System Development for Fuzzy Project Scheduling

Hongqi Pan and Chung-Hsing Yeh
School of Business Systems
Monash University
Clayton Campus, VIC 3800
AUSTRALIA

Abstract: - Fuzzy resource-constrained project scheduling with multiple performance modes covers the majority of possible situations in practice. This type of scheduling is a typical complex combinatorial optimization problem. Exact approaches may not be viable. To solve this type of problem efficiently and effectively, four metaheuristic approaches, fuzzy GA alone, fuzzy GA with tabu, fuzzy SA and fuzzy SA with tabu have been developed and built into a system in this research. Each approach has its own advantage depending on circumstances. The system developed has practical significance in solving fuzzy project scheduling in realistic settings, providing a number of efficient and effective approaches.

Key-Words: - Fuzzy sets, Fuzzy project scheduling, metaheuristic, genetic algorithms, simulated annealing, tabu mechanism, system development

1 Introduction
Project scheduling has been widely applied in many areas such as engineering, building industry, defence projects and software engineering. However, many realistic project scheduling problems often have a number of flexible performance options of activities under resource constraints throughout the project. In addition, fuzziness also inherently exists in project scheduling because, often, there is insufficient precise information on activities in a project or because historical information on similar types of activities is not available. Project scheduling with these concerns is significantly meaningful in practice and is defined as fuzzy multi-mode resource-constrained project scheduling (FMMRCPS). A recent literature review has indicated that vigorous research on FMMRCPS model is needed because of the dearth of research in this field [1]. This study develops four metaheuristic approaches, fuzzy GA, fuzzy GA with tabu, fuzzy SA and fuzzy SA with tabu, and these approaches are built into a system of FMMRCPS, using the object-oriented programming methodology. This system developed makes it easier for practitioners to solve realistic FMMRCPS under different circumstances efficiently and effectively.

2 FMMRCPS Formulation
In the FMMRCPS model, a given project comprises of J activities (j = 1, 2, ..., J), which are subject to the constraints of precedence relationships and resource limitation. Activity j can be executed in mode m of Mj modes (m = 1, 2, ..., Mj) with its corresponding resource requirements \{ k_{jm}, \forall r \in R \}. Once an executive mode is assigned to an activity and it is scheduled, any mode changes or interruption of the activity is not permitted, that is, the activity becomes non-preemptive. Let t be the fuzzy time period, and x_{jmx} be a binary variable, whose value is defined as

\[
x_{jmx} = \begin{cases} 
1 & \text{if activity } j \text{ is performed in mode } m \text{ at time } t \\
0 & \text{otherwise}
\end{cases}
\]

Let \( \tilde{H} \) be the heuristically determined fuzzy project completion time, that is to add up fuzzy duration times of all activities for mode 1 in a project. It can be expressed as

\[
\tilde{H} = \sum_{j=1}^{J} \tilde{d}_{j1}, \quad m = 1
\]  

Let \( E\tilde{T}_j \) and \( L\tilde{T}_j \) be the fuzzy earliest and latest finish times of activity j respectively. FMMRCPS for minimising the project completion time can be formulated as

\[
\min \sum_{m=1}^{M_j} \sum_{x_{jmx} \neq 0} \tilde{t} \times x_{jmx}
\]

subject to

\[
\sum_{m=1}^{M_j} \sum_{x_{jmx} \neq 0} x_{jmx} = 1, \quad j = 1, 2, ..., J
\]  

\[
\sum_{m=1}^{M_j} \sum_{x_{jmx} \neq 0} (\tilde{t} - \tilde{d}_{jmx}) \times x_{jmx} - \sum_{m=1}^{M_j} \sum_{x_{jmx} \neq 0} \tilde{t} \times x_{jmx} \geq 0, \quad j = 1, 2, ..., J
\]  

\[
\sum_{j=1}^{J} \sum_{m=1}^{M_j} k_{jmx} \sum_{x_{jmx} \neq 0} x_{jmx} \leq K, \quad r = 1, 2, ..., R; \quad \tilde{t} = 1, ..., \tilde{H}
\]
As shown above, an FMMRCPS problem can be expressed as a fuzzy 0-1 programming model with fuzzy constraints. Objective function (2) is to ensure that the project will be completed as soon as possible. Constraints (3) allow an activity to be scheduled only once over the whole project. Constraints (4) describe the precedence relationships among activities of a project; Constraints (5) state that the resource usage cannot exceed the resource availability. In the FMMRCPS model, there are two decisions required to be made: (a) selecting activity modes, and (b) determining the sequence of activities under the conditions that satisfy precedence relationships and resource constraints. Because there are a great number of fuzzy constraints in the FMMRCPS model, when \( |J| > 3 \), it becomes a complex combinatorial optimisation problem of complete NP-hardness [2].

As FMMRCPS is a complex combinatorial optimization problem, exact approaches are not viable. Metaheuristic approaches seem to be attractive in solving FMMRCPS problems of practical sizes efficiently and effectively [3]. Four metaheuristic approaches developed in a system to FMMRCPS will be explained in the following section.

3 Metaheuristic Approaches
Realistically-sized FMMRCPS problems are impossible to solve by any exact approaches as it is claimed that a newly developed powerful approach can only solve up to a maximum of 30 activities with two performance modes in a scheduling project [4]. Nowadays metaheuristic approaches have been widely applied in many large, complex optimisation problems. This kind of approach is a class of approximate methods that is not intended to explore an immense solution space, rather randomly searching in the whole space. This class of heuristics may be concisely described as “walk through neighbourhoods”, a search trajectory through the solution domain of a problem. In the following subsections, the four metaheuristic approaches developed for FMMRCPS will be described.

3.1 Fuzzy GA
Genetic Algorithms (GA) were initially introduced by Holland for the process of biological evolution [5]. It is the intelligent exploitation of a random search that is characterised by a parallel search of the entire space against a conventional point-by-point search for finding optimal solutions.

In developing a fuzzy GA, a designed chromosome contains three sub-chromosomes: (a) mode assignment, (b) activity priority, and (c) details of fuzzy start and finish times of each activity as well as the fuzzy project completion time. The specially designed chromosomes can directly reflect all necessary scheduling information without any interpretation for a solution. Let \( \Phi' \) be a chromosome \( \gamma \), representing a possible schedule solution in the \( x \)th generation. The 1st and 2nd rows describe a mode \( m_j \) and a priority value \( \omega_j \) assigned to each activity \( j \) in chromosome \( \gamma \). The 3rd row indicates the fuzzy start \( S_{T_j} \) and finish \( F_{T_j} \) times of each activity \( j \), together with the last column for the trailing information about fuzzy project completion time \( PC_{T} \) of chromosome \( \gamma \). A chromosome representing a solution to a FMMRCPS problem can be symbolised as

\[
\Phi' = \begin{bmatrix}
    m_1 & \ldots & m_j & \ldots & m_n \\
    \omega_1 & \ldots & \omega_j & \ldots & \omega_n \\
    S_{T_1}, F_{T_1} & \ldots & S_{T_j}, F_{T_j} & \ldots & S_{T_n}, F_{T_n} & PC_{T}
\end{bmatrix}
\]

(6)

To visually describe a chromosome for representing a possible schedule to FMMRCPS, Figure 1 depicts an example of a chromosome containing three sub-chromosomes in a project containing 8 activities.

![Solution presentation by a chromosome](image)

Fig. 1 Solution presentation by a chromosome

For mode operations in offspring generations, the system allows the user to choose one out of two categories: either assigning modes independently or mode assignment based on search strategies. The option of assigning modes independently is that the mode changes rely only on the user preset and is not concerned with the trends of solution changes during offspring generation, whereas mode assignment concerned with search strategies is based on the current conditions in generated solutions from the recent past to the present.

For priority operations in the 2nd sub-chromosomes, three genetic operations are to be applied: (a) mutation, (b) crossover, and (c) neighbourhood swaps. The mutation operation is used to alter only two genes from the 2nd sub-chromosome of a parent while generating the new 2nd sub-chromosome for a child. When this
operation is applied, the generated $2^{nd}$ sub-chromosome of the offspring has the similar genetic structure to that of the parent with only difference in two genes. This operation is to produce new chromosomes with only small variations into the population for the purpose of a local search area, when the trend of generated schedules has been improved in the recent searching. The crossover operation combines the features of the $2^{nd}$ sub-chromosomes of two parents to generate that of an offspring. Therefore, the $2^{nd}$ sub-chromosome of a child generated is not similar to that of either parent. This operation is often applied to situations where different search spaces should be explored in order to avoid trapping in local optima. Neighbourhood swaps can be viewed as a series of mutations by pair-wise interchanges to generate a set of the $2^{nd}$ sub-chromosomes of offspring based on a randomly selected gene in a location of a parent. Neighbourhood swaps are used to improve scheduling solutions by searching the neighbourhood from an initial solution.

Whilst generating the next generation, all current chromosomes are evaluated against a certain measure of fitness. The fitter chromosomes are selected in the artificial version of the naturally survival phenomenon referred to as the survival of the fittest and the elimination of weakness. A specific fitness function is developed to determine the chances of survival based on the status of the fuzzy project completion time of individual chromosomes. That means, the shorter project completion time should have a higher value presented in the specifically designed fitness function. Let $v'_i$ represents the value of fitness for chromosome $\gamma$ in the $\lambda$th generation. The fitness function for chromosome $\gamma$ in the $\lambda$th generation can be measured as

$$v'_i = \frac{R(\tilde{f}_{i,\max}) - R(\tilde{f}_{i,\min}) + \alpha}{R(f_{i,\max}) - R(f_{i,\min}) + \alpha} \quad (7)$$

Let $\tilde{f}_{i,\max}$ and $\tilde{f}_{i,\min}$ be the maximum and minimum fuzzy project completion times respectively in the $\lambda$th generation, and $\tilde{f}_{i,\gamma}$ represent the objective function of the fuzzy project completion time for chromosome $\gamma$ in that generation. The ranking indices of $\tilde{f}_{i,\max}$, $\tilde{f}_{i,\min}$ and $\tilde{f}_{i,\gamma}$ are denoted as $R(\tilde{f}_{i,\max})$, $R(\tilde{f}_{i,\min})$ and $R(\tilde{f}_{i,\gamma})$ respectively. $\alpha$ is a parameter, and chosen as a positive real number within the open interval $(0, 1)$. It prevents the denominator of formula (7) from being zero.

In the beginning of fuzzy GA, all initial solutions in the population are first generated randomly. To avoid generating the same chromosomes initially, a mechanism is built into the system to ensure that all chromosomes generated in the generation are different from each other. In the construction of the subsequent generations, a specified number of fit chromosomes are selected in the generation, based on the fitness function given in Formula (7). During each generation, mode assignment and activity priority operations are applied in the generation of offspring until a good solution that satisfies certain criteria, is found.

### 3.2 Fuzzy GA with Tabu

The procedure of fuzzy GA with tabu is similar to that of fuzzy GA alone except that fuzzy GA with tabu incorporates the tabu mechanism. The tabu search was proposed by Glover [6, 7], for solving complex optimisation problems. One of the main functions of tabu is the use of adaptive memory to guide the search behaviour that is the hallmark of tabu search. The adaptive memory feature of tabu records information on search history that can be exploited in the search process. In this research, a short-term memory of tabu, referred to as tabu list, is adopted in fuzzy GA with tabu. The use of tabu can directly exclude the search alternatives classified as forbidden that have been visited recently, thus avoiding visiting the same solutions more than once. The size of tabu list can be set randomly by the system or specified by the user. However, the size of tabu has to be carefully decided, since a large tabu list may take too much time in evaluating the current search against every previous stored search. Experiments have demonstrated that a good size of tabu list may be around the population size divided by 10.

### 3.3 Fuzzy SA

Simulated annealing (SA) is a heuristic optimisation technique for tackling complex combinatorial problems where a large number of variables are involved. The metaphor of SA comes from metallurgy and is based on the thermal process for obtaining a low energy level while a metal is cooling gradually from molten to solid state. The mathematical approach for annealing was developed by Metropolis et al [8]. The solution representation is important component in fuzzy SA. The designed solution representation should not only represent the nature of a specific FMMRCPSP problem, but also suit neighbourhood operation in fuzzy SA. The solution presentation designed here
contains four elements: (a) a list of mode assignments of activities for each stage of partial schedules, (b) a list of sets of priority values to the activities in each partial schedule, (c) a list of all stages of partial schedules, and (d) the objective function of the project completion time.

The first element in the solution presentation is the mode assignment list. This element is the list of sets of specific modes assigned for activities on partial schedules. In addition, the mode selected for an activity should be within the available modes of the activity. Let $mode_{list}$ be a list of sets of modes assigned to activities on partial schedules, and let $PS_n$ denote a partial schedule in stage $n$ ($n = 1, 2, ..., N$). The mode assignment list for partial schedules in each stage can be mathematically defined as

$$mode_{list} = \sum_{i = 1}^{N} \{j \in PS_n \mid (m_j), j \} \quad m_j = 1, 2, ..., M; j = 1, 2, ..., J \quad (8)$$

The second element is a list of sets of priority values assigned to activities in each partial schedule. To simply assigning priority values to activities rather than employing priority rules, a priority value randomly assigns a different integer to each activity for generating the initial solution at the beginning of fuzzy SA. That is, if a project contains 100 activities, a different integer from 1 to 100 will be assigned to each activity. This element is used to manipulate priority values to specific or all activities when the operation on activity priorities is required. Let $priority_{list}$ be a list of activity priorities in each partial schedule and $v_j$ be a priority value assigned to activity $j$. The list of sets of activity priorities in every partial schedule of the whole schedule can be represented as

$$priority_{list} = \sum_{i = 1}^{N} \{j \in PS_n \mid (v_j), j \} \quad j = 1, 2, ..., J \quad (9)$$

The third element of the solution presentation records all details of partial schedules in each stage of scheduling. Let $PS_{list}$ be a list of sets of partial schedules for every stage from 1 to $N$, and let $A(\tilde{t}_j)$ and $C(\tilde{t}_j)$ denote the active set and complete set at fuzzy partial schedule time $\tilde{t}_j$ of stage $n$ in scheduling respectively. The active set, $A(\tilde{t}_j)$ is the set for placing those activities that are being scheduled and have not been finished at the fuzzy partial schedule time $\tilde{t}_j$ of stage $n$ whereas the complete set is the set where the activities have been completed at the fuzzy partial schedule time $\tilde{t}_j$ of stage $n$. The list of all partial schedules in every stage of scheduling can be represented as

$$PS_{list} = \sum_{i = 1}^{N} \{(A(\tilde{t}_j), C(\tilde{t}_j)) \mid j \in A(\tilde{t}_j) \cup C(\tilde{t}_j)\} \quad (10)$$

The fourth element of the solution presentation represents the objective function of fuzzy project scheduling, $f(i)$. The objective function of project scheduling here is the fuzzy project completion time that can be represented as the partial schedule time $PS\tilde{t}_n$ at the last stage $N$, plus the longest fuzzy finish time of activity $j$ among those activities in the complete set $C(\tilde{t}_j)$ for the last stage $N$ of scheduling. This element can be expressed as

$$f(i) = PS\tilde{t}_n + \max\{F\tilde{t}_j \mid j \in C(\tilde{t}_j)\} \quad (11)$$

The above defined solution presentation has a number of merits. The solution representation can initially indicate the fuzzy project completion time as the objective function of scheduling. It also reflects all necessary information required by FMMRCPS, including both the mode assignment and activity priorities, as well as the sequence of activities of partial schedules for every fuzzy scheduled time. Furthermore, the solution representation can be manipulated efficiently and effectively in fuzzy SA to generate a neighbouring solution.

To apply the acceptance probability of Metropolis’ criterion, the fuzzy project completion time has to be defuzzified. Let $\tilde{f}(i+1)$ be a new fuzzy neighboring solution generated from the current fuzzy solution $\tilde{f}(i)$. The values of the fuzzy objective function in the current $\tilde{f}(i)$ and neighborhood $\tilde{f}(i+1)$ solutions are converted into ranking indices $R(\tilde{f}(i))$ and $R(\tilde{f}(i+1))$ respectively. Let $MC(T)$ denote Metropolis’ criterion at the current temperature $T$ for the acceptance probability of a new solution in fuzzy SA for solving FMMRCPS problems. $MC(T)$ can be expressed as

$$MC(T) = \begin{cases} 1 & \text{if } R(\tilde{f}(i+1)) \leq R(\tilde{f}(i)) \text{ or } R(\tilde{f}(i+1)) > R(\tilde{f}(i)) \text{ and } T > 0 \text{ or } R(\tilde{f}(i+1)) > R(\tilde{f}(i)) \text{ and } T = 0 \text{ or } R(\tilde{f}(i+1)) > R(\tilde{f}(i)) \text{ and } T < 0 \text{ and } \exp(-|T|) \geq R(\tilde{f}(i)), \\ \exp(|T|) & \text{if } R(\tilde{f}(i+1)) > R(\tilde{f}(i)) \text{ and } T > 0 \text{ and } \exp(-|T|) < R(\tilde{f}(i)) \end{cases} \quad (12)$$

To generate neighbouring solutions, first of all the system picks up a partial schedule time $\tilde{t}_j$ as a selected fuzzy time point, from the set of partial schedule times of the current solution, as expressed in Formulae (8) and (9). Once the system selects the partial schedule time $\tilde{t}_j$, a series of neighbouring solutions are generated through perturbations. This is accomplished by mode assignment, or activity priority operations, or a combination of both operations.
Once the two new neighboring solutions have been processed in the annealing procedure, and after they have been obtained by fuzzy forward and backward scheduling respectively, the Markov chain length, $L_{\text{markov}}$, is reduced by one. Such a procedure will be repeated for the number of iterations specified by $L_{\text{markov}}$ until the Markov chain length becomes zero. Thus, the Metropolis algorithm is attempting to generate and inspect a number of neighboring solutions in $L_{\text{markov}}$ at the same temperature $T$ in order to dig for a better solution. The fuzzy SA will be terminated if one of the stopping criteria has been satisfied or the temperature level has reached the lowest temperature $T_{\text{final}}$. When fuzzy SA is terminated, the solution stored in the variable of the best solution in the system becomes the optimal or approximately optimal solution to FMMRCPS.

### 3.4 Fuzzy SA with Tabu

To prevent the reproduction of the same solutions generated from recent searches, tabu mechanism is incorporated into fuzzy SA. Tabu for short-term memory is employed in fuzzy SA with tabu. The tabu list keeps track of solution attributes for representing previous searches in order to avoid cycling in the same search space. The attributes of recently visited solutions are labelled as tabu active and solutions containing these attributes become tabu. Apart from the tabu mechanism, fuzzy SA with tabu is similar to fuzzy SA alone. Fuzzy SA incorporating a short-term memory function can forbid transitions in solutions if the attributes are already contained in the solutions generated. Therefore, the merit of fuzzy SA with tabu is to ensure that the search procedure does not revisit solutions previously generated. In addition, the tabu mechanism expands the search for better neighboring solutions although sometimes worse solutions may be generated. Each time a new neighboring solution is generated that is not forbidden, it moves to the top of the tabu list and the bottom solution in the tabu list is dropped off. The size of tabu can be selected to be either fixed or changeable in the system. In the case of changeable tabu size, the size is variable, depending on the status of the current search. If new neighboring solutions are generated that have not recently appeared in the tabu list, the size of tabu will be reduced as the current search is in a definitely different search area. If newly generated neighboring solutions have appeared frequently in the tabu list, the size of tabu needs to be increased as the search may be moving towards a previous search area.

### 4 System Development

The system development for FMMRCPS is a very complex programming process, involving a number of complicated procedures for processing activities with multiple performance modes under both precedence relationships and resource constraints in a fuzzy environment. To facilitate logical understanding and maintenance of the large system to FMMRCPS, the entire structure of the system uses traditional structure programming techniques along with the modular concepts for each individual task [9].

In the system development, some common tasks may frequently be used by different approaches or different stages of scheduling such as fuzzy forward and backward scheduling mechanisms, fuzzy arithmetic and fuzzy ranking. The other tasks may will be modified or be added with extra functions for future approaches. To make a concise structure of the system to FMMRCPS, The methodology of modular design is a suitable way of taking up these concerns. The feature of a module has its own unique task or function. The advantage of applying this methodology in such a system development is that any modification on a particular module will not affect other modules in the system. The other benefit of using modular design is that many common tasks treated as modules can be easily reusable.

During the system development, each module has been verified for correctness by application to the sample data before the modules were assembled together. Thus, the system will reliably implement any of the four metaheuristic approaches in handling FMMRCPS. The system for FMMRCPS is implemented using four stages: (a) the input of data for a project manually or data to be generated by the system, (b) the input of parameters' values for the selected approach, (c) implementing the selected approach and (d) the result output as either a single result or the results with statistical analyses.

Figure 2 shows the overall structure of the system to FMMRCPS. The system is decomposed into five main parts: (a) interfaces for the user input, (b) individual main structures for each approach built into the system, (c) perturbation algorithms for manipulating activity priorities and modes, (d) fuzzy scheduling mechanism, and (e) output interfaces.

During the system development, most situations that might occur in project scheduling are taken into consideration. In terms of activity duration times, the system provides several flexible options to handle either fuzzy or crisp times, or situation containing both type of data. The system also provides choices for dealing with either single mode
or multiple mode resource-constrained project scheduling. To set up project instances, one, or a set of projects, can be generated manually by the user or generated automatically by the system. In addition, the system has the facilities for statistical analysis for evaluating the approaches that are developed.

![Diagram](image)

Fig. 2 The system integrating four metaheuristic approaches to FMMRCPS

The object-oriented programming (OOP) methodology is used in the whole system development. The OOP is an efficient approach of developing software of this kind. The new generation of VB.net is coded in the system because there are a number of benefits. First, the system written in VB.net is much more secure than many other programming languages. Second, the system written in this language is able to be run on any computers and operating systems. Third, the system written in VB.net not only can be used on a standalone computer but, importantly, can be shared by different departments across an organisation on the network base, used by multiple users.

Overall, the salient features of the system developed for FMMRCPS can be summarised below:

- The system provides a user friendly interface;
- The system can handle both crisp and fuzzy activity duration times;
- The system can deal with both classical project scheduling with single mode and multi-mode project scheduling;
The system can generate project information either manually or automatically;

- The system provides several metaheuristic approaches for solving FMMRCPS;
- The system allows intensive experiments to be conducted for evaluating individual approaches or for determining the values of parameters of individual approaches;
- The system can provide a crisp report about the details of experimental results and the relevant information such as CPU time for individual runs;
- The system not only can be run on standalone computers, but also can be distributed through networks and used by multiple users;
- The system is compact and allows common modules to be shared, because of the application of the modular concept during system development;
- The system structure is well organised and is easy to maintain with understandable coding;
- The system provides a number of open options for adding other approaches.

5 Conclusion

FMMRCPS is a realistic project scheduling model, allowing activities to have flexible performance modes under resource constraints where activity duration times are fuzzy and each mode of an activity has its unique fuzzy duration time. This scheduling model covers most practical situations, and it is a generalised case in resource-constrained project scheduling. However, it is more complex than classical single mode project scheduling. To tackle FMMRCPS under realistic settings, four metaheuristic approaches are developed and built into the system. The system developed is a complete system with user-friendly interfaces from data input to result output, and with detailed statistical analysis for individual FMMRCPS approaches. The system is reliable because each module has been tested logically and technically from the sample project data, and the whole system has also been tested by several project instances with either crisp or fuzzy data or the combination of both. The FMMRCPS system developed has practical significance in proving a useful tool for solving any practically sized project scheduling problems where the activity duration times are fuzzy and activities can be performed in one of several executive modes under precedence relationships and resource constraints.

These metaheuristic approaches developed in the system have their own advantages when applied in different circumstances. Fuzzy GA alone is suitable for conditions where a schedule result is required immediately. Fuzzy GA with tabu may get better results than fuzzy GA alone, but it requires significant computation time. It can be used when the time required for solution is less important. If the user is familiar with the parameter settings, the fuzzy SA is a better approach than fuzzy GA alone when the computation time is of concern. The fuzzy SA with tabu outperforms any other approaches. But it requires much more computational time than any of others. This approach can be used in conditions where a very good result is required without limitation on time to solution.

References:


