Research Article

Project Delivery of Goods With Limited Resources and Minimum Time Using Fuzzy Logic Method

Paryati[1], K. Martin Sagayam[2], Krit Salahddine[3], Ahmed A. Elngar[4], Hanafy M. Ali[5]

[1]UPN “Veteran" Yogyakarta, Country Indonesia

[2]Karunya Institute of Technology and Sciences', Country India

[3]Ibn Zohr University, Agadir, Country Marocco

[4]Beni-Suef University, Faculty of Computers and Artificial Intelligence', Beni-Suef city, 62511, Egypt,

[5]Computer and Systems Engineering Department, Faculty of Engineering, Minia University, El Minia, Egypt.

Article History: Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; Published online: 5 April 2021

Abstract

In solving the problem of fuzzy application for scheduling delivery of goods with a limited power source and minimal time, the problems that are still not considered in the RCPS problem modeling are the uncertainty characteristics of the parameters of the timing of activities in the delivery project. Even though this can be solved by using the PERT (Project Evaluation and Review Technoloque) method with a probabilistic approach, this technique still ignores the limitations of the supply of resources. Actually, a probabilistic approach can be used, if previously provided data about the experience in completing similar projects. But if the project is a new project or the techniques and methodologies used to complete a new project, such as new techniques and methodologies in software engineering, among others: object-oriented design and programming, computer-aided software design, user interface management systems, fourth generation languages, etc., then the probabilistic approach is not suitable.

In this situation, the decision maker must be able to estimate the cost and time duration, of all activities in the project based on existing experience, related to the level of knowledge they have, about new techniques and methodologies to be applied, and the level of human resource expertise. which are available. This method of estimating project costs and time, which is more precise, uses representations in the form of fuzzy numbers, namely fuzzy sets in the set of real numbers that are normal, convex, and closed intervals. The delivery time is modeled as a fuzzy number of LR types IKiri, IKanan, α, β, with three values of α-cuts E = 0.3, L = 0.7, and I = 1.0. The fuzzy transformation model is based on three pairs of inferior and superior values from α-cuts. The priority for scheduling delivery of goods is based on the smallest early start time EST value. Resources are solved by serial and parallel models. The smallest makespan value is used to determine the best solution. Goods delivery settlement uses fuzzy operations, namely arithmetic operations and relation operations.

Analysis of the output oftware based on testing with test scenarios (table ^ .50) for some input data shows that the parallel method is better than the serial method. This is indicated by the large difference in makespan value generated from the two methods. Based on input data from a software development project, the serial method gives makespan values in the range between 675.0 and 867.0, while the parallel method gives makespan values in the range between 116.0 and 259.0. The analysis of the output software in a fuzzy Gantt Chart representation shows that an activity can be scheduled with varying degrees of optimism. The degree of optimistic activity scheduling can be graded in linguistic terminology between two extreme pessimistic and optimistic values, namely very pessimistic, pessimistic, slightly pessimistic, slightly optimistic, optimistic, very optomistic.

Keywords: Fuzzy Logic, Scheduling, PERT, Makespan.

1. INTRODUCTION

Project scheduling is making the implementation of every activity in a project by optimizing the efficiency of using time and available resources, but the suitability of precedence between activities is still met. 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. Then these two methods are combined into one which is known as the PERT-CPM method. The planning phase begins by dividing (break down) the project into several activities, determining resource requirements and the duration of each activity, followed by creating a network of activities that present predictions among activities in graphical form. The scheduling phase aims to determine the start and end times of each activity according to the prediction 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.

Actually, a probabilistic approach can be used, if previously provided data about the experience in completing similar projects. But if the project is a new project or the techniques and methodologies used to complete a new project, (such as new techniques and methodologies in software engineering, among others: object-oriented design and programming, computer-aided software design, user interface management systems, fourth generation languages, etc.), then the probabilistic approach is not suitable. In this situation, the decision maker must be able to estimate the cost and time duration, of all activities in the project based on existing experience, related to the level of knowledge they have, about new techniques and methodologies to be applied, and the level of human resource expertise. which are available. This method of estimating project costs and time, which is more precise, uses representations in the form of fuzzy numbers, namely fuzzy sets in the set of real numbers that are normal, convex, and closed intervals.

2. LITERATURE REVIEW

2.1 Project Scheduling Phase.

The planning phase begins with dividing the project into activities, determining resource requirements and duration, followed by creating a network of activities. The scheduling phase aims to determine the start and end times of each activity, and can show activities that are on a 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. Project scheduling problems are characterized by four components, as follows:

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

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

2.2 Resource.

R resources are divided into three categories, as follows:

a. Renewable resources, consisting of p types, namely Rpr, ………., Rpr with inventory limits of Nlr, ……., Npr units for each time period t.

b. Non-renewable resources, consisting of v types, namely Rln, ………., Rvn with each supply limit of months, ………., Bvn units, and decreasing in proportion to the number of uses.

c. Doubly-constrained resources, consisting of u types, namely Rld, ………., Rud with renewable inventory limit of Nld, ………., Npd each nonrenewable initial inventory limit unit as much as Bld, …… …., Bvd unit.

2.3 Activity Forming Mode.

In the planning phase, the project is divided (break down) into several activities. Each activity has a specific role in the project, that is, the same activity cannot be repeated several times.
The resource requirements for an activity Zj in the activity set Z are usually discrete. If viewed from the point of view of resource requirements, each activity can be formed in several modes of activity formation. An activity Zj with activity formation mode m will require multi-category resources, rrmkj, rnmlj, rdmhj, where k = l, ……, p, l = l, …… .., v, h = l,…., u, and the duration pmj. There is a logical relationship between the supply of resources (logistics) and the duration of an activity, that is, if the logistics are limited (for renewable) or less smooth (for non-renewable), the duration will increase.
Information about the mode of formation of this activity is very useful for optimization of resource use, especially for the category of renewable resources, because this resource can be used concurrently for several activities with a specific activity formation mode, provided that for each mode of activity formation there is nothing preemitive.
There are two models of activity networks, namely:

a. Activity-On-Node (A-O-N) Model.
b. The Activity-On-Arc (A-O-A) model.

2.4 Criteria.

Criteria are minimizing project makespan, maximum lateness, resource profile smoothness, mean flow time, net present value, and project cost.
Project scheduling classification, as follows:

a. Project scheduling with multi-category resources.
b. Project scheduling with the formation of multi-mode activities.
c. Project scheduling with a multi-criteria criterion function.
d. Project scheduling with multi-category resources, multi-mode activities, and multi-criteria functions.

2.5 Terminology project scheduling.

Namely :

a. Ready time.
b. Duration.
c. Time delay.
d. Start time.
e. Time's up.
f. EST is the time the activity starts at the earliest.

2.6 Fuzzy Data Representation

Uncertainty aspects are currently being studied a lot in efforts to develop a system model that is closer to the real world. Each model must pay attention to three basic characteristics as benchmarks for the approach to the real world, namely the level of complexity, the level of reliability, credibility, and the uncertainty. Uncertainty has an important role in maximizing the usefulness of a system model. In general, the more able a system is to tolerate uncertainty, the less complexity and more reliable it will be.

2.7 Fuzzy Sets

Fuzzy set theory to represent this form of vague uncertainty, namely an object that has membership with imprecise boundaries imprecision membership in a fuzzy set is not based on the theory of bi-value logic. Aristotelian which is crips affirmation = full and denial = not member, but based on a certain membership degree of membership or membership function. The membership function is represented by a number value that lies in the interval [0,1], with the extreme values 0 and 1 indicating the denial and affirmation of its membership, respectively.

2.8 Determination of Fuzzy Set Membership Functions

The fuzzy set membership function can be determined directly or indirectly by involving one or many experts. There are several methods that can be used to determine the fuzzy set membership function, namely: Direct Method with one Expert, Direct Method with Many Experts, Indirect Method with one Expert, Indirect Method with Many Experts.

2.9 Fuzzy Numbers

Fuzzy number M is a fuzzy set A in the set of real universe $\Re$ which meets the following conditions :

a. A fuzzy set must be in the form of a normal fuzzy set, because if the fuzzy number is expressed as a set of real numbers around x, it is certain that the membership function of x itself is 1.

b. Fuzzy set A must be a closed interval for each membership function $\mu A(x) \in [0,1]$, so that each fuzzy number must be in the form of a convex fuzzy set.

c. Support for fuzzy set A must be in the form of a closed (bounded) interval, so that each convex fuzzy set is not necessarily categorized as a fuzzy number, because the support may be in the form of open or half open intervals.

2.10 Representation of Fuzzy Numbers

When viewed from the characteristics of the membership function, the representation of fuzzy numbers is divided into two categories, as follows:

a. Linear Representation

There are two linear representations, namely:

a. Trapezoidal Figure 1a.
b. Triangular Figure 1b.

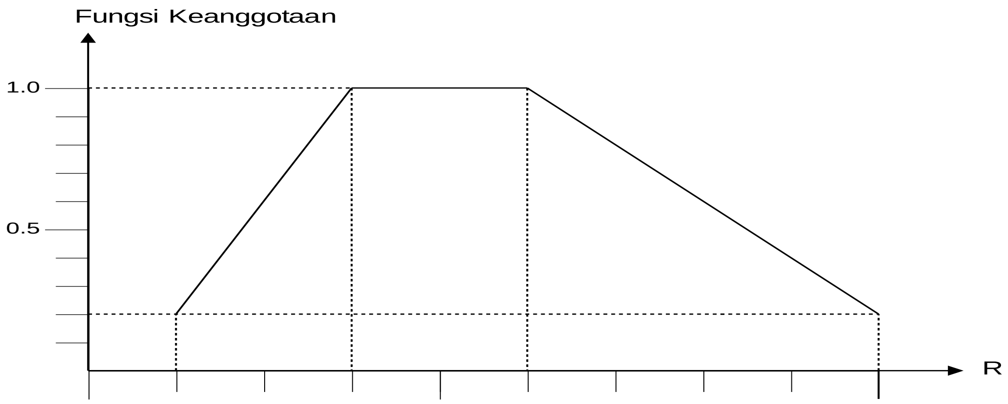

Figure 1a. Trapezoidal Fuzzy Number

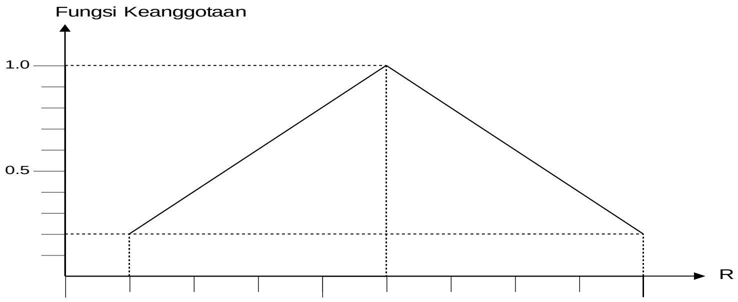

Figure 1b. Triangular Fuzzy Number

b. Representation Arround

The arround representation is also called a bell curve because it is shaped like a bell. There are three classes of bell curves, namely PI-Curve, Beta-Curve, and Gaussian-Curve. The curves can be balanced Figure 2.

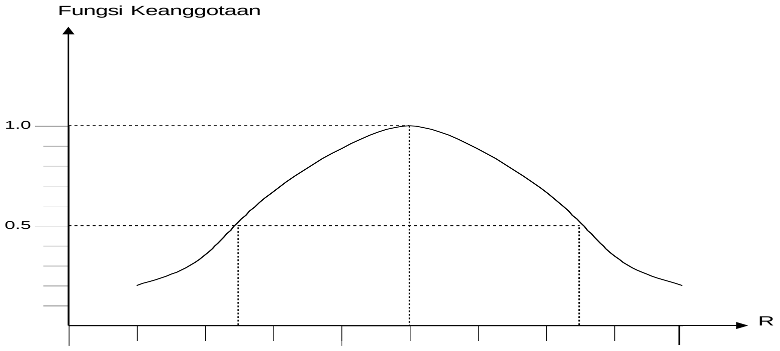

Figure 2. Balanced Bell Curve

2.11 Systems engineering work cycle using the waterfall model.

In simple terms the system engineering work cycle can be described as follows: System Engineering, Analysis, Design, Code, Testing.

2.12 Project Scheduling Methods.

Much research has been done regarding scheduling problems. In general, scheduling problem solving methods, including project scheduling can be categorized as follows:

a. Manual Scheduling

This method, although it requires high effort, can incorporate unstructured human expertise into the scheduling process.

b. Computer Simulation Scheduling

This method uses statistical functions to simulate scheduling behavior in the real world using a computer device as a tool.

c. Mathematical Approach

This method uses operational research (operation research), such as Integer programming, Dynamic Programming, and heuristic methods (eg, neighborhood search).

d. New Method

This category of methods includes:

- Metaheuristic, based on intensification or diversification to achieve problem solving results (eg taboo search, simulated annealing, and genetic algotithm).
- Partial Enumeration, using Beam search.
- Bottleneck, the translation of the OPT principle.
- Application of an expert system (expert system), neural network (neutral nertwork), and a combination of AI, OR, DSS.

e. Dynamic Bottleneck

This method includes an estimate of the costs that must be incurred in the event of delays in the implementation of one or more activities on the way.

3. RESEARCH METHOD

3.1 Project Scheduling Methods

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

3.2 Fuzzy Number Modeling Steps

The approach that is often used to solve optimization problems with several parameters is uncertain as follows :

a. Find a suitable model for each problem parameter whose value is uncertain.

b. The transformation of each problem parameter with uncertain value becomes deterministic.

c. Find a solution for each deterministic parameter and check the results if the results are not yet 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, put forward by Dubois D. and Prade H. [Hap-93, Hap-96] with the following notation:

M = (IKiri, IKanan, α, β) LR

[IKiri, IKanan] is the core (core) of fuzzy M number, with IKiri and IKanan being the left (inferior) and right (superior) limits of the core interval, respectively. Whereas α and β are the left and right sides of the fuzzy number, respectively.

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

L [(II - x) / α] if x <II
μM (x) = 1 if x Є [IKiri, IKanan] α, β Є R +
R [(x-IKanan) / β] if x> IKanan
L [0] = R [0] = 1 and L [1] = R [1] = 0

The flat tioe fuzzy number model with three α-cut (I = 1.0, L = 0.6, E = 0.2) and five smooth parts (a, b, c, d, e) can simplify this problem .

Alpha Cut

- For every α = I and μ (x) = I, x is definite.
- For each α = L and μ (x) ≥ L, x is Eligible.
- For each α = E and μ (x) <E, the x value is ignored.

Step (2)

Each α-cut has a left (inferior) and a right (superior) limit, so that from the three α-cuts, six deterministic real number values are obtained, namely EKiri, LKiri, IKiri, IKanan, LKanan, and EKanan.

Step (3)

Each value of the six deterministic values is solved in the usual way, so that we get six deterministic solutions. If the six deterministic solutions are combined, the results are obtained in fuzzy number terminology.

3.3 Defining Fuzzy Number Arithmetic Operators

There are several arithmetic operators of fuzzy numbers that need to be defined in this problem. If x and y represent two fuzzy numbers based on the definition of this problem. Then the following applies:

Addition

Increment is defined as follows:

x + y = ((x.EKiri + y.EKiri), (x.LLKiri + y.LKiri), (x.IKiri + y.IKiri), (x.IKRan + y.IKRan), (x.LKRight + y. L right), (x. PRESS + y. RIGHT))

Addition

The reduction is defined as follows:

x - y = ((x.EKiri - y.EKiri), (x.LLKiri - y.LKiri),
(x.IKiri - y.IKiri), (x.IKanan - y.IKanan),
(x. L right - y. L right), (x. press - y. right))

Maximum

The maximum is defined as follows:

Max (x, y) = (max (x.EKiri, y.EKiri), max (x.LLKiri, y.LKiri),
max (x.IKiri, y.IKiri), max (x.IKanan, y.IKanan),
max (x.LRight, y.LRight), max (x.PRESS, y. EKRIGHT))

Minimum

The minimums are defined as follows:

Min (x, y) = (min (x.EKiri, y.EKiri), min (x.LLKiri, y.LKiri),
min (x.IKiri, y.IKiri), min (x.IKanan, y.IKanan),
min (x.LRight, y.LRight), min (x.Pressure, yRight))

4. IMPLEMENTASI PROGRAM

4.1 Test Results at the Implementation Stage

Testing at the implementation stage includes class testing, integrated testing and testing with test cases.

Class Test Results.

Class test checks every service (procedure or function) in a class randomly based on the initial status and the final status. If the initial status of a service in a class is known, then test conditions are created based on its initial state. Borland C ++ Builder provides an exception and handler command block to check the success / failure of executing program instruction code. This command block is inserted in the service code section to check the success / failure of the service implementation. For example, test the class to check the success / failure of the memory allocation of the free store object creation with the new commands on the MSClick and MPclick services of the TFormJP class.

```
void_fastcall  TFormJP: : MSClick(TObject* Sender)
(
   Try
(
        If(!JPF→Enable) delete JP ;
        JPF = new TJP ;
        BacaTabSD ( ) ;
        BacaTabK ( ) ;
        JPF→MetodeSerial ( ) ;
        tampilkanHasil ( ) ;
        JPF→Enable = files ;
    }
    Catch (std ; ; bab_alloc)
    {
       Showmessage(*Alokasi gagal !*) ;
     }
}
/ / -----------------------------------------------------------------------------------------------------------
------------
void_fastcall  TFormJP: : MPClick(TObject* Sender)
(
   Try
(
        If(!JP→Enable) delete JP ;
        JP = new TJP ;
        BacaTabSD ( ) ;
        BacaTabK ( ) ;
        JP→MetodeParalel ( ) ;
        tampilkanHasil ( ) ;
        JP→Enable = files ;
    }
    Catch (std ; ; bab_alloc)
    {
       Showmessage(*Alokasi gagal !*) ;
    }
}
/ / -----------------------------------------------------------------------------------------------------------
------------
```

Figure 3. Class Test Program on MSClick and MPClick Services

The TSumberDaya class represents a resource that is sent as a free-store object whose existence is stored in the TTabSD container class. The TSumberDay class declaration is shown in the Program below.

```
//------------------------------------------------------------------------------------------------------------
-------------
//                                   DEKLARASI  KELAS TSumberDaya
//------------------------------------------------------------------------------------------------------------
-------------
#include "URlProyek. H"

Class TSumberdaya : public TElProyek
(
Public ;                               / /   User declarations

//------------------------------------------------------------------------------------------------------------
-------------
//                                  CONSTRUCTOR dan DESTRUCTOR
//------------------------------------------------------------------------------------------------------------
-------------
         TSumberDaya ( ) ;
         TSumberDaya ( const TNomor, const TNama, const TJenis, const TStok) :
         -TSumberDaya ( ) ;
//------------------------------------------------------------------------------------------------------------
-------------
//                  PROPERTY READ dan WRITE NILAI ATRIBUT
//------------------------------------------------------------------------------------------------------------
-------------

                  _property TJenis Jenis = (read= Atrjenis, write=Setjenis) :
                  _property TStok stokawal = (read= Atrstokawal, write=Setstokawal) :
                  _property TStok stoksisa = (read= Atrstoksisa, write=Setstoksisa) :
                  _property Tcacah cacah= (read= Atrcacah, write=Setcacah) :
                  _property TDeFuzzy maka t= (read= Atrmaka_t, write=Setmaka_t) :
                  _property TLProfilSD* LProfilSD = (read=AtrProfilSD,write=SetLProfilSD) :
                  _property TSumberdaya* Next = (read=AtrNext,write=SetNext) :

Protectd:                              / /   User declarations

//------------------------------------------------------------------------------------------------------------
-------------
//                                               SET NILAI ATRIBUT
```

```
//--------------------------------------------------------------------------------------------------------------
-------------

        Void SetJenis (const TJenis*) ;
        /* I . S.: Sebuah objek dari kelas TSumberDaya  telah diciptakan
           F . S.       : Atribut AtrAwal telah di-set
           Proses       : Set atribut AtrJenis                                    */

        Void Setawal (const TStok *) ;
        /* I . S.: Sebuah objek dari kelas TSumberDaya  telah diciptakan
           F . S.       : Atribut AtrJenis telah di-set
           Proses       : Set atribut AtrAwal                                     */
```

Figure 4. TSumberDaya Class Declaration

5. RESULTS AND DISCUSSION

5.1 Integrated Test Results

Integrated test checks each service in all classes and the relationship between services in one class and services in other classes. Integrated test results are determined based on changes in status and attribute values in the interrelated classes. The pad software classes are interconnected. The classes in this JP software have functioned according to the specification requirements.

5.2 Test Results with Test Cases

Testing with test cases aims to increase confidence in the reliability of the JP software. The selected test cases must be solid to represent all possible events. This JP software test uses a scenario approach with input data validation test cases to get the output results
The test case uses examples of project data with known outputs, namely agricultural projects, and sample small projects.

Table 1. Test Cases for Input Data and Output Results

No	Activities	Action
1	Create project data	1.Select the Project Menu
2	Updating data on resources and / or activities	2.Select Submenu Create project data
3	Save input data to a specific file	1.Right-click the resource table form or activity table to activate it.
4	Create a project schedule	2.Select update menu
5	Validate the output data	3.Select the add submenu if it is added at the end of the line, the insert submenu if it is inserted at a certain line, and the delete submenu if it is deleted on a certain line, and the delete submenu if it is deleted on the selected line.

6	Validation of output data in a graphical representation	4.Fill in the table rows that are added or inserted with input data whose output is known.
7	Test window facilities	1.Select the project menu
8	Extract project data from	2.Select the save submenu as
9	Create a project schedule (project data from a file)	3.Specify a directory and file name.
10	Validation of output data (project data from files)	4.Right-click the Save button if it is saved and right-click the Cancel button if it's canceled
11	Validation of output data in a graphical representation (project data from files)	1.Select the menu for schedule
12	Exit	2.Select subm, enu serial method or parallel method.

5.3 Analysis of the output of software based on testing with test scenarios.

Analysis of the output of software based on testing with test scenarios (table ^ .50) for some input data shows that the parallel method is better than the serial method. This is indicated by the large difference in makespan value generated from the two methods. Based on input data from a software development project, the serial method gives makespan values in the range between 675.0 and 867.0, while the parallel method gives makespan values in the range between 116.0 and 259.0.

5.4 The analysis of the output of software in a fuzzy.

The analysis of the output of software in a fuzzy representation shows that an activity can be scheduled with varying degrees of optimism. The degree of optimistic activity scheduling can be graded in linguistic terminology between two extreme pessimistic and optimistic values, namely very pessimistic, pessimistic, slightly pessimistic, slightly optimistic, optimistic, very optomistic.

5.5 analysis of the results of the graphical output of the resource use profile from software.

Based on the analysis of the results of the graphical output of the resource use profile from software, it can be shown that the available resources have not been used optimally, especially for the category of nonrenewable resources. Analysis of the software output in the Gantt Chart fuzzy graph representation shows that an activity can be scheduled with varying degrees of optimism. The degree of optimistic activity scheduling can be graded in linguistic terminology between two extreme pessimistic and optimistic values, namely very pessimistic, pessimistic, slightly pessimistic, a little bit optimistic, optimistic, very optimistic. If the Kb activity which is the predecessor of the Kb activity has been scheduled with a degree of optimistic ra, then the Kb activity can be scheduled with an optimistic degree of rb ≥ ra.

6. CONCLUSION

Based on the results of the analysis obtained :

a. The project scheduling output from the software that has been made can be concluded that every activity in the fuzzy project scheduling must be scheduled with a degree of optimism that is more than or equal to the degree of optimistic predecessor.
b. Based on the analysis of the results of the resource use profile output from this software, it can be seen that the resources have not been used optimally, especially for the category of non-renewable resources.
c. For the development of this software, it is possible to enter the optimization aspect of resource use into one of the criteria to be achieved.
d. The delivery time is modeled as a fuzzy number of LR types IKiri, IKanan, α, β, with three values of α-cuts E = 0.3, L = 0.7, and I = 1.0.
e. The fuzzy transformation model is based on three pairs of inferior and superior values from α-cuts.
f. The priority for scheduling delivery of goods is based on the smallest early start time EST value.
g. Resources are solved by serial and parallel models.
h. The smallest makespan value is used to determine the best solution.
i. Goods delivery settlement uses fuzzy operations, namely arithmetic operations and relation operations.

BIBLIOGRAPHY

[1] Arifudin, R., 2011, "Optimizing Project Scheduling by Balancing Costs Using a Combination of CPM and Genetic Algorithms", Journal of Informatics Society, 2 (4).

[2] Adi Nugroho, 2020, Object-Oriented Software Engineering using the USDP Method, Andi Publisher, Yogyakarta.

[3] Cox E., 2014, The Fuzzy Systems Handbook, AP Proffesional, Cambridge.

[4] Fonseca, Carlos M, and Fleming, 2015, Genetic Algorithm Formulation Objective Optimazation: Formulation, discussion and Generallization, Dept. Automation & Controller System Engineering, University of Sheffield, UK.

[5[Hamzah M., Unas S.E., and Widiarsa, 2012, "Construction project scheduling using the FLASH method (Fuzzy Logic Application for Schudulling)", Jurnal Pengairan, 2 (1).

[6] Hasan, H., Jantje B.M., and Pingkan A.K.P. 2016. "Factors Causing Delays in Construction Projects and Alternative Completions (Case Study in Manado Town Square II)", Jurnal Sipil Statik, 4 (11).

[7] Huda, A.M., 2014, "Planning Scheduling for PT Gersik Jaatama's Office Building with the Fuzzy Logic Application Scheduling (FLASH) Method", MATRIK Journal, 16 (2), 23-36.

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

[9] Pressman, R.S, 2015, Software Engineering Available Online, Mc Graw Hill, London.

[10] Rinaldi Munir, Leony Lidya, 2016, Agorithm and Programming in Pascal, C, and C ++, Sixth Edition (Latest), Informatics, Cimahi West Java.