Approximation methods to solve a single machine scheduling problem with fuzzy due date to minimize multi-objective functions

Hanan Ali Chchan 1, Mustafa Talal Kadhim2

1Department of Mathematics, College of Sciences, University of Mustansiriyah, Baghdad
2Department of Mathematics, College of Sciences, University of Mustansiriyah, Baghdad

Abstract: In this paper present three method ant colony optimization (ACO), particle swarm optimization (PSO) and bees algorithm optimization (BAO) to the solution multi-objective function of single machine problem with the fuzzy due date. The objective function to minimize total completion time and maximum lateness with a fuzzy due date. By a computer simulation used to compare the performance of each algorithm with another one from where accuracy and time.

1. Introduction

The concept of fuzzy decision making introduced by bellman and zadeh[1] in 1970, different requisitions of the fluffy principle will choice making issues have been introduced. In 1974 Tanaka et al. [2] and in 1979 Zimmermann[3] were detailed fluffy mathematical programming issues. In 1989 W.Szwarc and J.J.Liu[4] were found approximation solution to flow shop of m machine and n jobs where m ≤ 8 and n ≤ 8. In 1990 Inuiguchi, M., Ichishashi, H. and Tanaka [5] were propose many approaches in the field of fuzzy mathematical programming. In any case, many promising and intriguing areas stay to be explored in the field of fluffy combinatorial improvement. In 1992 Ishii et, al. [6] for scheduling problem were introduced the concept of fuzzy due date. In 1994 HisaoShibuchi, Naohisa Yamamoto.[7] were solve NP-hard problem by approximation methods and compare between descent, simulated annealing and taboo search algorithms are applied to the problem. In 1999Andreas Bauer, Bernd Bullnheimer [8] were solve NP-hard problem by used ant colony optimization methods and developed it. In 2003[9] G. Celano, A. Costa And S. Fichera were developed genetic algorithm to solved fuzzy flow shop scheduling problem. In 2005 [10] Hong Wang was applied branch and bound method to got exact solution and approach to artificial intelligence search techniques and compare between them. In 2006[11] Hamid Allaoui and Samir Lamouri were using Johnson’s algorithms to found approximation solution for some flow shop scheduling problem formatted by makespan for two machine. In 2008 [12] BabakJavadi and al. were proposed model to solved minimize the weighted mean completion time and the weighted mean earliness to no wait flow shop scheduling problem . In 2010[13] K Sheibani was The proposed technique comprises of two stages: masterminding the positions in need request and afterward building a grouping for flow shop scheduling problem to makespan criterion. In 2012[14] H. F. Abdullah was found approximation solution for two machine flow shop scheduling problem to minimized total earliness by proposes a new algorithms. CengizKahraman a, OrhanEngin and Mustafa KerimYilmaz[15] were solved multi objective function formatted by minimized the average tardiness and the number of tardy jobs to fuzzy flow shop scheduling problem by found new artificial immune system algorithms. In 2014 [16] J.Behnamiana , S.M.T. FatemiGhomiz were solved bi- objective hybrid scheduling problem formulated by minimized maximum completion time and sum of earliness and tardiness for flow shop scheduling problem by using some algorithms of local search as genetic algorithms and particle swarm optimization to found approximation solution. DonyaRahmani, Reza Ramezanianand Mohammad SaidiMehrabad[17] were studied fuzzy flow shop scheduling problem formulated by minimized total flow shop and total tardiness to considered provide release time, process time and a more realistic model by using genetic algorithm. B. Naderi, M. Aminnayeri, M. Piri and M.H. Ha’iriYazdi[18] were studied multi-objective no-wait flow shop scheduling problem to makespan and total tardiness formatted by F/nwt/TT.Cmax by using three type of local search greedy, moderate and curtailed fashions. In 2017[19] they studied development in flow shop scheduling problem under uncertainties depicts the distinctive arrangement draws near introduced in the writing and present status of exploration. At last, a few headings for future examination . In 2018 [20] ChiwenQu ,Yanming Fu,Zhongjun Yi,and Jun Tanwere solved no-wait flow shop scheduling problem to minimize the maximum accomplished time

2. Preliminaries

In this section we review the concept of fuzzy set theory
2-1 Definition (fuzzy set)\[9\]
The subset \( S \) of \( X \) is a fuzzy set if \( S = \{(x, \mu(x)): x \in X\} \) where \( \mu(x) \) is membership function define by \( \mu(x): X \to [0, 1] \)

2-2 Definition (support)\[9\]
A fuzzy set \( S \) is said to be support if \( S \) is a set of all a point \( x \in X \) such that
\[ \text{Supp}(S) = \{x \in X: \mu(x) > 0\} \]

2-3 Definition (core)
Let \( S \) is a fuzzy set a core of \( S \) is a set of all \( x \in X \) such that \( \mu(x) = 1 \)
\[ \text{core}(S) = \{x \in X: \mu(x) = 1\} \]

2-4 Definition (normal)
Let \( S \) is a fuzzy set is said to be normal if \( \exists x \in X \) such that \( \mu(x) = 1 \)

2-5 Definition (\( \alpha \) cut)
Let \( S \) is a fuzzy set \( \alpha \)-cut define by the following
\[ S_\alpha = \{x \in X: \mu(x) \geq \alpha\} \text{ where } \alpha \in [0, 1] \]

2-6 Definition (convex fuzzy set)
Let \( S \) is fuzzy set is said to be convex fuzzy set if every \( x_1, x_2 \in S_\alpha \) and \( \alpha \in [0, 1] \) and satisfy the following condition
\[ f(yx_1 + (1-y)x_2) \geq f(x_1) \wedge f(x_2) \]

2-7 Definition (fuzzy number)\[10\]
Let \( S \in R \) is a fuzzy subset is said to be fuzzy number if satisfy the following condition:
\[ \text{ if a fuzzy set is normal} \]
\[ \text{ ii. If the membership } \mu(x) \text{ is quasi concave this mean} \]
\[ \mu(sx + (1-s)y) \geq \min \{\mu(x), \mu(y)\} \]
\[ \text{ iii. The membership function } \mu(x) \text{ is semi continuous this mean} \]
\[ \{x \in R: \mu(x) \geq \alpha \} \text{ this set is closed in } R \text{ for } \alpha \in [0, 1] \]

2-8 Definition (triangular fuzzy number)
Let \( S \) be a fuzzy set define by \( S = (s_l, s_c, s_u) \) with a membership function define by
\[
\mu_S(x) = \begin{cases} 
0 & \text{if } x < s_l^l \\
\frac{x - s_l^l}{s_c^l - s_l^l} & \text{if } s_l^l \leq x < s_c^l \\
\frac{s_c^l - x}{s_c^l - s_l^l} & \text{if } s_c^l \leq x < s_c^u \\
\frac{s_u^l - x}{s_u^l - s_c^l} & \text{if } s_c^u \leq x < s_c^u \\
1 & \text{if } x \geq s_c^u \\
0 & \text{if } x \leq s_l^l 
\end{cases}
\]
Is called triangular fuzzy number.

3. Problem formulation

Suppose there are \( n \)-jobs scheduling on single machine each job has a processing time, and triangular fuzzy due date \( \bar{d}_j \). On a machine all a jobs are available to be processed and starts without interrupted. Let a sequence \( \sigma \) be a sequence of jobs processed on single machine to minimized total completion time and maximum lateness with a fuzzy due date.

Now, let the triangular fuzzy number \( (s_1, s_2, s_3) \), we using distance measure
Let \( A = [\overline{a}_a, \overline{a}_a] \) and \( B = [\overline{b}_a, \overline{b}_a] \), than
\[
\overline{d}(\overline{A}, \overline{B}) = \frac{1}{2} \int_0^1 ((\overline{a}_a - \overline{b}_a)^2 + (\overline{a}_a - \overline{b}_a)^2) da + \frac{1}{2} \int_0^1 ((\overline{a}_a - \overline{b}_a)^2 + (\overline{a}_a - \overline{b}_a)^2) da
\]
Where
\[
(\chi)^+ = \begin{cases} 
\chi & \text{if } \chi \geq 0, \\
0 & \text{if } \chi < 0 
\end{cases}
\]
And
the following lateness function:

\[
L(C_j, \bar{D}_j) = \frac{1}{2} \int_{0}^{1} ((C_j - d_{j_a}^{-})^+ + (C_j - \bar{d}_{j_a}^{-})^+) \, d\alpha + \frac{1}{2} \int_{0}^{1} ((C_j - d_{j_a}^{+})^- + (C_j - \bar{d}_{j_a}^{+})^-) \, d\alpha
\]

Where \([d_{j_a}^{-}, \bar{d}_{j_a}^{-}]\) according to \(\alpha\)-cut

To derive the fuzzy lateness cost function we have four cases:

**Case 1:**

- If \(C_j < d_{j_c}^{+}\)
  - For \(C_j - (d_{j_c}^{+} + (d_{j_c}^{-} - d_{j_f}^{-})\alpha)\)
  - If \(\alpha = 0\) then \(C_j - d_{j_f}^{-} < 0\)
  - If \(\alpha = 1\) then \(C_j - d_{j_c}^{+} < 0\)
  - For \(C_j - (d_{j_u}^{+} + (d_{j_c}^{-} - d_{j_u}^{-})\alpha)\)
  - If \(\alpha = 0\) then \(C_j - d_{j_u}^{-} < 0\)
  - If \(\alpha = 1\) then \(C_j - d_{j_u}^{+} < 0\)

Then by equation (1) we get

\[
L(C_j, \bar{D}_j) = \frac{1}{2} \int_{0}^{1} ((C_j - d_{j_a}^{-})^+ + \bar{d}_{j_a}^{-}) \, d\alpha
\]

\[
= \frac{1}{2} \int_{0}^{1} ((C_j - (d_{j_c}^{+} + (d_{j_c}^{-} - d_{j_f}^{-})\alpha)) + (C_j - (d_{j_u}^{+} + (d_{j_c}^{-} - d_{j_u}^{-})\alpha)) \, d\alpha
\]

\[
= \frac{1}{2} \int_{0}^{1} \left[ C_j \alpha - d_{j_f}^{-} \alpha - \frac{1}{2} d_{j_c}^{-} \alpha^2 + \frac{1}{2} d_{j_f}^{+} \alpha^2 + C_j \alpha - d_{j_u}^{-} \alpha - \frac{1}{2} d_{j_c}^{-} \alpha^2 + \frac{1}{2} d_{j_u}^{+} \alpha^2 \right] \, d\alpha
\]

\[
= \frac{1}{2} \int_{0}^{1} \left[ 2C_j - \frac{1}{2} d_{j_f}^{+} - d_{j_c}^{-} \right] \, d\alpha
\]

\[
= C_j - \frac{1}{4} \left( d_{j_f}^{+} + 2d_{j_c}^{-} + d_{j_u}^{+} \right)
\]

**Case 2:**

- If \(d_{j_f}^{-} \leq C_j < d_{j_c}^{+}\) then:
  - For \(C_j - (d_{j_f}^{-} + (d_{j_c}^{-} - d_{j_f}^{-})\alpha)\)
  - If \(\alpha = 0\) then \(C_j - d_{j_f}^{-} < 0\)
  - If \(\alpha = 1\) then \(C_j - d_{j_c}^{-} < 0\)
  - For \(C_j - (d_{j_u}^{-} + (d_{j_c}^{-} - d_{j_u}^{-})\alpha)\)
  - If \(\alpha = 0\) then \(C_j - d_{j_u}^{-} < 0\)
  - If \(\alpha = 1\) then \(C_j - d_{j_c}^{-} < 0\)
  - Than \(C_j - (d_{j_f}^{-} - d_{j_f}^{-})\alpha - d_{j_f}^{-} \geq 0\)
  - Than \(C_j - d_{j_f}^{-} < \alpha \leq \frac{C_j - d_{j_f}^{-}}{d_{j_f}^{-} - d_{j_f}^{-}} \)

Then \([0, \frac{C_j - d_{j_f}^{-}}{d_{j_f}^{-} - d_{j_f}^{-}}] \geq 0, \frac{C_j - d_{j_f}^{-}}{d_{j_f}^{-} - d_{j_f}^{-}}, 1\]

by using equation (1)

\[
L(C_j, \bar{D}_j) = \frac{1}{2} \int_{0}^{1} ((C_j - d_{j_a}^{-})^+ + (C_j - \bar{d}_{j_a}^{-})^+) \, d\alpha + \frac{1}{2} \int_{0}^{1} ((C_j - d_{j_a}^{+})^- + (C_j - \bar{d}_{j_a}^{+})^-) \, d\alpha
\]
\[
\frac{c_j-a_j^l}{a_j^l-a_j^l} = \frac{1}{2} \int_0^1 (C_j - (d_j^l + (d_j^l - d_j^l)\alpha)) d\alpha + \frac{1}{2} \int_0^1 \frac{c_j-a_j^l}{a_j^l-a_j^l} (C_j - (d_j^l + (d_j^l - d_j^l)\alpha)) d\alpha + \frac{1}{2} \int_0^1 (C_j - (d_j^u + (d_j^l - d_j^u)\alpha)) d\alpha
\]

\[
= \frac{1}{2} \int_0^1 (C_j - d_j^l\alpha - \frac{1}{2}(d_j^l - d_j^l)\alpha^2) [\frac{c_j-a_j^l}{a_j^l-a_j^l}] d\alpha + \frac{1}{2} \int_0^1 (C_j - d_j^u\alpha - \frac{1}{2}(d_j^l - d_j^u)\alpha^2) [\frac{c_j-a_j^l}{a_j^l-a_j^l}] d\alpha
\]

\[
= \frac{1}{2} [C_j\alpha - d_j^l\alpha - \frac{1}{2}d_j^l\alpha^2] + \frac{1}{2} [(2C_j - \frac{1}{2}d_j^u - \frac{1}{2}d_j^u) - \frac{1}{4}(2d_j^l + 2d_j^l + d_j^u)]
\]

Case (3)
If \(d_j^l \leq C_j \leq d_j^u\) then:

For \(C_j - (d_j^l + (d_j^l - d_j^l)\alpha)\)

If \(\alpha = 0\) then \(C_j - d_j^l > 0\)
If \(\alpha = 1\) then \(C_j - d_j^l < 0\)
If \(\alpha = 1\) then \(C_j - d_j^l > 0\)

Then \(C_j - (d_j^u + (d_j^l - d_j^u)\alpha) = 0\)

\[
(C_j - d_j^l)\alpha \geq d_j^u - C_j
\]

\[
\alpha \geq \frac{d_j^u - C_j}{d_j^u - d_j^l}
\]

Then \(\frac{d_j^u - C_j}{d_j^u - d_j^l} \leq 0, \quad \frac{d_j^u - C_j}{d_j^u - d_j^l} \geq 0\)

Then by equation (1)

\[
L(C_j, D_j) = \frac{1}{2} \int_0^1 ((C_j - d_j^l) + (C_j - d_j^l))^2 d\alpha + \frac{1}{2} \int_0^1 ((C_j - d_j^l) - (C_j - d_j^l))^2 d\alpha
\]

\[
= \frac{1}{2} \int_0^1 [(C_j - d_j^l + (d_j^l - d_j^l)\alpha) d\alpha + \frac{1}{2} \int_0^1 [C_j - (d_j^u + (d_j^l - d_j^u)\alpha)] d\alpha
\]

\[
+ \frac{1}{2} \int_0^1 \frac{d_j^u - C_j}{d_j^u - d_j^l} (C_j - (d_j^u + (d_j^l - d_j^u)\alpha)) d\alpha
\]

\[
= \frac{1}{2} [C_j\alpha - d_j^l\alpha - \frac{1}{2}(d_j^l - d_j^l)\alpha^2] + \frac{1}{2} [C_j\alpha - d_j^u\alpha - \frac{1}{2}(d_j^l - d_j^u)\alpha^2] + \frac{1}{2} [C_j\alpha - d_j^u\alpha - \frac{1}{2}(d_j^l - d_j^u)\alpha^2] + \frac{1}{2} [C_j\alpha - d_j^u\alpha - \frac{1}{2}(d_j^l - d_j^u)\alpha^2]
\]

Then we get

\[
C_j - \frac{1}{4} (d_j^l + 2d_j^l + d_j^u)
\]

Case (4)
: if \( d_j^u < C_j \)
For \( C_j - (d_j^l + (d_j^c - d_j^l) \alpha) \)
If \( \alpha = 0 \) then \( C_j - d_j^l > 0 \)
If \( \alpha = 1 \) then \( C_j - d_j^c > 0 \)
For \( C_j - (d_j^u + (d_j^c - d_j^u) \alpha) \)
If \( \alpha = 0 \) then \( C_j - d_j^u > 0 \)
If \( \alpha = 1 \) then \( C_j - d_j^c > 0 \)
Then by equation (1) we get
\[
L(C_j, \bar{B}_j) = \frac{1}{2} \int_0^1 ((C_j - d_j^l)^+ + (C_j - d_j^c)^+) \, d\alpha
\]
\[
= \frac{1}{2} \int_0^1 \left( (C_j - (d_j^l + (d_j^c - d_j^l) \alpha) + C_j - (d_j^u + (d_j^c - d_j^u) \alpha)) \, d\alpha \right)
\]
\[
= \frac{1}{2} \left[ C_j - d_j^l \alpha - \frac{1}{2} d_j^c \alpha^2 + \frac{1}{2} d_j^l \alpha^2 + C_j - d_j^u \alpha + \frac{1}{2} d_j^c \alpha^2 + \frac{1}{2} d_j^u \alpha^2 \right]_0^1
\]
\[
= \frac{1}{2} \left[ 2C_j - d_j^l - d_j^c - \frac{1}{2} d_j^u \right]
\]
\[
= C_j - \frac{1}{4} [d_j^l + 2d_j^c + d_j^u]
\]

Where \( C_j \) is completion time of jobs \( j \) under a sequence \( \delta \)

Using the traditional notion, we denote by

\[
1 \bar{B}_j = TFN \setminus \sum_{j=1}^{n} C_j + L_{\text{max}}
\]

the problem formulated by

\[
\text{Min } F = \text{Min} \sum_{j=1}^{n} C_j + \bar{L}_{\text{max}}
\]

subject to:

\[
C_j \geq P_j; \quad j=1,2,\ldots,n \quad \text{.........1)}
\]

\[
C_j = C_{j-1} + P_j; \quad j=2,3,\ldots,n
\]

\[
L_j = C_j - d_j
\]

4. **Local search algorithms**

In this section we will used three different methods of local search to solved multi-objective function on single machine formulated by

\[
1 \bar{B}_j = TFN \setminus \sum_{j=1}^{n} C_j + L_{\text{max}}
\]

Suppose \( T \) is a finite set and let a function \( f: T \rightarrow R \) has a solution \( t' \in T \) with \( f(t') \in f(t) \) for all \( t \in T[21] \). Local search is an iterative strategy which moves starting with one arrangement in \( S \) then onto the next as long as essential. To methodically look through \( S \). The potential moves from an answer \( s \) to next arrangement ought to be limited somehow or another. A local search(neighborhood) define by moves from initial solution by some sequence neighborhood change until found local optimum with improve each time value of the objective function. A local search starting with any feasible solution there exists a sequence of move to reach optimal solution. Neighborhood structures assume a vital part in local search as the time intricacy of a hunt depends on the size of the Neighborhood and the computational cost of the moves. This choice leads to the well-known iterative improvement method which may be formulated as follows:

4-1 Ant Colony Optimization Algorithms

**Step 0:** Initialization. Define the user specified parameters; the number of decision variables (n) (this number is sum of the number of green times as stage numbers at each intersection, the number of offset times as intersection numbers and common cycle time), the constraints for each decision variable, the size of ant colony (m), search space value (β) for each decision variable.
Step 1: Set t=1
Step 2: Generate the random initial signal timings, \( \psi(c, \theta, \phi) \)
within the constraints of decision variables.
Step 3: Distribute the initial green timings to the stages according to distribution rule as mentioned above.
At this step, randomly generated green timings at Step 2 are distributed to the stages according to generated cycle
time at the same step, minimum green and intergreen time.
Step 4: Get the network data and fixed set of link flows for TRANSYT-7F traffic model.
Step 5: Run TRANSYT-7F.
Step 6: Get the network PI. At this step, the PI is determined using TRANSYT-7F traffic model.
Step 7: If \( t = t_{\text{max}} \)
then terminate the algorithm; otherwise, \( t = t + 1 \)
and go to Step 2

4-2 The Bees Colony Optimization Algorithm

INPUT: \( n, ss, e, \text{nep}, \text{nsp}, \) Maximum of iterations.
Step 1. Initialize population with random solutions.
Step 2. Evaluate fitness of the population.
Step 3. REPEAT
Step 4. Select sites for neighborhood search.
Step 5. Recruit bees for selected sites (more bees for best e sites) and evaluate fitness’s.
Step 6. Select the fittest bee from each patch.
Step 7. Assign remaining bees to search randomly and evaluate their fitness’s.
Step 8. UNTIL stopping criterion is met.

4-3 Particle Swarm Optimization (PSO) Algorithm

Step 1. Initialize a population of particles with random positions and
velocities on d-dimensions in the problem space.
Step 2. PSO operation includes:
a. For each particle, evaluate the desired optimization fitness function
in d variables.
b. Compare particle's fitness evaluation with its pbest. If current
value is better than pbest, then set pbest equal to the current value,
and pai equals to the current location xi.
c. Identify the particle in the neighborhood with the best success so
far, and assign it index to the variable g.
d. Change the velocity and position of the particle according to
equations (2.1a) and (2.1b).
Step 3. Loop to step (2) until a criterion is met.

5. Computational results

In this section we using local search methods by using coding of matlab virgin R2017a were tasted and runs
on computer Pentium IV at 2.400GH\(z\), 4.00GB. in the below table given the results of optimal values by ant colony
optimization algorithms(ACO), the bees colony optimization algorithm(BA) and particle swarm optimization
algorithm(PSO) were \( n = 10, 50, 100, 150, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1500 \) as the following table
\( n: \) no. of jobs,
Ex: no. of examples,
PSO: particle swarm optimization algorithm
ACO: ant colony optimization algorithms
BA: the bees colony optimization algorithm
Time: The execution time of the problem (by seconds).

<table>
<thead>
<tr>
<th>n</th>
<th>X</th>
<th>ACO</th>
<th>TIME</th>
<th>BA</th>
<th>TIME</th>
<th>PSO</th>
<th>TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
<td>359.25</td>
<td>1.5</td>
<td>332.75</td>
<td>79.033</td>
<td>335.75</td>
<td>2.713</td>
</tr>
<tr>
<td></td>
<td>547.25</td>
<td>1.5</td>
<td>41</td>
<td>535.5</td>
<td>80.415</td>
<td>538.5</td>
<td>2.772</td>
</tr>
<tr>
<td>----</td>
<td>--------</td>
<td>-----</td>
<td>----</td>
<td>-------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>475.75</td>
<td>1.4</td>
<td>32</td>
<td>447.75</td>
<td>78.841</td>
<td>475.75</td>
<td>2.997</td>
</tr>
<tr>
<td></td>
<td>482</td>
<td>1.5</td>
<td>04</td>
<td>481</td>
<td>80.359</td>
<td>482</td>
<td>2.712</td>
</tr>
<tr>
<td></td>
<td>327.25</td>
<td>1.5</td>
<td>20</td>
<td>321.25</td>
<td>79.450</td>
<td>321.25</td>
<td>2.566</td>
</tr>
<tr>
<td>50</td>
<td>19047.75</td>
<td>5.3</td>
<td>71</td>
<td>12219</td>
<td>315.685</td>
<td>12898.5</td>
<td>9.250</td>
</tr>
<tr>
<td></td>
<td>11367</td>
<td>4.0</td>
<td>32</td>
<td>10794</td>
<td>314.183</td>
<td>11375.25</td>
<td>10.376</td>
</tr>
<tr>
<td></td>
<td>11394.75</td>
<td>4.1</td>
<td>06</td>
<td>10943</td>
<td>305.246</td>
<td>11351.75</td>
<td>9.460</td>
</tr>
<tr>
<td></td>
<td>10158.5</td>
<td>4.1</td>
<td>84</td>
<td>9264.25</td>
<td>328.664</td>
<td>9957</td>
<td>9.580</td>
</tr>
<tr>
<td></td>
<td>9486.75</td>
<td>4.1</td>
<td>45</td>
<td>8655.25</td>
<td>317.797</td>
<td>8834.75</td>
<td>9.558</td>
</tr>
<tr>
<td>100</td>
<td>40466.25</td>
<td>7.9</td>
<td>47</td>
<td>40045.5</td>
<td>640.410</td>
<td>40356</td>
<td>18.707</td>
</tr>
<tr>
<td></td>
<td>40746</td>
<td>7.6</td>
<td>92</td>
<td>39718.75</td>
<td>667.534</td>
<td>40353.25</td>
<td>18.925</td>
</tr>
<tr>
<td></td>
<td>47280.5</td>
<td>16.</td>
<td>254</td>
<td>47090.25</td>
<td>720.777</td>
<td>47819.75</td>
<td>18.811</td>
</tr>
<tr>
<td></td>
<td>41339.75</td>
<td>16.</td>
<td>811</td>
<td>39605.25</td>
<td>888.121</td>
<td>40185.5</td>
<td>26.512</td>
</tr>
<tr>
<td></td>
<td>42929.75</td>
<td>0.2</td>
<td>30</td>
<td>41354</td>
<td>895.917</td>
<td>41590</td>
<td>19.122</td>
</tr>
<tr>
<td>150</td>
<td>106426.75</td>
<td>12.</td>
<td>704</td>
<td>101898</td>
<td>1035.324</td>
<td>102993.25</td>
<td>29.996</td>
</tr>
<tr>
<td></td>
<td>107677.5</td>
<td>12.</td>
<td>643</td>
<td>104638.75</td>
<td>976.389</td>
<td>106106.75</td>
<td>29.279</td>
</tr>
<tr>
<td></td>
<td>98332.5</td>
<td>11.</td>
<td>621</td>
<td>94492.5</td>
<td>1094.254</td>
<td>97691</td>
<td>36.839</td>
</tr>
<tr>
<td></td>
<td>105958.5</td>
<td>14.</td>
<td>306</td>
<td>102151.25</td>
<td>1016.971</td>
<td>104651</td>
<td>29.078</td>
</tr>
<tr>
<td></td>
<td>105167.5</td>
<td>11.</td>
<td>713</td>
<td>103512.75</td>
<td>1014.326</td>
<td>104676</td>
<td>29.548</td>
</tr>
<tr>
<td>200</td>
<td>194745.75</td>
<td>13.</td>
<td>483</td>
<td>189046</td>
<td>1214.389</td>
<td>192709.5</td>
<td>36.380</td>
</tr>
<tr>
<td></td>
<td>197056.75</td>
<td>13.</td>
<td>807</td>
<td>192809.75</td>
<td>1217.649</td>
<td>194437.5</td>
<td>35.132</td>
</tr>
<tr>
<td></td>
<td>185655.75</td>
<td>13.</td>
<td>405</td>
<td>180032</td>
<td>1202.807</td>
<td>181268.25</td>
<td>34.713</td>
</tr>
<tr>
<td></td>
<td>177984.75</td>
<td>13.</td>
<td>260</td>
<td>170486</td>
<td>1221.464</td>
<td>173903.25</td>
<td>34.989</td>
</tr>
<tr>
<td></td>
<td>195065.75</td>
<td>12.</td>
<td>449</td>
<td>185542.75</td>
<td>1236.416</td>
<td>189810</td>
<td>35.042</td>
</tr>
<tr>
<td>300</td>
<td>458314</td>
<td>15.</td>
<td>151</td>
<td>449284</td>
<td>1319.829</td>
<td>454636.5</td>
<td>37.921</td>
</tr>
<tr>
<td></td>
<td>439736.5</td>
<td>15.</td>
<td>196</td>
<td>434399.25</td>
<td>1315.083</td>
<td>439248.25</td>
<td>38.041</td>
</tr>
<tr>
<td></td>
<td>442944.5</td>
<td>13.</td>
<td>564</td>
<td>437213.5</td>
<td>1313.724</td>
<td>440301.25</td>
<td>38.925</td>
</tr>
<tr>
<td></td>
<td>456717</td>
<td>14.</td>
<td>818</td>
<td>450442.5</td>
<td>1031.251</td>
<td>454151</td>
<td>37.634</td>
</tr>
<tr>
<td></td>
<td>464816.5</td>
<td>15.</td>
<td>309</td>
<td>460248.25</td>
<td>1316.386</td>
<td>461392</td>
<td>39.110</td>
</tr>
<tr>
<td>400</td>
<td>744315</td>
<td>15.879</td>
<td>735855</td>
<td>1479.297</td>
<td>745847</td>
<td>44.631</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>----------</td>
<td>--------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>783312.25</td>
<td>15.808</td>
<td>781248.75</td>
<td>1468.212</td>
<td>782342.5</td>
<td>44.284</td>
<td></td>
<td></td>
</tr>
<tr>
<td>781578</td>
<td>15.501</td>
<td>767425.25</td>
<td>1450.819</td>
<td>774938.25</td>
<td>41.870</td>
<td></td>
<td></td>
</tr>
<tr>
<td>785648</td>
<td>16.215</td>
<td>771059.5</td>
<td>1356.176</td>
<td>775431.5</td>
<td>40.409</td>
<td></td>
<td></td>
</tr>
<tr>
<td>785648</td>
<td>17.130</td>
<td>771059.5</td>
<td>1381.142</td>
<td>775431.5</td>
<td>41.457</td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>1203537.25</td>
<td>17.349</td>
<td>1180203.2</td>
<td>1488.563</td>
<td>1197999.2</td>
<td>45.540</td>
<td></td>
</tr>
<tr>
<td>1188662</td>
<td>15.987</td>
<td>1177537.2</td>
<td>1489.234</td>
<td>1182720.7</td>
<td>44.830</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1242205.25</td>
<td>16.018</td>
<td>1238348</td>
<td>1486.792</td>
<td>1251027.5</td>
<td>42.651</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1216479.25</td>
<td>16.662</td>
<td>1218784.8</td>
<td>1488.936</td>
<td>1222257</td>
<td>42.877</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1153541.25</td>
<td>16.499</td>
<td>1145806.7</td>
<td>1492.290</td>
<td>1157400.2</td>
<td>42.892</td>
<td></td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>1728530.75</td>
<td>17.593</td>
<td>1710070.7</td>
<td>1631.703</td>
<td>1724363.7</td>
<td>46.377</td>
<td></td>
</tr>
<tr>
<td>1728553</td>
<td>17.661</td>
<td>1714604.5</td>
<td>1563.504</td>
<td>1725398.5</td>
<td>47.658</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1791190</td>
<td>19.074</td>
<td>1782975.5</td>
<td>1562.558</td>
<td>1789210.2</td>
<td>46.155</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1742749.75</td>
<td>18.370</td>
<td>1716733</td>
<td>1559.189</td>
<td>1737821.5</td>
<td>46.089</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1727570.25</td>
<td>18.896</td>
<td>1712956.5</td>
<td>1558.625</td>
<td>1733046</td>
<td>46.586</td>
<td></td>
<td></td>
</tr>
<tr>
<td>700</td>
<td>2256608.75</td>
<td>18.942</td>
<td>2234514.7</td>
<td>1749.084</td>
<td>2249483.7</td>
<td>51.035</td>
<td></td>
</tr>
<tr>
<td>2489438.5</td>
<td>19.565</td>
<td>2451308.7</td>
<td>1741.403</td>
<td>2474523.2</td>
<td>49.727</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2484713</td>
<td>19.717</td>
<td>2473856.2</td>
<td>1748.992</td>
<td>2494122.5</td>
<td>52.060</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2413287.5</td>
<td>19.003</td>
<td>2385101.2</td>
<td>1745.376</td>
<td>2405379.5</td>
<td>51.769</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2457635.25</td>
<td>19.265</td>
<td>2445536.5</td>
<td>1767.265</td>
<td>2452116.7</td>
<td>50.072</td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>3104373.5</td>
<td>21.660</td>
<td>3077404.7</td>
<td>1913.805</td>
<td>3093468.5</td>
<td>54.059</td>
<td></td>
</tr>
<tr>
<td>3262420.75</td>
<td>20.521</td>
<td>3229612.5</td>
<td>1893.188</td>
<td>3262025.5</td>
<td>52.988</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3116125.25</td>
<td>20.541</td>
<td>3086304.5</td>
<td>1839.503</td>
<td>3127753.2</td>
<td>54.074</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3211890</td>
<td>20.490</td>
<td>3182970</td>
<td>1814.507</td>
<td>3199328.2</td>
<td>52.170</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3058229</td>
<td>20.328</td>
<td>3042580.2</td>
<td>1848.626</td>
<td>3060590.5</td>
<td>53.444</td>
<td></td>
<td></td>
</tr>
<tr>
<td>900</td>
<td>3889877.5</td>
<td>23.545</td>
<td>3847421.7</td>
<td>2070.062</td>
<td>3865368.5</td>
<td>59.861</td>
<td></td>
</tr>
<tr>
<td>4120634</td>
<td>23.630</td>
<td>4095036.7</td>
<td>2081.621</td>
<td>4116886</td>
<td>58.882</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3953056.5</td>
<td>33.680</td>
<td>3954258.2</td>
<td>2233.430</td>
<td>3962795</td>
<td>82.996</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3829251</td>
<td>24.487</td>
<td>3796587.5</td>
<td>2906.174</td>
<td>3799322.7</td>
<td>62.339</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6. Conclusion

In this paper we solved problem Q which NP-heard on single machine with fuzzy due date by using local search methods to get approximation solution and a compare between ant colony optimization algorithms(ACO), the bees colony optimization algorithm (BA) and particle swarm optimization algorithm (PSO) form where Accuracy value of objective function and time of processing in coding of matlab.

Reference
10. for the Single Machine Total Tardiness Problem, 0-7803-5536-9/9991999 IEEE.
13. Hamid Allaoui and Samir Lamouri, USING JOHNSON’S ALGORITHM TO APPROXIMATE SOME FLOW SHOPS SCHEDULING PROBLEMS WITH UNAVAILABILITY PERIODS, Copyright c2006 IFAC
22. Chiwen Qu, Yanming Fu, Zhongjun Yi, and Jun Tan, Solutions to No-Wait Flow Shop Scheduling Problem Using the Flower Pollination Algorithm Based on the Hormone Modulation Mechanism, Hindawi Complexity Volume 2018, Article ID