Multi-Criteria Scheduling Optimization Using Fuzzy Logic

Yaron Klein
Department of Electrical Engineering - Systems
Tel-Aviv University
klein@eng.tau.ac.il

Gideon Langholz
Department of Electrical Engineering - Systems
Tel-Aviv University
langholz@eng.tau.ac.il

ABSTRACT

In this paper we introduce a fuzzy based engine for the scheduling problem. We assume that a human planner is unable to determine the weights for a multi-objective system but is able only to compare between pairs of objectives according to a limited vocabulary. We convert this vocabulary to fuzzy terminology and apply a fuzzy based optimum search for a solution to the scheduling problem. We demonstrate our approach using a real world scheduling system.

1. INTRODUCTION

Scheduling has been examined in the operations research literature since the early fifties. It can be defined as allocation of resources over time to perform a collection of tasks.

In general, scheduling is the problem of sequencing a set of operations and allocating them to temporal slots without violating technical and capacitive constraints. These operations may be operations on machines in a factory, train or car movement in transportation, lessons at school, or computer processes performed on the CPU. By using an evaluation function to differentiate the feasible solutions, the scheduling goal is to find a good or perhaps the best solution.

The scheduling problem

The multiple-criteria project scheduling problem is characterized by four components: a set of resources *R*, set of activities *A*, set of precedence constraints C, and a set *Q* of project performance measures (objectives, criteria):

The set *R* consists of *p* types of renewable resources. The set *A* is composed of *n* activities, which have discrete resource requirements. It is assumed that preemption of activities is not allowed. The set *C* includes constraints pertaining to the relationships between objects from *A* and *R* and among themselves. The set *Q* consists of time and cost criteria: project makespan, resource utilization smoothness, maximum lateness, mean flow time, NPV and project cost. Each criterion is associated with a weight that specifies its importance in the overall decision.

Speaking generally, the scheduling problem is to find an allocation (considered in time) of resources from set *R* to activities from set *A* such that all activities in A are completed, the constraints are satisfied, and the best compromise between criteria from set *Q* is reached.

Importance of objectives

The planner's role in solving the scheduling problem is to model the problem. While modeling the system, i.e., constraints, resources and tasks, does not pose a problem to the planner, prioritizing the objectives requires expert knowledge. Thus the question remains how to correctly map the importance factor to the ratio scale.

The most common and simple method of weighting the user's preferences is by setting a numeric weight for each objective, thus establishing a cost function. That is, for each objective function $\mathrm{o}_i(x) \in Q$, we set a corresponding weight w_i such that the cost function for a valid solution *X is:*

$$f(X) = \sum_{i=1}^{n} w_i \cdot o_i(X) \qquad (1\text{-}1)$$

and the best solution *X* will be the one with the minimal cost. However, setting the weights for the various objectives is not a trivial task for the user. To this end we review a method that efficiently extracts the weights from the user's knowledge.

AHP - Saaty's approach: We assume that the planner cannot assign absolute weights to objectives, but can only compare pairs of objectives according to limited vocabulary. We can then apply the procedure developed by Saaty [9] to compute the absolute importance factors for each objective from paired comparisons between objectives. Basically, to compute absolute importance factors for *m* objectives, we pairwise compare these *m* objectives with each other, their relative importance being written in an *m*-by-*m* matrix, and use the eigenvalues of the matrix are used to calculate the absolute weights.

Saaty [9] proposes to use the importance attributes shown in Table 1-1 "due to the human ability to make effective quantitative distinctions (only between) five attributes".

Table 1-1. Relative importance table.

Relative Importance	Human readable definition
1	equal importance
3	weak importance of one over another
5	essential or strong importance
7	very strong or demonstrated importance
9	absolute importance
2,4,6,8	intermediate values between two adjacent scale judgments

2. SCHEDULING OPTIMIZATION

Iterative improvement methods combined with other heuristic approaches that improve a given schedule by searching in the neighborhood for a better schedule, can supply a good scheduling solution. In this section we describe two of the most popular improvement methods, *Tabu Search* and *Simulated Annealing*.

Tabu search

Tabu search is a rich set of methodologies for building extended neighborhood procedures, with particular emphasis on avoiding a local optimum. Why do we get caught in local optima? Because once we reach the best state within a neighborhood (local optimum), further moves will increase the objective function and hence, none will be chosen. How can this be fixed? By always taking the best move available, even if it increases the cost of the objective. This is basically the diversification move, because intensification in momentarily of no advantage.

Now, of course, if we move out of the local optimum, then on the very next move we can probably do best by moving right back. However, we must force to continue diversifying for a few more moves. The approach that tabu search employs is to keep a list of the last *m* moves (Glover mentions $m = 7$ as typical) and not to allow them to be repeated while they remain on the list.

First approaches of tabu search have investigated the whole neighborhood [10], however in realistic industrial applications this is not possible.

Simulated Annealing

Simulated annealing is an optimization method based on ideas from statistical physics where the annealing of a solid to its ground state (i.e., the state of minimum energy) is simulated.

In simulated annealing one selects randomly one modification operator and decides weather the resulting schedule can be accepted as a new candidate for further search. To escape local optima, a decrease of the evaluation is allowed by a probability that reduces during search. If the evaluation of a schedule s_i is c_i, then the algorithm accepts a neighbor s_j with an evaluation function of c_j, with the probability of

$$P = \min\left(1, e^{-\frac{c_j - c_i}{t}}\right) \qquad (2\text{-}1)$$

where *t* (the temperature) is a positive control parameter which is decreased during the execution of the algorithm. The probability of accepting a new schedule is small if the difference between both evaluations is large. In the beginning, when the temperature is high, it is more likely that a larger difference is accepted. The global strategy is to search first randomly over the whole search space for schedules. Later the search is more restricted to find a near optimal solution.

The decision how fast the annealing temperature decreases influences the time that the algorithm needs to solve a problem and how good the solution will be. The different annealing temperatures can be given explicitly by a fixed set of constants or by an implicit function. Also, the duration in which the search uses a certain temperature can be set. Often these settings are called the annealing schedule.

3. FUZZY SET THEORY IN SCHEDULING

In this section we describe the use of fuzzy set theory in the scheduling problem. We introduce the terms fuzzy constraints, fuzzy objectives, and fuzzy decision in contrast to the conventional definitions.

Fuzzy constraints, fuzzy goals and fuzzy decision.

We start by reviewing the fundamentals of fuzzy decision making as they were first set by Bellman and Zadeh in [1] in which three basic concepts were introduced: fuzzy goal, fuzzy constraint, and fuzzy decision. In the following section we introduce the framework of decision making in a fuzzy environment.

Let *X* be a given set of possible alternatives, which contains the solution of a decision making problem under considerations. A fuzzy goal (objective) *G* is a fuzzy set on *X* characterized by the membership function

$$\mu_G : X \to [0,1] \qquad (3\text{-}1)$$

A fuzzy constraint *C* is a fuzzy set on *X* characterized by the membership function

$$\mu_C : X \to [0,1] \qquad (3\text{-}2)$$

In other words, in definition of the fuzzy decision, there is no difference between the fuzzy goals and the fuzzy constraints.

Fuzzy weighting and preferences

The introduction of soft constraints requires the use of mathematical operations such as aggregation and weighting. Dealing with several soft constraints and objectives, we should determine the importance of them. Weighting can be done in the traditional (crisp) way by assigning numerical weights, or by using linguistic terminology with fuzzy set theory.

The problem of the importance of constraints was addressed from several aspects. Yager [7] argues that ranking or weighting of objectives can be achieved by linear ordering, intervals, relative ratios, or absolute rating. Fargier [3] took a similar approach to Yager and proposed to order constraints with respect to each other by giving them priority degree. Yager's [7] approach is slightly different in that he does not use numbers but a finite set of ordered symbols to represent the weights of the criteria using the following linear ordering:

perfect ≥ *very high* ≥ *high* ≥ *medium* ≥ *low* ≥ *very low* ≥ *lowest*

4. OPTIMIZATION WITH FUZZY RULE BASE

The AHP procedure described in section 1 solves the interface problem with the planner. However, the weights produced are passed on to the search engine with the assumption that the same units are used. This is not the case in normal scheduling problems. In the following section we introduce a search engine that receives the pairwise comparisons in a fuzzy way and produces a solution based only on the comparisons.

Fuzzy comparison of objectives

We define fuzzy terminology applied to Saaty's (Table 1-1) notations using fuzzy numbers. Each notation from Table 1-1 is defined as a fuzzy number such that the terminology includes the set:

{*equal, weak more important, weak less important, essential more important, essential less important, very strong more important, very strong less important, absolute more important, absolute less important*} =
$\left\{\tilde{1}, \tilde{3}, -\tilde{3}, \tilde{5}, -\tilde{5}, \tilde{7}, -\tilde{7}, \tilde{9}, -\tilde{9}\right\} = A$.

where $\tilde{a}$ is a fuzzy number as shown in Figure 4-1.

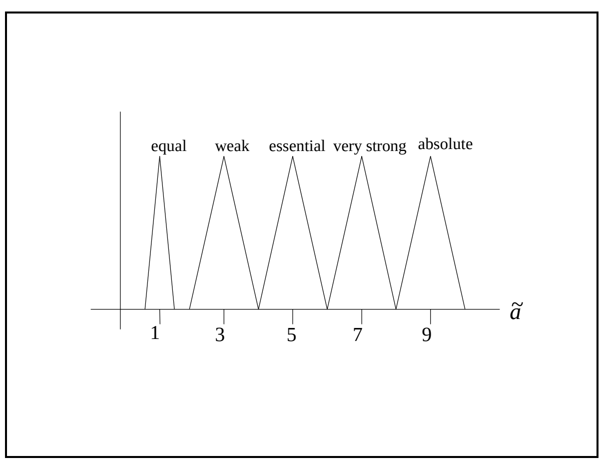

Figure 4-1. Fuzzy terminology for pairwise comparison

We consider a set of *n* (crisp) objective functions, $\{f_i(\bar{x})\}_{i=1}^n$, where:

- $\bar{x}$ is a solution of the scheduling problem.
- f_i maps a solution to a real value: $f_i : \bar{x} \to \mathrm{R}$.

The planner determines a set of $n-1$ pairwise comparisons $\tilde{a}_{i,j} \in A$ using the fuzzy vocabulary *A*, between the objectives f_i and f_j.

Fuzzy comparison of solutions

In order to use any algorithm, we have to define a mechanism to determine whether solution $\bar{x}_1$ is better or worse than solution $\bar{x}_2$. Let *improvement* be a fuzzy variable with vocabulary of three terms *R*={*better*, *equal*, *worse*}.

Definition 4-1. The improvement of solution $\bar{x}_2$ with respect solution $\bar{x}_1$ under the objective function f_i is the ratio:

$$improvement = \frac{f_i(\bar{x}_2) - f_i(\bar{x}_1)}{f_i(\bar{x}_1)} \qquad (4\text{-}1)$$

where $f_i(\bar{x})$ is the score of solution $\bar{x}$. The membership function corresponding to *improvement* is illustrated in Figure 4-2.

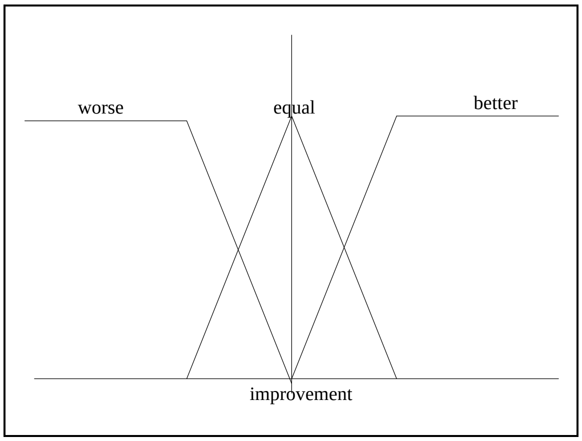

Figure 4-2. Improvement membership function

Comparison of solutions regarding two objective function: Consider a system with two objective functions, f_1 and f_2, and a fuzzy variable $\tilde{a}_{1,2}$ that determines the importance ratio between the objectives. Assume two solutions, $\bar{x}_1$ and $\bar{x}_2$. Then, $f_1(\bar{x}_2)$ can be *better* than, *equal* to, or *worse* than $f_1(\bar{x}_1)$ with a membership according to Figure 4-2.

Definition 4-2. We define a fuzzy variable *r* according to the following rule base:

- *if* $f_1(\bar{x}_2)$ *is better than* $f_1(\bar{x}_1)$ *and* $f_2(\bar{x}_2)$ *is better than* $f_2(\bar{x}_1)$ *then* $r = 1$
- *if* $f_1(\bar{x}_2)$ *is better than* $f_1(\bar{x}_1)$ *and* $f_2(\bar{x}_2)$ *is equal than* $f_2(\bar{x}_1)$ *then* $r = \frac{\tilde{a}_{1,2} - 1}{\tilde{a}_{1,2} + 1}$
- *if* $f_1(\bar{x}_2)$ *is better than* $f_1(\bar{x}_1)$ *and* $f_2(\bar{x}_2)$ *is worse than* $f_2(\bar{x}_1)$ *then* $r = \frac{\tilde{a}_{1,2} - 1}{\tilde{a}_{1,2} + 1}$
- *if* $f_1(\bar{x}_2)$ *is equal than* $f_1(\bar{x}_1)$ *and* $f_2(\bar{x}_2)$ *is better than* $f_2(\bar{x}_1)$ *then* $r = \frac{1}{\tilde{a}_{1,2} + 1}$
- *if* $f_1(\bar{x}_2)$ *is equal than* $f_1(\bar{x}_1)$ *and* $f_2(\bar{x}_2)$ *is equal than* $f_2(\bar{x}_1)$ *then* $r = 0$
- *if* $f_1(\bar{x}_2)$ *is equal than* $f_1(\bar{x}_1)$ *and* $f_2(\bar{x}_2)$ *is worse than* $f_2(\bar{x}_1)$ *then* $r = \frac{-1}{\tilde{a}_{1,2} + 1}$
- *if* $f_1(\bar{x}_2)$ *is worse than* $f_1(\bar{x}_1)$ *and* $f_2(\bar{x}_2)$ *is better than* $f_2(\bar{x}_1)$ *then* $r = \frac{1 - \tilde{a}_{1,2}}{\tilde{a}_{1,2} + 1}$
- *if* $f_1(\bar{x}_2)$ *is worse than* $f_1(\bar{x}_1)$ *and* $f_2(\bar{x}_2)$ *is equal than* $f_2(\bar{x}_1)$ *then* $r = \frac{-\tilde{a}_{1,2}}{\tilde{a}_{1,2} + 1}$
- *if* $f_1(\bar{x}_2)$ *is worse than* $f_1(\bar{x}_1)$ *and* $f_2(\bar{x}_2)$ *is worse than* $f_2(\bar{x}_1)$ *then* $r = -1$

Recall that $\tilde{a}_{1,2}$ is a fuzzy number. Applying the min and max operators and center of gravity as defuzzification function, then *r* is a number in the range $[-1,1]$.

Proposition 4-1. By using an α-cut, we can determine that:

- $\bar{x}_1$ *is better than* $\bar{x}_2$ *if* $r > \alpha$
- $\bar{x}_1$ *is worse than* $\bar{x}_2$ *if* $r < -\alpha$
- $\bar{x}_1$ *is equal than* $\bar{x}_2$ *if* $-\alpha < r < \alpha$

for two objective functions, f_1 and f_2, and a fuzzy variable $\tilde{a}_{1,2}$ that determines the importance ratio between the objectives.

Note: For $\alpha = 0$, we get only *better* and *worse* results.

Multi-objective function comparison: Consider a system with *n* objective functions. Each pair of functions f_i and f_j, together with the fuzzy ratio $\tilde{a}_{i,j}$ produces a number $r_{i,j}$ in the interval $[-1,1]$.

Definition 4-3. The overall comparison between multi-objective functions is the aggregation of the pairs' results

$$r = \sum r_{i,j} \qquad (4\text{-}2)$$

Proposition 4-2. By using an α-cut, we can determine that:

- $\bar{x}_1$ *is better than* $\bar{x}_2$ *if* $r > \alpha$
- $\bar{x}_1$ *is worse than* $\bar{x}_2$ *if* $r < -\alpha$
- $\bar{x}_1$ *is equal than* $\bar{x}_2$ *if* $-\alpha < r < \alpha$

for *n* objective functions, $f_1, \ldots, f_n$ and fuzzy variables $\{\tilde{a}_{i,j}\}_{i=1}^n$ that determines the importance ratio between the objectives.

Note: For $\alpha = 0$, we get only *better* and *worse* results.

Fuzzy Tabu Search

In this section we introduce the use of fuzzy based tabu search engine. Earlier we saw a method for comparing between two feasible solutions. We can now apply this technique to the tabu search method.

We assume that there are *n* objective functions $f_1,...,f_n$, and that the user sets the fuzzy variables $\left\{\tilde{a}_{i,j}\right\}_{i=1}^{n}$ that determine the importance ratio between the objectives.

Algorithm 4-1. Fuzzy Tabu Search:

1. Obtain a random solution $\bar{x}_0$.
2. Set $\bar{x}_i = \bar{x}_0$.
3. Set $\bar{x}_{best} = \bar{x}_0$.
4. Set a tabu list of up to *m* solutions. Reset the list (i.e., the list is empty).
5. Find a feasible solution $\bar{x}_{i,0}$ in the neighborhood of $\bar{x}_i$ such that $\bar{x}_{i,0}$ is not in the tabu list.
6. Set $\bar{x}_{i,next} = \bar{x}_{i,0}$.
7. For every solution $\bar{x}_{i,j}$ in the neighborhood of $\bar{x}_i$:
8. If according to Proposition 4-2 with $\alpha = 0$, $\bar{x}_{i,j}$ is *better* than $\bar{x}_{i,next}$ and $\bar{x}_{i,j}$ is not in the tabu list, then:
9. $\bar{x}_{i,next} = \bar{x}_{i,j}$.
10. Set $\bar{x}_i = \bar{x}_{i,next}$.
11. Put $\bar{x}_i$ in the tabu list.
12. If $\bar{x}_i$ is *better* than $\bar{x}_{best}$, then $\bar{x}_{best} = \bar{x}_i$.
13. Repeat steps 5-12 *I* times.

After *I* iterations, $\bar{x}_{best}$ holds the best solution.

Fuzzy Simulated Annealing

In this section we introduce the use of fuzzy based simulated annealing engine. While in tabu search we took the best solutions in the neighborhood, in simulated annealing we take one of the feasible solutions according to some probability. The probability to choose solution *i* over *j* is given by (2-1). However, since we do not have an aggregated cost function, we should use an alternative approach as shown in the following paragraphs.

Fuzzy Simulated Annealing Algorithm:

Definition 4-4. Let *temperature* be the fuzzy variable

$$temperature \in \{hot,\ warm,\ cool,\ freeze\}$$

whose membership is shown in Figure 4-3.

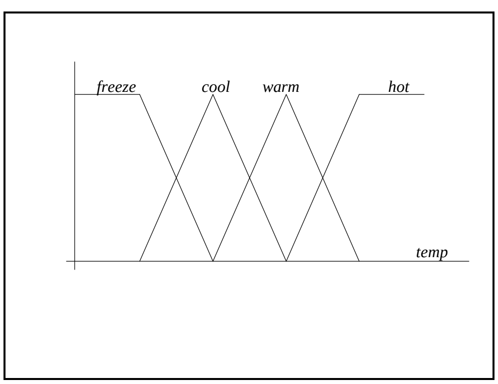

Figure 4-3. Temperature membership function

In order to replace the exponent in (2-1), we introduce a new method for choosing the next solution. Assume there are *N* feasible solutions, $\bar{x}_i,...,\bar{x}_N$, such that $\bar{x}_i$ is *better* than $\bar{x}_j$ if $i < j$, i.e., the solutions are sorted according to Proposition 4-2 with $\alpha = 0$. Then, the temperature determines the probability of the selection according to the following rules:

- *In hot temperature, the selection is random, i.e., equal probability.*
- *In cold temperature, select the best solution.*

We can translate these rule of thumb into the following fuzzy rule-base:

- *if temp is hot then* $prob(s_i)$ *is* $1/N$ (e^{0i})
- *if temp is warm then* $prob(s_i)$ *is* $e^{-i/N}$
- *if temp is cool then* $prob(s_i)$ *is* $e^{-5i/N}$
- *if temp is freeze then* $prob(s_i)$ *is 1 for* $i = 0$ *and 0 else* ($e^{-\infty i}$)

where s_i is the *i*th solution in the sorted list of solutions.

Algorithm 4-2. Fuzzy Simulated Annealing.

1. Obtain a random solution $\bar{x}_0$.
2. Set $\bar{x}_i = \bar{x}_0$.
3. Set $\bar{x}_{best} = \bar{x}_0$.
4. Find *N* feasible solutions, $\bar{x}_{i,1},...,\bar{x}_{i,N}$ in the neighborhood of $\bar{x}_i$.
5. Sort the solutions such that $\bar{x}_{i,j}$ is *better* than $\bar{x}_{i,k}$ if $k > j$ (i.e., the solutions are sorted according to Proposition 4-2 with $\alpha = 0$).
6. For every solution $\bar{x}_{i'}$ in the neighborhood of $\bar{x}_i$:
7. If $\bar{x}_{i'}$ is *better*, then $\bar{x}_{i,N}$, then:
8. $\bar{x}_{i,N} = \bar{x}_{i'}$
9. Sort the new group of solutions such that $\bar{x}_{i,j}$ is *better* than $\bar{x}_{i,k}$ if $k > j$ (i.e., the solutions are sorted according to Proposition 4-2 with $\alpha = 0$).
10. If $\bar{x}_{i,1}$ is *better*, then $\bar{x}_{best} = \bar{x}_{i,1}$.
11. Use the rule base to set probabilities to the *N* solutions $\bar{x}_{i,1},...,\bar{x}_{i,N}$.
12. Choose one of the solutions at random according to the probabilities and set it as new $\bar{x}_i$.
13. Repeat steps 4-12 *I* times.

5. CASE STUDY: MANPOWER SCHEDULING

In this section we describe a real world scheduling problem and use it to illustrate our approach. The system considered is that of manpower scheduling in an air cargo terminal at Tel-Aviv airport.

In general, system tasks include steady permanent tasks and temporary tasks that are unique to this month only. The system planner gets a list of tasks at the beginning of each month. Having updated data on all employees and several criteria for scheduling, the planner’s job is to plan a good schedule that satisfies all requirements for this month within the criteria.

The system also produces a variety of scheduling reports such as the scheduling itself, weak points, billing reports etc.

System definition

This section introduces the approach to the scheduling problem of cargo transport in the airport. We specify the system structure and its requirements in order to construct the scheduling process.

System tasks: The system includes a database of possible tasks and resources. From this database we construct the schedule.

In the airport, there are several sites for cargo transport. These sites include a northern ramp, southern ramp, import deck, export deck, etc. The site list is fixed and does not change from one week to another. However, on rare occasions, a temporary site can be opened and therefore, the first table in the scheduling database includes all the standard and non-standard sites.

The work shifts include the shifts' hours. The standard shifts include the morning shift from 6:45 to 15:45, the afternoon shift from 15:45 to 23:45, and the night shift from 23:45 to 6:45. Other shifts include the morning-noon shift and noon-night shift. The system planner can define new types of shifts for a certain week on demand. Hence, the second table in the database is a list of both standard and special shifts.

The system includes several patterns that define collections of positions. For example, the pattern "standard morning - northern ramp" includes a list that is used usually every day in the northern ramp in the morning hours. Other patterns can be less applicable or unique for some week. The table in the database includes the following entries:

1. Quantity: The number of personnel of this type.
2. Profession: The profession type, e.g., head of shift, forklift operator, etc.

The next table defines the system's task for the week. It connects entries from the other tables to construct job definitions. The task table includes the following entries:

1. Site: As defined in the first table.
2. Shift: As defined in the second table.
3. Pattern: As defined in the third table.
4. Status: Can be fixed, active, inactive.

The status field categorizes the task into one of the following types: A fixed task is a permanent task that the system planner is not authorized to cancel. An active task is a permanent task that the system planner can delete. An inactive task is a task that the system planner can add in order to enhance activity.

The last table along with the previous ones define the system requirement in a one to one fashion. All system activities as defined in these tables need to be fulfilled. In addition there is a general parameters' list that adds constraints to the requirements. This list includes two parameters: Maximum working time, i.e., the maximum time a person can work continuously, and the minimum rest time, i.e., the minimum time between two shifts per person.

System's resources: The planner has a list of all system resources, i.e., all persons of the company and their details. Each employee has a file with his/her personal information. This information contains personal data, working skills, and contract type.

The first entry is the employee's profession. The profession field includes the professions relevant to each site. For each site the employee can have one major profession and several secondary professions. In addition, the level of each profession must be defined. For example, John Smith's major profession in the northern ramp is forklift driver level 1, whereas his profession in the export ramp is shift assistance manage level 2.

The second entry is the availability of the employee. Listed are the days in which the employee is unavailable in the next month as well as the days that are inconvenient.

The third entry is the contract status and salary of the employee. A permanent employee gets a fixed salary (working up to 200 hours a month) and overtime (hourly based pay) for every hour beyond 200 hours a month. A contractor earn on an hour base, i.e., the salary is linear to the working hours.

Other parameters such as phone number, date of birth, are relevant for the reports' producing phase.

Scheduling criteria

The planner chooses from the solutions' domain to the scheduling problem the one with the best score according to several criteria.

The first criterion is the quality of the solution. This criterion includes the number of hours worked by employees not in their major profession. The second criterion is the total cost. Basically, his is basically the money that this month will cost by salaries. The third criterion is convenience. It includes the hours worked by employees during their inconvenient days.

Scheduling solution

Table 5-1 and Table 5-2 show the results for the manpower (shift) scheduling case study. Table 5-1 uses crisp weights for the scheduling criteria whereas Table 5-2 uses fuzzy (linguistic) vocabulary. The upper line in Table 5-2 pertains to our tabu search algorithm whereas the second line pertains to our fuzzy simulated annealing algorithm.

Table 5-1. Shift scheduling results using crisp preferences

Weights	Money	Quality	Conv
Money (weight=1) Quality (weight=1) Convenience (weight=1)	586707 587932	1557 1585	432 793
Money (weight=1) Quality (weight=20) Convenience (weight=20)	587226 586562	334 414	95 77

Table 5-2. Shift scheduling results using fuzzy preferences

Fuzzy scheduling	Money	Quality	Conv
Money is equal to Quality. Money is equal to Convenience. Quality is equal to Convenience.	589867 589549	116 220	8 35
Money is *very strong* less important compared to Quality. Money is *essential* less important compared to Convenience. Quality is *weak* more important compare to Convenient.	588745 590965	69 58	60 51

As we can see from the results, the fuzzy approach is much more intuitive and produces scheduling scenarios that are more appropriate. The crisp approach, which entails the selection of weights is more time consuming because it requires a relatively large number of iterations to obtain the "optimal" weights.

6. CONCLUSION

In this paper we have extended the tabu search and simulated annealing search algorithms to enable the use of fuzzy (linguistic) information. We applied these fuzzy search engines to the scheduling problem.

Simulations using a real world scheduling problem indicate that the fuzzy approach is not only more intuitive but produces scheduling scenarios that are more appropriate.

7. REFERENCES

[1] R. E. Bellman and L. A. Zadeh. "Decision making in a fuzzy environment." *Management Science*, 17-B(4):141-164, December 1970.

[2] De Werna, and A. Hertz, "A Tabu search techniques: A tutorial and an application to neural networks", *OR Spektrum* 11, pp. 131-141, 1989.

[3] H. Fargier, J. Lang, and T. Schiex. "Selecting preferred solutions in fuzzy constraint satisfaction problems". In *First European Congress on Fuzzy and Intelligent Technologies*, EUFIT'93, pp. 1128-1134, Aachen, Germany, September 1993.

[4] M. Laguna, J. Barnes and F. Glover, Tabu search methods for a single machine scheduling problem, *Journal of International Manufacturing* 2, pp. 63-73, 1991.

[5] W. Slany, Scheduling as a fuzzy multiple criteria optimization problem. Ph.D. Thesis (tech. rep. CD-TR 94/62), 1994.

[6] I. B. Turksen, Three fuzzy theory approaches for scheduling system design. In *Fuzzy Information Engineering*. (Yager, Prade, Dubuis eds)1997.

[7] R. Yager. A new methodology for ordinal multiobjective decision based on fuzzy sets. *Decision Sciences*, 12:589-600, 1981.

[8] R. Yager. Constrained OWA aggregation. *Fuzzy Sets and Systems* 81 (1996) 89-101

[9] T. L. Satty. *The analytic hierarchy process*. McGraw-Hill, 1980

[10] M. Widmer and A. Hertz. A new heuristic method for the flow shop sequencing problem, *European Journal of Operation Research* 41, 1989.

[11] H. J. Zimmermann. Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets and Systems*, 1, pp. 45-55, 1978.