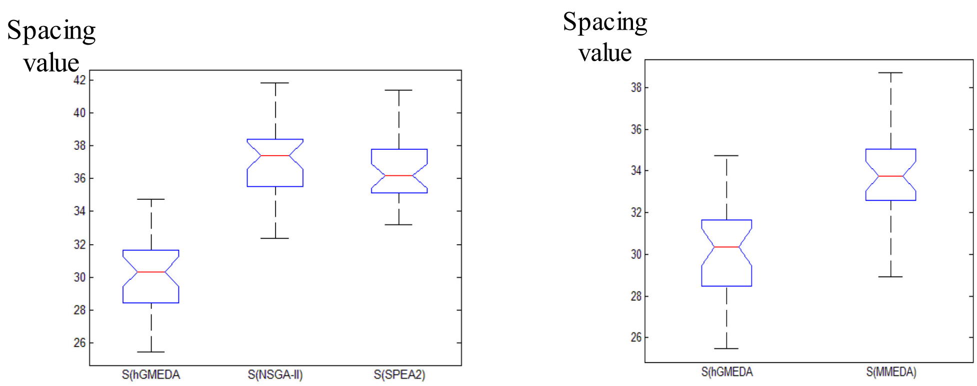
where d_i is the nearest distance of individual i in solution set S.

The result of *SP* values are shown in Table 4.4. It shows that hGMEDA has smaller *SP* than other two methods, which demonstrates that our proposal is better on distribution performance. With the combined sampling strategies, for both the edge region and the central region, hGMEDA can keep the solution with diversity. Meanwhile, a simple mechanism to preserve the diversity evenly through dynamic adjustment on D-Fitness is adopted, so that hGMEDA can achieve satisfactory dispersion performance. In algorithm of MMEDA, the solutions are always sampled by the probability model, which is more stable one. Compared with MMEDA, hGMEDA employs GA in the first stage, which can provide more "random" solutions. That's the reason why hGMEDA has better distribution performance.

In Fig. 4.5, it shows the boxplot of spacing by hGMEDA with other three methods on two benchmark problems #n041_1 and #n042_2.



#n041_1



#n042_1

Fig. 4.5 Boxplot of spacing by hGMEDA, NSGA-II, SPEA2 and MMEDA on #n041_1 and #n042_1

4.2.4 Computation time

The computation costs of the multi-objective optimization algorithms mainly depend on the fitness evaluations, ranking and distance calculation. We need to compare four methods with the same termination criterion that reaches 1000 generations. As shown in Fig. 4.6, the mean computation time of NSGA-II, SPEA2, MMEDA and hGMEDA are 215.7s, 236.3s, 197.6s, and 176.1s respectively.

We take m as the number of objectives, N as the population size. In the algorithm of NSGA-II, it needs mN^2 times comparisons to find the relationship of domination, and the time complexity of NSGA-II is O(mN^2). In SPEA2, for each individual, the kth nearest distance is calculated, so

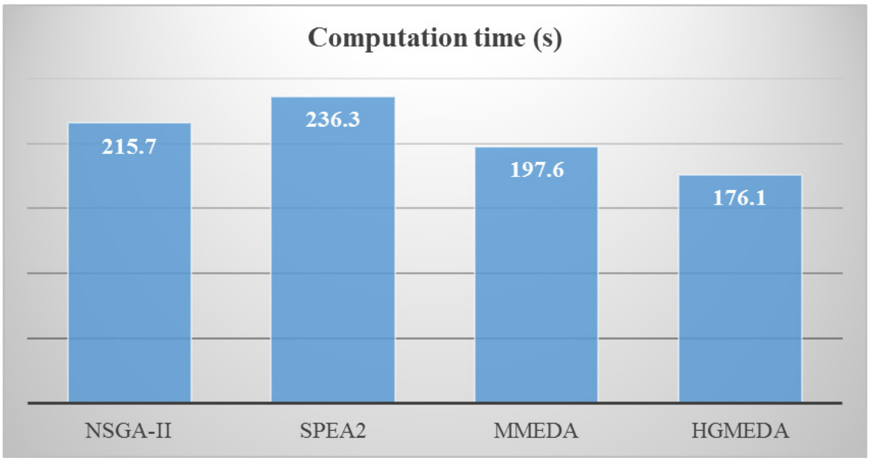


Fig. 4.6 Comparison on calculation time

that the time complexity is O(mN^2logN). Without calculation for crowded distance, the time spending for fitness calculation in MMEDA is slightly small. We have to spend extra time to estimate the Markov structure for partial decision variables. However, in the evolving process of Markov network based EDA, we do not need to update the structure in every generation. As a result, although the Markov network cost longer time, the fitness functions of hGMEDA can save time a lot, totally our proposal has smaller computation time. Take hGMEDA and MMEDA for comparison, hGMEDA can reduce CPU time about 10.9% due to co-evolutionary paradigm.

4.3 Summary

In chapter 3 and 4, two types of empowered Markov network based EDA are developed for solving the multi-objective RCSPs. First proposal is multi-objective Markov network based EDA (MMEDA), in which the framework of multi-objective optimization algorithm with a combined fitness assignment function. Second one is two-stage hybrid GA and MMEDA (hGMEDA) to enhance the computational efficiency, which is inspired by the idea of cooperative co-evolutionary paradigm with sequential evolving process. Furthermore, two kinds of problem-specific local search for makespan and load balancing are proposed to increase solutions quality. The experiment results demonstrate that, compared with NSGA-II and SPEA2, our proposal hGMEDA can improve 17.00%, 22.48% on coverage, 7.59%, 10.28% on generational distance and 12.20%, 10.01% on spacing averagely. Furthermore, hGMEDA can reduce CPU time about 10.9% than MMEDA, about 18.4% and 25.5% faster than NSGA-II and SPEA2 respectively.

Chapter 5
Multi-objective Robust Scheduling Method based on MMEDA for RCSP

This chapter gives a detailed description of our proposal, one robust scheduling method based on hGMEDA. In this chapter, we discussed the manner of robust schedule and two kinds of robust measures on time-based-robust and capacity-based-robust are defined. Next, a multi-phase scheduling method to make robust scheduling is developed and explained in detail.

5.1 Robustness measure

To deal with different level of uncertainties in production scheduling problems, different manner of schedules are produced. In this study, we focus on the medium uncertainty, and try to develop a proactive or robust schedule, which is a more practical and common situation in real-world problems.

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 robustness has to be considered at the same time. In general, RCSP with uncertainty can be viewed as three group objectives: time-based, resource-based and robust-based.

In job shop environment, the robustness is often defined as the difference between expected value objective (e.g., makespan) and actual ones [85]. In RCSP, except the duration of project, the resource usage is also need to take into consider. For example, the deviation of the actual starting time of each operation and the expected one is to be minimized, or minimize the resource flow network for the problems with unrestricted resource availability [86].

In order to well and fully describe the robustness of RCSP, we proposed two kinds of robust measures for RCSP: time-based-robust and capacity-based-robust.

5.1.1 Time-based-robust measure (TRM)

In order to measure the robustness based on time criterion, one popular way is slack-based. There are two kind slack time in previous studies: total slack time and free slack time [101]. In this study, we use the concept of total slack time, which represents the ability of keeping expected makespan, defining as the difference between the possible earliest starting time of one activity and its corresponding possible latest starting time.

In Fig. 5.1, 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 *A*2 while the red area is for activity *A*3. In previous studies, most of them thought the slack time for *A*2 and *A*3 are equal, because the lengths of time period are same. However, from the view of resource allocation in RCSP, *A*3 requires more amount of resource than *A*2, in other words, if *A*3 delayed, more resources should be held by it and impact to the system is larger than delay of *A*2. Meanwhile,

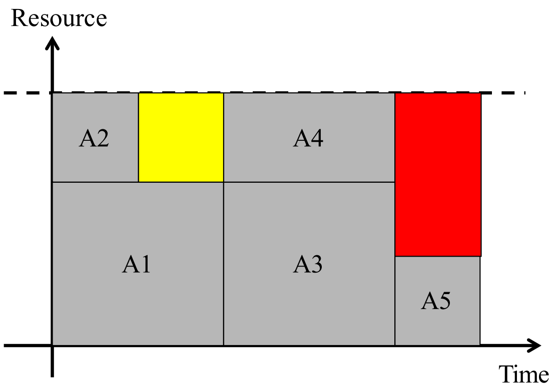
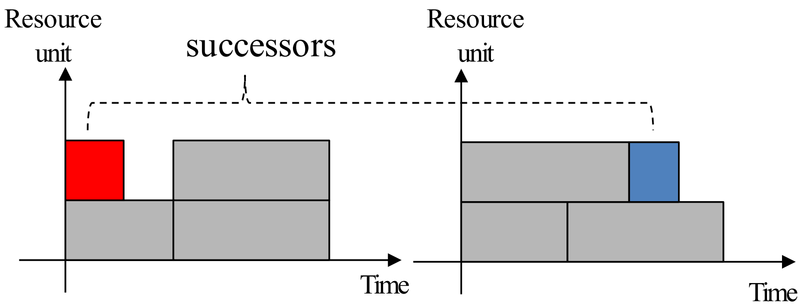
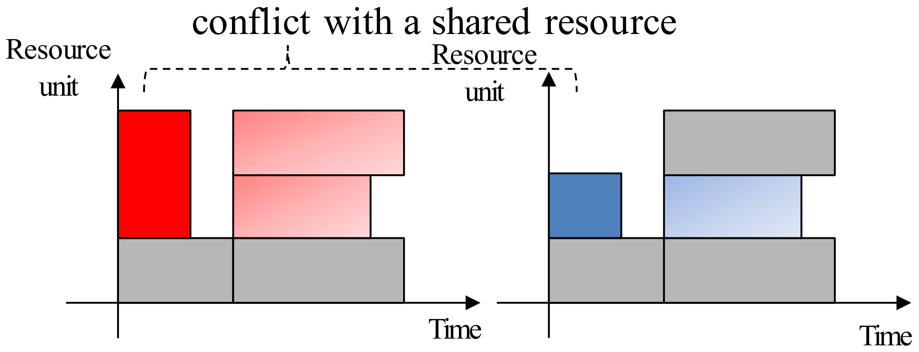


Fig. 5.1 An illustrative example of slack time



(a)



(b)

Fig. 5.2 Amount of successors and conflict resource

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. 5.2, it shows two typical conditions of slack time. In Fig. 5.2(a), two activities with red colour and blue colour 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. 5.2(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.

$$TRM: \ \sum_{j=1}^{N} s_j NSucc_j \sum_{k=1}^{K} r_{jk} \tag{5.1}$$

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.

5.1.2 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 hours per day and 5 days per week), which is the capacity of time-adjusted resource.

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

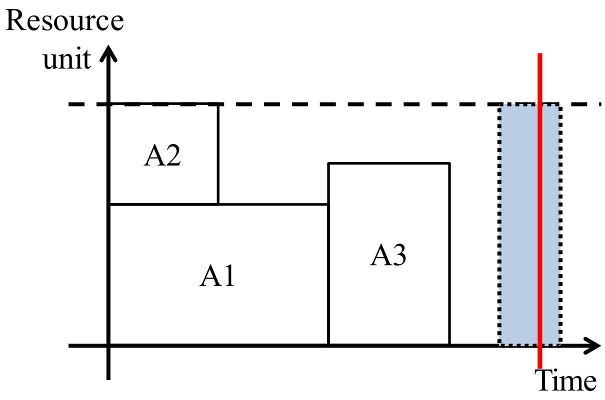


Fig. 5.3 Resource capacity exceeding with uncertainty

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. 5.3, there is one schedule which containing 3 activities ($A1$, $A2$ and $A3$) and the resource 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 that the total working time is also uncertainty but follows normal distribution. One target of the schedule is try 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:

$$CRM: \ prob_{\xi \in \Xi}\left(\sum_{t=1,\ldots,horizon;\, j=1,\ldots,N} (d_j^{\xi} \times x_{jt}) \le G_k \right) \ge \text{threshold} \tag{5.2}$$

where G_k is the goal value for resource k, threshold is the confidence level, such as 80%.

5.2 Problem formulation of robust RCSP

The deterministic RCSP has been explained in previous chapter, here we focus on the uncertainty of duration time.

From the previous literatures, there are some probability distribution used in robust optimization algorithms [102], such as normal distribution, Poisson distribution, and uniform distribution [103].

In this study, we take normal distribution as the probability model for duration uncertainty, which is the most popular and has been widely used in recent researches. As shown in Fig. 5.4, it shows the curve of an illustrated example of normal distribution. Usually, the normal distribution is represented as:

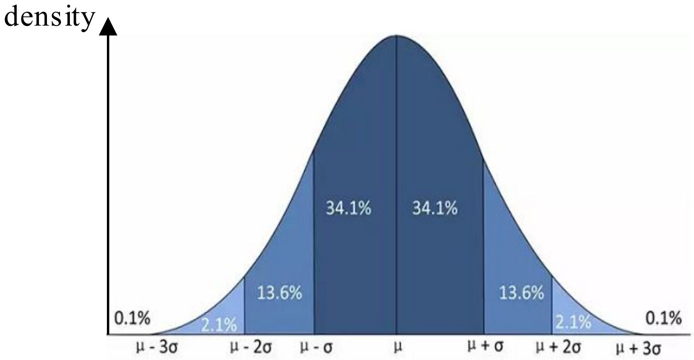


Fig. 5.4 An illustrated example of normal distribution

$$f = N(\mu, \sigma^2) \tag{5.3}$$

where μ and σ^2 represent mean value and variance (squared scale) respectively, μ is used to decide the averaged value and σ can control the uncertainty level high or low.

For RCSP with duration time uncertainty, we model the problem with bi-objective of makespan minimization and time based robustness (TRM) maximization, together with the chance constraint of capacity based robustness (CRM), which is a very generic model applicable to different kinds of applications:

- *Objective:*

$$\min \left\{ \frac{\sum_{\xi \in \Xi} \max_{j=1,\ldots,N} (c_j^{\xi})}{|\Xi|} \right\} \tag{5.4}$$

$$\max \left\{ \sum_{j=1}^{N} s_j NSucc_j \sum_{k=1}^{K} r_{jk} \right\} \tag{5.5}$$

- *Subject to:*

$$\begin{aligned} & prob_{\xi \in \Xi}\Big(\sum_{t=1,\ldots,horizon;\, j=1,\ldots,N} (d_j^{\xi} \times x_{jt}) \le G_k \Big) \ge \text{threshold} \\ & k = 1,\ldots,K; \xi \in \Xi \end{aligned} \tag{5.6}$$

where c_j^{ξ} is the completion time of operation j on scenario ξ.

The complete mathematical model will be given in next chapter, including nations, decision variables, objectives and constraints.

The complete mathematical model for MRCPSP with duration uncertainty is given as following:

- *Index*

i	activity index, i =1, …, N
m	mode index, m =1, …, M_j
k	resource index, k =1, …, K
ξ	scenario set, ξ = 1,…, Ξ

- *Parameter*

N	the total amount of activities
M_j	the total amount of modes for activity j
K	the total amount of resources
N_k	capacity of resource k
Ξ	the total amount of scenarios
G_k	the recommendation level of resource k
d_{jm}^{ξ}	for scenario ξ, the duration time of activity j with mode m
s_{jm}^{ξ}	for scenario ξ, the starting time of activity j
c_j^{ξ}	for scenario ξ, the completion time of activity j
r_{jkm}	usage of resource k for activity j selecting mode m
p_j	predecessors set of activity j
$NSucc_j$	the total amount of successors of activity j
s_j	slack time of activity j

- *Decision Variable*

$$x_{jmt} = \begin{cases} 1 & \text{activity } j \text{ is executed at time } t \text{ with mode } m; \\ 0 & \text{otherwise.} \end{cases}$$

- *Objective:*

$$\min \left\{ \frac{\sum_{\xi \in \Xi} \max_{j=1,\dots,N} (c_j^{\xi})}{|\Xi|} \right\} \tag{5.7}$$

$$\max \left\{ \sum_{j=1}^{N} s_j NSucc_j \sum_{k=1}^{K} \sum_{m=1}^{M_j} r_{jkm} \right\} \tag{5.8}$$

- *Subject to:*

$$\sum_{i=1}^{M_i} \sum_{t=s_i^{\xi}}^{c_i^{\xi}} t \cdot x_{imt} \le \sum_{j=1}^{M_j} \sum_{t'=s_i^{\xi}}^{c_i^{\xi}} (t' - d_{jm}) \cdot x_{jmt'}, \quad j = 1,\dots,N; i \in p_j; \xi \in \Xi \tag{5.9}$$

$$\sum_{m=1}^{M_j} \sum_{t=s_j^{\xi}}^{c_j^{\xi}} x_{jmt} = 1, \;\; j = 1,\dots.,N; \xi \in \Xi \tag{5.10}$$

$$prob_{\xi \in \Xi}\Big(\sum_{t=1,\dots,horizon; j=1,\dots,N; m=1,\dots,M_j} (d_{jm}^{\xi} \times x_{jmt}) \le G_k \Big) \ge \text{threshold}, \quad k = 1,\dots,K; \xi \in \Xi \tag{5.11}$$

$$x_{jmt} \in \{0,\ 1\}, \quad j = 1,\dots.,N; m = 1\dots M_j; t = 1\dots\text{horizon} \tag{5.12}$$

$$s_j^{\xi} \geq 0,\ c_j^{\xi} \geq 0, j = 1,...., N; \xi \in \Xi \tag{5.13}$$

Inequality (6.3) presents the constraints of precedence relation among activities. Equation (6.4) guarantees that one activity has to choose one of its corresponding modes to execute. Inequality (6.5) states the chance constraint of the capacity-based robustness. Equation (6.6) and (6.7) represent the nonnegative restrictions.

5.3 Two-phased robust scheduling method based on hGMEDA (robust hGMEDA)

When we try to solve one RCSP with uncertainty, there are several points have to be concerned:

a) Resource capacitated constraint & resource allocation;

b) Precedence relation constraint & sequencing;

c) Robust optimization & robustness measures;

d) Uncertainty evaluation & simulation on scenario-based;

e) Chance constraint (optional);

f) Multi-objective optimization (optional);

g) So on.

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.

The strategy of robust hGMEDA is shown in Fig. 5.5.

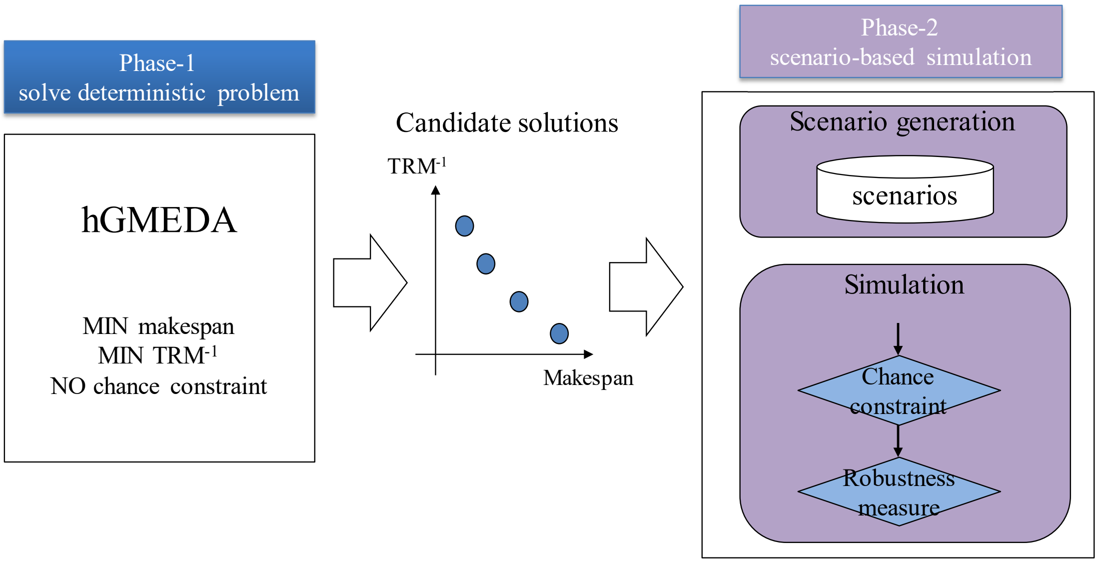


Fig. 5.5 Strategy of robust hGMEDA

5.3.1 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.

Same to the deterministic multi-objective RCSP solved in chapter 3 and 4, in phase-1, we take hGMEDA to calculate some candidate solutions.

Here we have to mention that, in chapter 3 we have discussed that, there are some non-Pareto solutions in the archive. One 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.

Meanwhile, after solutions are sampled, a problem-specific local search is applied to increase the quality of each candidate solution.

If the critical path is kept, the makespan cannot be shorten. As a result, we try to make a new schedule with smaller makespan by breaking the existing critical path. Different to JSP or FJSP, in project scheduling, it has a high probability that there are many 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 time based robustness. So that our local search is based on critical path with considering slack time.

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 problems, the solutions before and after the local search are both kept. One reason is for multi-objective problems, local search maybe improve one objective while decrease another one, but both solutions could be good or Pareto ones. Second reason is that, for robust scheduling problems, more optimal solutions may be not robust ones or cannot satisfy the chance constraints, in other words, we have to keep more candidate solutions.

Fig. 5.6 shows the pseudo code of local search by moving activity for reducing makespan. The purpose of moving activity is to change the position of one activity to other assignable position with the constraints of other activities existing. Since new schedule is obtained by

```
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. 5.6 Local search by moving activity deleting one activity and moving it to another position, the new makespan must be not 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. Then we have to calculate each time interval with the equation (5.14):

$$Time\_Interval_i = \sum_{j \in TI_i} s_j \times NSucc_j \times r_{jkm} \tag{5.14}$$

where TI_i represent the set of the activities which take the time interval i as their slack time period.

Finally, we select some promising solutions based on multi-objective, and update the archive. After generations, the solutions in the archive will be used as the candidate solutions for next phase.

5.3.2 Phase-2: solve the uncertainty problem by scenario based simulation

In phase-2, it contains 2 main steps. In step 1, some scenarios are generated. As shown in Fig. 5.7, 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.

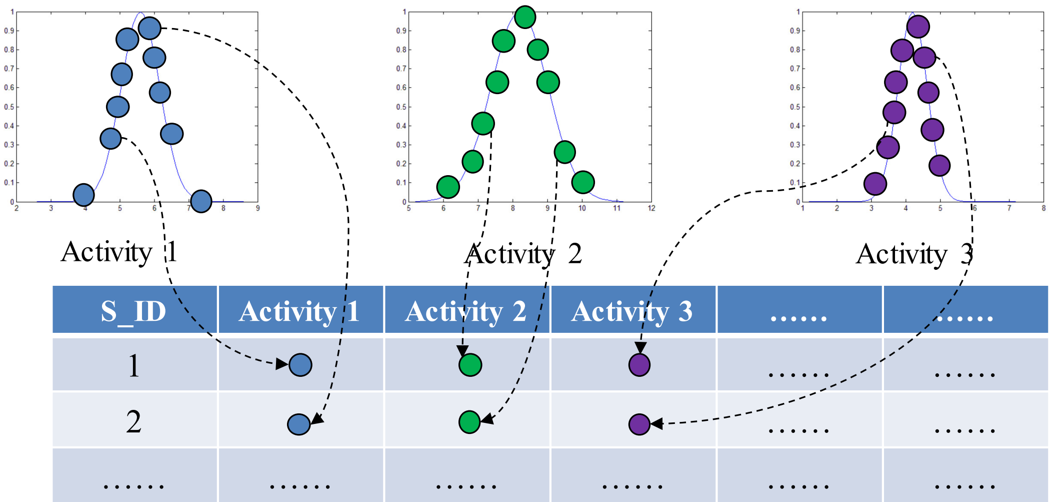


Fig. 5.7 Scenario generation

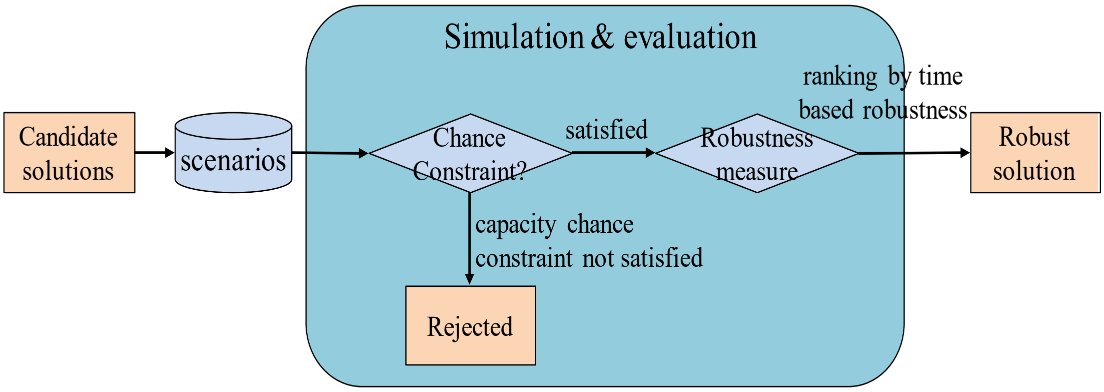


Fig. 5.8 Flowchart of simulation

In step 2, firstly, by scenario based simulation, we evaluate each candidate solution whether to satisfy chance constraint, and reject one which fails to satisfy. Next, based on the robustness measure, finally the robust schedule is selected.

The flowchart of simulation is shown in Fig. 5.8.

Here we briefly discuss how to generate the final robust schedules. Depending on the problem setting or decision made by project manager, there could be three possible ways.

a) Pareto optimization solutions of schedule:

We have already received some alternative solutions in archive, and some unsatisfied solutions are cleaned out by checking on chance constraint. The simple way is, we collect all the remaining solutions by Pareto dominated checking, and finally the solutions belong to Pareto set are all kept as the solutions. That means every solution remaining could be the optimal one in some conditions in future. Because in phase-1, multiply objectives contain both makespan and robustness measures, so that the Pareto optimization solutions have the potential to be the most robust schedule.

b) One single robust schedule by weighted average:

Another way is making multiply objectives as one objective, with the weighting given to each objective by decision makers or problem experts. It is a difficult way to give suitable weighting to each objective. However, it is still one possible way to get one final solution.

c) One single robust schedule by robust measure:

The third way is, for the alternative solutions checked by chance constraints, we evaluate them by scenario-based again (it could be with the same scenarios or different scenarios re-sampled), with the objective of new robustness measure or original one in phase-1. For example, after some solutions are eliminated for violating some chance constraints, one robust measure is employed to all the remaining solutions (the robust measure could be same with one objective in phase-1, or another one), and the final robust solution is selected based on the objective value of robust measure.

In this study, we choose second way to decide our robust solution, which is the most reasonable way. Tolerant of uncertainty is very important issue for robust scheduling problems, however, for any scheduling problems, makespan should be consider in high priority. Because it is not the key topic, in this study, we do not discuss how to decide the weights.

5.4 Evolving procedure of proposed scheduling method

In Fig. 5.9, it illustrates the general evolving procedure of proposed robust scheduling method based on hGMEDA.

To solve the RCSP under uncertainty, there are three steps in phase-1. In step-1, based on objective of makespan, some sequencing solutions are generated by GA. In step-2, by using MMEDA, the solutions of resource allocation are produced. In step-3, to combine these two sub-solutions and evaluate them by D-Fitness to get Pareto solutions. All these 3 steps are performed

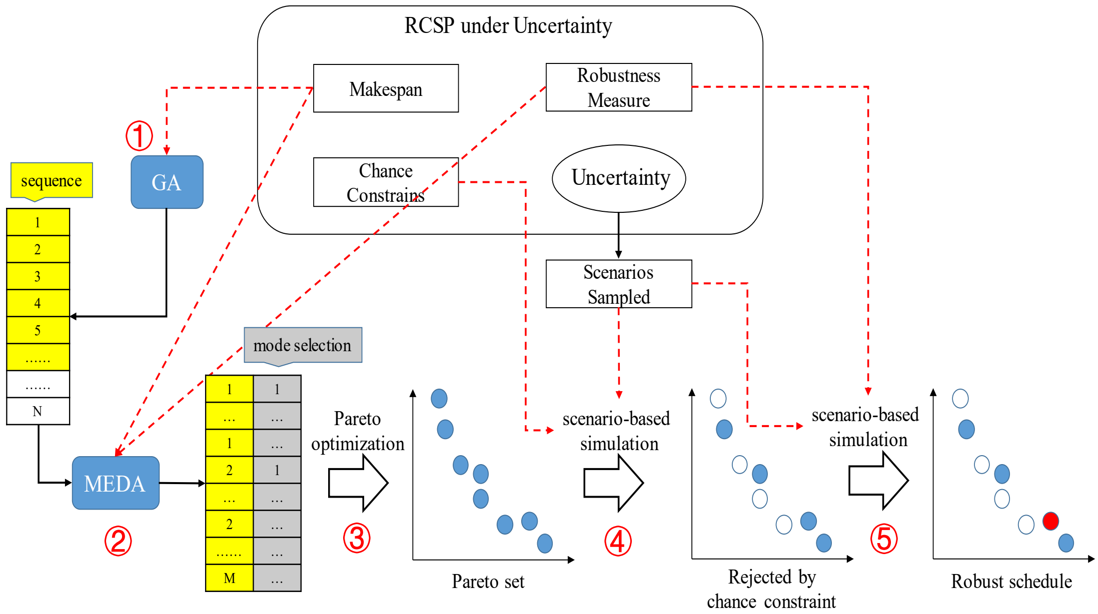


Fig. 5.9 The evolving process of robust scheduling method for RCSP

with the deterministic manner. Based on the information of multiply objectives and capacitated constraints, by using hGMEDA, the alternative solutions are achieved (shown as blue nodes).

In phase-2, it contains 2 steps. In step-4, some solutions are eliminated by checking chance constraints, which represented as the open circle. In step-5, based on the robustness measure, the solution colored red has the highest objective value, so that being selected as the final robust schedule.

Here we pay more attentions to the definition and difference between objective of time based robustness (TRM) in phase-1 and robustness measure in phase-2.

In phase-1, we treat the problem as bi-objective: minimizing makespan and maximizing TRM. If the scheduling system has longer slack time, the ability of absorb the disruption will be increased, especially to protect the expected makespan.

In phase-2, we try to decide which one is the robust schedule based on the robustness measure. One possible way is to use the regret of makespan, which means the difference between expected makespan and actual ones. After one schedule produced, the expected makespan can be calculated as the makespan for every activity choose its averaged duration. With the duration changing, the actual makespan may be increase or decrease. If the difference between these two makespan is smaller, the schedule is more robust for keeping makespan.

From the above example, we can clearly found that, objective and robust measure are in different definitions, but these two aiming to make the same contribution. The maximization of slack time has a high probability leading to small difference between expected makespan and actual one.

However, if we set the regret of makespan as one objective in phase-1, it becomes very difficult to solve the problem as the deterministic manner and without simulation by scenarios. Therefore, that’s one reason why it is possible for us to use two criteria in two phases.

In this study, because this is not the key point of the research, we use the same measure of time based robustness TRM in both phase-1 and phase-2.

Chapter 6
Experimental Evaluation on Resource Constrained Robust Project Scheduling

6.1 Introduction

In this chapter, we take the application of MRCPSP with duration uncertainty for case study, and to evaluate the solutions given by our proposal to demonstrate the searching ability of robust hGMEDA and tolerant of uncertainty of the robust solution.

We still take the same case study in chapter 4, making experiments on benchmark problem data set PSPLIB [96]. In the benchmark PSPLIB, for each activity, the duration of completion time is one constant value. To create the problems as MRCPSP with duration uncertainty, we revised the duration time as the normal probability $N(\mu, \sigma^2)$, where μ is the original duration time in benchmark problems, and we set σ as 10% of μ.

For solving project scheduling problems with uncertainty, some algorithms are developed, however, most of them belong to heuristic methods. Igelmund et al. [104] developed a method based on selection policy named pre-selective to minimize the cost for the total project. For PERT project, Golenko developed a novel resource constrained scheduling model [105], where the activities have random durations. Fawzan developed a newly designed robust measure to slove RCPSP with two objective of maximizing robustness and minimizing makespan [106], by generating an approximate set with using Tabu search. Roel Leus developed a heuristic method by using the algorithms of relaxation on scenarios [107], which enables the decision maker to produce a schedule with acceptable value of objective with each scenario.

To demonstrate the efficiency performance of our proposal fairly, some experiments are performed to compare robust hGMEDA with two meta-heuristic methods, including our proposed algorithm deterministic hGMEDA and one typical MOEA SPEA2.

For SPEA2, we model the scheduling problem by GA-based representation with two sub-chromosome including (1) priority value of each activity for deciding sequencing and (2) mode id for each activity for deciding mode selection. The detail coding manner for SPEA2 is coming from a method proposed by Wang [26].

6.2 Experiment and discussion

We design three experiments to demonstrate, first one is conducted to make comparisons on the optimality of expected makespan. Secondly, we evaluate the variance of makespan of our schedule compared with deterministic ones to evaluate time based robustness. Thirdly, we evaluate the capacity based robustness by comparing the percentage of satisfaction of chance constraint.

All algorithms were implemented by JAVA language and conducted on Intel Core i3 with 4G memory. For each algorithm and each benchmark problem, we evaluate the mean result with 30 trials. To make the same environment and fairly comparisons, the major parameters of methods are shown in Table 6.1.

Table 6.1 The parameters of compared algorithms for robust scheduling

	Gen.	**Pop.**	**Operator**	**Parameter**
SPEA2	1000	100	Crossover(P_c) Mutation(P_m)	$P_c = 0.80$ $P_m = 0.20$ $k = 2$
MMEDA **hGMEDA**	1000	100	Tournament(k) Gibbs sampling Local search	*promisingRate* = 0.7 *elimRate* = 0.1 $k = 2$

6.2.1 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 equation (6.1):

$$E(C_{\max}) = \frac{\sum_{\xi \in \Xi} \max_{j=1,\dots,N}(c_j^{\xi})}{|\Xi|} \tag{6.1}$$

In the Table 6.2, there is 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.

Fig. 6.1 shows the results of expected makespan for problem #n041_1. Based on the scenario-based simulation, our proposal finally choose the robust one instead of the solutions with the highest time based robustness objective value under deterministic environment. In other words,

Table 6.2 Comparison on expected makespan of robust hGMEDA, hGMEDA and SPEA2

Problem	Mean value of makespan [time unit] (30 trials)			Improvement	
	Robust hGMEDA	hGMEDA	SPEA2	Decreased (with hGMEDA)	Decreased (with 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%

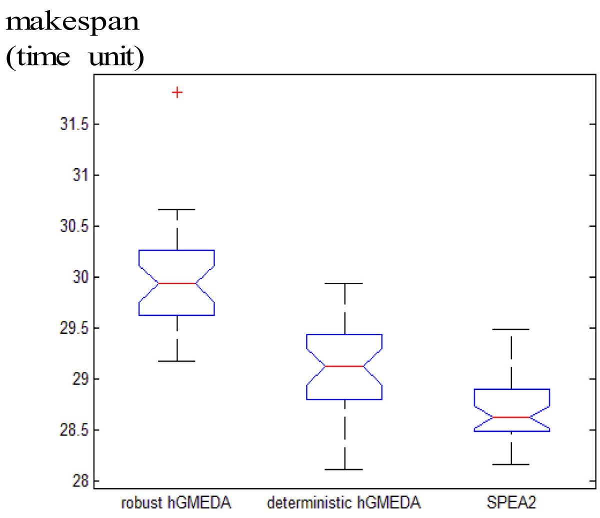


Fig. 6.1 Boxplot of expected makespan by robust hGMEDA, hGMEDA and SPEA2 on #n041_1

our solution pays more attention on robust under uncertainty, and would sacrifice some optimality of makespan.

6.2.2 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 equation (6.2), and we evaluate the mean result with 30 trials.

$$\text{Variance} = \frac{\sum_{\xi \in \Xi} \left| MK^{\xi} - MK^{*} \right|}{|\Xi|} \tag{6.2}$$

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 6.3, 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.

In Fig. 6.2, it shows the boxplot figure of variance of makespan by three different methods. From the figure of results on benchmark problem #n041_1, with 30 trials, not only the mean value of variance of robust hGMEDA is smaller, but also the standard deviation is smaller. In other words, the solutions given by robust hGMEDA have high ability of tolerant ability for uncertainty.

Table 6.3 Comparison on variance of makespan of robust hGMEDA, hGMEDA and SPEA2

Problem	Mean value of Variance [time unit] (30 trials)			Decrease of Variance	
	Robust hGMEDA	hGMEDA	SPEA2	Decreased (with hGMEDA)	Decreased (with 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%

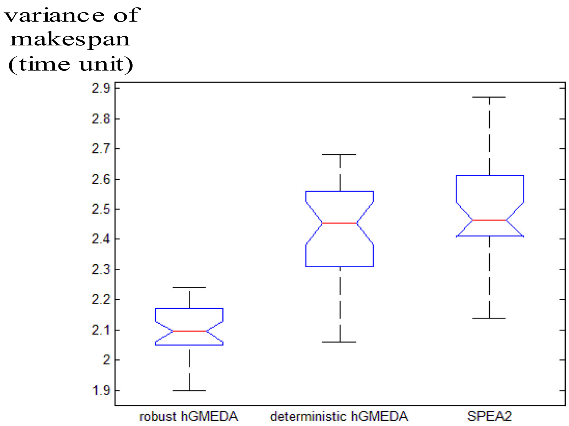


Fig. 6.2 Boxplot of variance of makespan by robust hGMEDA, hGMEDA and SPEA2 on #n041_1

We summary the two objectives of makespan and time-based-robustness together in Fig. 6.3. From the figure, averagely, compared with deterministic hGMEDA, our proposal can increase the robustness 9.39% with the cost of 3.71% increase of makespan. For scheduling method based on SPEA2, our proposal can increase the robustness 12.37% with the cost of 3.61% increase of makespan.

6.2.3 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

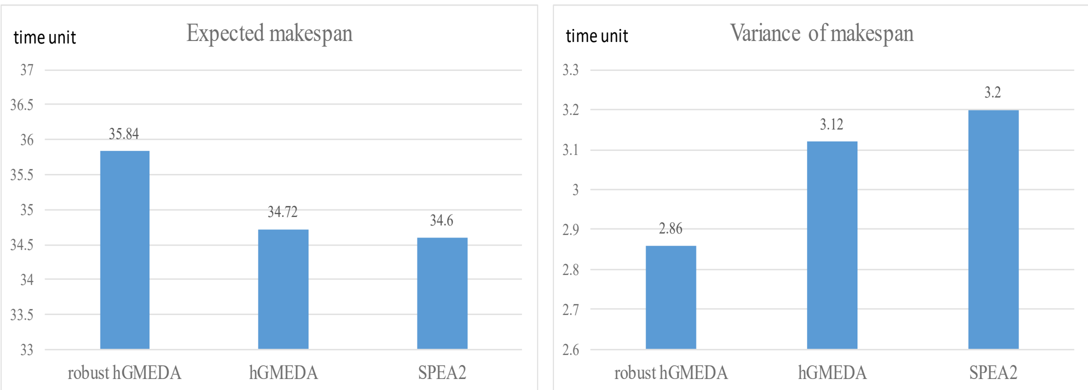


Fig. 6.3 Expected makespan and variance of makespan

Table 6.4 Comparison on percentage of satisfying chance constraint of robust hGMEDA, hGMEDA and SPEA2

Problem	Mean value of Percentage (30 trials)			Improvement	
	Robust hGMEDA	hGMEDA	SPEA2	Increased (with hGMEDA)	Increased (with 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%

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 6.4. 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.

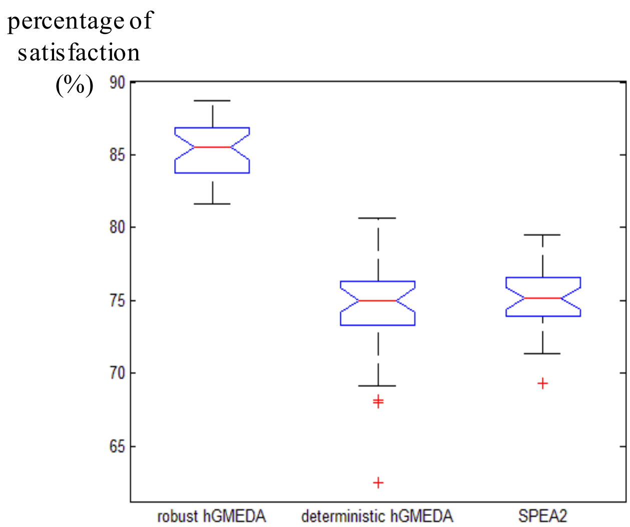


Fig. 6.4 Boxplot of percentage of satisfying chance constraint by robust hGMEDA, hGMEDA and SPEA2 on #n041_1

From the Fig. 6.4, it shows the results for problem #n041_1, due to the checking on the chance constraint, our robust method can achieve the percentage always higher than 80%, while other two methods only focus on two objectives without considering satisfying chance constraint, the percentage is lower than our proposal.

6.2.4 Discussion

From a viewpoint of modelling, a great number of real-life problems such as railway and airline scheduling problems, or course scheduling problems, can be modelled as variations of RCSPs. In the project of large building construction or chemical plant manufacturing, the most important feature is different skilled workers involved and acted as main resource which could affect the duration of completion time for each activity. One activity can be finished by several workers, and its completion time could be shorten with more workers engaged. As a result, this kind of flexibility of configuration for human power could be modeled as multi-mode. Meanwhile, the company policy on overtime could be viewed as recommendation level considering the health of workers, which is one goal to satisfy.

The effect of our system could be classified into two aspects: a) The solutions given by our algorithm can fully utilize the workers employed for this project, with a high reliability under the uncertainty environment. b) Based on the final results we received, some analysis could be further conducted. With the knowledge, we could know whether we have to employ or fire some workers for budget management. For example, if the project manager finds out the overtime threshold cannot be satisfied, then he/she can understand that more workers are needed to complete this project, else the project manager has to relax the threshold. The project manager notices that, in the satisfied solution, some kind of skilled-workers have lots of spare time, someone has to be fired to reduce payments. It’s a kind of trade-off between expend of resources and robustness we try to achieve.

The output of our proposal can provide feasible and satisfied schedules, giving the opportunity for the project manager to select one based on his own perspective, meanwhile, the solutions can also be analyzed to understand the situation of this project more clearly, especially on the utilization of resource for the budget management.

6.3 Summary

One robust scheduling method of robust hGMEDA is presented to deal with the uncertainties of activities in RCSP. The new robustness measures of time-based robustness and capacity-based robustness are introduced and a stochastic multi-objective optimization method based on scenario-based simulation is also proposed. The numerical experiment results demonstrated that our proposal can improve the time based robustness and capacity based robustness of RCSP solutions.

Chapter 7
Conclusion

7.1 Conclusion

It is well known that Resource Constrained Scheduling Problem (RCSP) with considering of resource utilization, makespan or budget management is important practically, however, to get an executable feasible scheduling solution is usually a complex NP-hard multi-objective combinatorial optimization, because, to find optimal scheduling of RCSP, it should be considered not only to minimize the makespan but also need to make capacity load balancing among the resources with satisfying the resource constraints.

Furthermore, in real-world resource constrained scheduling problems, parameters such as activity durations and resource requirements, originated from a great number of potential sources, and disruption of the original schedule, are seldom precisely known. These uncertainties incur high costs by resource idleness, high inventory, and missing deadlines. Therefore, dealing with uncertainty in a scheduling environment becomes another critical problem, which has significant impacts on productivity, customer satisfaction and profitability.

Conventionally, multi-objective evolutionary algorithms were developed based on GA with fitness assignment function of Pareto selection, however optimality and calculation efficiency of the conventional methods are not satisfiable because of complexity of RCSP and the method cannot handle the uncertainties mentioned above.

Estimation of Distribution Algorithm (EDA), as a class of population-based optimization algorithm, has been proved to get higher optimality than conventional Evolutionary Algorithm (EA), such as Genetic Algorithm (GA). The key idea of EDA is to build a constructed probabilistic model of the distribution of good solutions and guides further search behavior based on the model. Furthermore, probabilistic graphical models (PGMs) are used to represent the interaction behavior among the discrete decision variables, to improve the learning ability of the probabilistic model and to improve performance of EDA. Markov network based EDA (MEDA) was proposed where the Markov network is used as a PGM to model the stochastic interrelation among decision variables with the assumption of neighborhood relations.

Firstly, in this study, we enhance MEDA for multi-objective optimization to solve RCSP of multi-objective scheduling problems and propose multi-objective Markov network based EDA (MMEDA) to find Pareto optimal solution set by introducing new fitness assignment functions. Two-stage architecture of hybridizing GA and MMEDA (hGMEDA) is also proposed to improve the calculation efficiency of MMEDA.

Secondly, in order to deal with these uncertainties, a multi-phase robust scheduling method (robust hGMEDA) based on hGMEDA is proposed for robust scheduling. The two measures of time-based robustness and capacity-based robustness are introduced and a stochastic robust multi-objective optimization method by using scenario-based simulation is proposed. Applicability and effectiveness of the proposed methods are demonstrated through applications of resource constrained project scheduling problems.

Chapter 1 introduces the background, objective of our research and outline of the dissertation.

Chapter 2 gives a review of the conventional meta-heuristic algorithms proposed for solving RCSP, especially Estimation of Distribution Algorithm and its extension of PGMs based on EDA. Furthermore, some conventional multi-objective evolutionary algorithms and robust scheduling approaches are presented briefly.

Chapter 3 makes the illustrations on the idea and method that enhance MEDA for multi-objective optimization and also can improve its calculation efficiency for searching Pareto solution set.

Firstly, multi-objective MEDA (MMEDA) is proposed where novel fitness function is introduced on MEDA to find Pareto solution set. Two kinds of fitness assignment functions are combined to improve calculation time and diversity of Pareto solutions compared with conventional multi-objective evolutionary algorithms. And the heuristic method including two type local search are proposed to empower the conventional MEDA by improving the quality of candidate solutions.

Secondly, in order to further improve the calculation efficiency of proposed MMEDA, the algorithm hybrid GA and MMEDA (hGMEDA) is developed to solve resource constrained scheduling problems. Inspired by the cooperative co-evolutionary, in hGMEDA, a two-stage architecture based on sequential co-evolutionary paradigm is proposed. In the first stage, GA is employed to find feasible solutions for sequencing sub-problem without resource capacitated, because GA can provide more "random" solutions and higher diversity of solutions. In the second stage, based on the partial solutions given by stage-1, MMEDA is adopted to find optimal resource allocation and calculate the Pareto optimal solution set by using Markov network model of the stochastic interrelation between resources and activities.

Chapter 4 demonstrates our proposal of MMEDA and hGMEDA with the application of RCSP. In this chapter, a multi-mode resource constrained project scheduling problem (MRCPSP), which is a typical application of multi-objective RCSP, is solved by the scheduling method based on our proposed algorithm of hGMEDA, and the performance of our proposal is demonstrated with comparative results on the optimality and diversity of Pareto scheduling solutions by comparing with two typical and popular multi-objective evolutionary methods of NSGA-II and SPEA2. Five cases of MRCPSP which have different activity network structures with 22 activities and 3-mode constraints are solved by the proposed method and the experimental results demonstrate that our proposal can improve about 17.00%, 22.48% on coverage, 7.59%, 10.28% on generational distance and 12.20%, 10.01% on spacing averagely, compared with NSGA-II and

SPEA2, respectively. Calculation time of proposed method of hGMEDA is also compared with the conventional methods and MMEDA, and hGMEDA reduces calculation time about 18.4%, 25.5% and 10.9% for NSGA-II, SPEA2 and MMEDA respectively.

Chapter 5 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.

Chapter 6 demonstrates our proposal of robust hGMEDA with the application of one scheduling problem under uncertainty. In this chapter, a typical application of MRCPSP under duration uncertainty is studied. Several experiments are conducted on the benchmark problems of MRCPSP with duration uncertainty, and advantage of proposed robust hGMEDA is demonstrated by comparing robust measures between schedule solutions generated by robust hGMEDA and other two deterministic scheduling methods of hGMEDA and SPEA2 under the same condition. The numerical results show that our proposal robust hGMEDA provides the expected makespan larger than other two methods of 3.17% and 3.61%, but decreases the variance of makespan about 9.39% and 12.37% averagely, compared with deterministic hGMEDA and SPEA2. And robust hGMEDA improves the percentage of satisfying the threshold on capacity based robustness about 11.5%, 10.2%, compared with the solutions given by the scheduling methods based on hGMEDA and SPEA2, respectively.

7.2 Future work

For algorithm of MMEDA and hGMEDA, there are many parameters to set: EDA related, Markov-network-related, multi-objective-related, robust-optimization-related. In our future work, some researches on parameter tuning could be conducted to increase the accuracy of our approaches.

Secondly, it is one research direction to improve the performance by using other types of structure estimating algorithms or sampling methods. For example, if we can find the cause-effect relationship among variables in scheduling problems, Bayesian network based approach may become more convincing due to its directed structure, which is stronger relation than neighborhood.

Thirdly, decision support system by analyzing the Pareto solutions could be another research direction in future.

References

[1] X. Hao, H. W. Lin, and T. Murata, "Application of negotiable evolutionary algorithm in flexible manufacturing planning and scheduling," in *Industrial Engineering and Engineering Management (IE&EM), 2010 IEEE 17Th International Conference on*, 2010, pp. 496–500.

[2] M. Nawaz, E. E. Enscore, and I. Ham, "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem," *Omega*, vol. 11, no. 1, pp. 91–95, 1983.

[3] D. Applegate and W. Cook, "A computational study of the job-shop scheduling problem," *ORSA J. Comput.*, vol. 3, no. 2, pp. 149–156, 1991.

[4] F. Pezzella, G. Morganti, and G. Ciaschetti, "A genetic algorithm for the flexible job-shop scheduling problem," *Comput. Oper. Res.*, vol. 35, no. 10, pp. 3202–3212, 2008.

[5] W. Shen and D. H. Norrie, "Agent-based systems for intelligent manufacturing: a state-of-the-art survey," *Knowl. Inf. Syst.*, vol. 1, no. 2, pp. 129–156, 1999.

[6] S. Hartmann and D. Briskorn, "A survey of variants and extensions of the resource-constrained project scheduling problem," *Eur. J. Oper. Res.*, vol. 207, no. 1, pp. 1–14, 2010.

[7] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. R. Kan, "Optimization and approximation in deterministic sequencing and scheduling: a survey," *Ann. Discrete Math.*, vol. 5, pp. 287–326, 1979.

[8] W. Herroelen and R. Leus, "Project scheduling under uncertainty: Survey and research potentials," *Eur. J. Oper. Res.*, vol. 165, no. 2, pp. 289–306, 2005.

[9] T. J. R. Johnson, "An algorithm for the resource constrained project scheduling problem," Thesis, Massachusetts Institute of Technology, 1967.

[10] E. Demeulemeester and W. Herroelen, "A branch-and-bound procedure for the multiple resource-constrained project scheduling problem," *Manag. Sci.*, vol. 38, no. 12, pp. 1803–1818, 1992.

[11] E. L. Demeulemeester and W. S. Herroelen, "New benchmark results for the resource-constrained project scheduling problem," *Manag. Sci.*, vol. 43, no. 11, pp. 1485–1492, 1997.

[12] J. P. Stinson, E. W. Davis, and B. M. Khumawala, "Multiple resource–constrained scheduling using branch and bound," *AIIE Trans.*, vol. 10, no. 3, pp. 252–259, 1978.

[13] A. Mingozzi, V. Maniezzo, S. Ricciardelli, and L. Bianco, "An Exact Algorithm for the Resource-Constrained Project Scheduling Problem Based on a New Mathematical Formulation," *Manag. Sci.*, vol. 44, no. 5, pp. 714–729, Spring 1998.

[14] M. Bartusch, R. H. Möhring, and F. J. Radermacher, "Scheduling project networks with

resource constraints and time windows," *Ann. Oper. Res.*, vol. 16, no. 1, pp. 199–240, 1988.

[15] B. De Reyck and others, "A branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations," *Eur. J. Oper. Res.*, vol. 111, no. 1, pp. 152–174, 1998.

[16] J. E. Kelley, "The critical-path method: Resources planning and scheduling," *Ind. Sched.*, vol. 13, pp. 347–365, 1963.

[17] R. Alvarez-Valdes and J. M. Tamarit, "Heuristic algorithms for resource-constrained project scheduling: A review and an empirical analysis," *Adv. Proj. Sched.*, vol. 134, 1989.

[18] O. Oguz and H. Bala, "A comparative study of computational procedures for the resource constrained project scheduling problem," *Eur. J. Oper. Res.*, vol. 72, no. 2, pp. 406–416, 1994.

[19] R. Kolisch and S. Hartmann, "Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis," in *Project scheduling*, Springer, 1999, pp. 147–178.

[20] C. E. Bell and J. Han, "A new heuristic solution method in resource-constrained project scheduling," *Nav. Res. Logist. NRL*, vol. 38, no. 3, pp. 315–331, 1991.

[21] S. Hartmann, "A competitive genetic algorithm for resource-constrained project scheduling," *Nav. Res. Logist. NRL*, vol. 45, no. 7, pp. 733–750, 1998.

[22] K. Bouleimen and H. Lecocq, "A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version," *Eur. J. Oper. Res.*, vol. 149, no. 2, pp. 268–281, 2003.

[23] K. Nonobe and T. Ibaraki, "Formulation and tabu search algorithm for the resource constrained project scheduling problem," in *Essays and surveys in metaheuristics*, Springer, 2002, pp. 557–588.

[24] D. Merkle, M. Middendorf, and H. Schmeck, "Ant colony optimization for resource-constrained project scheduling," *IEEE Trans. Evol. Comput.*, vol. 6, no. 4, pp. 333–346, 2002.

[25] B. Jarboui, N. Damak, P. Siarry, and A. Rebai, "A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems," *Appl. Math. Comput.*, vol. 195, no. 1, pp. 299–308, 2008.

[26] L. Wang and C. Fang, "An effective estimation of distribution algorithm for the multi-mode resource-constrained project scheduling problem," *Comput. Oper. Res.*, vol. 39, no. 2, pp. 449–460, 2012.

[27] N. Damak, B. Jarboui, P. Siarry, and T. Loukil, "Differential evolution for solving multi-mode resource-constrained project scheduling problems," *Comput. Oper. Res.*, vol. 36, no. 9, pp. 2653–2659, 2009.

[28] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[29] P. Larrañaga, H. Karshenas, C. Bielza, and R. Santana, "A review on probabilistic graphical models in evolutionary computation," *J. Heuristics*, vol. 18, no. 5, pp. 795–819, 2012.

[30] X. Hao, J. Tian, H. W. Lin, and T. Murata, "An Effective Markov Random Fields based Estimation of Distribution Algorithm and Scheduling of Flexible Job Shop Problem," *IEEJ Trans. Electron. Inf. Syst.*, vol. 134, no. 6, pp. 796–805, 2014.

[31] P. Brucker, A. Drexl, R. Möhring, K. Neumann, and E. Pesch, "Resource-constrained project scheduling: Notation, classification, models, and methods," *Eur. J. Oper. Res.*, vol. 112, no. 1, pp. 3–41, 1999.

[32] K. Neumann and J. Zimmermann, "Methods for resource-constrained project scheduling with regular and nonregular objective functions and schedule-dependent time windows," in *Project Scheduling*, Springer, 1999, pp. 261–287.

[33] S. Shakya, R. Santana, and J. A. Lozano, "A markovianity based optimisation algorithm," *Genet. Program. Evolvable Mach.*, vol. 13, no. 2, pp. 159–195, 2012.

[34] W. Herroelen and R. Leus, "Robust and reactive project scheduling: a review and classification of procedures," *Int. J. Prod. Res.*, vol. 42, no. 8, pp. 1599–1620, 15 2004.

[35] G. E. Vieira, J. W. Herrmann, and E. Lin, "Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods," *J. Sched.*, vol. 6, no. 1, pp. 39–62, Jan. 2003.

[36] N. Mladenović and P. Hansen, "Variable neighborhood search," *Comput. Oper. Res.*, vol. 24, no. 11, pp. 1097–1100, 1997.

[37] T. Back, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[38] J. H. Holland, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, vol. viii. Oxford, England: U Michigan Press, 1975.

[39] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[40] F. Glover and M. Laguna, "Tabu Search," in *Handbook of Combinatorial Optimization*, D.-Z. Du and P. M. Pardalos, Eds. Springer US, 1998, pp. 2093–2229.

[41] J. Kennedy, "Particle Swarm Optimization," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Springer US, 2011, pp. 760–766.

[42] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Trans. Syst. Man Cybern.*, vol. 16, no. 1, pp. 122–128, 1986.

[43] H. Mühlenbein and G. Paass, "From recombination of genes to the estimation of distributions I. Binary parameters," *Parallel Probl. Solving Nature—PPSN IV*, pp. 178–187, 1996.

[44] P. Larrañaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for*

evolutionary computation, vol. 2. Springer Science & Business Media, 2002.

[45] S. Baluja, "Population-Based Incremental Learning. A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning," Jun. 1994.

[46] H. Mühlenbein and G. Paaß, "From recombination of genes to the estimation of distributions I. Binary parameters," in *Parallel Problem Solving from Nature — PPSN IV*, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds. Springer Berlin Heidelberg, 1996, pp. 178–187.

[47] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 287–297, Nov. 1999.

[48] C. González, J. A. Lozano, and P. Larrañaga, "Mathematical modelling of UMDAc algorithm with tournament selection. Behaviour on linear and quadratic functions," *Int. J. Approx. Reason.*, vol. 31, no. 3, pp. 313–340, Nov. 2002.

[49] M. Pelikan and H. Muehlenbein, "The Bivariate Marginal Distribution Algorithm," in *Advances in Soft Computing*, R. Roy, T. Furuhashi, and P. K. Chawdhry, Eds. Springer London, 1999, pp. 521–535.

[50] S. Baluja and S. Davies, "Combining Multiple Optimization Runs with Optimal Dependency Trees," Jun. 1997.

[51] J. S. De Bonet, C. L. Isbell, P. Viola, and others, "MIMIC: Finding optima by estimating probability densities," *Adv. Neural Inf. Process. Syst.*, pp. 424–430, 1997.

[52] H. M uhlenbeinH, T. Mahnig, and G.-F. Informationstechnik, "Convergence theory and applications of the factorized distribution algorithm," *J. Comput. Inf. Theory*, vol. 7, no. 1, pp. 19–32, 1999.

[53] R. Etxeberria and P. Larranaga, "Global optimization using Bayesian networks," in *Second Symposium on Artificial Intelligence (CIMAF-99)*, 1999, pp. 332–339.

[54] M. Pelikan, "Hierarchical Bayesian Optimization Algorithm," in *Hierarchical Bayesian Optimization Algorithm*, Springer Berlin Heidelberg, 2005, pp. 105–129.

[55] R. Santana, "A Markov Network Based Factorized Distribution Algorithm for Optimization," in *Machine Learning: ECML 2003*, N. Lavrač, D. Gamberger, H. Blockeel, and L. Todorovski, Eds. Springer Berlin Heidelberg, 2003, pp. 337–348.

[56] S. K. Shakya, "DEUM: A framework for an Estimation of Distribution Algorithm based on Markov Random Fields," The Robert Gordon University, 2006.

[57] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 2014.

[58] R. Kindermann, J. L. Snell, and others, *Markov random fields and their applications*, vol. 1. American Mathematical Society Providence, RI, 1980.

[59] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie, "Dependency

networks for inference, collaborative filtering, and data visualization," *J. Mach. Learn. Res.*, vol. 1, pp. 49–75, 2001.

[60] M. Frydenberg, "The chain graph Markov property," *Scand. J. Stat.*, pp. 333–353, 1990.

[61] P. Larrañaga, H. Karshenas, C. Bielza, and R. Santana, "A review on probabilistic graphical models in evolutionary computation," *J. Heuristics*, vol. 18, no. 5, pp. 795–819, Aug. 2012.

[62] X. Hao, L. Lin, M. Gen, and C.-F. Chien, "An effective Markov network based EDA for flexible job shop scheduling problems under uncertainty," in *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*, 2014, pp. 131–136.

[63] X.-W. Chen, G. Anantha, and X. Lin, "Improving Bayesian network structure learning with mutual information-based node ordering in the K2 algorithm," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 5, pp. 628–640, 2008.

[64] X. Hao, H. W. Lin, X. Chen, and T. Murata, "Cooperative bayesian optimization algorithm: a novel approach to multiple resources scheduling problem," *IEEJ Trans. Electron. Inf. Syst.*, vol. 132, no. 12, pp. 2007–2018, 2012.

[65] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliab. Eng. Syst. Saf.*, vol. 91, no. 9, pp. 992–1007, 2006.

[66] W. Zhang, "Research on scheduling problems in machining and assembly processes using multiobjective evolutionary algorithms," 2011.

[67] D. M. Jaeggi, G. T. Parks, T. Kipouros, and P. J. Clarkson, "The development of a multi-objective Tabu Search algorithm for continuous optimisation problems," *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1192–1212, 2008.

[68] P. Czyżżak and A. Jaszkiewicz, "Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization," *J. Multi-Criteria Decis. Anal.*, vol. 7, no. 1, pp. 34–47, 1998.

[69] J. E. Fieldsend and S. Singh, "A multi-objective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence," 2002.

[70] N. Khan, D. E. Goldberg, and M. Pelikan, "Multi-objective Bayesian optimization algorithm," in *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, 2002, pp. 684–684.

[71] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen, and others, *Evolutionary algorithms for solving multi-objective problems*, vol. 5. Springer, 2007.

[72] A. Ben-Tal and A. Nemirovski, "Robust optimization–methodology and applications," *Math. Program.*, vol. 92, no. 3, pp. 453–480, 2002.

[73] A. L. Soyster, "Technical note—convex programming with set-inclusive constraints and applications to inexact linear programming," *Oper. Res.*, vol. 21, no. 5, pp. 1154–1157, 1973.

[74] M. Sniedovich, "Wald's maximin model: a treasure in disguise!," *J. Risk Finance*, vol. 9, no.

3, pp. 287–291, 2008.

[75] L. J. Savage, "The theory of statistical decision," *J. Am. Stat. Assoc.*, vol. 46, no. 253, pp. 55–67, 1951.

[76] P. Kouvelis and G. Yu, *Robust discrete optimization and its applications*, vol. 14. Springer Science & Business Media, 2013.

[77] A. J. Davenport and J. C. Beck, "A survey of techniques for scheduling with uncertainty," *Unpubl. Manuscr. Available Httptidel Mie Utoronto Capublications Php*, 2000.

[78] S. Ghosh, "Guaranteeing fault tolerance through scheduling in real-time systems," Citeseer, 1996.

[79] H. Gao, M. S. Fox, W.-Y. Chiang, and S. Hikita, "Building robust schedules—an empirical study of single machine scheduling with uncertainty," *Soumis À Manag. Sci.*, 1995.

[80] S. V. Mehta, "Predictable scheduling of a single machine subject to breakdowns," *Int. J. Comput. Integr. Manuf.*, vol. 12, no. 1, pp. 15–38, 1999.

[81] S. V. Mehta and R. M. Uzsoy, "Predictable scheduling of a job shop subject to breakdowns," *IEEE Trans. Robot. Autom.*, vol. 14, no. 3, pp. 365–378, 1998.

[82] A. Davenport, C. Gefflot, and C. Beck, "Slack-based techniques for robust schedules," in *Sixth European Conference on Planning*, 2014.

[83] V. Jorge Leon, S. David Wu, and R. H. Storer, "Robustness measures and robust scheduling for job shops," *IIE Trans.*, vol. 26, no. 5, pp. 32–43, 1994.

[84] R. L. Daniels and P. Kouvelis, "Robust scheduling to hedge against processing time uncertainty in single-stage production," *Manag. Sci.*, vol. 41, no. 2, pp. 363–376, 1995.

[85] M. T. Jensen and others, *Robust and flexible scheduling with evolutionary computation*. BRICS, 2001.

[86] W. Herroelen and R. Leus, "The construction of stable project baseline schedules," *Eur. J. Oper. Res.*, vol. 156, no. 3, pp. 550–565, 2004.

[87] L. Wang, S. Wang, Y. Xu, G. Zhou, and M. Liu, "A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem," *Comput. Ind. Eng.*, vol. 62, no. 4, pp. 917–926, May 2012.

[88] K. Deb, K. Sindhya, and J. Hakanen, "Multi-objective optimization," in *Decision Sciences: Theory and Practice*, CRC Press, 2016, pp. 145–184.

[89] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceeding of the First International Conference on Genetic Algorithms and Their Applications, 1985*, 1985.

[90] W. Westera, "Decathlon, towards a balanced and sustainable performance assessment method," 2006.

[91] J. K. Lenstra, *Local search in combinatorial optimization*. Princeton University Press, 2003.

[92] P. Hansen and N. Mladenović, "Variable Neighborhood Search," in *Search Methodologies*, E. K. Burke and G. Kendall, Eds. Springer US, 2005, pp. 211–238.

[93] J. Tian, X. Hao, and T. Murata, "Markov Network based Multi-objective EDA and its Application for Resource Constrained Project Scheduling," *IEEJ Trans. Electron. Inf. Syst.*, vol. 136, no. 3, pp. 290–298, 2016.

[94] W. Michiels, E. Aarts, and J. Korst, *Theoretical aspects of local search*. Springer Science & Business Media, 2007.

[95] X.-C. Hao, J.-Z. Wu, C.-F. Chien, and M. Gen, "The cooperative estimation of distribution algorithm: a novel approach for semiconductor final test scheduling problems," *J. Intell. Manuf.*, vol. 25, no. 5, pp. 867–879, 2014.

[96] R. Kolisch and A. Sprecher, "PSPLIB-a project scheduling problem library: OR software-ORSEP operations research software exchange program," *Eur. J. Oper. Res.*, vol. 96, no. 1, pp. 205–216, 1997.

[97] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[98] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[99] D. A. Van Veldhuizen, "Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations," in *Evolutionary Computation*, 1999.

[100] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," Massachusetts Institute of Technology, 1995.

[101] S. Goren and I. Sabuncuoglu, "Robustness and stability measures for scheduling: single-machine environment," *IIE Trans.*, vol. 40, no. 1, pp. 66–83, 2008.

[102] S. L. Janak, X. Lin, and C. A. Floudas, "A new robust optimization approach for scheduling under uncertainty: II. Uncertainty with known probability distribution," *Comput. Chem. Eng.*, vol. 31, no. 3, pp. 171–195, 19 2007.

[103] Y. Wang, L. Lin, M. Gen, L. Sun, and H. Kawakami, "A Co-cooperative Evolutionary Algorithm for Flexible Scheduling Problem under Uncertainty," *Procedia Comput. Sci.*, vol. 61, pp. 515–520, 2015.

[104] G. Igelmund and F. J. Radermacher, "Preselective strategies for the optimization of stochastic project networks under resource constraints," *Networks*, vol. 13, no. 1, pp. 1–28, 1983.

[105] D. Golenko-Ginzburg and A. Gonik, "Stochastic network project scheduling with non-consumable limited resources," *Int. J. Prod. Econ.*, vol. 48, no. 1, pp. 29–37, 1997.

[106] M. A. Al-Fawzan and M. Haouari, "A bi-objective model for robust resource-constrained

project scheduling," *Int. J. Prod. Econ.*, vol. 96, no. 2, pp. 175–187, 2005.
[107] C. Artigues, R. Leus, and F. T. Nobibon, "Robust optimization for resource-constrained project scheduling with uncertain activity durations," *Flex. Serv. Manuf. J.*, vol. 25, no. 1–2, pp. 175–205, 2013.

Research Achievements

Journal Paper

[1] Jing Tian, Xinchang Hao, Tomohiro Murata, “Robust Optimization Method based on hybridization of GA and MMEDA for Resource Constraint Project Scheduling with Uncertainty”, pp.957-966, IEEJ Transactions on Electronics, Information and Systems, Vol. 137 (2017) No. 7, 2017.

[2] Jing Tian, Xinchang Hao, Tomohiro Murata, “Markov Network based Multi-objective EDA and its Application for Resource Constrained Project Scheduling”, pp.290-298, IEEJ Transactions on Electronics, Information and Systems, Vol. 136 (2016) No. 3, 2016.

[3] Xinchang Hao, Jing Tian, Hao Wen Lin, Tomohiro Murata, “An Effective Markov Random Fields based Estimation of Distribution Algorithm and Scheduling of Flexible Job Shop Problem”, pp.796-805, IEEJ Transactions on Electronics, Information and Systems, Vol. 134 (2014) No. 6, 2014.

International Conference Paper

[1] Jing Tian, Tomohiro Murata, “Robust Scheduling for Resource Constraint Scheduling Problem by Two-stage GA and MMEDA”, pp.1042-1047, 5th IIAI International Congress on Advanced Applied Informatics, 2016.

[2] Jing Tian, Xinchang Hao, Tomohiro Murata, “Multi-objective EDA for Multi-mode Project Scheduling with Resource Constraint”, pp.7-12, The Fourth Asian Conference on Information Systems, 2015.

[3] Jing Tian, Xinchang Hao, Tomohiro Murata, “Markov Network based EDA and Robust Project Scheduling with Multi-mode Resource Constraint and Rework Uncertainty”, pp.363-370, The Third Asian Conference on Information Systems, 2014.

[4] Jing Tian, Xinchang Hao, Tomohiro Murata, “Robust Scheduling by Hybrid Method of Policy Tree and Estimation of Distribution Algorithm and Application to Resource Constrained Project Scheduling Problem with Rework Uncertainty”, pp.371-377, The Second Asian Conference on Information Systems, 2013.

[5] Xinchang Hao, Jing Tian, Hao Wen Lin, Tomohiro Murata, “Markov Random Fields based Estimation of Distribution Algorithm and Application of Sharable Resource Constrained Scheduling Problem”, pp.62-67, The Second Asian Conference on Information Systems, 2013.

[6] Tao Xu, Jing Tian, Tomohiro Murata, “Research on personalized recommendation in E-commerce service based on Data Mining”, pp.313-317, The International Multi Conference of Engineers and Computer Scientists 2013, 2013.

[7] Xinchang Hao, Haowen Lin, Jing Tian, Tomohiro Murata, “Cooperative Estimation of Distribution Algorithm for Flexible Job Shop Problem with Configurable Machines”, pp.239-244, The First Asian Conference on Information Systems, 2012.

[8] Jing Tian, Tomohiro Murata, “Noisy GA Resampling on Evolved Parameterized Policies for Stochastic Constraint Programming”, pp.1439-1442, The 2nd International Conference on Computer Science and Service System, 2012.

[9] Jing Tian, Tomohiro Murata, “Decision Support of Maintenance Strategy for Stable Service System based on Fuzzy Logic”, pp.117-122, The 4th Japan-China Joint Symposium on Information Systems, 2011.

[10] Tian Jing, Murata Tomohiro, “Multi-objective Flexible Job Shop Scheduling with Uncertain Processing Time and Machine Available Constraint based on Hybrid Optimization Approach”, pp.581-586, 2010 IEEE International Conference of Automation and Logistics, 2010.