Ann Oper Res
DOI 10.1007/s10479-008-0489-9

Generating artificial chromosomes with probability control in genetic algorithm for machine scheduling problems

Pei-Chann Chang · Shih-Hsin Chen · Chin-Yuan Fan · V. Mani

© Springer Science+Business Media, LLC 2008

Abstract In this paper, a novel genetic algorithm is developed by generating artificial chromosomes with probability control to solve the machine scheduling problems. Generating artificial chromosomes for Genetic Algorithm (ACGA) is closely related to Evolutionary Algorithms Based on Probabilistic Models (EAPM). The artificial chromosomes are generated by a probability model that extracts the gene information from current population. ACGA is considered as a hybrid algorithm because both the conventional genetic operators and a probability model are integrated. The ACGA proposed in this paper, further employs the "evaporation concept" applied in Ant Colony Optimization (ACO) to solve the permutation flowshop problem. The "evaporation concept" is used to reduce the effect of past experience and to explore new alternative solutions. In this paper, we propose three different methods for the probability of evaporation. This probability of evaporation is applied as soon as a job is assigned to a position in the permutation flowshop problem. Experimental results show that our ACGA with the evaporation concept gives better performance than some algorithms in the literature.

Keywords Evolutionary algorithm with probabilistic models · Single machine scheduling · Total deviations · Flowshop machine scheduling · Artificial chromosomes

P.-C. Chang (✉)
Department of Information Management, Yuan-Ze University, 135 Yuan Tung Road, Chung-Li, Taiwan, ROC 32026
e-mail: iepchang@saturn.yzu.edu.tw

S.-H. Chen
Department of Electronic Commerce Management, Nanhua University, 32, Chungkeng, Dalin, Chiayi, 62248, Taiwan, ROC
e-mail: shihhsin@mail.nhu.edu.tw

C.-Y. Fan
Department of Industrial Engineering and Management, Yuan-Ze University, Chung-Li, Taiwan, ROC
e-mail: s948906@mail.yzu.edu.tw

V. Mani
Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India
e-mail: mani@aero.iisc.ernet.in

Published online: 06 December 2008

1 Introduction

Genetic algorithm is basically an implicit probability model (Baluja and Davies 1998), which mainly preserves better chromosomes by selection; however, a better building block is not guaranteed for preserving better chromosomes. It is because gene is evaluated within a chromosome (Harik et al. 1999). There are some researchers, such as Glover and Kochenberger (1996), who argued that some gene structures are similar in some better chromosomes from the traveling salesman problem (TSP) of Lin and Kernighan (1973) which supports this idea. Their experimental result shown that there were 85% segments in any two local optimal solutions were identical. Therefore, among all the branches of genetic algorithms, Evolutionary Algorithms Based on Probabilistic Models (EAPM) fulfills the gap of genetic algorithm. EAPM is different from past genetic algorithm in extracting chromosomes structure from parents, which is the exact position of each gene and then to produce offspring. This concept stemmed from Ackley (1987) and Syswerda (1993) and then was developed by Baluja (1995), Baluja and Davies (1998), Harik et al. (1999), Muhlenbein and Paaß (1996), Baraglia et al. (2001), Larrañaga and Lozano (2002), Zhang et al. (2005), Rastegar and Hariri (2006), and Lozano (2006). These algorithms are classified in Estimation of Distribution Algorithm (EDA) by Zhang et al. (2005).

Based on the authors' previous research Chang et al. (2008b), an artificial chromosome Genetic Algorithm (ACGA) was proposed which is related to some EAPM methods. The artificial chromosomes are generated by probability model that extracts the gene information from current population. The main difference is the ACGA that can be treated as a hybrid algorithm because the conventional genetic operators and probability model are integrated into together. From the previous results in single machine scheduling problem which minimizes the earliness/tardiness penalty with distinct due date, ACGA is able to produce a very satisfactory results.

In this paper, ACGA is not only applied to solve the permutation flowshop problem but also employs an evaporation concept, which is widely used in Ant Colony Optimization (ACO), as a probability control tool. The pheromone evaporation mitigates past experience so that it increases the diversity and to prevent stagnation of searching process for latter-on constructed solutions (Corne et al. 1999). In addition, it is useful to prevent pheromone concentration in optimal path from being excessively (Sim and Sun 2003). Therefore, since none of the EAPM papers discusses this issue, this paper proposes three different methods to evaporate the probability as soon as a job is assigned to a position. The paper is organized as follows: Sect. 2 the literature survey reviewing the papers in single machine scheduling, flowshop scheduling and Evolutionary Algorithm with Probabilistic Models. Section 3 is the methodology which includes the ACGA algorithm and the evaporation methods. The experimental results are shown in Sect. 4 and the conclusion is drawn in Sect. 5.

2 Literature survey

2.1 The single machine scheduling problems with earliness and tardiness penalties

As a generalization of weighted tardiness scheduling, the problem is strongly NP-hard in Lenstra et al. (1975). The earlier works in this problem was due to Chang (1999), Chang and Lee (1992a, 1992b), Wu et al. (1993). Both exact algorithm and heuristic approaches have been proposed. Among the exact approaches, branch-and-bound algorithms were presented by Abdul-Razaq and Potts (1988), Li (1997), Liaw (1999). The lower bounding procedure of Abdul-Razaq and Potts (1988) was based on the sub-gradient optimization approach and the dynamic programming state-space relaxation technique, whereas Li (1997) and Liaw (1999) used Lagrangian relaxation and the multiplier adjustment method. Valente and Alves (2007) presented a branch-and-bound algorithm based on a decomposition of the problem into weighted earliness and weighted tardiness sub-problems. Two lower bound procedures were presented for each sub-problem, and the lower bound for the original problem was then simply the sum of the lower bounds for the two sub-problems. Later on, Akturk and Ozdemir (2001), Sourd and Kedad-Sidhoum (2003), Sourd and Kedad-Sidhoum (2007), Jouglet et al. (2008) developed various dominance rules to solve the problem. These rules were employed in branch-and-bound algorithm which enhances the fathoming procedure or to be combined with other meta-heuristic, such as Chang et al. (2009) Integrated the DPs with GA in the problem of earliness/tardiness scheduling problems.

Among the heuristics, Ow and Morton (1989) developed several dispatch rules and a filtered beam search procedure. In Valente and Alves (2005), they presented an additional dispatch rule and a greedy procedure, and also consider the use of dominance rules to further improve the schedule obtained by the heuristics. A neighborhood search algorithm was also presented by Li (1997). Belouadah et al. (1992) dealt with the similar problem with a different objective in minimizing the total weighted completion time. Apart from these algorithms, some metaheuristics were developed (Michalewicz et al. 1996; Lee et al. 1997; Dimopoulos and Zalzala 2000). For example, Chang et al. (2008a) developed a new algorithm, termed as Electromagnetism-Like algorithm, dealing with the single machine scheduling problems with the consideration of earliness/tardiness. Electromagnetism-like algorithm was originally proposed by Birbil and Fang (2003), which was able to solve continuous problem while Electromagnetism-like algorithm should corporate random key method to solve the sequential problems. Electromagnetism-like algorithm will diversify the inferior solutions while the genetic algorithm operator, i.e. the crossover, recombines better solutions. From the experiment results presented the hybrid algorithm is better than using the Electromagnetism-Like algorithm alone.

2.2 Flowshop scheduling problems

In the operations research literature, flowshop scheduling is one of the most well studied problems in the area of scheduling (Murata et al. 1996). A permutation flowshop scheduling problem (PFSP) concerns that n jobs are processed on m machines in the same order to meet one or some specified objectives. Baker (1974) summarized the assumptions of permutation flowshop scheduling problems. Therefore most of the research works emerged to develop effective heuristics and metaheuristics.

Framinan et al. (2004) reported a review and classification of the heuristics for permutation flowshop scheduling problems. Hejazi and Saghafian (2005) presented a complete survey of flowshop scheduling problems and contributions from 1954 to 2004. This survey concerned some exact methods, constructive heuristics, metaheuristics, and evolutionary approaches. This paper is a good reference for n/m/p/Cmax. Ruiz and Maroto (2005) provided a comprehensive review and evaluation of permutation flowshop heuristics. For example, they did extensive comparisons in the flowshop scheduling problems, including tabu search, simulated annealing, genetic algorithms, iterated local search, and hybrid techniques Through reading these review articles, it is apparent that heuristics developed for PFSPs have proposed a remarkable contribution.

Leaving the traditional structures behind, this work intends to improve the effectiveness and efficiency of genetic search by embedding more feedback information in the evolutionary process by incorporating probabilistic models. Next section presents the evolutionary algorithm with probabilistic models.

2.3 Evolutionary algorithm with probabilistic models

Genetic Algorithm can be treated as an implicit probabilistic model whereas there are some algorithms which are able to estimate unknown probability distribution. These algorithms are named evolutionary algorithm with probabilistic models (EAPM), probabilistic model-building genetic algorithms (PMBGAs), or estimation of distribution algorithms (EDAs) (Pelikan et al. 2002). EAPM is a brand new branch of GA, which learns useful population information from the promising solutions and then samples new offspring. The advantage of EAPM is the probabilistic model which enables the offspring representing this solution structure. EAPM has gained more and more attentions in the academic field so far.

Baluja (1995) and Baluja and Davies (1998) discussed the population-base incremental learning (PBIL) algorithm and combining optimizers with mutual information tree (COMIT) respectively. PBIL updates statistic information generation by generation, which is an extremely simple algorithm and there is no interdependent parameters. COMIT selects initial solutions intelligently and then combines the hill-climbing algorithm or PBIL into together. Later on, Muhlenbein and Paaß (1996) proposed a famous algorithm named Univariate Marginal Distribution Algorithm (UMDA). UMDA is similar to PBIL and cGA.

Harik et al. (1999) proposed compact genetic algorithm (cGA). The genetic vector is the corresponding probability distribution and each gene is generated independently. Compared cGA with simple GA, cGA requires $l * \log_2(n+1)$ bits and GA needs $l * n$. In addition, it differs from the PBIL algorithm because cGA takes the fix length of updating strategy of probability vector while PBIL. cGA provides a good stimulation for the design of genetic algorithm.

Later on, there is an important concept emphasized in guided mutation (Zhang et al. 2005). Although the probabilistic model extracted the parental distribution, which provided global information of evolutionary direction; however, the probabilistic model doesn't aware the local information. In order to conquer this problem, a proportion of genes are copied into the new chromosome and the rest of genes are selected by the probabilistic model. Therefore, the contribution of this research showed an important concept when we design an evolutionary algorithm with probabilistic models.

Above probabilistic models assumed there is no interaction between/among variables. When interactions exist, EAPM requires more complex model to solve it. For the pairwise interactions, MIMIC, Dependency Tree, and BMDA are proposed. To cover the multivariate interactions, Harik (1999) developed extended compact genetic algorithm (ECGA) which is the extended version of cGA. ECGA actually adopts the marginal product models (MPMs) rather than a vector. This model is able to make the exposition simpler, and enables the linkage of the variables. The other algorithm in this class is Bayesian optimization algorithm (BOA) in Pelikan et al. (1999). Finally, for extensive review of evolutionary algorithm based on probabilistic models, please refer to Larrañaga and Lozano (2002), Lozano (2006), and Pelikan et al. (2002).

This paper attempts to develop a novel approach to generate artificial chromosomes, which can be embedded in the original GA procedure to speed up the convergence procedure in solving the scheduling problems. The approach attempts to extract the superior gene information from the current population to generate new offspring. By injecting these new generated artificial chromosomes into the process, the convergence rate and solution quality of the GA searching procedure will be improved greatly.

3 Generating artificial chromosomes by mining gene structures

In tradition, the scheduling problems were solved by some exact algorithms, such as Branch and Bound algorithm, Dynamic programming, and Lagrangian relaxation. However, owing to the computational complexity, these methods could only solve small size problems. As a result, this research proposed an artificial chromosome genetic algorithm (ACGA) with probabilistic control to solve the single machine and flowshop scheduling problems. This algorithm is able to capture the parental distribution and to generate solutions by the probabilistic model. Furthermore, this proposed algorithm considers the issue of diversity preservation of the probabilistic models. Thus, this characteristic makes it different from other previous EAPM algorithms.

The primary procedure of ACGA is to collect gene information first and to use the gene information to generate artificial chromosomes. Before collecting the gene information, ACGA collects the chromosomes whose fitness is better by comparing the fitness value of each chromosome with average fitness value of current population. Thus, the average fitness is calculated. A detailed procedure of the ACGA algorithm is depicted in Fig. 1.

There are two parameters to be decided in this algorithm, which are *startingGen* and *interval*. The first parameter *startingGen* is to determine the starting time of generating artificial chromosomes. The main reason is that the probabilistic model should be only applied to generate better chromosomes when the searching process reaches a more stable state. As a result, the probability model is applied after some generations. Later on, artificial chromosomes are not generated in each generation because it takes more computational time since the proportional selection takes $O(n^2)$ time complexity for each solution. Consequently, *interval* controls the time interval of artificial chromosomes generated. A set of experiments for parameter configuration has been set up by Design-of-Experiment (DOE). DOE will examine the significance of each factor. According to these preliminary results, both factors have no significant difference. Therefore, the *startingGen* and *interval* are set to 500 and 50 in later experiments, respectively.

The following Sect. 3.1 explains the proposed algorithm in detail. First, a step by step procedure is applied to explain how to establish a probabilistic model. Then in Sect. 3.2, an instance is applied to explain how to generate an offspring by the probabilistic model.

3.1 Establishing a probabilistic model

Suppose a population has M strings $X^1, X^2, \dots, X^M$ at current generation t, which is denoted as Population(t). Then, X_{ij}^k is a binary variable in chromosome k, which is shown in (1).

$$X_{ij}^k = \begin{cases} 1 & \text{if job } i \text{ is assigned to position } j, \\ 0 & \text{otherwise,} \end{cases} \quad i = 1, \dots, n;\ j = 1, \dots, n \tag{1}$$

The fitness of these M chromosomes is evaluated and the gene information is collected from N best chromosomes where $N \le M$. The N chromosomes are set as $M/2$ in this research. The purpose of only selecting N best chromosomes from population is to prevent the quality of the probabilistic model from being down-graded by inferior chromosomes. Let $P_{ij}(t)$ be the probability of job i to show up at position j at current generation. Our probability model is similar to PBIL where the $P_{ij}(t)$ is updated as follows:

$$P_{ij}(t+1) = \frac{1}{N} \sum_{k=1}^{N} X_{ij}^k, \quad i = 1, \dots, n,\ j = 1, \dots, n \tag{2}$$

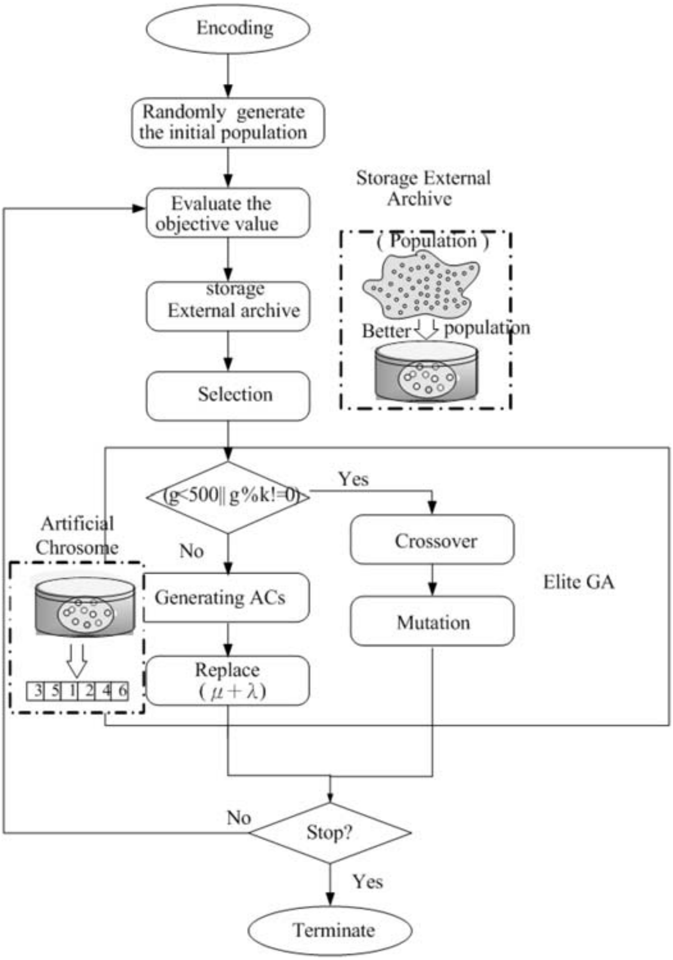

Fig. 1 The framework of the ACGA

For the probabilistic matrix of all jobs at different positions, they are written as the (3)

$$P(t+1) = \begin{pmatrix} P_{11}(t+1) & \cdots & P_{1n}(t+1) \\ \vdots & \ddots & \vdots \\ P_{n1}(t+1) & \dots & P_{nn}(t+1) \end{pmatrix} \tag{3}$$

To demonstrate the working theory of the artificial chromosome generation procedure, a 5-job problem is illustrated in Fig. 2. Suppose there are ten solutions (chromosomes) whose fitness is better than average fitness. Then, we accumulate the gene information from these ten chromosomes to form a dominance matrix. As shown in the left-hand side of Fig. 2, there are two job 1, two job 2, two job 3, one job 4, and three job 5 on position 1. Again, there are three job 1, one job 2, two job 3, three job 4, and one job 5 on position 2. The procedure will repeat for the rest of the position. The dominance matrix contains the gene information from better chromosomes and they are illustrated in the right-hand side of Fig. 2. Finally, the

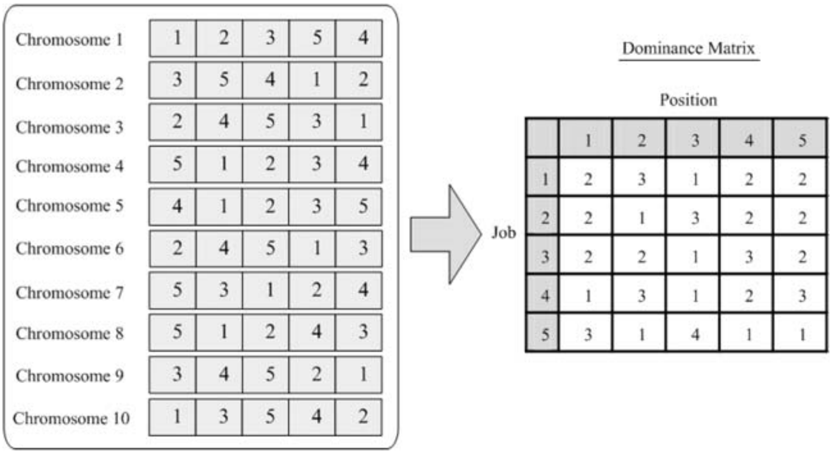

Fig. 2 To collect gene information and converted into a dominance matrix

value of each element is divided by N which is the total number of chromosomes selected. Consequently, the probabilistic matrix P is constructed.

3.2 Generating offsprings by the parental distribution

As soon as the probabilistic matrix P is built, jobs are assign onto each positions by proportional selection. Through this proportional selection, Zhang and Muhlenbein (2004) showed if the distribution of the new elements capture the parents well, global optimal will be obtained, and a factorized distribution algorithm converges globally under proportional selection. The assignment sequence for each position is assigned in random sequence, which will be able to diversify the artificial chromosomes. The assignment procedure is determined as follows:

S: A set of shuffled sequence which determines the sequence of each position is assigned a job.
Ω: is the set of un-arranged jobs.
J: The set of arranged jobs. J is empty in the beginning.
θ: A random probability is drawn from $U(0, 1)$.
i: A selected job by proportional selection.
k: The element index of the set S.

$S \leftarrow$ shuffled the job number $[1 \ldots n]$
$J \leftarrow \Phi$
while $k \neq \Phi$ **do**
    $\theta \leftarrow U(0, 1)$
    Select a job i satisfies $\theta \leq P_{ik} / \sum_{i \in \Omega} P(i, k)$
    $J(k) \leftarrow i$
    ProbabilityControl(i, k)
    $\Omega \leftarrow \Omega \backslash i$
    $S \leftarrow S \backslash k$
end while

Step 7 is the probability control, which stresses the diversification for the probabilistic models. This paper proposed three different methods proposed for probability control. Without using the probability control, it is a generic ACGA. The detail descriptions of probability control are shown in Sect. 3.3. Finally, since the time-complexity of the proportional selection is $O(n^2)$, it spends more time than using crossover operator. As a result, it is the reason why this paper hybridizes the probabilistic model with genetic operators that can avoid the excessive computational efforts.

3.3 Probability control

In this section, there are some proposed methods which discuss the diversification for the probabilistic models. As far as the diversification concerns, the idea is to create diversified artificial chromosomes by mitigating the probability of job i assigned to a position j by a proportional selection. The main reason of reducing the probability value is when there is a P_{ij} which very closes to 1, the proportional selection might always select this job to this position again. It may cause the problem of stagnation or jumping into local optimal. Hence, our idea is motivated by the pheromone control in Ant Colony Optimization (ACO) (Corne et al. 1999). The pheromone control methods are categorized into evaporation (Alves and Almeida 2007), aging (Schoonderwoerd et al. 1997), and limiting and smoothing pheromone (Stutzle et al. 2000). Sim and Sun (2003) mentioned that evaporation methods may not have the weaknesses belonged to aging, and limiting and smoothing pheromone. Consequently, the concept of evaporation is applied here. These three different evaporation methods are introduced as follows.

3.3.1 Constant evaporation

The first method comes up with a parameter α which exponentially decreases the probability of the job i at position j. Parameter α is a constant value, which is set to a small value (e.g., 0.05). In order to distinguish this method from the latter, it is called the constant evaporation. The equation is shown as follows:

$$P_{ij} = P_{ij} - P_{ij} * \alpha \tag{4}$$

3.3.2 Current best objective evaporation

The second method takes the advantage of the current best objective value among all solutions generated. Thus, $(1/\mathit{currentBestObj}) * \alpha$ constitutes a little increasing value than the $P_{ij} * (1 - \alpha)$. The setting of $\alpha = 0.05$ is the same as the constant evaporation. This method is called the best objective evaporation and it is shown in (5)

$$P_{ij} = P_{ij} * (1 - \alpha) + (1/\mathit{currentBestObj}) * \alpha \tag{5}$$

3.3.3 Max-Min evaporation

Finally, we substitute the current best objective value by the difference of maximum and minimum objective value among all chromosomes in the current population. Hence the method is called max-min range evaporation which is as shown in (6).

$$P_{ij} = P_{ij} * (1 - \alpha) + \left(\frac{1}{\max \mathit{Obj} - \min \mathit{Obj}} \right) * \alpha \tag{6}$$

3.4 Replacement strategy

After injecting artificial chromosomes into the population, we apply $\mu + \lambda$ replacement strategy, which combines previous parent population and the new generated artificial chromosomes into the gene pool. Then, we select better μ chromosomes from the combined population and consequently, better solutions are preserved to the next generation.

4 Experimental results

In this section, the performance of ACGA with probability control, i.e., the constant evaporation, current best objective evaporation, and Max-Min evaporation is compared with other algorithms published in the literature. Furthermore, these algorithms solved the single machine scheduling problems and flowshop scheduling problems, which were taken from Sourd and Kedad-Sidhoum (2003) and Reeves (1995), respectively. Each instance is replicated 30 times by each algorithm. Sections 4.1 and 4.2 demonstrate the experimental results of the single machine problems and flowshop scheduling problems, respectively.

4.1 Results of single machine problems

There are numerous data sets published in the literature for the single machine scheduling problems, including 20, 30, 40, 50, 60, and 90 jobs. Each data set of 20 jobs up to 50 jobs contains 49 instances (problems) whereas there are only 9 instances in the data set of 60 jobs and 90 jobs. We carried out our experiments on these total 214 instances. The stopping criterion is the number of examined solutions, which is 100,000 solutions. The parameters of GA include the crossover rate, mutation rate, and population size which are determined in our preliminary experiments. They are set up as 0.8, 0.5, and 100, respectively. ACGA with probability control will be compared with Genetic Algorithm with elitism (GA), Genetic Algorithm with Dominance Properties (GADP) in Chang et al. (2009), the original version of ACGA (Chang et al. 2008b), and the hybrid framework of ACGA with Dominance Properties (ACGADP). GADP applies a set of dominance properties to generate a good initial population in the beginning and it is able to enhance the exploration ability of Genetic Algorithm. As a result, ACGADP means the initial solutions are constructed by dominance properties and the rest of evolutionary process is conducted by ACGA.

In the experimental tests, there is no significant difference among these three evaporation methods, i.e., ACGA with the three evaporation methods. Since ACGA with the current best objective evaporation has a better result in average performance, it is compared with GA, GADP, ACGA, and ACGADP. Some selected results are shown at Table1 which shows the min, average, and max of the algorithms. However, there is still without enough information to determine which algorithm is statistically significant. Consequently, ANOVA (Analysis of Variance) and Duncan grouping test are employed to further distinguish the performance of the algorithms. The complete test results are available on our website.[1]

To distinguish the performance of the proposed algorithm from the other four algorithms in the literature, ANOVA was performed. As shown in Table 2, the first column source indicated the factors; DF is the degree of freedom; SS is sum of squares; F is the value of F test, and Pr is the probability of the statistic significance (Montgomery 2001). The

[1] http://ppc.iem.yzu.edu.tw/publication/sourceCodes/InjectionArtificialChromosomes/

Table 1 Selected results of single machine scheduling problems: Job 20, 30, 40, 50, 60, and 90

Instance	GA			GADP			ACGA			ACGADP			ACGA+Current Best		
	Min	Avg.	Max	Min	Avg.	Max	Min	Avg.	Max	Min	Avg.	Max	Min	Avg.	Max
sks252a	4052	4195	4508	3947	3947	3947	3947	3979	4067	3947	3947	3947	3947	3975	4067
sks255a	2388	2489	2787	2372	2372	2372	2372	2380	2388	2372	2373	2388	2372	2383	2388
sks258a	1184	1250	1371	1184	1242	1248	1184	1200	1248	1184	1193	1248	1184	1197	1248
sks352a	7588	8203	9063	7395	7395	7395	7392	7394	7395	7392	7394	7395	7392	7394	7395
sks355a	6202	6849	7693	6056	6068	6212	6056	6065	6193	6056	6058	6058	6056	6057	6058
sks358a	3104	3283	3787	3069	3074	3076	3069	3073	3076	3069	3073	3076	3069	3073	3076
sks452a	11804	12634	14053	11367	11367	11367	11367	11406	11581	11367	11367	11367	11367	11390	11528
sks455a	6573	7566	9435	6405	6405	6405	6405	6427	6666	6405	6405	6405	6405	6438	6666
sks458a	4424	5587	8331	4294	4303	4319	4294	4321	4391	4294	4300	4306	4294	4321	4391
sks552a	23491	24827	26241	22863	22863	22863	22863	22894	23148	22863	22863	22863	22863	22890	23013
sks555a	10877	12233	14626	10207	10243	10446	10187	10216	10299	10207	10209	10226	10187	10227	10299
sks558a	5776	7345	9358	5269	5298	5416	5269	5269	5269	5269	5269	5269	5269	5269	5269
sks622a	43930	45018	46017	43048	43048	43048	43048	43120	43479	43048	43048	43048	43048	43089	43369
sks625a	25563	26672	27957	25253	25253	25253	25229	25260	25307	25253	25253	25253	25229	25259	25453
sks628a	17463	18431	20707	17047	17057	17123	17047	17059	17162	17047	17055	17123	17047	17055	17172
sks652a	31292	33022	35426	30801	30801	30801	30801	30871	31080	30801	30801	30801	30801	30890	31080
sks655a	17409	19137	23546	16158	16158	16158	16158	16218	16635	16158	16158	16158	16158	16228	16612
sks658a	9948	12715	18469	9623	9623	9626	9623	9655	9724	9623	9623	9626	9623	9643	9705
sks682a	38930	39722	41137	38836	38940	39109	38714	38749	38863	38744	38923	39109	38714	38756	38852
sks685a	38736	39744	41345	38084	38096	38166	38084	38103	38166	38084	38090	38166	38084	38101	38166
sks688a	34456	35826	37229	33551	33654	33665	33551	33639	33665	33551	33646	33665	33551	33624	33665
sks922a	91516	93966	96966	88994	89606	90514	88841	88894	89067	88866	89315	90493	88841	88887	89078
sks925a	74327	76438	79979	72038	72045	72141	72038	72065	72114	72038	72055	72120	72038	72069	72286
sks928a	38676	41879	49091	33825	33992	34159	33830	33973	34138	33903	34019	34195	33835	33948	34054
sks952a	73718	76863	79847	68150	68188	68441	68150	68288	68674	68150	68179	68253	68150	68251	68408
sks955a	35647	40444	45820	30660	30664	30700	30582	30683	31312	30590	30663	30697	30582	30641	30785
sks958a	23553	30662	39623	19945	19972	20028	19950	20025	20201	19954	20012	20108	19957	20013	20142
sks982a	100092	102345	105794	98613	99041	99349	98613	98644	98832	98613	99081	99349	98613	98644	98834
sks985a	82254	84966	87173	78296	78442	78532	78296	78414	78520	78296	78445	78557	78296	78423	78502
sks988a	88094	91422	96318	81984	81993	82097	81984	82002	82053	81984	81995	82045	81984	81999	82056

Table 2 ANOVA result of the Method Comparisons in single machine scheduling problems

Source	DF	SS	Mean square	F value	Pr > F
Instance	213.00	6.85E+12	3.22E+10	327224.00	<0.0001
Method	4.00	6.98E+09	1.75E+09	17757.60	<0.0001
Instance*	852.00	1.24E+10	14607067.82	148.64	<0.0001
Error	31030.00	3.05E+09	98270.56		
Corrected total	32099.00	6.87E+12			

Table 3 The Duncan grouping result for the five algorithms

Duncan grouping		Mean	*N*	Method
	A	13982.894	6420	GA
	B	12827.096	6420	GADP
	B			
C	B	12816.471	6420	ACGADP
C				
C		12813.276	6420	ACGA
C				
C		12811.868	6420	ACGA+Best Objective Evaporation

probability of the statistic significance shows that there exists significant difference among these methods. As a result, Duncan grouping results is further applied in Table 3. When there are two factors share the same alphabet, it means they are in the same group and there is no significant difference for methods in the same group. On the other hand, there is statistical difference between these two methods since they are in different groups. In our case, there is no much difference among ACGA with best objective evaporation, ACGA and ACGADP. However, three of them are better than GADP and GA. To conclude the results, ACGA with evaporation consideration or ACGA is the best algorithm, ACGADP and GADP are ranked second, and GA is the worst.

4.2 Results of flowshop scheduling problems

The instances of this flowshop scheduling problem are available from OR-Library.[2] From the results of the three variants of ACGA, there is no significant difference among these three evaporation methods. However, ACGA with the Max-Min evaporation method has the lowest average in average. Thus, ACGA with the Max-Min evaporation method is compared with GA and the original version of ACGA. The results are shown in Table 4.

ANOVA is employed to evaluate the statistical significance of these three algorithms and the results are shown in Table 5. The test shows that the factor, i.e., these three methods, is significant. Therefore, Duncan grouping method is further applied to distinguish the grouping of these algorithms, which is demonstrated in Table 6. The Duncan grouping presents three of them are in different group, which means the ACGA with the Max-Min evaporation is the best, ACGA is the intermediate, and GA is the worst.

[2] http://people.brunel.ac.uk/~mastjjb/jeb/info.html

Table 4 Reeves flowshop results

Instance	n, m	opt	GA			ACGA			ACGA+Max-Min		
			Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
rec01	20, 5	1247	1249	1252.20	1280	1247	1248.90	1249	1247	1248.90	1249
rec03	20, 5	1109	1109	1112.30	1117	1109	1110.50	1111	1109	1110.90	1117
rec05	20, 5	1242	1245	1251.20	1273	1245	1245	1245	1245	1245.80	1269
rec07	20, 10	1566	1584	1586.80	1626	1566	1576.90	1584	1566	1577.30	1584
rec09	20, 10	1537	1538	1568.70	1589	1537	1554.80	1574	1537	1557.80	1574
rec11	20, 10	1431	1431	1445.40	1476	1431	1440.60	1473	1431	1439.30	1469
rec13	20, 15	1930	1936	1959.80	1981	1935	1949.70	1968	1930	1949.20	1961
rec15	20, 15	1950	1961	1980.50	2011	1950	1969.10	1993	1951	1969.90	2018
rec17	20, 15	1902	1919	1957.40	2009	1902	1938.30	1974	1911	1936.10	1960
rec19	30, 10	2093	2124	2152.30	2197	2111	2132	2171	2099	2133.60	2162
rec21	30, 10	2017	2050	2063.30	2103	2046	2051.80	2096	2046	2051.80	2084
rec23	30, 10	2011	2041	2070.60	2104	2023	2055.40	2086	2021	2048	2074
rec25	30, 15	2513	2554	2595.50	2660	2556	2588.50	2637	2532	2586.10	2633
rec27	30, 15	2373	2402	2438.90	2484	2396	2426.20	2523	2397	2421.60	2446
rec29	30, 15	2287	2324	2367.20	2423	2298	2363.20	2414	2309	2347.40	2383
rec31	50, 10	3045	3124	3185.30	3264	3121	3150.80	3239	3106	3153	3255
rec33	50, 10	3114	3140	3180.90	3231	3139	3153	3219	3140	3156.10	3201
rec35	50, 10	3277	3277	3308.90	3370	3277	3282.20	3308	3277	3280.90	3288
rec37	75, 20	4951	5210	5274.70	5351	5156	5245.60	5353	5169	5243	5319
rec39	75, 20	5087	5266	5338.40	5442	5248	5337.60	5453	5247	5315.50	5379
rec41	75, 20	4960	5215	5289.40	5356	5175	5270.70	5349	5168	5268.80	5370

Table 5 ANOVA result of the Method Comparisons

Source	DF	Type I SS	Mean square	F value	Pr > F
Instance	20	3072650812.00	153632541.00	332227.00	<.0001
Method	2	95664.00	47832.00	103.44	<.0001
Instance*Method	40.00	42819.00	1070.00	2.31	<.0001
Error	1827	844865.00	462.00		
Corrected	1889	3073634160.00			

5 Conclusions

This research presented an evolutionary algorithm with probabilistic models, which is a hybrid framework combining the probabilistic model and genetic operators together. From these experiments in single machine and flowshop scheduling problems, ACGA outperformed other algorithms in the literature. Part of the reason is that the probabilistic model can extract parental distribution and then generates good artificial solution following the parental distribution. Consequently, the proposed algorithm can provide very promising results.

Finally, the key of improving the evolutionary algorithm with probabilistic models is based on the establishment of a good probabilistic model. However, previous researches did

Table 6 The Duncan grouping result for the three algorithms

Duncan grouping	Mean	*N*	Method
A	2541.89	630.00	GA
B	2528.14	630.00	ACGA
C	2525.75	630.00	ACGA+Max-Min Evaporation

not take this issue into consideration. Therefore, the major contribution of this research is to mine the structure of the chromosomes generated in previous generations by establishing a more solid probabilistic model through the application of different data mining techniques such as clustering, classification or fuzzy methods. In addition, the idea of probability control is applied to further improve the performance of evolutionary algorithm. The probability control can further enhance ACGA to create a much diversified population through the probability evaporation. Consequently, the intensive experiments in the single machine scheduling problems and the flowshop scheduling problems are very satisfactory and convincing. We expect to apply the ACGA with probability control to other combinatorial problems in the near future.

Acknowledgements We appreciate that Prof. Shaw S.Wang provided some constructive comments for this paper.

References

Abdul-Razaq, T., & Potts, C. (1988). Dynamic programming state-space relaxation for single-machine scheduling. *The Journal of the Operational Research Society*, *39*(2), 141–152.

Ackley, D. H. (1987). *A connectionist machine for genetic hillclimbing*. Dordrecht: Kluwer Academic.

Akturk, M. S., & Ozdemir, D. (2001). A new dominance rule to minimize total weighted tardiness with unequal release dates. *European Journal of Operational Research*, *135*(2), 394–412.

Alves, M. J., & Almeida, M. (2007). MOTGA: A multiobjective Tchebycheff based genetic algorithm for the multidimensional knapsack problem. *Computers and Operations Research*, *34*(11), 3458–3470.

Baker, K. R. (1974). *Introduction to sequencing and scheduling*. New York: Wiley.

Baluja, S. (1995). An empirical comparison of seven iterative and evolutionary function optimization heuristics.

Baluja, S., & Davies, S. (1998). Fast probabilistic modeling for combinatorial optimization. In *Proceedings of the fifteenth national/tenth conference on artificial intelligence/innovative applications of artificial intelligence table of contents* (pp. 469–476).

Baraglia, R., Hidalgo, J., Perego, R., NUCE I, & CNR P (2001). A hybrid heuristic for the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, *5*(6), 613–622.

Belouadah, H., Posner, M., & Potts, C. (1992). Scheduling with release dates on a single machine to minimize total weighted completion time. *Discrete Applied Mathematics*, *36*(3), 213–231.

Birbil, Ş., & Fang, S. C. (2003). An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization*, *25*(3), 263–282.

Chang, P. C. (1999). A branch and bound approach for single machine scheduling with earliness and tardiness penalties. *Computers and Mathematics with Applications*, *37*(10), 133–144.

Chang, P. C., & Lee, H. C. (1992a). A greedy heuristic for bicriterion single machine scheduling problems. *Computers and Industrial Engineering*, *22*(2), 121–131.

Chang, P. C., & Lee, H. C. (1992b). A two phase approach for single machine scheduling: minimizing the total absolute deviation. *Journal of the Chinese Institute of Engineers*, *15*(6), 735–742.

Chang, P. C., Chen, S. H., & Fan, C. Y. (2008a). A novel electromagnetism-like algorithm in single machine scheduling problem with distinct due dates. *Expert Systems with Applications* 39(3).

Chang, P. C., Chen, S. H., & Fan, C. Y. (2008b). Mining gene structures to inject artificial chromosomes for genetic algorithm in single machine scheduling problems. *Applied Soft Computing Journal*, *8*(1), 767–777.

Chang, P. C., Chen, S. H., & Mani, V. (2009). A hybrid genetic algorithm with dominance properties for single machine scheduling with dependent penalties. *Applied Mathematical Modeling*, *33*(1), 579–596.

Corne, D., Dorigo, M., Glover, F., Dasgupta, D., Moscato, P., Poli, R., & Price, K. (1999). *New ideas in optimization*. New York: McGraw-Hill.

Dimopoulos, C., & Zalzala, A. (2000). Recent developments in evolutionary computation for manufacturing optimization: problems, solutions, and comparisons. *IEEE Transactions on Evolutionary Computation*, *4*(2), 93–113.

Framinan, J. M., Gupta, J. N. D., & Leisten, R. (2004). A review and classification of heuristics for permutation flow-shop scheduling with makespan objective. *Journal of the Operational Research Society*, *55*(12), 1243–1255.

Glover, F., & Kochenberger, G. (1996). Critical event tabu search for multidimensional knapsack problems. In *Meta-heuristics: theory & applications*. Dordrecht: Kluwer Academic.

Harik, G. (1999). Linkage learning via probabilistic modeling in the ECGA. *Urbana*, *51*, 61, 801.

Harik, G., Lobo, F., & Goldberg, D. (1999). The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, *3*(4), 287–297.

Hejazi, S., & Saghafian, S. (2005). Flowshop-scheduling problems with makespan criterion: a review. *International Journal of Production Research*, *43*(14), 2895–2929.

Jouglet, A., Savourey, D., Carlier, J., & Baptiste, P. (2008). Dominance-based heuristics for one-machine total cost scheduling problems. *European Journal of Operational Research*, *184*(3), 879–899.

Larrañaga, P., & Lozano, J. A. (2002). *Estimation of distribution algorithms: a new tool for evolutionary computation*. Dordrecht: Kluwer Academic.

Lee, C. Y., Lei, L., & Pinedo, M. (1997). Current trends in deterministic scheduling. *Annals of Operations Research*, *70*, 1–41.

Lenstra, J., Kan, A., & Brucker, P. (1975). Complexity of machine scheduling problems. Stud. integer Program, Proc. Workshop Bonn.

Li, G. (1997). Single machine earliness and tardiness scheduling. *European Journal of Operational Research*, *96*(3), 546–558.

Liaw, C. F. (1999). A branch-and-bound algorithm for the single machine earliness and tardiness scheduling problem. *Computers and Operations Research*, *26*(7), 679–693.

Lin, S., & Kernighan, B. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, *21*(2), 498–516.

Lozano, J. A. (2006). *Towards a new evolutionary computation: advances in the estimation of distribution algorithms*. Berlin: Springer.

Michalewicz, Z., Dasgupta, D., Le Riche, R., & Schoenauer, M. (1996). Evolutionary algorithms for constrained engineering problems. *Computers & Industrial Engineering*, *30*(4), 851–870.

Montgomery, D. C. (2001). Design and analysis of experiments.

Muhlenbein, H., & Paaß, G. (1996). *From recombination of genes to the estimation of distributions I. Binary parameters*. *Lecture Notes in Computer Science* (Vol. 1141, pp. 178–187). Berlin: Springer.

Murata, T., Ishibuchi, H., & Tanaka, H. (1996). Genetic algorithms for flowshop scheduling problems. *Computers & Industrial Engineering*, *30*(4), 1061–1071.

Ow, P. S., & Morton, T. E. (1989). The single machine early/tardy problem. *Management Science*, *35*(2), 177–191.

Pelikan, M., Goldberg, D. E., & Cantu-Paz, E. (1999). BOA: The Bayesian optimization algorithm. In *Proceedings of the genetic and evolutionary computation conference GECCO-99* (Vol. 1, pp. 525–532).

Pelikan, M., Goldberg, D. E., & Lobo, F. G. (2002). A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, *21*(1), 5–20.

Rastegar, R., & Hariri, A. (2006). A step forward in studying the compact genetic algorithm. *Evolutionary Computation*, *14*(3), 277–289.

Reeves, C. R. (1995). A genetic algorithm for flowshop sequencing. *Computers and Operations Research*, *22*(1), 5–13.

Ruiz, R., & Maroto, C. (2005). A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, *165*(2), 479–494.

Schoonderwoerd, R., Holland, O. E., Bruten, J. L., & Rothkrantz, L. J. M. (1997). Ant-based load balancing in telecommunications networks. *Adaptive Behavior*, *5*(2), 169.

Sim, K. M., & Sun, W. H. (2003). Ant colony optimization for routing and load-balancing: survey and new directions. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, *33*(5), 560–572.

Sourd, F., & Kedad-Sidhoum, S. (2003). The one-machine problem with earliness and tardiness penalties. *Journal of Scheduling*, *6*(6), 533–549.

Sourd, F., & Kedad-Sidhoum, S. (2007). A faster branch-and-bound algorithm for the earliness-tardiness scheduling problem. *Journal of Scheduling* (pp. 1–10).

Stutzle, T., Hoos, H. H. et al. (2000). MAX-MIN ant system. *Future Generation Computer Systems*, *16*(8), 889–914.

Syswerda, G. (1993). Simulated crossover in genetic algorithms. *Foundations of Genetic Algorithms*, *2*, 239–255.

Valente, J. M. S., & Alves, R. A. F. S. (2005). Improved heuristics for the early/tardy scheduling problem with no idle time. *Computers and Operations Research*, *32*(3), 557–569.

Valente, J. M. S., & Alves, R. A. F. S. (2007). Heuristics for the early/tardy scheduling problem with release dates. *International Journal of Production Economics*, *106*(1), 261–274.

Wu, S. D., Storer, R. H., & Chang, P. C. (1993). One-machine rescheduling heuristics with efficiency and stability as criteria. *Computers and Operations Research*, *20*(1), 1–14.

Zhang, Q., & Muhlenbein, H. (2004). On the convergence of a class of estimation of distribution algorithms. *IEEE Transactions on Evolutionary Computation*, *8*(2), 127–136.

Zhang, Q., Sun, J., & Tsang, E. (2005). An evolutionary algorithm with guided mutation for the maximum clique problem. *IEEE Transactions on Evolutionary Computation*, *9*(2), 192–200.