International Journal of Innovative Computing, Information and Control
Volume **5**, Number **12**, December **2009**
ICIC International ©2009 ISSN 1349-4198
pp. 1–ISII08-013

A GUIDED MEMETIC ALGORITHM WITH PROBABILISTIC MODELS

Shih-Hsin Chen[1], Pei-Chann Chang[2], Qingfu Zhang[3] and Chin-Bin Wang[1]

[1]Department of Electronic Commerce Management
Nanhua University
No. 32, Chungkeng, Dalin, Chiayi 62248, Taiwan
{ shihhsin; cbwang }@mail.nhu.edu.tw

[2]Department of Information Management
Yuan-Ze University
135 Yuan-Dong Rd., Taoyuan 32026, Taiwan
iepchang@saturn.yzu.edu.tw

[3]Department of Computing and Electronics System
University of Essex
Wivenhoe Park, Colchester, CO4 3SQ, UK
qzhang@essex.ac.uk

Received July 2008; revised December 2008

Abstract. *Due to the combinatorial explosions in solution space for scheduling problems, the balance between genetic search and local search is an important issue when designing a memetic algorithm* [23] *for scheduling problems. The main motivation of this research is to resolve the combinatorial explosion problem by reducing the possible neighborhood combinations using guided operations to remove these inferior moves. We proposed a new algorithm, termed as a Guided memetic algorithm, which is one of the algorithms in the category of evolutionary algorithm based on probabilistic models (EAPMs). The algorithm explicitly employs the probabilistic models which serves as a fitness surrogate. The fitness surrogate estimates the fitness of the new solution generated by a local search operator beforehand so that the algorithm is able to determine whether the new solution is worthwhile to be evaluated again for its true fitness. This character distinguishes the proposed algorithm from previous EAPMs. The single machine scheduling problems are applied as test examples. The experimental results show that the Guided memetic algorithm outperformed elitism genetic algorithm significantly. In addition, the Guided memetic algorithm works more efficiently than previous EAPMs and Elitism Genetic algorithm. As a result, it is a new break-through in genetic local search with probabilistic models as a fitness surrogate.*

1. **Introduction.** As defined by Nareyek [33], there are two search-paradigms for search: refinement search and local search. The refinement search is iteratively narrowing process alternating between commitment and propagation while local search conducts a search by iteratively changing an initial state. The advantage of refinement search is that it can be understood easily. Local search, however, can be usually computed very fast in solving larger problems. It provides better exploiting information for realistic problems that have more complex objective functions. Therefore, many researchers utilized the local search to enhance the search ability. Nevertheless, a genetic local search takes the advantages of both the refinement search and local search [17, 15, 36]. Ishibuchi et al. [23] advocated that the importance of maintaining a balance between genetic search and local search. The main reason is that the local search operator blindly changes the initial state

which reduces the number of function evaluations available for genetic search because the possible neighborhood combinations are very large. As a result, the motivation of this research is to propose a new genetic local search algorithm which will prune away bad moves. This new approach can maintain the balance between genetic search and local search. In addition, the efficiency and effectiveness of the genetic local search algorithm is expected.

To derive these expected results, this paper develops a new approach by embedding the probabilistic model with genetic local search operator which is termed as a Guided Memetic Algorithm. The probabilistic model is served as a fitness surrogate and we can determine whether it is worthwhile to evaluate the real fitness of a new solution generated by local search operator in advance. If a movement is likely to lead to an inferior solution, the movement is abandoned. Thus, the total amount of computation time can be greatly reduced by avoiding those bad moves generated for the local search since the combinations of local search are huge. The balance between the genetic search and local search can be maintained by using the probabilistic models.

The proposed algorithm actually belongs to the Evolutionary Algorithm based on Probabilistic models (EAPMs) which is one of the most popular evolutionary algorithms in recent years [7, 8, 20, 32, 35, 41]. EAPM explicitly builds a probabilistic model to present the parental distribution. EAPM also generates solutions by sampling from the probabilistic model. It is important to know how EAPM works. There are some researchers, such as Glover and Laguna [18], who argued that certain gene structures resembles those in other superior better chromosomes. In [11], they presented the probability of each jobs at different positions graphically in single machine scheduling problems, which shows that the salient genes appeared after some generations are executed in the population. It means that some jobs are gradually converged into fixed positions. Lin and Kernighan [29] solved the traveling salesman problem (TSP) and the results shows that 85% of the segments in any two local optimal solutions were identical. Due to EAPM extracts genetic information from parents on the fly, EAPM might be a promising method able to capture the population information and manipulate the building block, hence, solve hard optimization problems efficiently [40].

The probabilistic model in this research, however, is not applied directly to generate new solutions because sampling from probabilistic model requires higher computational cost in solving sequencing problems (See [41, 3, 11]). Instead, a guided operation will be applied as a fitness surrogate to study the feasibility of a new solution. As a result, the proposed algorithm combines the advantages of probabilistic model and local search to work more efficiently than refinement search [33]. The computational efficiency of the algorithm is therefore expected.

The rest of the paper is organized as follows: Section 2 is the literature review of scheduling problems, particularly for single machine scheduling problems with the minimization of earliness/tardiness cost. Section 3 is the detail explanations of the Guided Memetic Algorithm, Section 4 is the experimental results whereas the proposed algorithm was evaluated by using the single machine scheduling problems with the objective of minimizing the earliness/tardiness cost, Section 5 is the discussions and conclusions of this research.

2. **Literature Review of the Single-machine Scheduling Problems.** Scheduling problems arise in almost every area of human endeavor and defy any simple and succinct classification [19]. Various scheduling problems can be found in [2, 6, 5, 16, 21, 22, 12]. As a generalization of single-machine scheduling to minimize the weighted tardiness [26], the single-machine scheduling problem to minimize the total weighted earliness and tardiness costs is strongly NP-hard. The earlier works on this problem were due to [10, 14, 39].

[9] dealt with a similar problem with the objective of minimizing the total weighted completion time. The problem is the same as that discussed in [9]. Later on, [4] developed various dominance rules to solve the problem.

The earliness/tardiness scheduling problem with equal release dates and no idle time has been considered by several authors. Both exact and heuristic approaches have been proposed to solve the problem. Among the exact algorithm approaches, branch-and-bound algorithms were presented by [1, 27, 28]. The lower bounding procedure of Abdul-Razaq and Potts [1] is based on the sub-gradient optimization approach and the dynamic programming state-space relaxation technique, whereas [27] and [28] used Lagrangian relaxation and the multiplier adjustment method. Among these heuristics, [34] developed several dispatching rules and a filtered beam search procedure. In [38], the authors presented an additional dispatching rule and a greedy procedure. They also considered the use of dominance rules to further improve the schedule obtained by the heuristics. A neighborhood search algorithm was presented by [27].

Some research has developed dominance properties (DPs) for this category of problems [28, 37, 31, 30, 24]. DPs are employed in branch-and-bound algorithms to enhance the fathoming procedure to be combined with other meta-heuristics, such as integrating DPs with GA to solve the scheduling problem with earliness/tardiness penalties [13].

3. **Self-Guided Memetic Algorithm.**

EAPM extracts the gene variable structure from population distribution and expresses it with a probabilistic model [25]. This research embed the probabilistic model with local search operator to reduce the number of bad moves. The probabilistic model serves as a fitness surrogate and it can evaluate the figure of merit beforehand. It means that the probabilistic model will calculate the probability difference of selected genes located at different positions to determine the movement of genes decided by local search operator. As a result, this proposed algorithm will exploit the solution space efficiently instead of blindly searching the solution space. In this paper, we proposed a guided memetic algorithm to solve scheduling problems and a procedure is named guided local search. The following figure is the pseudo code of proposed algorithm.

Population: A set of solutions
generations: The maximum number of generations
$P(t)$: Probabilistic model
t : Generation index

1. Initialize *Population*
2. $t \leftarrow 0$
3. Initialize $P(t)$
4. **while** $t < generations$ **do**
5. EvaluateFitness(*Population*)
6. Selection/Elitism (*Population*)
7. $P(t+1) \leftarrow$ BuildingProbabilityModel(*Selected Chromosomes*)
8. Crossover()
9. Mutation()
10. Guided Local Search()
11. $t \leftarrow t+1$
12. **end while**

FIGURE 1. Algorithm1: The main procedures of Guided Memetic Algorithm

Step 2 initializes the probability matrix $P(t)$ by using $1/n$ where n is the problem size. Step 7 builds the probabilistic model $P(t)$ after the selection procedure. The details of

$P(t)$ are explained in Section 3.1. Step 8 and Step 9 explain the standard crossover and mutation procedures. In this research, two-point center crossover and swap mutation are applied in the crossover and mutation procedures in scheduling problems. In Step 10, probabilistic model will evaluate the goodness of the move generated by local search, which is shown in Section 3.2.

The following sections explain the proposed algorithm in details, which at first explain how to establish a probabilistic model and how the probabilistic model reduces the number of bad moves in the local search procedure.

3.1. **Establishing a probabilistic model.** Suppose a population has M strings X^1, X^2 ,..., 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 Eq 1.

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

The fitness of these M chromosomes is evaluated and the gene information is collected from N best chromosomes where $N < M$. The N chromosomes are set as $M/2$ which is a thumb-of-rule. The purpose of selecting only 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 appear at position j at the current generation. As in PBIL [7], the probability $P_{ij}(t)$ is updated as follows:

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

where $\lambda \in (0, 1]$ is the learning rate. The larger λ is, the more the gene information of the current population contributes. For the probabilistic matrix of all jobs at different positions, they are written as Eq 3.

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

Therefore, the expected fitness of a solution X^k can be estimated from the probabilistic matrix, by means of joint probability distribution shown as follows:

$$P_{t+1}(X^k) = P_{t+1}(X_{[1],1}^k, \ldots, X_{[n],n}^k) \approx \prod_{p=1,i=[p]}^{n} P_{t+1}(X_{ip}^k) \tag{4}$$

where $[p]$ is the job at Position p in Chromosome X^k.

In general, the procedures of establishing the probabilistic model include

1. Evaluating fitness of all M solutions,
2. Selecting N best solutions, and
3. Learning the joint distribution of the selected solutions. After a probabilistic model is built up, the probabilistic model can serve as a surrogate of fitness evaluation since the $\prod_{p=1,i=[p]}^{n} P_{t+1}(X_{ip}^k)$ estimates the goodness of the solution X^k.

As soon as the probabilistic model $P(t+1)$ simplified as P in the rest of sections, it is embedded with local search operator. It is shown in the next section.

3.2. **Guided memetic algorithm.** This research enables the probabilistic difference as the fitness surrogate to function in the local search procedure. This section demonstrates that the probabilistic model embeds with the local search operator as an example. The concept of this algorithm is that before the genes are moved using local search operator, the probabilistic differences of the moved genes are evaluated before the real fitness. When the fitness surrogate judges the new solution which is no better than the original one, the proposed algorithm won't evaluate the real fitness of the new solution generated by local search operator.

On the other hand, if the fitness surrogate indicates that the new solution is better than the original one, the real fitness value of the modified solution is evaluated. It is obvious that, when the fitness is further improved, the original solution is replaced by the new solution. Because the combination of local solution space is huge, the advantage of the fitness surrogate would first screen out some bad moves in advance so the genetic local search would not waste too many search efforts in searching bad moves.

Some local search strategies are adopted in this study. First of all, the procedure does systematic search of the solution space. There is a heuristic rule that the neighborhood size is up to $maxNeighborhood$, say 3, 5, or 7. This parameter configuration is done by Design-of-Experiment. As a result, the local search will focus on the neighbors which are not too far away. Second, only elite solutions are considered for local search because it is costly for inferior solutions [23]. The local search method utilizes the 2-Opt exchanging in the end. It is, to be sure, not limited to 2-Opt method since the probabilistic difference can be applied to any local search method. The following pseudo code describes how the procedure works.

X: A selected solution
Y: Solution X is modified into a new solution.
i and j: The range of 2-Opt search
length: The problem size
Δ: The probabilistic difference of solution X and Y
maxExaminedSoln: The maximum number of examined solutions
ExaminedSoln: The number of solutions has been examined
X: Select an elite solution
$f(X)$ *and* $f(Y)$: Real fitness of the solution X and Y
$maxNeighborhood$: Determine the maximum neighborhood size

1. FOR$i = 1$ to length
2. FOR$j = (i + 1)$ to $(i + maxNeighborhood)$
3. STATE $Y \leftarrow$ 2-Opt(X, i, j)
4. STATE $\Delta \leftarrow$ FitnessSurrogate (X, Y)
5. IF$\Delta > 0$ and *ExaminedSoln*$<$ *maxExaminedSoln*
6. IF$f(X) > f(Y)$
7. $X \leftarrow Y$
8. *ExaminedSoln* ++
9. ENDIF
10. ENDIF
11. ENDFOR
12. ENDFOR

Figure 2. The detail procedures of Guided Local Search

The first step is to randomly select an elite solution to do local search. It then does a systematic local search by 2-Opt method. In Step 4, fitness surrogate is employed for the probabilistic difference. It follows that this function evaluates the probabilistic difference of genes moved by the local search. After that, when the probability difference is positive and the number of current examined solution is less than the maximum number of the examined solution, the algorithm proceeds to calculate the real fitness of the new solution. If the new solution is better than the original solution, the original solution is replaced by the new one in Step 7.

There is a six-job problem to present how the guided memetic algorithm works. When there is an elite solution {536124} and the cut points are at Position 2 and 5, the new solutions will be {521634} (see Figure Figure 3.).

$$X \text{ (Original solution): } \{5|3612|4\} \rightarrow Y \text{ (New Solution): } \{5|2163|4\}$$

Figure 3. An example of 2-Opt local search

The fitness surrogate examines whether the New Solution Y is better than the Elite solution. Eq 5 is the join probability of the Original solution X solution and the New Solution Y.

$$\begin{aligned} P_X &\approx (\prod_{p\in(1or6),g=[p]}^{n} P(X_{gp}))(\prod_{p\notin(1or6),g=[p]}^{n} P(X_{gp})) \\ P_Y &\approx (\prod_{p\in(1or6),g=[p]}^{n} P(Y_{gp}))(\prod_{p\notin(1or6),g=[p]}^{n} P(Y_{gp})) \end{aligned} \tag{5}$$

Eq.(5) can be simplified as Eq.(6) because $\prod_{p\in(1or6),g=[p]}^{n} P(X_{gp})$ and $\prod_{p\in(1or6),g=[p]}^{n} P(Y_{gp})$ are equal and the two terms are greater than or equal to zero.

$$\begin{aligned} P'_X &\approx \prod_{p\notin(1or6),g=[p]}^{n} P(X_{gp}) \\ P'_Y &\approx \prod_{p\notin(1or6),g=[p]}^{n} P(Y_{gp}) \end{aligned} \tag{6}$$

One more step of preventing the probability of a job at a position is zero which makes the joint probability become insensitive, Eq.(6) is further revised as Eq.(7).

$$\begin{aligned} P''_X &\approx \sum_{p\notin(1or6),g=[p]}^{n} P(X_{gp}) = (p_{32} + p_{63} + p_{14} + p_{25}) \\ P''_Y &\approx \sum_{p\notin(1or6),g=[p]}^{n} P(Y_{gp}) = (p_{22} + p_{13} + p_{64} + p_{35}) \end{aligned} \tag{7}$$

At last, we subtract P''_X from P''_Y and we have the final result Δ in Eq.(8).

$$\Delta = P''_X - P''_Y \tag{8}$$

If Δ is positive, the new solution Y is rejected because it might be inferior to the original one. Otherwise, the solution is potentially superior to the original solution. As a results, this new solution is double checked by the real fitness function. When the new solution is better than the original one, it replaces the original solutions.

This approach brings a benefit, which is when local search are put on too many efforts in exploiting solution space, the quota for genetic search are reduced. It is the reason why [23] discussed the balance between a genetic search and a local search. As a result, the local search with the fitness surrogate would detect bad moves although it might run the risk of finding a local optimal. Because there is no research that utilizes the concept of probabilistic model as a fitness surrogate, it is natural that no research brought forward this approach in conducting local search. Consequently, this idea is also a novel one which can be a good tool for local search algorithms.

4. **Experimental Results.** In order to evaluate the performance of the Guided Memetic Algorithm with probabilistic models, it was compared with some algorithms in literature. These algorithms are used to test the single machine scheduling problems and these instances are taken from [37]. This single machine scheduling problems with the consideration of minimizing the earliness and tardiness were evaluated. The problem definitions is shown in Section 4.1. These algorithms were implemented by Java 2 (With JBuilder JIT compiler) on Windows 2003 server (Intel Xeon 3.2 GHZ). We test the scheduling problems provided by [37], whereas there are job 20, 30, 40, 50, 60, and 90. In all the experiments, each instance was replicated 30 times. Sections 4.2 presented the empirical results of single machine scheduling.

4.1. **Problem statements of single machine scheduling problem.** In this paper, a deterministic single machine scheduling problem without release date is investigated and its objective is to minimize the total sum of earliness and tardiness penalties. A detailed formulation of the problem is described as follows: A set of n independent jobs $\{J_1, J_2, \cdots, J_n\}$ has to be scheduled without preemptions on a single machine that can handle one job at a time at most. The machine is assumed to be continuously available from time zero onwards and unforced machine idle time is not allowed. Job $J_j, j = 1, 2, \cdots, n$ becomes available for the processing at the beginning, requires a processing time p_j and should be completed on its due date d_j. For any given schedule, the earliness and tardiness of J_j can be respectively defined as $E_j = \max(0, d_j - C_j)$ and $T_j = \max(0, C_j - d_j)$, where C_j is the completion time of J_j.

The objective is then to find a schedule that minimizes the sum of the earliness and tardiness penalties of all jobs $\sum_{j=1}^{n} (\alpha_j E_j + \beta_j T_j)$ where α_j and β_j are the earliness and tardiness penalties of job J_j. The inclusion of both earliness and tardiness costs in the objective function is compatible with the philosophy of just-in-time production, which emphasizes producing goods only when they are needed. The early cost may represent the generating cost of completing a product early, the deterioration cost for perishable goods or a holding (stock) cost for finished goods. The tardy cost can represent rush shipping costs, lost sales and loss of goodwill. It is assumed that no unforced machine idle time is allowed, so the machine is only idle when no job is currently available for processing. This assumption reflects a production setting where the cost of machine idleness is higher than the early cost stemming from completing any job before its due date, or the capacity of the machine is limited when compared with its demand, so that the machine must be kept running all the time. Some specific examples of production arrangements with these characteristics are provided by [34] and [39]. The set of jobs is assumed to be ready to process jobs at the beginning which is a characteristic of the deterministic problem.

4.2. **Results of single machine scheduling problems.** The proposed algorithm, guided memetic algorithm, is compared with Genetic Algorithm with Dominance Properties (GADP), Artificial Chromosome with Genetic Algorithm (ACGA), and Guided Mutation (EA/G) which can be found in [13], [11], and [41]. All of them evaluated 100,000 solutions in the single machine scheduling problems. Table 1 is the partial results of the basic statistics results of these algorithms. For more detail results, please refer to our website [1].

TABLE 1. Single machine scheduling problems: Job 20, 30, 40, and 50

	GA			GADP			ACGA			EA/G			Guided GLS		
instance	Min	Avg.	Max	Min	Avg.	Max	Min	Avg.	Max	Min	Avg.	Max	Min	Avg.	Max
sks222a	5286	5401.7	5643	5286	5291	5298	5286	5288.8	5298	5286	5289.8	5298	5286	5291.9	5298
sks225a	3958	4173.5	4389	3958	3958.6	3977	3958	3958	3958	3958	3959.3	3977	3958	3958.6	3977
sks228a	2085	2155.6	2749	2085	2085	2085	2085	2085	2085	2085	2085	2085	2085	2085	2085
sks252a	4052	4195.3	4508	3947	3947	3947	3947	3979	4067	3947	3991	4067	3947	3979	4067
sks255a	2388	2488.5	2787	2372	2372	2372	2372	2380	2388	2372	2380.4	2388	2372	2383.5	2397
sks258a	1184	1249.9	1371	1184	1241.6	1248	1184	1200.4	1248	1184	1186.1	1248	1184	1205.3	1248
sks282a	4348	4435	4695	4348	4353.1	4355	4348	4350.8	4355	4348	4350.1	4355	4348	4351.3	4355
sks285a	4452	4642.6	4895	4452	4452	4452	4452	4452	4452	4452	4452	4452	4452	4452	4452
sks288a	3421	3517.6	3847	3421	3421	3421	3421	3421	3421	3421	3421	3421	3421	3421	3421
sks322a	11623	12066	12916	11568	11572	11622	11568	11577	11622	11568	11576	11622	11568	11577	11622
sks325a	7615	8151.8	9650	7587	7703.2	7904	7587	7587	7587	7587	7642.3	7929	7587	7587	7587
sks328a	3195	3556.2	4810	3164	3164	3164	3164	3164	3164	3164	3164	3164	3164	3164	3164
sks352a	7588	8203.2	9063	7395	7395	7395	7392	7393.8	7395	7392	7398.4	7416	7392	7394.1	7395
sks355a	6202	6849.2	7693	6056	6067.9	6212	6056	6065.4	6193	6056	6062.1	6212	6056	6062.5	6212
sks358a	3104	3282.6	3787	3069	3074.4	3076	3069	3073.4	3076	3069	3074.8	3106	3069	3073.2	3076
sks382a	11157	11319	11558	11140	11152	11160	11140	11149	11222	11140	11152	11222	11140	11157	11222
sks385a	9157	9212.1	9389	9148	9148	9148	9148	9148	9148	9148	9148	9148	9148	9148	9148
sks388a	11321	11499	11789	11317	11317	11317	11317	11317	11317	11317	11320	11340	11317	11317	11317
sks422a	25656	26211	27462	25656	25658	25712	25656	25659	25704	25656	25661	25712	25656	25662	25704
sks425a	12725	13592	15198	12601	12604	12605	12601	12606	12668	12601	12613	12682	12601	12611	12643
sks428a	7237	7741.1	8761	7129	7129.4	7141	7129	7129	7129	7129	7134.1	7146	7129	7129	7129
sks452a	11804	12634	14053	11367	11367	11367	11367	11406	11581	11367	11396	11581	11367	11403	11528
sks455a	6573	7566	9435	6405	6405	6405	6405	6426.7	6666	6405	6424.6	6545	6405	6428.9	6527
sks458a	4424	5587	8331	4294	4302.9	4319	4294	4320.5	4391	4294	4319.3	4381	4294	4323.7	4394
sks482a	19656	20122	20605	19559	19580	19735	19559	19573	19735	19559	19568	19593	19559	19566	19572
sks485a	15459	16023	16576	15256	15309	15480	15256	15338	15480	15256	15316	15480	15256	15342	15480
sks488a	17374	17999	18668	16862	16881	16906	16862	16863	16888	16862	16863	16888	16862	16863	16888
sks522a	29485	30623	32340	29309	29322	29507	29309	29312	29396	29309	29326	29398	29309	29336	29398
sks525a	25693	26113	26713	25433	25436	25517	25433	25438	25469	25433	25444	25496	25433	25437	25469
sks528a	11154	12037	13121	10798	10821	10823	10798	10838	11129	10798	10824	10860	10798	10878	11129
sks552a	23491	24827	26241	22863	22863	22863	22863	22894	23148	22863	22906	23143	22863	22885	23013
sks555a	10877	12233	14626	10207	10243	10446	10187	10216	10299	10187	10224	10299	10187	10222	10301
sks558a	5776	7345	9358	5269	5297.7	5416	5269	5269	5269	5269	5269	5269	5269	5269	5269
sks582a	28375	29154	30551	27939	28117	28352	27939	27947	28056	27939	27959	28167	27939	27978	28167
sks585a	25200	25862	26447	24828	24830	24853	24828	24839	24853	24828	24836	24853	24828	24838	24853
sks588a	25453	26422	28197	24844	24846	24856	24844	24845	24861	24844	24849	24870	24844	24847	24861

[1] http://mail.nhu.edu.tw/∼shihhsin/publications/sourceCodes/InjectionArtificialChromosomes/Results.htm

To verify the difference of the algorithms, ANOVA is employed in Table 2. Because there is significant difference among these methods, in order to justify the performance of these algorithms, Duncan pair-wise comparison is employed in Table 3.

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

Source	DF	Type I SS	Mean Square	F Value	Pr > F
instances	213	6.85E+12	32172844813	325037	<.0001
method	4	6.93E+09	1732362401	17501.8	<.0001
instances*method	852	12281545276	14414959.24	145.63	<.0001
Error	31030	3071416622	98982.17		
Corrected Total	32099	6.88E+12			

Duncan Grouping test shows that there is significant difference between/among subjects if they share different alphabet. Otherwise, there are no differences between/among the subjects. In Table 3, Duncan comparisons indicated ACGA which performed as well as EA/G. On the other hand, GADP and guided memetic algorithm are the second group in this comparisons and GA is the worst in the single machine scheduling problems.

Table 3. Duncan Grouping in testing objective values of single machine scheduling problems

Duncan Grouping	Mean	N	method
A	13982.894	6420	GA
B	12831.934	6420	Guided Memetic Algorithm
B			
B	12827.096	6420	GADP
C	12813.66	6420	EA/G
C			
C	12813.276	6420	ACGA

After we tested the ANOVA results in minimizing objective values, the computational time is also examined. Table 4 and Table 5 are ANOVA and Duncan grouping results of the CPU time comparisons, respectively. Since the CPU time is significant, Duncan Grouping results indicated that Guided Memetic Algorithm works more efficient than others, particularly the ACGA and EA/G. The reason is that ACGA and EA/G requires $O(n^2)$ time when they generate a new solution by sampling from probabilistic model in solving sequential problems. Consequently, these results show that Guided Memetic Algorithm is very attractive because the algorithm performs well and work efficiently than Standard Genetic Algorithm.

5. **Discussions and Conclusion.** The paper unveils a new concept of in local search algorithm by using the EAPMs, which is to apply the probabilistic model evaluating the figure of merit of a new solution beforehand generated by local search. Compared with previous EAPM algorithms, the probabilistic model is able to determine the goodness of the new solutions, instead of using sampling from probabilistic model to generate solutions. Although this concept can be employed in many aspects, this paper embedded this concept with local search operator as an example. As a result, Guided Memetic Algorithm with probabilistic models enables the local search to process without blindly searching. The proposed algorithm works efficiently than previous EAPMs and the elite

TABLE 4. ANOVA result of the computation time in single machine scheduling problems

Source	DF	Type I SS	Mean Square	F Value	Pr > F
instances	213	1.40E+05	657.88	4717.43	<.0001
method	4	481180.13	120295.03	862589	<.0001
instances*method	852	261996.18	307.51	2205.01	<.0001
Error	31030	4327.39	0.14		
Corrected Total	32099	887633.05			

TABLE 5. Duncan Grouping for these algorithms in testing CPU time in single machine scheduling problems

Duncan Grouping	Mean	N	method
A	11.06	6420	EA/G
B	2.57	6420	GADP
C	1.62	6420	ACGA
D	1.21	6420	GA
E	0.64	6420	Guided Memetic Algorithm

Genetic Algorithm. The proposed algorithm solved single machine scheduling problems with minimization of earliness/tardiness cost. The experimental result indicated that the proposed algorithm indeed performed better than elite Genetic algorithm. Although it doesn't outperform the previous EAPMs, the proposed algorithm, however, works more efficiently than elite Genetic algorithm. It is because the evaluation of the probabilistic difference of local search operator takes only a constant time. When it comes to previous EAPMs, the time complexity is $O(n^2)$. Thus, this approach doesn't lead to extra or excessive computational efforts. Based on this pioneer research, researchers are able to design an operator which integrates the proposed concept in the near future.

REFERENCES

[1] T. Abdul-Razaq and C. Potts, Dynamic programming state-space relaxation for single-machine scheduling, *The Journal of the Operational Research Society*, vol.39, no.2, pp.141-152, 1988.

[2] J. Adams, E. Balas and D. Zawack, The shifting bottleneck procedure for job shop scheduling, *Management Science*, vol.34, no.3, pp.391-401, 1988.

[3] U. Aickelin, E. K. Burke and J. Li, An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering, *Journal of the Operational Research Society*, vol.58, no.12, pp.1574-1585, 2007.

[4] M. J. Alves and M. Almeida, MOTGA: A multiobjective Tchebycheff based genetic algorithm for the multidimensional knapsack problem, *Computers and Operations Research*, vol.34, no.11, pp.3458-3470, 2007.

[5] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting and R. Vijayakumar, Providing quality of service over a shared wireless link, *Communications Magazine, IEEE*, vol.39, no.2, pp.150-154, 2001.

[6] K. Baker and G. Scudder, Sequencing with earliness and tardiness penalties: A review, *Operations Research*, vol.38, no.1, pp.22-36, 1990.

[7] S. Baluja and S. Davies, Fast probabilistic modeling for combinatorial optimization, *Proc. of the 15th National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence Table of Contents*, pp.469-476, 1998.

[8] R. Baraglia, J. Hidalgo, R. Perego, I. CNUCE and P. CNR, A hybrid heuristic for the traveling salesman problem, *IEEE Transactions on Evolutionary Computation*, vol.5, no.6, pp.613-622, 2001.

[9] H. Belouadah, M. Posner and C. Potts, Scheduling with release dates on a single machine to minimize total weighted completion time, *Discrete Applied Mathematics*, vol.36, no.3, pp.213-231, 1992.
[10] P. C. Chang, A branch and bound approach for single machine scheduling with earliness and tardiness penalties, *Computers and Mathematics with Applications*, vol.37, no.10, pp.133-144, 1999.
[11] P. C. Chang, S. H. Chen and C. Y. Fan, Mining gene structures to inject artificial chromosomes for genetic algorithm in single machine scheduling problems, *Applied Soft Computing Journal*, vol.8, no.1, pp.767-777, 2008.
[12] P. C. Chang, S. H. Chen and K. L. Lin, Two-phase sub population genetic algorithm for parallel machine-scheduling problem, *Expert Systems with Applications*, vol.29, no.3, pp.705-712, 2005.
[13] P. C. Chang, S. H. Chen and V. Mani, A hybrid gnetic algorithm with dominance properties for single machine scheduling with dependent penalties, *Applied Mathematical Modeling*, vol.33, no.1, pp.579-596, 2009.
[14] P. C. Chang and H. C. Lee, A greedy heuristic for bicriterion single machine scheduling problems, *Computers and Industrial Engineering*, vol.22, no.2, pp.121-131, 1992.
[15] I. Essafi, Y. Mati and S. Dauzere-Peres, A genetic local search algorithm for minimizing total weighted tardiness in the job-shop scheduling problem, *Computers and Operations Research*, vol.35, no.8, pp.2599-2616, 2008.
[16] T. Furusho, T. Nishi and M. Konishi, Distributed optimization method for simultaneous production scheduling and transportation routing in semiconductor fabrication bays, *International Journal of Innovative Computing, Information and Control*, vol.4, no.3, pp.559-575, 2008.
[17] J. Gao, L. Sun and M. Gen, A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems, *Computers and Operations Research*, vol.35, no.9, pp.2892-2907, 2008.
[18] F. Glover and G. Kochenberger, Critical even tabu search for multidimensional knapsack problems, *Meta-Heuristics: Theory & Applications*, 1996.
[19] J. Gupta, An excursion in scheduling theory: an overview of scheduling research in the twentieth century, *Production Planning and Control*, vol.13, no.2, pp.105-116, 2002.
[20] G. Harik, F. Lobo and D. Goldberg, The compact genetic algorithm, *IEEE Transactions on Evolutionary Computation*, vol.3, no.4, pp.287-297, 1999.
[21] Y. Hirashima, A Q-learning system for container transfer scheduling based on shipping order at container terminals, *International Journal of Innovative Computing, Information and Control*, vol.4, no.3, pp.547-558, 2008.
[22] X. Hu, M. Huang and A. Zeng, An intelligent solution system for a vehicle routing problem in urban distribution, *International Journal of Innovative Computing, Information and Control*, vol.3, no.1, pp.189-198, 2007.
[23] H. Ishibuchi, T. Yoshida and T. Murata, Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling, *IEEE Transactions on Evolutionary Computation*, vol.7, no.2, pp.204-223, 2003.
[24] A. Jouglet, D. Savourey, J. Carlier and P. Baptiste, Dominance-based heuristics for one-machine total cost scheduling problems, *European Journal of Operational Research*, vol.184, no.3, pp.879-899, 2008.
[25] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, Kluwer Academic Publishers, 2002.
[26] J. Lenstra, A. Kan and P. Brucker, Complexity of machine scheduling problems, *Stud. integer Program., Proc. Workshop Bonn*, 1975.
[27] G. Li, Single machine earliness and tardiness scheduling, *European Journal of Operational Research*, vol.96, no.3, pp.546-558, 1997.
[28] C. F. Liaw, A branch-and-bound algorithm for the single machine earliness and tardiness scheduling problem, *Computers and Operations Research*, vol.26, no.7, pp.679-693, 1999.
[29] S. Lin and B. Kernighan, An effective heuristic algorithm for the traveling-salesman problem, *Operations Research*, vol.21, no.2, pp.498-516, 1973.
[30] X. Luo, C. Chu and C. Wang, Some dominance properties for single-machine tardiness problems with sequence-dependent setup, *International Journal of Production Research*, vol.44, no.17, pp.3367-3378, 2006.
[31] X. Luo and F. Chu, A branch and bound algorithm of the single machine schedule with sequence dependent setup times for minimizing total tardiness, *Applied Mathematics and Computation*, vol.183, no.1, pp.575-588, 2006.
[32] H. Muhlenbein and G. Paaß, From recombination of genes to the estimation of distributions I. Binary parameters, *Lecture Notes in Computer Science*, vol.1141, pp.178-187, 1996.

[33] A. Nareyek, *Constraint-Based Agents: An Architecture for Constraint-Based Modeling and Local-Search-Based Reasoning for Planning and Scheduling in Open and Dynamic Worlds*, Springer, 2001.
[34] P. S. Ow and T. E. Morton, The single machine early/tardy problem, *Management Science*, vol.35, no.2, pp.177-191, 1989.
[35] J. Peña, V. Robles, P. Larrañaga, V. Herves, F. Rosales and M. Perez, GA-EDA: Hybrid evolutionary algorithm using genetic and estimation of distribution algorithms, *Innovations In Applied Artificial Intelligence: 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2004*, Ottawa, Canada, 2004.
[36] A. S. Ramkumar, S. G. Ponnambalam, N. Jawahar and R. K. Suresh, Iterated fast local search algorithm for solving quadratic assignment problems, *Robotics and Computer-Integrated Manufacturing*, vol.24, no.3, pp.392-401, 2008.
[37] F. Sourd and S. Kedad-Sidhoum, The one-machine problem with earliness and tardiness penalties, *Journal of Scheduling*, vol.6, no.6, pp.533-549, 2003.
[38] J. M. S. Valente and R. A. F. S. Alves, Improved heuristics for the early/tardy scheduling problem with no idle time, *Computers and Operations Research*, vol.32, no.3, pp.557-569, 2005.
[39] S. D. Wu, R. H. Storer and P. C. Chang, One-machine rescheduling heuristics with efficiency and stability as criteria, *Computers and Operations Research*, vol.20, no.1, pp.1-14, 1993.
[40] Q. Zhang and H. Muhlenbein, On the convergence of a class of estimation of distribution algorithms, *IEEE Transactions on Evolutionary Computation*, vol.8, no.2, pp.127-136, 2004.
[41] Q. Zhang, J. Sun and E. Tsang, An evolutionary algorithm with guided mutation for the maximum clique problem, *IEEE Transactions on Evolutionary Computation*, vol.9, no.2, pp.192-200, 2005.