Fuzzy Distance Function and its Applications on Cost Function $\sum w_j C_j^2 + \widetilde{L}_{\max}(C_j, \widetilde{D}_j)$

Hanan A. Cheachan
Department of Mathematics,
College of Sciences,
University of Mustansiriyah,
Baghdad,

Faria A. Cheachan
Department of Mathematics,
College of Sciences,
University of Mustansiriyah,
Baghdad,

Hussam A.A. Mohammed
Department of Mathematics,
College of Education University of Kerbala,

Abstract:-

This paper seek to determine the feasibility of using fuzzy distance function concepts which introduced by Lam and Cai to solve single machine scheduling multi objective functions subject to job dependents due dates and we compare and test different local search method (Threshold Accepted (TA), Tabu Search (TS) and Simulated Annealing (SA)) computational experience 1000 jobs with reasonable time.

المستخلص:-

في هذا البحث نريد أن نبحث عن جدوى العمليةِ لاستعمال مفاهيمِ دالة المسافةِ الضبابيةِ التي قدّمتْ مِن قِبل Lam و Cai لحَلّ مسائل جدولة الماكنة الواحدة والتي تكون ذات دوال متعددة الأهداف معتمدة على أزمان مثالية للأعمال مستقلة ونحن نُقارنُ ونَختبرُ طريقةَ بحثِ محليّةِ مختلفةِ (Threshold Accepted (TA), Tabu Search (TS) and Simulated Annealing (SA)) تجربة حسابية لـ 1000 عمل بالوقتِ المعقولِ.

1. Introduction:-

In 1965 concept of fuzzy sets was introduced by Zadeh [16]. The notions of fuzzy metric spaces were introduce by Kramosil and Michalek [11]. From then on many researches on fuzzy metric spaces have been carried out [4,5]. Lam and Cai gave a fuzzy function for measuring the distance between fuzzy numbers and also showed by experiments their distance function given very good approximation to the expected distance is numerous situations [12]. We study the problem of scheduling n jobs in a single machine with multi-objective and subjected to job-dependents due dates. Recently, some scheduling models with fuzzy due dates have been studied. Relevant works include that of Ishii et. al. [9], Han et. al. [6], Ishibuchi et. al. [7,8], Stanfield et. al. [15] and Lam and Cai [13]. The objective of [6,7,8], are to maximize the total degree of satisfaction with respect to fuzzy due dates, while [15] seeks to find the optimal schedule among those that do not exceed the maximum acceptable possibility of lateness in a problem involving fuzzy due dates. In [13], they were in interested in the direct generalization of the traditional nonlinear lateness cost function with the due dates being fuzzy numbers also they gave a notation of lateness cost function with the due dates. In this paper we are interested in the direct generalization of the traditional multi-objective function measure with the due dates being fuzzy numbers, specifically we will consider the situation when a due date is represented by a triangular fuzzy number.

2. Preliminaries:-

In this section we give some basic concepts that we needed then later.

2.1 Definition [13]:-

Let $\mathbb{R}$ be real line. A fuzzy set $\widetilde{A}$ from $\mathbb{R}$ into [0,1] which satisfy the following conditions:-

1- There exists $x_0 \in \mathbb{R}$ such that $\widetilde{A}(x_0) = 1$.

2- $\widetilde{A}(\lambda x + (1-\lambda)y) \geq \min\{\widetilde{A}(x), \widetilde{A}(y)\}$, where $x, y \in \mathbb{R}$ and $\lambda \in [0,1]$ is said to be a fuzzy number.

2.2 Note [13]:-

1- The support of a fuzzy number $\tilde{A}$ is the set of points x in R such that $\tilde{A}(x) > 0$, we assume support of fuzzy number $\tilde{A}$ as a closed bounded subset [a,b] of R.

2- an α -cut of a fuzzy number $\tilde{A}$ is the set of points x in R such that $\tilde{A}(x) \geq \alpha$, we assume on α -cut of a fuzzy number $\tilde{A}$ as a closed interval $[\underline{a}_\alpha, \overline{a}_\alpha]$.

3- The fuzzy number $\tilde{A}$ is defined, in general, as follows:

$$\tilde{A}(x) = \begin{cases} 0 & for \quad x < a \\ f_{\tilde{A}}(x) & for \quad a \leq x < c \\ 1 & for \quad c \leq x \leq d \\ g_{\tilde{A}}(x) & for \quad d < x \leq b \\ 0 & for \quad b < x \end{cases}$$

Where $f_{\tilde{A}}$ *and* $g_{\tilde{A}}$ are respectively, non-decreasing, and non-increasing functions.

4- The distance between fuzzy numbers is defined as follows:

$$\tilde{d}(\tilde{A}, \tilde{B}) = \tilde{d}_\tau(\tilde{A}, \tilde{B}) + \tilde{d}_\varepsilon(\tilde{A}, \tilde{B})$$

$$\tilde{d}\tau(\tilde{A}, \tilde{B}) = \frac{1}{2}\int_0^1 \{(\underline{a}_\alpha - \underline{b}_\alpha)^+ + (\overline{a}_\alpha - \overline{b}_\alpha)^+\} d\alpha,$$

$$(x)^+ = \begin{cases} x \ \ if \ \ x < 0 \\ 0 \ \ if \ \ x < 0 \end{cases}$$

$$\tilde{d}_\varepsilon(\tilde{A}, \tilde{B}) = \frac{1}{2}\int_0^1 \{(\underline{a}_\alpha - \overline{b}_\alpha)^- + (\overline{a}_\alpha - \overline{b}_\alpha)^-\} d\alpha,$$

$$(x)^- = \begin{cases} 0 \ \ if \ \ x \geq 0 \\ x \ \ if \ \ x < 0 \end{cases}$$

and $[\underline{a}_\alpha, \overline{a}_\alpha]$, $[[\underline{b}_\alpha, \overline{b}_\alpha]$ are respectively, the α -cuts of $\tilde{A}$ *and* $\tilde{B}$.

3. Problem Formulation:-

Suppose that there are n independent jobs to be processed on a single machine. Job j, $j = 1, \ldots, n$ requires p_j and w_j units of processing time and weight respectively. Each job is assigned with a fuzzy due date $\tilde{D}_j$, which is a triangular fuzzy number (TFN). The machine can process at most job at a time, and the problem is to determine a sequence λ to process the jobs so that $\sum_{j=1}^{n} w_{(\sigma_j)} C_{(\sigma_j)} + \tilde{L}_{\max}$.

Specifically, we address the case where $\tilde{D}_j$ is a TFN which is defined in terms of three numbers $[d_j^l, d_j^c, d_j^u]$ as follows:

$$\tilde{D}_j(x) = \begin{cases} 0 & if \ x < d_j^l \\ \dfrac{(x - d_j^l)}{(d_j^c - d_j^l)} & if \ d_j^l \leq x < d_j^c \\ \dfrac{(d_j^u - x)}{(d_j^u - d_j^c)} & if \ d_j^c \leq x < d_j^u \\ 0 & if \ d_j^u \leq x \end{cases}$$

The possible range of the fuzzy due date is $[d_j^l, d_j^u]$, when the maximum value occurs at the point d_j^c. Accordingly, the range $[d_j^l, d_j^u]$, and the point d_j^c are called the support and the core of $\tilde{D}_j$, respectively.

Assuming the due date is a crispy number d_j for the time being then the cost function we are interested to study has the following form $\sum_{j=1}^{n} w_j C_j^2 + \tilde{L}_{\max}$

Where C_j and w_j are respectively the completion time and weights of job *j*, $j = 1, \ldots, n$.

If the due date is a fuzzy number, then $\tilde{L}_{\max}$ is a fuzzy function and our problem become $\sum w_j C_j^2 + \tilde{L}_{\max}(C_j, \tilde{D}_j)$ using the traditional notion, we denote the problem formulated in this sections as $1|\tilde{D}_j = TFN|\sum w_j C_j^2 + \tilde{L}_{\max}(C_j, \tilde{D}_j)$.

4. Model Development:-

In this section, we give the formal of cost function $\sum w_j C_j^2 + \tilde{L}_{\max}(C_j, \tilde{D}_j)$ which is depended in work. We start to give formal for $\tilde{L}(C_j, \tilde{D}_j)$. In [13] gave the notation of use fuzzy distance in $\tilde{L}(C_j, \tilde{D}_j)$ without details here we give some details of derive $\tilde{L}(C_j, \tilde{D}_j)$ and find the final formula of $\tilde{L}(C_j, \tilde{D}_j)$ which is needed later. By replacing fuzzy distance $\tilde{d}$ by $\tilde{L}$, $\tilde{A}$ with C_j, the crisp completion time of job *j* and $\tilde{B}$ *with* $\tilde{D}_j$ fuzzy due date of the job *j*, the lateness of the job can be evaluated using the following lateness function:

$$\tilde{L}(C_j, \tilde{D}_j) = \tilde{d}_\tau(C_j, \tilde{D}_j) + \tilde{d}_\varepsilon(C_j, \tilde{D}_j)$$

$$= \frac{1}{2}\int_0^1 \{(C_j - \underline{d}_{j\alpha})^+ + (C_j - \overline{d}_{j\alpha})^+\} d\alpha + \frac{1}{2}\int_0^1 \{(C_j - \underline{d}_{j\alpha})^- + (C_j - \overline{d}_{j\alpha})^-\} d\alpha$$

Where $[\underline{d}_{j\alpha}, \overline{d}_{j\alpha}]$ is the α-cut of $\tilde{D}_j$

Hence:

$$\tilde{L}(C_j, \tilde{D}_j) = \frac{1}{2}\int_0^1 \{(C_j - [(d_j^c - d_j^l)\alpha + d_j^l])^+ + (C_j - [-(d_j^u - d_j^c)\alpha + d_j^u])^+\} d\alpha +$$

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

We must discuss the following four cases:-

1- If $C_j < d_j^l$ the $c_j - [(d_j^c - d_j^l)\alpha + d_j^l] < 0$ and $C_j - [-(d_j^u - d_j^c)\alpha + d_j^u] < 0$

In this case:

$$\tilde{L}(C_j, \tilde{D}_j) = \frac{1}{2}\int_0^1 \{(C_j - [(d_j^c - d_j^l)\alpha + d_j^l]) + (C_j - (-(d_j^u - d_j^c)\alpha + d_j^u)\} d\alpha$$

$$= \frac{1}{2}[2C_j - d_j^c - \frac{1}{2}d_j^l - \frac{1}{2}d_j^u]$$

$$= C_j - [\frac{1}{2}d_j^c + \frac{1}{4}d_j^l + \frac{1}{4}d_j^u]$$

Thus $\tilde{L}(C_j, \tilde{D}_j) = C_j - \frac{1}{4}[2d_j^c + d_j^l + d_j^u]$

2- If $d_j^l \le C_j < d_j^c$ then $C_j - [-(d_j^u - d_j^c)\alpha + d_j^u] < 0$ for each α

In this case:

$$\tilde{L}(C_j, \tilde{D}_j) = \frac{1}{2}\int_0^1 \left(C_j - \left[\left(d_j^c - d_j^l\right)\alpha + d_j^l\right]\right)^+ d\alpha + \frac{1}{2}\int_0^1 \left(C_j - \left[\left(d_j^c - d_j^l\right)\alpha + d_j^l\right]\right)^- d\alpha$$

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

$$= \frac{1}{2}\int_0^{\frac{C_j - d_j^l}{d_j^c - d_j^l}} C_j - [(d_j^c - d_j^l)\alpha + d_j^l] d\alpha + \frac{1}{2}\int_{\frac{C_j - d_j^l}{d_j^c - d_j^l}}^1 C_j - [(d_j^c - d_j^l)\alpha + d_j^l] d\alpha + \frac{1}{4}[2C_j - d_j^u - d_j^c]$$

$$= C_j - \frac{1}{2}d_j^c - \frac{1}{4}d_j^l - \frac{1}{4}d_j^u$$

Thus, $\tilde{L}(C_j, \tilde{D}_j) = C_j - \frac{1}{4}[2d_j^c + d_j^l + d_j^u]$

3- If $d_j^c \le C_j \le d_j^u$ then $C_j - [(d_j^c - d_j^i)\alpha + d_j^l] \ge 0$ for each α

In this case

$$\tilde{L}(C_j, \tilde{D}_j) = \frac{1}{2}\int_0^1 \{(C_j - [(d_j^c - d_j^l)\alpha + d_j^l)^+ + (C_j - [-(d_j^u - d_j^c)\alpha + d_j^u])^+\} d\alpha$$

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

$$= \frac{1}{2}C_j - \frac{1}{4}d_j^c - \frac{1}{4}d_j^i + \frac{1}{2}\int_{\frac{d_j^u - C_j}{d_j^u - d_j^c}}^1 (C_j - [-(d_j^u - d_j^c)\alpha + d_j^u])^+ d\alpha + \frac{1}{2}\int_0^{\frac{d_j^u - C_j}{d_j^u - d_j^c}} (C_j - [-(d_j^u - d_j^c)\alpha + d_j^u])^- d\alpha$$

$$= C_j - \frac{1}{2}d_j^c - \frac{1}{4}d_j^l - \frac{1}{4}d_j^u$$

Thus:

$$\tilde{L}(C_j, \tilde{D}_j) = C_j - \frac{1}{4}[2d_j^c + d_j^l + d_j^u]$$

4- If $d_j^u \le C_j$ then $C_j - [(d_j^c - d_j^l)\alpha + d_j^l] > 0$ and $C_j - [-(d_j^u - d_j^c)\alpha + d_j^u] > 0$

In this case:

$$\tilde{L}(C_j, \tilde{D}_j) = \frac{1}{2}\int_0^1 \{(C_j - [(d_j^c - d_j^l)\alpha + d_j^l)^+ + (C_j - [-(d_j^u - d_j^c)\alpha + d_j^u])^+\} d\alpha$$

$$= C_j - \frac{1}{2}d_j^c - \frac{1}{4}d_j^l - \frac{1}{4}d_j^u$$

Thus:

$$\tilde{L}(C_j, \tilde{D}_j) = C_j - \frac{1}{4}[2d_j^c + d_j^l + d_j^u]$$

Therefore, from four cases we have $\tilde{L}(C_j, \tilde{D}_j) = C_j - \frac{1}{4}[2d_j^c + d_j^l + d_j^u]$

5. Local Search Techniques:-

In this section we study local search techniques which are useful tools for solving multi-objective single machine scheduling $1|\tilde{D}|\sum_{j=1}^{n} w_j C_j^2 + \tilde{L}_{\max}$.

For a given finite set S and given function Z: $S\rightarrow \mathbb{R}$ one has find solution $s' \in S$ with $Z(s') \leq Z(s)$ for all $s \in S$ [1]. Local search is an iterative procedure which moves from one solution in S to another as long as necessary. In order to systematically search through S. The possible moves from a solution s to next solution should be restricted in some way. To describe such restrictions, we introduce a neighborhood structure $N: S \rightarrow 2^s$ on S. For each $s \in S, N(s)$ describes the subset of solutions which can be reached in one step by moving from s. The set $N(s)$ is called the neighborhood of s. A neighborhood structure N may be represented by a directed graph G= (V,A) where V = S and $(s,t) \in A \leftrightarrow t \in N(S)$, G is called the neighborhood graph of the neighborhood structure. Usually it is not possible to store the neighborhood graph because S has an exponential size. To overcome this difficulty, a set H of allowed modifications $F: S \rightarrow S$ is introduced. For a given solution s, the neighborhood of s can now be defined by $N(s) = \{F(s) \mid F \in H\}$ using these definitions using these definitions, a local search method may be described as follows. In each iteration we start with a solution $s \in S$ and choose a solution $s' \in N(s)$. Based on the values $Z(s)$ and $Z(s')$, we choose a starting solution of the next iteration. According to different criteria used for the choice of the starting solution of the next iteration we get different types of local search techniques. The simplest choice is to take the solution with the smallest value of the cost function. This choice leads to the well-known iterative improvement method which may be formulated as follows.

6. Algorithm Iterative Improvement [14]

1. Choose an initial solution $s' \in S$
2. Repeat
3. $s = s'$
4. Generate the best solution $s' \in N(s)$.
5. Until $Z(s') \geq Z(s)$

A method which seeks to avoid being trapped in a local minimum is simulated annealing. It is a randomized method because:

- s' is chosen randomly from $N(s)$, and in the i-th step s' is accepted with probability

$$\min\left\{1, \exp\left(\frac{-Z(s') - Z(s)}{c_i}\right)\right\}$$

Where (c_i) is a sequence of positive control parameters with $\lim_{n\to\infty} c_i = 0$.

7. Algorithm Simulated Annealing [10]

1. i=0
2. Choose an initial solution $s \in S$
3. Best:=$Z(s)$;
4. $s' := s$;
5. Repeat
6. Generate randomly a solution $s' \in N(s)$.
7. If random $[0,1] < \min\left\{1, \exp\left(\frac{-Z(s') - Z(s)}{c_i}\right)\right\}$ Then $s := s'$
8. If $Z(s') <$ best then Begin
9. $s := s'$
10. Best:= $Z(s')$ End
11. $c_{i+1} = g(c_i)$;
12. i=i+1
13. Until stop criterion

A detailed discussion of how one should define the control function g and the stop criterion for practical application is described in Aarts and Lenstra [1]. One possibility is to stop offer a given amount of computation time. A variant of simulated annealing is the threshold acceptance method [2]. It differs from s' is accepted if the difference $Z(s')-Z(s)$ is smaller than some non-negative threshold *t*, *t* is a positive control parameter which is gradually reduced.

8. Threshold Acceptance Algorithm [2]

1. *i*:=0
2. Choose an initial solution $s \in S$
3. Best:=*Z*(*s*)
4. $s := s'$
5. REPEAT
6.Generate randomly a solution
 $s' \in N(s)$.
7.If $Z(s')-Z(s)$<t then $s := s'$
8. If $Z(s') <$ best then
 Begin
9. $s := s'$
10. Best:= $Z(s')$
 End
11. $t_i - 1 : g(t_i)$
12. *i*=*i*+1
13. Until stop criterion *g* is non-negative function with *g*(*t*)<*t* for all *t*.

9. Tabu-search Algorithm [3]

1. Choose an initial solution $s \in S$
2. Best:=*Z*(*s*)
3. Tabu-list:=Q
4. REPEAT
5. Cand:={ $s' \in N(s) \mid$ the move from *s* to *s'* is not tabu or *s'* satisfies the aspiration criterion}
6. Generate a solution $s' \in cand(s)$
7. Update the tabu list
8. $s := s'$
9. If Z(s) < best then
 Begin
10. Best :=Z(*s*)
 END
11. Until stop criterion

Different stopping criteria and procedures for updating the tabu list T can be developed. We also have the freedom to choose a method for generating a solution $s' \in cand(s)$. A simple strategy is to choose the best possible s' with respect to function *Z*. $Z(s') = \min\{Z(s') \mid s' \in cand(s)\}$.

However, this simple strategy can be much too time consuming. Since the cardinality of the set *cand*(*s*) may be very large. For these reasons we may restrict our choice to a subset V$\subseteq$*cand*(*s*).
$Z(s') = \min\{Z(s') \mid s' \in V\}$.

10. Computational Results:-

Local search methods were tested by coding then in Matlab R2009b and runs on a Pentium IV at 2.00GHz, 2.92GB computer. The tested problem instances are generated as follows:

For n = 10, 20, 30, 50, 100, 200, 500 & 1000, and integer p_j for $j \in N = \{1,2,...,n\}$ is generated by randomly selecting integers from interval [1,10] in our experiments, problem instances of 5-10 jobs were randomly. The processing times were generated uniformly in the range [10,30]. The fuzzy due dates were generated randomly with the support in the range of [1,W], where W was also randomly selected among the values of {10, 20,30,40,50}.

In the following table (1) show the efficiency local search heuristic methods (Threshold Accepting (TH), Tabu Search (TS) and Simulated Annealing (SA)) have been approached in terms of comparable rate of value. TH gives the best solution for the half iterations then the SA comes in the second and TS was worst.

Table (1) Compares of local search methods

N	SA	TH	TS
10	251142.5	251142.5	251142.5
20	4097678	4096478.75	4098747.48
30	25642835.58	25606847.03	25618424.18
50	239641176.1	241958052.75	240931717.48
100	21191969369.28	21233398463.83	21309455939.13
200	431131991642.35	430567821821.43	431820301406.78
500	31380084989051.1	31118145892512.3	31792336699031.7
1000	2955526952579040	2974434398976050	2966719399295070

We plot our results by using coding Matlab programming (semilogy plot data as logarithmic scales for the y-axis). In figure (1a) the difference between our approach LS it's not clear because the dig result and small difference therefor, in figure (1b) show the difference between LS for 50 jobs.

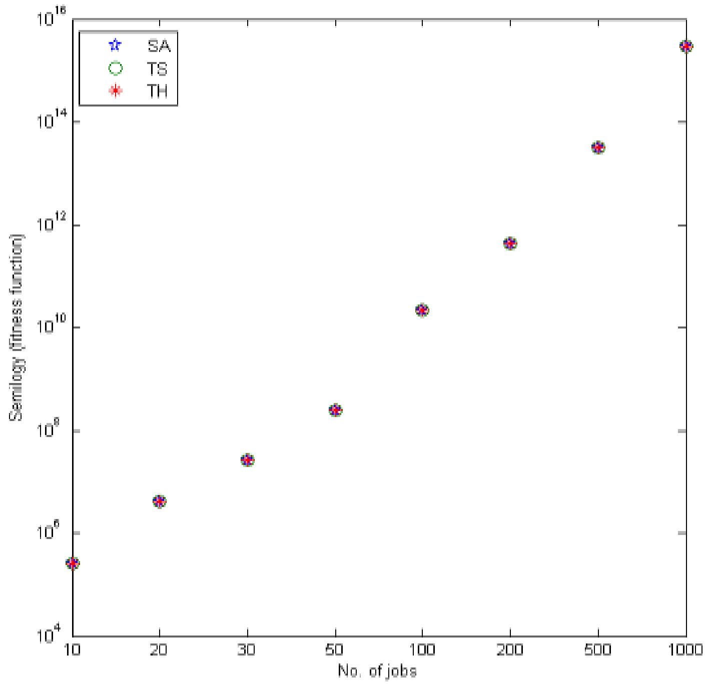

Figure (1a) Local search

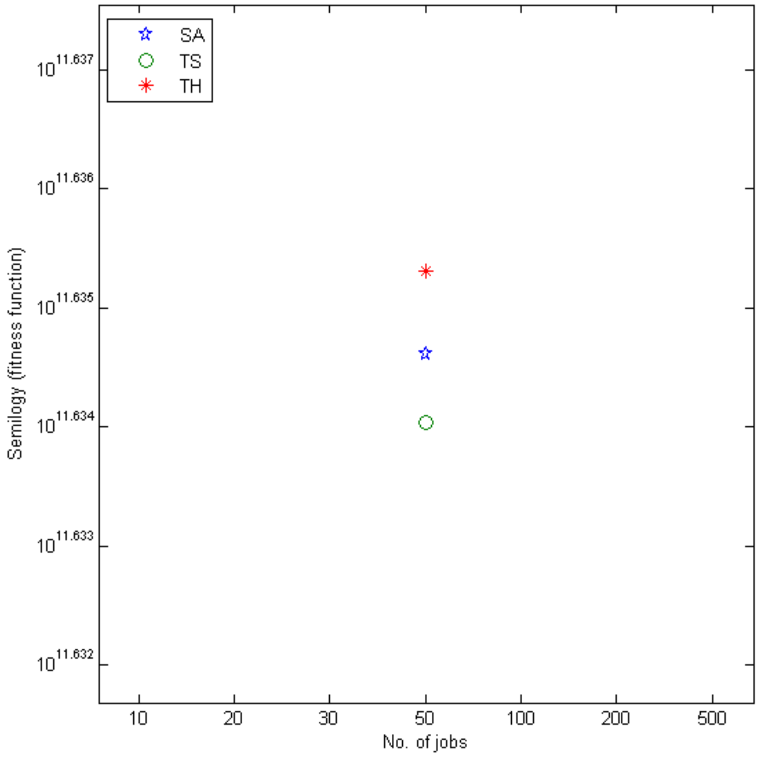

Figure (1b) Local search for 50 jobs

In the following table (2) show the efficiency local search heuristic methods (SA, TH and TS) have been approached in terms of comparable rate of times. TH gives the best times for the small jobs and SA gives the best times for the large jobs while TS gives the worst times.

Table (2) Compares times of local search methods

N	SA	TH	TS
10	0.024628	0.024408	0.067109
20	0.024236	0.023673	0.07474
30	0.024374	0.023866	0.087488
50	0.025001	0.024556	0.130415
100	0.024671	0.024712	0.225104
200	0.025081	0.026379	0.509513
500	0.029063	0.045633	1.570016
1000	0.035899	0.060478	3.122682

11. Concluding Remarks:-

We have developed a new model to formulate the situation where jobs with fuzzy due dates are to be scheduled on a single machine. The local search methods used to solve all the large problems the result show the robustness and flexibility of local search heuristics.

References:-

[1] Aarts, E. H. L. and Lenstra, J. K., "*Local search in combinatorial optimization*", Wiley Inter Science, New York, 1997.

[2] Dueck, G. and Scheuer, T., "*Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing*", Journal of Computational Physics, Vol. 90, PP.161-175.

[3] Glover, F., "*A user's guide to tabu search*", Annals of Operations Research, Vol. 41, PP. 3-28, 1994.

[4] Gregori, V. and Romaguera, S., "*Some properties of Fuzzy metric spaces*", Fuzzy Sets and Systems, Vol.115, PP.485-489, 2000.

[5] Gregori, V. and Romaguera, S., "*Fuzzy Quasi-metric spaces*", Applied General Topology, Vol.5, PP.129-136, 2004.

[6] Han, S., Ishii, H. and Fujii, S., "*On machine scheduling system with fuzzy due dates*", European Journal Operation Research, Vol.79, PP. 1-12, 1994.

[7] Ishibuchi, H., Yamanoto, Murata, T. and Tanaka, H., "*Genetic algorithms and neighborhood search algorithm, for fuzzy flow shop scheduling problems*", Fuzzy Sets and System, Vol. 94, PP.81-100, 1994.

[8] Ishibuchi, H., Yamamoto, S., Misak and Tanaka, H., "*Local search algorithm for flow shop scheduling with fuzzy due dates*", International Journal Product Ecom., Vol.33, PP. 53-66, 1994.

[9] Ishii, H., Tada, M., and Masuda, T., "*Two scheduling problems with fuzzy due dates*", Fuzzy Set and Systems, Vol.46, PP.339-347,1992.

[10] Kirkpatrick, A., Gelatt, C., Vechi, M., "*Optimization by simulated annealing*", Science, Vol. 220, PP.671-680, 1983.

[11] Kramosil, O., Michalek, J., "*Fuzzy metric and statistical metric spaces*", kybemetica, Vol. 11, PP. 326-334, 1975.

[12] Lam, S. and Cai, X., "*Distance measures of fuzzy numbers computational intelligence and applications*", Springer, Berlin, PP.207-214, 1999.

[13] Lam, S. and Cai, X., "*Single machine scheduling with nonlinear lateness cost functions and fuzzy due dates*", Nonlinear Analysis Real World Application, Vol.3, PP.307-316, 2002.

[14] Lee, C., and Kim, Y., "*Search heuristic for resource constrained project scheduling*", Journal of Operational Research Society, Vol.47, PP. 678-689.

[15] Stanfield, P., King, R. and Joines, J., "*Scheduling arrivals to a production system in a fuzzy environment*", European Journal Operation Research, Vol.93, PP.75-87, 1996.

[16] Zadeh, L., "*Fuzzy sets*", Information and Control, Vol.8, PP.338-353, 1965.