Journal of College of Education for pure sciences(JCEPS)
Web Site: http://eps.utq.edu.iq/ Email: com@eps.utq.edu.iq
Volume 7, Number 3, September 2017

Using Genetic Algorithm to Minimize Single Machine Scheduling Problem

Sami .M Araibi

sami.mezal@utq.edu.iq

Department of Mathematics, College of Computer Science and Mathematics,Thi-Qar University, Thi-Qar, Iraq

Abstract

In this paper, the problem of scheduling *n* jobs on a single machine is considered. The aim in this study is to find the near optimal solution for the discounted total weighted completion time$\sum_{\text{F}1}^{\text{J}} w_{\text{F}}(1\ \ \text{F}\ \text{A}^{\text{F}\ \ \%\text{F}})$ with unequal release date by using Genetic algorithm. Three special cases are derived and proved that yield optimal solutions. Also, lower bound and upper bound that introduced in this study, in order to find the optimal solution by Branch and Bound algorithm and comparison with genetic algorithm. Results of extensive computational tests show that proposed (Genetic Algorithm) is effective in solving problems up to (2000) jobs at a time less than or equal to (10) minutes.

Keywords:single machine, scheduling, genetic algorithm, discounted total weighted completion time, unequal release date.

Journal of College of Education for pure sciences(JCEPS)
Web Site: http://eps.utq.edu.iq/ Email: com@eps.utq.edu.iq
Volume 7, Number 3, September 2017

إستخدام الخوارزمية الجينية لتصغير مسألة جدولة الماكنة الواحدة

سامي مزعل عريبي

sami.mezal@utq.edu.iq

قسم الرياضيات، كلية علوم الحاسوب و الرياضيات، جامعة ذي قار، ذي قار، العراق

الخلاصة:

تناولنا في هذا البحث دراسة مسألة جدولة n من النتاجات (Jobs) على ماكنة واحدة. الهدف من هذه الدراسة إيجاد حلول قريبة من الحل الأمثل لمسألة المجموع الوزني لتخفيض وقت الاتمام $\sum_{j=1}^{n} w_j\ (1-e^{-\alpha C_j})$ عندما تكون للنتاجات ازمنة تحضير غير متساوية باستخدام الخوارزمية الجينية. تم اشتقاق وبرهان ثلاث حالات خاصة والتي انتجت حلول مثلى. كذلك تم اقتراح قيد ادنى وقيد اعلى للمسألة من اجل استخدامها في خوارزمية التفرع والتقيد لايجاد الحل الأمثل ومقارنة النتائج مع الخوارزمية الجينية. نتائج الاختبارات الحسابية أثبتت بأن الخوارزمية الجينية فعالة في حل المسائل لغاية (2000) نتاج في وقت أقل او يساوي (10) دقيقة.

1. Introduction

This paper focuses on single machine scheduling with unequal release dates was considered to minimize the discounted total weighted completion time by using genetic algorithm, which can be characterized as $1/r_j/\sum_{j=1}^{n} w_j\ (1-e^{-\alpha C_j})$ using the Graham and Lawler classification [1]. Rothkopf andSmith (1984) [2] considered the total discounted weighted completion time as$1//\sum_{j=1}^{n} w_j\ (1-e^{-\alpha C_j})$. Their study was more general cost function and discounted rate of ,$0<\alpha<1$ of the problem can be solved as P-type optimally inpolynomial time by the Weighted Discounted Shorter Processing Time (WDSPT) rule. The rule that schedules the jobs in non-decreasing order of ratio: $\frac{1-e^{-\alpha p_j}}{w_j e^{-\alpha p_j}}$ (non-increasingorder of ratio:$\frac{w_j e^{-\alpha p_j}}{1-e^{-\alpha p_j}}$) [3]. Wang et. al (2006) considered the problem $F_m//\sum_{j=1}^{n} w_j\ (1-e^{-\alpha C_j})$ and show that its NP-hard [4].Yunqiang Yin et al. (2012) [5] showed that the total weighted discounted completion time is polynomials solvable and optimal by considering the effects of position-dependent learning and time-dependent deterioration simultaneously. The problem total weighted discounted completion time studied by Lin Li et al. (2013) [6] showed the heuristics according to the corresponding problems without learning effect. Guochen and Yuewu (2013) [7] showed the problem $1/DT,\ D_{psd}/\sum_{j=1}^{n} w_j\ (1-e^{-\alpha C_j})$ is past-sequence-dependent delivery time and deteriorating jobs. Hongjie Li et al. (2014) [8]studies the problem$1/p_j^X = a_j^X(1-kt)/\sum_{j=1}^{n} w_j^A\left(1-e^{-\alpha C_j^A}\right): f_{max}^B \le U$considers a scheduling environment in which there are two agents and a set of jobs, each of which belongs to one of the two agents and its actual processing time is defined as a decreasing linear function of its starting time. Each of the two agents competes to process its respective jobs on a single machine and has its own scheduling objective to optimize.

Glass and Potts (1996) [9]; Gupta et al. (2002) [10]; Riad (2004) [11]; Dirk and Christian (2004) [12]; Soukhal and Martineau (2005) [13] applied genetic algorithm to solve flow-shop machine scheduling problem, Murat and Ahmet (2003) [14] using genetic algorithm to consider two bicriteria scheduling problem on a single machine: minimizing flow time and number of tardy jobs, and minimizing flow time and maximum earliness. Mutsunori and Toshihide (1996) [15] propose to use dynamic programming in the process of obtaining new generation solution in the genetic algorithm, to evaluate the effectiveness of this approach, they choose three representative combinatorial optimization problems. Celso et al. (2005) [16] propose a tabu search – based heuristic and a genetic algorithm and consider the single machine scheduling problem with common due date. Performance was measured by the minimization of the weighted sum of earliness and tardiness penalties of jobs, Allahverdi and Aldowaisan (2004) [17] proposed a hybrid genetic heuristics, which can be used for the single criterion of makespan or maximum lateness, or the bicriteria problem.

2. Formulation of the Problem

The general problem of scheduling jobs on a single machine to minimize the discounted total weighted completion time that can be state as follows. A set of n independent jobs $N = \{1,2,\dots,n\}$which has to be scheduled without preemption on a single machine that can handle at most one job at a time. The machine is assumed to be continuously available from time zero onwards and no precedence relationship exists between jobs. Each job $j, j \in N$ has an integer processing timep_j , a release date r_j and a positive weighted w_j . For any given schedule $(1,2,\dots,n)$, the discounted total weighted completion time:$w_j\,(1 - e^{-\alpha C_j})$can be computed, where C_j be a completion time for job$j$, given by the relationship:

$$C_1 = p_1 + r_1$$

$C_j = \max\{C_{j-1}, r_j\} + p_j$ for $j = 2,\dots,n$

Our scheduling problem can be state mathematically more precisely as follows:

Given a schedule $\delta = (1,2,\dots,n)$, then for each job $j \in \delta$ can be calculate thediscounted total weighted completion time. The objective is to find a schedule,$\sigma = (\sigma_{(1)}, \sigma_{(2)}, \dots, \sigma_{(n)})$belong to a neighborhood of δthat minimize the cost $Z(\sigma)$, where

$$Z(\sigma) = \sum_{j=1}^{n} w_{\sigma(j)}\,(1 - e^{-\alpha C_{\sigma(j)}})$$

Let ζ be a set of all schedules, $|\zeta| = n!$, then we can formulate our problem in mathematical form as:

$$\left.\begin{array}{ll}
Z = \min\limits_{\sigma\in\zeta}\{Z(\sigma)\} = \min\limits_{\sigma\in\zeta}\left\{\sum\limits_{j=1}^{n} w_{\sigma(j)}\left(1 - e^{-\alpha C_{\sigma(j)}}\right)\right\} & \\
s.to & \\
C_{\sigma(j)} \ge r_{\sigma(j)} + p_{\sigma(j)} & , \ j = 1,\dots,n \\
C_{\sigma(j)} \ge C_{\sigma(j-1)} + p_{\sigma(j)} & , \ j = 2,\dots,n \\
w_{\sigma(j)} \ge 1 & , \ j = 1,\dots,n \\
p_{\sigma(j)} > 0\,, r_{\sigma(j)} \ge 0 & , \ j = 1,\dots,n \\
0 < \alpha < 1 &
\end{array}\right\} \dots (P)$$

3. Special Cases

Case 1: If $r_j = r \forall j,\ (j = 1,2,\dots,n)$, then WDSPT rule gives an optimal solution, for $1/r_j = r/\sum_{j=1}^{n} w_j\,(1 - e^{-\alpha C_j})$ problem.

Proof:

Consider the sequence $S = \delta jk\delta'$, where δ and δ' are partial sequences and j and k are two jobs with $\frac{w_j e^{-\alpha p_j}}{1-e^{-\alpha p_j}} \le \frac{w_k e^{-\alpha p_k}}{1-e^{-\alpha p_k}}$.

Let η be a completion time of last job in δ. Let $S' = \delta kj\delta'$ a new sequence (by interchange jobs j and k in original sequence).

Let $\tau = \max\{\eta, r\}$

In S: $C_j = \tau + p_j$ and $C_k = \tau + p_j + p_k$ (Since $\tau > r$)

$$w_j(1-e^{-\alpha C_j}) + w_k(1-e^{-\alpha C_k}) = w_j\left(1-e^{-\alpha(\tau+p_j)}\right) + w_k\left(1-e^{-\alpha(\tau+p_j+p_k)}\right) \quad \text{... (1)}$$

In S': $C_k = \tau + p_k$ and $C_j = \tau + p_k + p_j$ (Since $\tau > r$)

$$w_j(1-e^{-\alpha C_j}) + w_k(1-e^{-\alpha C_k}) = w_j\left(1-e^{-\alpha(\tau+p_k+p_j)}\right) + w_k\left(1-e^{-\alpha(\tau+p_k)}\right) \quad \text{... (2)}$$

From$(1)-(2)$ we get

$$w_j - w_j e^{-\alpha(\tau+p_j)} + w_k - w_k e^{-\alpha(\tau+p_j+p_k)} - w_j + w_j e^{-\alpha(\tau+p_k+p_j)} - w_k + w_k e^{-\alpha(\tau+p_k)}$$

$$= -w_j e^{-\alpha(\tau+p_j)} - w_k e^{-\alpha(\tau+p_j+p_k)} + w_j e^{-\alpha(\tau+p_k+p_j)} + w_k e^{-\alpha(\tau+p_k)}$$

$$= w_j e^{-\alpha(\tau+p_j)}(-1+e^{-\alpha p_k}) + w_k e^{-\alpha(\tau+p_k)}(1-e^{-\alpha p_j})$$

$$= e^{-\alpha\tau}(-1+e^{-\alpha p_k})(1-e^{-\alpha p_j})\left(\frac{w_j e^{-\alpha p_j}}{1-e^{-\alpha p_j}} + \frac{w_k e^{-\alpha p_k}}{-1+e^{-\alpha p_k}}\right)$$

$$= -e^{-\alpha\tau}(1-e^{-\alpha p_k})(1-e^{-\alpha p_j})\left(\frac{w_j e^{-\alpha p_j}}{1-e^{-\alpha p_j}} - \frac{w_k e^{-\alpha p_k}}{1-e^{-\alpha p_k}}\right) \ge 0$$

Thus, the sequence S' is preferred to the sequence S. Then WDSPT rule gives an optimal solution, for $1/r_j = r/\sum_{j=1}^{n} w_j(1-e^{-\alpha C_j})$ problem.

Case 2: If $r_j = r$ and $p_j = p\forall j,\ (j = 1,2,\dots,n)$, then the sequence which is sequencing the jobs in decreasing order of their weight, gives an optimal solution for $1/r_j = r, p_j = p/\sum_{j=1}^{n} w_j(1-e^{-\alpha C_j})$ problem.

Proof:

Consider the sequence $S = \delta jk\delta'$, where δ and δ' are partial sequences and j and k are two jobs with $w_j \le w_k$.

Let η be a completion time of last job in δ. Let $S' = \delta kj\delta'$ a new sequence (by interchange jobs j and k in original sequence).

Let $\tau = \max\{\eta, r\}$

In S: $C_j = \tau + p$ and $C_k = \tau + 2p$

$$w_j(1-e^{-\alpha C_j})+w_k(1-e^{-\alpha C_k})=w_j\left(1-e^{-\alpha(\tau+p)}\right)+w_k\left(1-e^{-\alpha(\tau+2p)}\right) \quad \dots (3)$$

In S': $C_k=\tau+p$ and $C_j=\tau+2p$

$$w_j(1-e^{-\alpha C_j})+w_k(1-e^{-\alpha C_k})=w_j\left(1-e^{-\alpha(\tau+2p)}\right)+w_k\left(1-e^{-\alpha(\tau+p)}\right) \quad \dots (4)$$

From$(3)-(4)$ we get

$$\begin{aligned} & w_j-w_je^{-\alpha(\tau+p)}+w_k-w_ke^{-\alpha(\tau+2p)}-w_j+w_je^{-\alpha(\tau+2p)}-w_k+w_ke^{-\alpha(\tau+p)} \\ &= -w_je^{-\alpha(\tau+p)}-w_ke^{-\alpha(\tau+2p)}+w_je^{-\alpha(\tau+2p)}+w_ke^{-\alpha(\tau+p)} \\ &= w_j\left(e^{-\alpha(\tau+2p)}-e^{-\alpha(\tau+p)}\right)-w_k\left(e^{-\alpha(\tau+2p)}-e^{-\alpha(\tau+p)}\right) \\ &= \left(e^{-\alpha(\tau+2p)}-e^{-\alpha(\tau+p)}\right)(w_j-w_k)=-e^{-\alpha(\tau+p)}(1-e^{-\alpha p})(w_j-w_k)\geq 0 \end{aligned}$$

Then, sequencing the jobs in decreasing order of their weight, gives an optimal solution for $1/r_j=r,p_j=p/\sum_{j=1}^{n}w_j\,(1-e^{-\alpha C_j})$ problem.

Case 3: If $r_j=r$, $p_j=p$ and $w_j=w\forall j,\ (j=1,2,\dots,n)$, then any order gives an optimal solution, for $1/r_j=r,p_j=p,w_j=w/\sum_{j=1}^{n}w_j\,(1-e^{-\alpha C_j})$.

Proof:

It is clear from case (1) and case (2).

4. The Upper Bound (UB)

The following algorithm (heuristic) has been proposed to obtain the upper bound for the problem (P).

Step (1): Order the jobs by SRD to obtain a sequence$\sigma=(\sigma_{(1)},\sigma_{(2)},\dots,\sigma_{(n)})$.

Step (2): If there exists a job i $(i=1,\dots,n)$ such that $C_{\sigma(i)}\geq r_{\sigma(k)}$, $k=i+1,i+2,\dots,n$ then a new sequence $\pi=(\pi_{(1)},\pi_{(2)},\dots,\pi_{(n)})$ which is obtained from σ after ordering the jobs $i+1,i+2,\dots,n$ according to WDSPT rule, otherwise go to Step (4).

Step (3): Compute $\sum_{j=1}^{n}w_{\pi(j)}\,(1-e^{-\alpha C_{\pi(j)}})$, go to Step (5).

Step (4): Compute $\sum_{j=1}^{n}w_{\sigma(j)}\,(1-e^{-\alpha C_{\sigma(j)}})$.

Step (5): Stop.

5. The LowerBound (LB)

In this section, the lower bound derived for problem (P). To obtain the lower bound for problem (P), the relaxation of constraints of objective function will be as follows:

We assume that $r^*=\min_{j\in N}\{r_j\}$, and applying Case (1). The optimal solution for the new problem is a lower bound for problem (P).

6.The Genetic Algorithm (GA)

Genetic algorithms are global search and optimization techniques modeled from natural genetics. They date back to the early work described by John Holland[18]**.** It works on a randomly generated candidate solution pool, which is usually called "population". Each encoded candidate solution is called "chromosome". During the searching process, the selection, crossover and mutation operators are executed repeatedly until the stop criteria is satisfied[19].

The main components of a Genetic algorithm are as follows [19][20]:

1. Initialization

Initially many individual solutions are generated randomly or produced with a good heuristic to form an initial population. The chromosomes should be encoded. There are many ways to encode the initial generation, for scheduling problems the natural representation is usually used. The natural representation consists of permutation of the jobs 1,2,…,n which defines the processing order of the jobs[20]. This population is called the current population. The population size depends on the nature of the problem; it is ranging from the size of 20 Lee and Kim[21] to 300 Delia Croce et al. [22].

2. Fitness (evaluation)

The measurement of chromosome fitness is based on the objective function (fitness).When a population is generated, each chromosome is evaluated and its fitness is calculated for each chromosome. Finally, each chromosome is assigned its fitness value along of the population size.

3. Selection

Natural selection of some chromosomes, by selection methods (according fitness value usually), chromosomes (parents) are selected from the population for combining to produce new chromosomes (children).

4. Reproduction

Reproduction (crossover and mutation) selects the best individuals from the population; crossover mates these individuals to produce one or more children and mutation makes small local changes.

- **Crossover**

It creates the information exchange mechanism among the genes of individuals, and contributes to the generation of new individuals with better gene combination for populations. What attaches crossover operating is the crossover rate, which is used to decide the number or proportion of individuals from crossover among the new generation. There are many kinds of crossover, some of them are:

a) One point crossover[23]:

The procedure of one-point crossover is to randomly generate a number (less than or equal to the chromosome length) as the crossover-position. Then keep the bits before the number unchanged and swap the bits after the crossover position between the two parents.

b) Two-point crossover (Partically matched crossover (PMX))[18]:

Two-point crossover calls for two points to be selected on the parent organism strings. Everything between the two points is swapped between the parent organisms, rendering two child organisms.

c) Linear order crossover (LOX)[20]:

This operator selects two random crossover points, the elements in the cross section of the first parent are removed from the second parent leaving some holes. These holes silde from the extremities towards the center until they reach the cross section. The cross section is then substituted with that of parent1. The other child is obtained similarly. LOX tends to maintain relative positions between the elements.

d) Homogeneous Mixture Crossover (HMX) [24]:

This crossover is given by the mixture of the two parents uniformly by making a set (m) from genes, the odd position from the first parent and the even position from the second parent. Then separate genes without repetition of the gene, since we read the set (m) from the left, if the gene $g$dose not exist in the child then keep it and put (0) in (m), otherwise we keep gene g in the second child and put (1) in (m), until (m) genes are exhausted. This way also gives two new children. This crossover preserves the absolute positions taken from one parent, and the relative positions of those from the other parent.

- **Mutation**

Mutation is the process of manipulating individual genes in a candidate solution. This is a completely random process that does not use any information from the parent chromosomes or the fitness of the candidate solution. Mutation uses random numbers to select genes, and then modifies the selected genes according to the strategy applied by the developer of the algorithm. This is a local search technique that tries to improve candidate solutions by looking at other candidates nearby in the solution space. Syswerda introduced a number of mutation operators [25]. In such an operator, two elements are selected at random.

Order-mutation: interchanges these two elements.

Position-mutation: places the second element before the first.

5) Termination[20]

There are several rules to decide if the GA cycle should be stopped: (1) The chromosomes in the current population are better than the ones in the previous population by calculating the average fitness of all chromosomes in every cycle, (2) The best chromosome in the current GA pool is better than the best one in the old GA pool by calculating the fitness value of the best chromosome in every cycle, or (3) The number of generation reaches the predetermined level.

7. Determine theGA Parameters

Genetic algorithm is general search and optimization method work on a population of feasible solutions (chromosomes) individuals to a given problem. In the following, we describe each of the mechanism for our scheduling problem briefly:

1- Initialization

The initial population can be generated randomly or can be constructed by using heuristic methods. In this paper we start with m = 120, 115 from them randomly and the remaining five are given by SPT rule, SRD rule, order the jobs according to non-decreasing order of $p_i - r_i$,order the jobs according to non-decreasing order of w_i and order the jobs according to non-decreasing order of p_i/w_i

2- New population

A new population is created by repeating the following substeps until the new population is completed.

a. Selection:

Selecting the individuals according to fitness value that will usually form the next generating's parents.

b. Crossover:

Homogeneous mixture crossover which is defined in section (6) is applied.

c. Mutation

Pairwise swap mutation is applied on each pair of parent solutions to generate two new solutions (children).

3- Termination Condition:

The GA procedure stops when a fixed number of generations (or iterations) are executed here (200) iterations at a time less than or equal to 10 minutes. This means that the GA procedure continues until the population is converged to a good, if not optimal solution to our problem (P).

8. Computational Experience

The GA algorithm is tested on problem (P) for finding the near optimal solution by coding it in Matlab R2010a and running on a personal computer Lenovo with Ram 8 GB. Test problems are generated as follows: For each job j, an integer processing time p_j and an integer weights w_j are generated from the discrete uniform distribution[1,10]. Also, for each job j, an integer release date r_j is generated from the discrete uniform distribution $[0, \alpha\, P]$, where [α = 0.125, 0.25, 0.50, 0.75, 1.00] and$P = \sum_{j=1}^{n} p_j$ .For each selected value of n, where n is the number of jobs, five problems were generated.

In the following table, we list 5 examples with different number of jobs each one is solved by GA algorithm and BAB algorithm.Thefirstcolumn(n)refers to number of jobs, the second column (*EX*)refers to the number of examples for each instance n, where $n \in \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$, the third column (*Optimal*) refers to the optimal values obtained by *BAB* algorithm for problem (*P*), the fourth column (*GA -Val*) refers to the values obtained by genetic algorithm for problem (*P*), the fifth column (*Status*)refers to the *GA-Val = optimal* (1) or not (0). No of opt. = number of examples that catch the optimal value.

Table (1): Comparison between the values found by Genetic algorithm and the optimal solution.

n	*EX*	*Optimal*	*GA-Val*	*Status*
5	1	15.4337	15.4337	1
	2	12.8122	12.8122	1
	3	22.5866	22.5866	1
	4	16.6100	16.6100	1
	5	31.8298	31.8298	1
No of opt.			5	
10	1	30.8856	30.8856	1
	2	53.2779	53.2779	1
	3	45.8647	45.8647	1
	4	54.9522	54.9522	1
	5	51.3897	51.3897	1
No of opt.			5	
15	1	85.4579	85.4579	1

	2	59.1973	59.8668	0
	3	78.5459	78.5459	1
	4	58.6539	58.6539	1
	5	77.0202	77.0202	1
No of opt.			4	
20	1	90.2415	90.2415	1
	2	85.1560	85.1560	1
	3	97.0930	97.1501	0
	4	108.9178	108.9202	0
	5	112.3470	112.3470	1
No of opt.			3	
25	1	107.8393	107.8393	1
	2	109.2682	109.2682	1
	3	107.3599	107.3599	1
	4	134.7675	134.7675	1
	5	126.8202	126.8784	0
No of opt.			4	
30	1	163.5850	163.5850	1
	2	134.9503	134.9503	1
	3	170.0225	170.0230	0
	4	133.4633	133.4682	0
	5	163.1448	163.1448	1
No of opt.			3	

Table (1): Continued

n	*EX*	*Optimal*	*GA-Val*	*Status*
35	1	173.6810	173.9573	0
	2	169.5568	170.0174	0
	3	195.2846	195.2998	0
	4	217.3255	217.6425	0
	5	176.0803	176.0803	1

No of opt.			1	
40	1	168.1377	168.1377	1
	2	203.9345	204.0166	0
	3	199.4398	199.4398	1
	4	222.1864	222.1882	0
	5	213.8191	213.8191	1
No of opt.			3	
45	1	234.2738	234.6415	0
	2	224.2128	224.2243	0
	3	237.8499	237.8499	1
	4	251.2239	251.2239	1
	5	282.0379	282.0379	1
No of opt.			3	
50	1	264.5121	264.6085	0
	2	270.3960	270.4173	0
	3	264.3606	264.3606	1
	4	290.1703	290.1703	1
	5	273.9374	273.9374	1
No of opt.			3	

From the above table (1) we conclude that: Genetic algorithm can obtainedoptimal solution for the problem (P) of size less than or equal (50) jobs.

Table (2): The performance of Genetic Algorithm for $n \in \{75\,, 100\,, 500\,,\ 1000\,, 2000\}$

n	*EX*	*GA-Val*	*Time*
75	1	381.3322	2.3477
	2	391.7309	2.2676
	3	437.1959	2.2517
	4	418.9927	2.1952
	5	359.5139	2.1992
100	1	556.6900	3.4094

	2	551.4581	3.4182
	3	535.5824	3.5062
	4	548.4623	3.3870
	5	559.5998	3.5047

Table (2): Continued

n	*EX*	*GA-Val*	*Time*
500	1	2747.2933	41.7488
	2	2582.3484	43.5865
	3	2646.2361	42.1467
	4	2809.9846	40.0611
	5	2742.9555	40.3235
1000	1	5524.8099	147.0670
	2	5463.3570	139.7997
	3	5406.3974	142.5698
	4	5430.3572	142.0966
	5	5460.3203	144.0388
2000	1	10871.3959	538.7002
	2	10962.2981	536.0907
	3	11018.0180	538.6969
	4	11052.7036	531.6272
	5	10691.9667	531.2761

From the above table (2) we conclude that: Genetic algorithm is effective in solving problems up to (2000) jobs.

9. Conclusions

This paper proposed a GA for solving the single machine scheduling to minimize the discounted total weighted completion time$w_j\,(1-e^{-\alpha C_j})$ with unequal release date. Several results concerning optimality of three solvable special cases are presented. Also, we obtained the upper bound for the problem $1/r_j/\sum_{j=1}^{n} w_j\,(1-e^{-\alpha C_j})$ and derive the lower boundwhich are used in branch and bound algorithm. We evaluate its performance by comparison with an optimal valueobtained by *BAB* algorithm for small size.The computational result shows that GA can obtain better solution.

10. References

[1] Graham R.L., Lawler E.L., Lenstra J.K., and Rinnooy K.A.,"Optimization and approximation in deterministic sequencing and scheduling: A survey", Annals of Discrete Mathematics 5, pp. 287-326 (1979).

[2] Rothkopf, M.H and Smith S.A, "There are no undiscovered priority index sequencing rules for minimizing total delay costs", Operations Research, 32, pp. 451–456, (1984).

[3] Pinedo, M.L., "Scheduling Theory, Algorithms, and Systems", Springer Science +Business Media, LLC., New York (2012).

[4] Ji-Bo, W., Feng, S., Bo, J., and Li-Yan, W., "Permutation flow shop scheduling with dominant machines to minimize discounted total weighted completion time", Applied Mathematics and Computation 182, pp. 947–954 (2006).

[5] Yunqiang, Y., Min, L., Jinghua, H., and Mengchu, Z., "Single-machine scheduling with job-position-dependent learning and time-dependent deterioration", Vol. 42, NO. 1, January (2012).

[6] Lin, L., Sheng-Wu Y,. Yu-Bin, W,. Yunzhang, H,. and Ping, J., " Single machine scheduling jobs with a truncated sum-of-processing-times-based learning effect", Int. J. Adv. Manuf. Technol. 67, pp. 261-267 (2013).

[7] Guochen, S., and Yuewu, L., " Single-machine scheduling with past-sequence-dependent delivery times and deteriorating jobs" Mathematica Aeterna, 3, (9), pp. 799 – 806, (2013).

[8] Hongjie, L., Zeyuan, L., and Yunqiang, Y., "Some single-machine scheduling problems with learning effects and two competing agents", The Scientific World Journal Volume Article ID 471016 (2014).

[9] Glass C.A. and Potts C.N., “A Comparison of local search methods for flow shop scheduling”, Annals operations Research 63, pp. 489-509, (1996).

[10] Jatinder N. D. Gupta, Karsten Hennig, Frank Werner, “Local search heuristics for two-stage flow shop problems with secondary criterion” Computers and Operations Research 29, pp. 123-149(2002).

[11] Riad Aggoune “Minimizing the makespan for the flow shop scheduling problem with availability constraints”, European Journal of Operational Research 153, pp. 534-543(2004).

[12] Dirk C. Mattfeld and Chirstian Bierwirth, “Discrete optimization an efficient genetic algorithm for job shop scheduling with tardiness objectives”, European Journal of Operational Research 155, pp. 616-630(2004).

[13] Soukhal,A. and P. Martinean. “Resolution of a scheduling problem in a flow-shop robotic cell”, European Journal of Operational Research 161, pp. 62-72(2005).

[14] Murat Koksalan and Ahmet Burak Keha, “Using genetic algorithm for single-machine bicriteria scheduling problems”, European Journal of Operational Research 145, pp. 543-556(2003).

[15] Mutsunori Yagiura and Toshihide Ibaraki, “Theory and methodology the use of dynamic programming in genetic algorithms for permutation problems”, European Journal of Operational Research 92, pp. 387-401(1996).

[16] Celso M.Hino, Debora P. Ronconi and Andre B. mends, “ Minimizing earliness and tardiness penalties in a single-machine problem with a common due date” European Journal of Operational Research 160, pp. 190-201(2005).

[17] Allahverdi, A. and Aldowaisan, T. “No-wait flow shops with bicriteria of makespan and maximum lateness”, European Journal of Operational Research 152, pp. 132-147(2004).

[18]Hummady L. Z., "Using genetic algorithm to solve (NP-Compete) problem", M.Sc. thesis Univ. of Al- Mustansiriyah, College of Science, Dep. of Mathematics (2005).

[19]Sun L., Cheng X. and Liang Y.," Solving job shopscheduling problem using genetic algorithm with penalty function", International Journal of Intelligent InformationProcessing 7, (2), pp. 65-77 (2010).

[20]Selman F. M.," Exact and local search methods for single machine problem", M.Sc. thesis Univ. of Al- Mustansiriyah, College of Science, Dep. of Mathematics (2008).

[21]Lee J.K. and Kim Y.D., "Search heuristic for resource constrained project scheduling", Journal of the operation research society 47, pp. 678-689 (1996).

[22] Della Croce F., Tadi R. and Valta G., "A Genetic algorithmforjob shop problems ". computer and OperationResearch 22, pp. 15-40(1995).

[23]Mutesher H. J.,"Flow shop scheduling problems usingexact and local search methods", M.Sc. Thesis Univ. of Al-Mustansiriyah, College of Science, Dep. of Mathematics (2008).

[24]Abbas, I.T., "The performance of multicriteria scheduling in one machine", M.Sc. Thesis Univ. of Al-Mustansiriyah, College of Science, Dep. of Mathematics (2009).

[25] Crauwels, H. "A comparative study of local search methods for one machine sequence problem". Ph.D. Thesis KatholiekeUniversity, Heverlee. Belgium (1998).