

FUZZY APPLICATION FOR SAND MINING PROJECT SCHEDULING WITH MINIMIZED MAKESPAN

Paryati1*

¹*Informatics Engineering, Faculty of Industrial Technology, UPN "Veteran" Yogyakarta, Indonesia
Street Babarsari No. 2 Tambakbayan Yogyakarta 55281
e-mail: yaya_upn_cute@yahoo.com, upnyaya@gmail.com

Abstract

The development of this software aims to solve the problem of resource allocation (renewable and nonrenewable) of sand mining in the Kulon Progo area with activities that have a logical sequence of implementation and do not hinder one another (nonpreemptable) to minimize the function of project completion time criteria. (makespan) in time parameters (ready time, duration time, deadline time, start time and end time) that are uncertain. The uncertainty of the time parameter is modeled as a (Left, Right, α , β) LR type fuzzy number, with three values of α -cuts $\{E = 0.2, L=0.6, \text{ and } I=1.0\}$. This fuzzy model is transformed into a deterministic model based on three pairs of inferior and superior values of α -cuts. Priority rules for scheduling activities based on the smallest EST (early start time) value. Conflicts over the use of resources are resolved in two ways, namely serial and parallel models. The best solution is determined based on the smallest makespan value. Solving this problem involves operations on fuzzy values, including arithmetic operations and relational operations that have certain rules. The JPF software provides two types of schedules, namely optimistic and pessimistic schedules.

Keywords: Project Scheduling, Fuzzy, Makespan, Optimistic, Pessimistic.

1. Introduction

Project scheduling with resource supply constraints (RCPS: Resource Constraints Project Scheduling) is aimed at being closer to the real world, especially for project scheduling in the sand mining area in Kulon Progo [Slo-89, Weg-90]. The RCPS models that have been developed include multi-category resources [Weg-80], multi-mode activity formation [Pat-90], multi-criteria function criteria [Slo-81, Slo-89], and a combination of the three models. the [Hap]. An aspect that has so far received little attention in RCPS problem modeling is the uncertainty characteristic of the implementation time parameters of each activity in a project. Actually, this has been handled in the PERT (Project Evaluation and Review Technoloque) method with a probabilistic approach [Tah-87], but this technique ignores resource supply constraints. A probabilistic approach can be used only if historical data is available regarding experience in implementing similar projects. If the project is something new or the techniques and methodology used to carry out the project are completely new (for example, several new techniques and methodologies in software engineering, including: object-oriented design and programming, computer-aided software design, user interface management systems, fourth generation languages, etc.), it is no longer possible to use a probabilistic approach. In such situations, a decision maker must be able to estimate the duration of each activity in a project based on the experience he has, related to the level of knowledge he has about the new techniques and methodologies and the level of expertise of the available human resources. This estimation method is more suitable to be represented in the form of fuzzy numbers, namely fuzzy sets in the universe of real numbers which are normal, convex and closed intervals. In this sand mining project scheduling system, the effect of transportation congestion results in uncertainty in some or all of the coefficients in the objective function, such as transportation costs or delivery times, which cannot be known with certainty. "A way to deal with uncertainty in decision making uses fuzzy principles (Lutfi A. Zadeh, 1965)".

2. Literature Review

2.1 Fuzzy Data Representation

Aspects of uncertainty are currently widely studied in efforts to develop system models that are closer to the real world. If in the traditional view, science is synonymous with certainty in all its manifestations (precision, specificity, sharpness, consistency, and so on) and all forms of uncertainty are considered unscientific, then in the modern view, uncertainty is considered as something essential that does not need to be avoided, because more representative of the real world.

Each model must pay attention to three basic characteristics as a benchmark for approaching the real world, namely the level of complexity, level of reliability, and uncertainty. Uncertainty has an important role in maximizing the usefulness of a system model. In general, the more a system is able to tolerate uncertainty, the less and the more the system's reliability increases.

The use of this form of uncertainty began in the 19th century through the laws of Newtonian Mechanics (physical processes at the molecular level). Subsequently, it was replaced by probability theory which is able to represent forms of uncertainty with fuzzy patterns.

2.2 Fuzzy Sets

In 1965, Lotfi A. Zadeh offered fuzzy set theory to represent this form of vague uncertainty, namely an object that has membership with imperfect boundaries (impression). Membership in a fuzzy set is not based on the crisp Aristotelian dual-valued logic theory (affirmation = full member and denial = not a member), but based on a certain degree of membership or membership function.

2.3 Determination of Fuzzy Set Membership Functions

Fuzzy set membership functions can be determined directly or indirectly by involving one or many experts. There are several methods that can be used to determine the membership function of fuzzy sets [Kli-95].

2.4 Project Scheduling

Project scheduling is making the implementation of each activity in a sand mining project by optimizing the efficient use of time and available resources, but the precedence between activities is still fulfilled. Project management has developed rapidly along with the determination of two analytical methods for planning, scheduling, and controlling in a project, namely the PERT method (Project Evaluation and Review Technique) and CPM (Critical Path Method). Both of these methods were developed in the same period (1956-1958) by two different groups. PERT was developed by a consulting firm used for scheduling research and development projects for the Polaris missile program within the United States Navy (US-Navy), while CPM was first developed by E.I. du Pont de Nemours & Company and further developed by Mauchly Associates for use in construction projects. PERT and CPM are purely time-oriented project scheduling methods, as they ignore resource supply constraints. The difference between these two methods is probabilistic, while the CPM method uses deterministic time estimation. Furthermore, these two methods are combined into one known as the PERT-CPM method. The planning phase begins with breaking down the project into several activities, determining the resource requirements and duration of each activity, followed by creating a network of activities that present the preferences among activities in graphic form. The scheduling phase aims to determine the start and end times of each activity according to the precedence that exists between these activities, and can show which activities are on the critical path in the network of activities that require special attention so that the project can be completed on time. The control phase aims to periodically evaluate the progress of project implementation, and if the implementation is too fast or too slow, then the schedule for subsequent activities must be reviewed. The project scheduling problem is characterized by four components, as follows:

$$S = \{R, Z, P, C\}$$

With a set of R resources, a set of Z activities, a set of preferences among P activities, and a set of C criteria functions as a measure of project implementation performance.

3. Method

3.1 Fuzzy Number Modeling

An approach that is often used to solve optimization problems with several parameters with uncertain values [Hap-93, Hap-94], as follows;

- 1) Find a suitable model for each problem parameter whose value is uncertain.
- 2) Transformation of each problem parameter which has an uncertain value becomes a deterministic value.
- 3) Find a solution for each deterministic parameter and check the results if the results are not feasible, then repeat step 2).

Step (1)

This problem uses the L-R flat type fuzzy number method which is included in the bell curve category (chapter II, section 2.1.2), proposed by Dubois D. and Prade H. [Hap-93, Hap-96] with the following notation:

$$M = (I_{\text{Left}}, I_{\text{Right}}, \alpha, \beta)_{LR}$$

$[I_{\text{Left}}, I_{\text{Right}}]$ is the core of the fuzzy number M , with I_{Left} and I_{Right} respectively the left (inferior) and right (superior) boundaries of the core interval. While α and β are the left and right sides of the fuzzy numbers, respectively.

The membership function of the L-R flat type fuzzy numbers is defined as follows:

$$\mu_M(x) = \begin{cases} L \left[\frac{I_{\text{Left}} - x}{\alpha} \right] & \text{if } x < I_{\text{Left}} \\ 1 & \text{if } x \in [I_{\text{Left}}, I_{\text{Right}}] \\ R \left[\frac{x - I_{\text{Right}}}{\beta} \right] & \text{if } x > I_{\text{Right}} \end{cases}$$

$$L[0] = R[0] = 1 \text{ and } L[1] = R[1] = 0$$

A tie flat fuzzy number model with three α -cuts ($I=1.0$, $L=0.6$, $E=0.2$) and five smooth parts (a,b,c,d,e) can simplify this problem (Figure 3.1 (a)).

Alpha Cut

For every $\alpha = I$ and $\mu(x) = I$, x is definite.

For each $\alpha = L$ and $\mu(x) \geq L$, x is worth Feasible.

For any $\alpha = E$ and $\mu(x) < E$, the value of x is ignored.

Step (2)

Each α -cut has a left (inferior) and right (superior) limit, so that from the three α -cuts, six deterministic real number values are obtained, namely E_{Kiri} , L_{Kiri} , I_{Kiri} , I_{Kanan} , L_{Kiri} , and E_{Kiri} .

3.2 Activity Formation Modes

This problem does not include the aspect of diversity of activity formation modes (one activity formation model).

3.3 Precedence among Activities

The model chosen for representation of preferences among activities in this problem is A-O-N (Activity-On-Node), on the grounds that this model is easier to implement in the environment of Object-Directed Analysis (OOA), Object-Directed Design (OOD), and Object-Directed Programming (OOP). The node is represented as an activity object which contains information about the number, name, processor, resource allocation, ready time, duration, due date, start time, and finish time. Arcs are represented as connection instances between activity objects whose information is stored in the activity's successor and predecessor objects.

3.4 Criteria

The criterion to be achieved in this problem is minimizing the project makespan, namely the time span for project implementation.

3.4.1 Fuzzy Time Parameters

All parameters related to time terminology in project scheduling are modeled as fuzzy numbers.

3.4.2 Minimizing Project Makespan

The heuristic function chosen to minimize makespan in this problem is MINEST, which is based on the smallest EST value.

3.4.3 Project Scheduling Method

There are two project scheduling methods, namely the serial method and the parallel method. The two methods have differences in how they handle resource conflicts. The serial method schedules activities based on their priority order sequentially. The parallel method tries to schedule activities based on their priority order, but if the activity that has a turn at this time cannot be scheduled, then the turn is given to the next priority order.

4. Discussion

4.1 Software Architecture

This software architecture consists of a project scheduling component and a user interface component. The project scheduling component provides two scheduling methods, namely the serial method and the

parallel method. The user interface component manages the reception of input data and the presentation of output results in graphical form.

4.2 Input Data

- a. Resource Table, consisting of number, name, type and quality of resources.
- b. Activity Table, consisting of number, name, precursor, resource allocation, ready time (optional), duration, and deadline (Optional).

4.3 Output Results

- a. Activity table, consisting of number, name, predecessor, resource allocation, ready time (optional), duration and deadline time (Optional), start time and finish time.
- b. Graph that represents the allocation of resource requirements for an activity (resource requirements).
- c. Graph that represents the profile of the use of certain resources within the time span of carrying out activities (resource profile).
- d. Graph that represents a network of activities (precedence constraints) with an A-O-N (Activity-On-Node) representation model.
- e. Graph that represents the fuzzy makespan of a project (project makespan).
- f. Graph that represents the activity implementation schedule in the form of a fuzzy Gantt chart.

4.4 Object Oriented Design

Object-Directed Design (OOD) aims to develop a complete system, including:

- a. Problem Domain Components
- b. Human Interaction Component
- c. Task Management Components
- d. Data Management Components

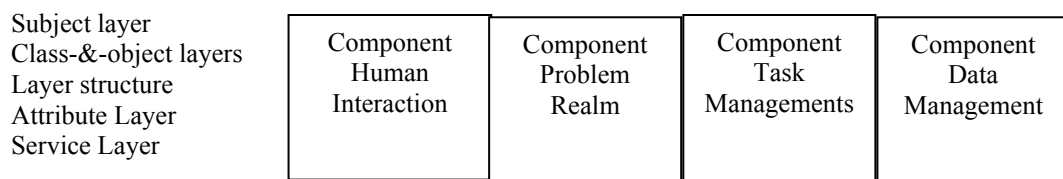


Figure 1. Multi-Layer and Multi-Component Model

Each component has a subject, class-&-object, structure, attribute, and service layers. the analysis phase describes the Problem Domain Components, while the Human Interaction Components, Task Management Components, and Data Management Components will be developed at this design stage.

5. Program Implementation

5.1 Programming

The software is implemented using the Borland C++ Builder programming language version 4.0 which is the development of the C++ language for Window-based programming environments. This tool provides a VCL (visual component library) class library that supports application programming in graphical displays, and a form designer to create graphical interfaces easily and quickly [Mis-97].

The C++ language is a development of the C language equipped with concepts from the Simula67, Algol68, and Ada languages. The concept of class is taken from Simula67 language, the concept of operator overloading and placement of declarations between instructions is taken from Algol68 language, and the concept of template and exception is taken from Ada language [Han-98].

5.2 Concepts of Classes and Objects

The concept of classes and objects in C++ aims to create new data types that are identical to the basic data types (eg int, float, char, string) owned by these languages. A basic data type consists of two components, namely a set of bits that represent an abstract value when the base type is declared and a set of operations that apply to it (eg +, -, *, and / in the data type float). A class also consists of a set of data bits (attributes) that represent the abstract value of the object of the class and the set of operations (services) it has. The concept of class and object can be expressed by analogy as follows:

Class vs object \equiv type vs variable

5.3 Class Definition

A class or object in C++ must be defined like a data structure by changing the struct keyword to class and declared in the filename_file.h header.

5.4 Input Data Classes

Input Data Presenting Screen Renderer Class

The TFormInput class provides data entry facilities from the keyboard. Validation of input data is filtered through object instantiation of the TMaskEdit class. Information about the presentation status of the display screen is built by object instances from the TLabel class. The OK and Cancel buttons are instantiated by the object instantiation of the TBitBtn class. classTMaskEdit, TLabel, and TBitBtn are provided by Borland C++ Builder. The TFormInput class declaration is shown below.

```
//-----  
---  
//  
//          DEKLARASI KELAS TFormMasukan  
//-----  
---  
Class TFormMasukan : public TForm  
(  
  _published:      // IDE-managed Components  
    TBitBtnpan* BitBtn1  
    TBitBtnpan* BitBtn2  
    TMaskEdit* MaskEdit1 ;  
    TLabel* Label ;  
  
    Void_fastcall Tampil(TObject* Sender)  
    Void_fastcall tutup(TObject* Sender, TCloseAction& Action)  
  
Public:      // User declarations  
//-----  
---  
//  
//          KONSTRUKTOR  
//-----  
---  
    _fastcall TFormMasukan (TComponent* Owner) :  
  
//-----  
---  
//  
//          SET DATA MASUKAN  
//-----  
---  
    Void masukan (String, string, string, string) :  
  
Protected:      // User declarations  
Private:      // User declarations  
  
};  
//-----  
---  
Extern PACKAGE TFormMasukan * FormMasukan ;  
//-----  
---
```

Figure 2. TFormInput Class Declaration Program

6. Test Results

6.1 Object-directed Tests

There are three tests that must be carried out in testing object-oriented software (OOT) [Pre-97], namely:

- 1) Testing is based on aspects of correctness and consistency.
- 2) Unit testing and integrated testing.
- 3) Testing with test cases.

6.2 Truth and Consistency Test

Correctness and consistency tests are carried out on the results of object-oriented analysis and design.

Correctness aspects include syntax rules and semantic rules. Analysis and design of the JPF software uses notation and writing syntax based on object-oriented analysis and design methods. Semantic correctness is determined based on the suitability between the analysis and design results with the software requirements specifications.

7. The consistency aspect relates to testing the message connection between classes. Consistency checks use the CRC (Class responsibility Collaboration) model in the form of a CRC index card. Each card contains the class name, responsibility for class services, and collaborations with other classes that send messages to request services from this class.

6.3 Test Results at the Implementation Stage

Tests at the implementation stage include class tests, integrated tests and tests with test cases.

```

void_fastcall TFormJPF: : MSerial1Click(TObject* Sender)
(
  Try
  (
    If(!JPF->Enable) delete JPF ;
    JPF = new TJPF ;
    BacaTabSD () ;
    BacaTabK () ;
    JPF->MetodeSerial () ;
    tampilkanHasil () ;
    JPF->Enable = files ;
  }
  Catch (std ; ; bab_alloc)
  {
    Showmessage("Alokasi gagal !") ;
  }
}
// -----
void_fastcall TFormJPF: : MParaleel1Click(TObject* Sender)
(
  Try
  (
    If(!JPF->Enable) delete JPF ;
    JPF = new TJPF ;
    BacaTabSD () ;
    BacaTabK () ;
    JPF->MetodeParalel () ;
    tampilkanHasil () ;
    JPF->Enable = files ;
  }
  Catch (std ; ; bab_alloc)
  {
    Showmessage("Alokasi gagal !") ;
  }
}
// -----

```

Figure 3. Class Test on the MSerialClick and MParallelClick Services

6.4 Integrated Test Results

The integrated test examines each service in all classes and the relationship between services in a class and services in another class. Integrated test results are determined based on changes in status and attribute values in interconnected classes. Software classes are interconnected classes.

The test case uses an example of project data whose output results are known, namely the sand mining project [Hap-96].

Table 1. Validation test cases for input data and output results.

No	Activity	Action
1	Create project data	(1) Select the Project Menu (2) Select the Create data project submenu
2	Updating resource data and/or activities	(1) Right-click the resource table or activity table form to activate. (2) Select the update menu (3) Select the add submenu if it is added at the end of a line, the insert submenu if it is inserted on a certain line, and the delete submenu if it is deleted on a certain line, and the delete submenu if deleted on the selected line. (4) Fill in the added or inserted table rows with input data whose output is known.
3	Saves input data to a specific file	(1) Select the project menu. (2) Select the save as submenu. (3) Specify the directory and file name. (4) Right-click the Save button if it is saved and right-click the cancel button if it is canceled.
4	Create a project schedule	(1) Select the create schedule menu. (2) Select subm, enu serial method or parallel method.
5	Validate output data	(1) Check the output data displayed in the start time and end time columns of the activity table. (2) Compare with known output data
6	Validate the output data in graphical representation	(1) Select the results display menu (2) Select the Activity grid, makespan, or Gantt Chart sunmenu (3) Compare with known graphical representations
7	Test the window facility	(1) Select the window menu (2) Select cascade, tile, arrange icons submenu, or collapse all. (3) Select a vertical or horizontal sub-menu if a tiled sub-menu is selected
8	Retrieves project data from files	(1) Select the project menu (2) Select the submenu retrieve project data (3) Look for the directory and file name (4) Right-click the open button if it is taken and right-click cancel if it is cancelled
9	Create a project schedule (project data from files)	(1) Select the create schedule menu (2) Select the serial method or parallel method submenu
10	Validate output data (project data from files)	(1) Check the output data displayed in the start and finish time lines of the activity table (2) Compare with known output results
11	Validate output data in graphical representation of project data from files	(1) Select the show results menu (2) Select the activity network submenu, makespan, or Gantt Chart (3) Compare it with a known graphical representation
12	Go out	(1) Select the project menu (2) Select the exit submenu

7. Conclusion

The results of the analysis and testing of the system that has been built are the output of project scheduling from this JPF software which is presented in a fuzzy Gantt Chart graphical representation. It can be concluded that every activity in fuzzy project scheduling must be scheduled with an optimistic degree greater than or equal to the optimistic degree of the predecessor. The uncertainty of the time parameter is modeled as a (Left, Right, α , β) LR type fuzzy number, with three values of α -cuts $\{E = 0.2, L=0.6, and I=1.0\}$. This fuzzy model is transformed into a deterministic model based on three pairs of inferior and superior values of α -cuts. Priority rules for scheduling activities based on the smallest EST (early start time) value. Conflicts over the use of resources are resolved in two ways,

namely serial and parallel models. The best solution is determined based on the smallest makespan value. Solving this problem involves operations on fuzzy values, including arithmetic operations and relational operations that have certain rules. The JPF software provides two types of schedules, namely optimistic and pessimistic schedules.

References

Adianto, A., & Miharja, R. (2019). Proposed Scheduling Of Yarn Production Using Neh Method And Johnson'S Algorithm To Minimize Production Time At Pt. Exercise The Gift Of True Independence. *Scientific Journal Of Industrial Engineering*, 6(3).

Annisya, S. D., & Saifudin, J. A. (2020). Analysis Of Refire Stone Production Scheduling Using The Campbell Dudek Smith (Cds), Nawaz Ensore Ham (Neh), And Palmer Method To Reduce Makespan At Pt. X. Juminten, 1(3), 165–176.

Baharuddin, A. V., Afris, W. H., & Saputri, Y. I. (2022). Measurement Of Standard Working Time In The Production Process At Ikm Kampar Galesong Donuts. *Journal Of Agro-Industry Engineering Research*, 1(1), 58–62.

Cox E., 2014, *The Fuzzy Systems Handbook*, AP Proffesional, Cambridge.

Iskandar, D., Masruri, A. A., & Saputra, D. (2018). Analysis Of Job Shop Production Scheduling In Smes In The Convection Sector Using The Tabu Search Algorithm Method (Case Study In Panca Konveksi). *Integration: Scientific Journal Of Industrial Engineering*, 3(2), 21–27.

Klir, G.J. and Folger T.A., 2018, *Fuzzy Sets, Uncertainly and Information*, Prentice Hall International Inc., London.

K.S., N. O. (2019). Production Scheduling Of Retreaded Tires Using The Branch And Bound Method And Campbell Dudeck Smith At Cv. New Nuance. *Valtech Journal*, 2(2), 108–116.

Latief, A., Melu, P. F., Lahay, I. H., & Hasanuddin, H. (2021). Measuring Employee Working Time In Crystal Ice Packaging Using The Time Study Method. *Jambura Industrial Review (Jirev)*, 1(2), 48–57.

Mashuri, C., Mujianto, A. H., Sucipto, H., & Arsam, R. Y. (2020). Application Of The Campbell Dudek Smith (Cds) Algorithm For Optimizing Production Time In Production Scheduling. *Jsinbis (Journal Of Business Information Systems)*, 10(2), 131–136.

Muharni, Y., Febianti, E., & Sofa, N. N. (2019). Minimizing Makespan In Flow Shop Scheduling For Parallel Machines For Steel Bridge B-60 Products Using The Longest Processing Time Method And Particle Swarm Optimization. *Industrial Services Journal*, 4(2).

Nuryawan, T., & Dwiwinarno, T. (2020). Standard Time Measurement For Achieving Productivity Case Study Of Making Elementary School Uniforms At Cv. Focus Production Tamansari, Kalasan, Sleman. *Effective Journal Of Economics And Business*, 11(2), 133–144.

Pranata, A., 2017, *Pemrograman Borland Delphi, Andi Offset*, Yogyakarta.

Pressman, R.J , 2017 , *Software Engineering A Practitional Approach*, Mc Graw Hill, London

Pradana, A. Y., & Pulansari, F. (2021). Analysis Of Measurement Of Working Time With a Stopwatch Time Study To Increase Production Targets At Pt. Xyz. *Juminten*, 2(1), 13–24.

Rahayu, M., & Juhara, S. (2020). Measuring Pen Assembling Standard Time Using Downtime Clocks During Work Design Analysis Practicum. *Education And Industrial Applications (Unistek)*, 7(2), 93–97.

Sari, E. M., & Darmawan, M. M. (2020). Measurement Of Standard Time And Workload Analysis In The Filling And Packing Process Of Lulur Mandi Products At Pt. Gloria Origita Cosmetics. *Asiimetrik Journal: Scientific Journal Of Engineering & Innovation*, 51–61.

Sari, P. M. (2020). Proposed Production Scheduling With The Dudek Smith Campbell Method For Personal Care Products At Pt. Lf Beauty Manufacturing Indonesia. *Journal Of Industrial Engineering Optimization (Joti)*, 2(2), 60–65.

Sholeh, M., Asih, E. W., & Sodikin, I. (2021). Optimal Work Scheduling To Minimize Delays At Pt Mandiri Jogja Internasional. *Journal Of Recreation*, 9(1), 35–42.

Sidabutar, S. N. S., Amin, M., & Putri, A. (2020). Scheduling Production Machine Operations Using The Cds (Campbell Dudek Smith) Method At Pt Tjokro Bersaudara Balikpapan.