**Generalization of “DKN Fuzzy Simpleks” Application Program.**

**S Rosyid1, Karyati2, Emut2, A M Abadi2**

Bachelor Student of Mathematics Study Programme, Yogyakarta State University, Indonesia1

Study Programme of Mathematics, Yogyakarta State University, Indonesia2

Corresponding author: karyati@uny.ac.id

**Abstract.**Generalization of “DKN Fuzzy Simpleks” application program was developed to solve trapezoidal fuzzy linear program (FLP) with an objective function to maximize. The method used in this application is fuzzy simplex. The research stage begins by examining the classical simplex method and fuzzy simplex. Both methods are adopted to arrange the algorithm of generalized fuzzy simplex that used to solve trapezoidal FLP with maximum unstandard case. Then the algorithm implemented in computer program with MATLAB based. The result obtain from this research is the algorithm of generalized fuzzy simplex can solve a maximum unstandard case of trapezoidal FLP. The algorithm can be used to computer program so that it could make new application, which is “Big M Fuzzy Simplex” as generalization of the application for DKN Fuzzy Simpleks.

**Keywords**: fuzzy linear programming, fuzzy simplex generalized, MATLAB, Big M Fuzzy Simplex

# Introduction

Applications or software make it easy to solve a problem, such as solving linear programming. “DKN (Dhoriva-Karyati:Niken) *Fuzzy* Simpleks” application is an application that is used to solve the problem of trapezoidal fuzzy number linear programming (FNLP) maximum pattern with the fuzzy simplex method. This method uses Maleki's ranking function. Maleki's ranking function gives good results for use in solving the fuzzy trapezoidal linear programing with the fuzzy simplex method [2].

The general model of linear programming problems is influenced by two factors, namely the limit of the quantity of supporting sources and the existence of an objective function that must be optimized with a pattern of maximizing or minimizing. The existence of these factors gives rise to various linear programming problem models. There is a linear programming model that has a maximizing objective function with a constraint function that involves more than equations or inequalities (≥) which is known as the maximum non-standard linear programming problem. Likewise, with the fuzzy linear program model, there is a linear fuzzy program model that requires a constraint function in the form of an equation or inequality (≥). The model is said to be a non-standard maximum pattern of fuzzy linear programming. Then the non-standard maximum pattern of fuzzy linear programming requires a method to find a solution.

Refers to the pattern of completion of non-standard maximum pattern trapezoidal FNLP with the fuzzy simplex method and simplex algorithm for a non-standard maximum pattern of linear programming [3] will be investigated regarding the completion of nonstandard pattern maximum trapezoidal PLF using the generalized fuzzy simplex method. Furthermore, based on this method, a MATLAB-based computer program was compiled as a generalization of the DKN Fuzzy Simplex application.

# The Generalized Fuzzy Simplex Method

*2.1 Algorithm generalized fuzzy simplex method*

The simplex method is one of the methods used to solve linear program problems. The solution to the problem of non-standard maximum linear programming using the classic simplex method was first introduced by Susanta [3]. Furthermore, the simplex method was developed to solve the fuzzy number linear programming. This method is known as fuzzy simplex. The fuzzy simplex method refers to the method researched by Karyati et al [1].

Referring to the two settlement methods, it can be given in general the stages of completing the trapezoidal FNLP for a non-standard maximum pattern using the fuzzy simplex method are known as follows:

1. Transform FNLP to the canonical form.
2. Arrange the resulting canonical form on a generalized fuzzy simplex table.
3. To test the optimality, it can be seen from the value on the $\tilde{Z}\_{j}-\tilde{C}\_{j}$ raw. The optimal solution is achieved when for all $j$, $\tilde{Z}\_{j}-\tilde{C}\_{j}≽\tilde{0}$. Reviewing the value on $\tilde{Z}\_{j}-\tilde{C}\_{j}$ calculated by the value of $\tilde{Z}\_{j}-\tilde{C}\_{j}$ ranking function. The ranking function used is Maleki's ranking function. If the optimal solution has not been achieved, the repair of the simplex table is carried out.
4. The repair simplex table is done by replacing the old base variable with the new base variable. Choose an entering variable by the most negative value on $R\left(\tilde{Z}\_{j}-\tilde{C}\_{j}\right)$ raw for new base variable. Choose a leaving variable by the most positive smallest $R\_{i}$ by $R\_{i}=\frac{B\_{i}}{a\_{ij}}$ with $a\_{ij}$ is positive number. using elementary row operations to change the element pivot must be equal to 1, and the other elements on the same row must be equal to 0.
5. Perform optimization test again. If the optimal solution has not been reached, repeat step 4 again.

*2.2 Numerical Simulation*

Given the trapezoidal FNLP problems following:

Example 1:

Maximize $\tilde{Z}=\left(3,5,2,2\right)x\_{1}+\left(2,4,2,2\right)x\_{2}$

Such that $x\_{1}+4x\_{2}\leq 4$

$x\_{1}+x\_{2}\geq 2$

$x\_{1},x\_{2}\geq 0$

Transform to the canonical form. The canonical form is structured by adding slack and surplus variables to the constraint function to become an equation. Base on non-standard maximum problem, the canonical form as follows:

Maximize $\tilde{Z}=\left(3,5,2,2\right)x\_{1}+\left(2,4,2,2\right)x\_{2}+\tilde{0}s\_{1}+\tilde{0}s\_{2}-\tilde{M}d\_{1}$

Such that $x\_{1}+4x\_{2}+s\_{1} =4$

$x\_{1}+x\_{2} +d\_{1}-s\_{2}=2$

$x\_{1},x\_{2}\geq 0$

Using the algorithm of generalized fuzzy simplex method, we get the initial generalized fuzzy simplex table on Table 1.

**Table 1.** The Initial Generalized Fuzzy Simplex for Example 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | $$\tilde{C}\_{j}$$ | (3,5,2,2)  | (2,4,2,2) | $$\tilde{0}$$ | -$\tilde{M}$  | $$\tilde{0}$$ | $$B\_{i}$$ | $$R\_{i}$$ |
| $$\tilde{C}\_{j}$$ | $$\overbar{x}\_{i} x\_{j}$$ | $$x\_{1}$$ | $$x\_{2}$$ | $$s\_{1}$$ | $$d\_{1}$$ | $$s\_{2}$$ |
| $$\tilde{0}$$ | $$s\_{1}$$ | 1 | 4 | 1 | 0 | 0 | 4 | 4 |
| -$\tilde{M}$  | $$d\_{1}$$ | 1 | 1 | 0 | 1 | -1 | 2 | 2 |
|  | $$\tilde{Z}\_{j}$$ | -$\tilde{M}$ | -$\tilde{M}$ | $$\tilde{0}$$ | -$\tilde{M}$ | $$\tilde{M}$$ | -$2\tilde{M}$ |  |
|  | $$\tilde{Z}\_{j}-\tilde{C}\_{j}$$ | (-m-5,-m-3,m+2,m+2) | (-m-4,-m-2,m+2,m+2) | $$\tilde{0}$$ | $$\tilde{0}$$ | $$\tilde{M}$$ |  |  |
|  | $$R(\tilde{Z}\_{j}-\tilde{C}\_{j})$$ | (-2m-8)/2 | (-2m-6)/2 | 0 | 0 | M |  |  |

According to the Table 1, variable $x\_{1}$ is selected as the entering variable. This variable is going to become a basis variable for the next iteration, as $d\_{1}$ is substituted as the ‘leaving variable’.

Then using elementary row operation to get the pivot element $a\_{21}$ become equal to 1 and another element on the same row be changed into zero. So, the next iteration tableau obtained as follow:

**Table 2.** Second Simplex Tableau Iteration

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | $$\tilde{C}\_{j}$$ | (3,5,2,2)  | (2,4,2,2) | $$\tilde{0}$$ | -$\tilde{M}$  | $$\tilde{0}$$ | $$B\_{i}$$ | $$R\_{i}$$ |
| $$\tilde{C}\_{j}$$ | $$\overbar{x}\_{i} x\_{j}$$ | $$x\_{1}$$ | $$x\_{2}$$ | $$s\_{1}$$ | $$d\_{1}$$ | $$s\_{2}$$ |
| $$\tilde{0}$$ | $$s\_{1}$$ | 0 | 3 | 1 | -1 | 1 | 2 | 2 |
| (3,5,2,2)  | $$x\_{1}$$ | 1 | 1 | 0 | 1 | -1 | 2 | - |
|  | $$\tilde{Z}\_{j}$$ | (3,5,2,2)  | (3,5,2,2)  | $$\tilde{0}$$ | (3,5,2,2)  | (-5, -3, 2,2) | (6,10,4,4) |  |
|  | $$\tilde{Z}\_{j}-\tilde{C}\_{j}$$ | (-2,2,4,4) | (-1,3,4,4) | $$\tilde{0}$$ | (3+M,5+M,2+M,2+M) | (-5, -3, 2,2) |  |  |
|  | $$R(\tilde{Z}\_{j}-\tilde{C}\_{j})$$ | 0 | 1 | 0 | 0 | -4 |  |  |

Based on Table 2, there is still negative value on $R(\tilde{Z}\_{j}-\tilde{C}\_{j})$ row, so that the simplex table is repaired again and the iteration is continued. Analogously selected $s\_{2} $as the entering variable and $s\_{1}$ as the leaving. The next iteration simplex table is shown in Table 3.

**Table 3.** The Optimum Simplex Iteration

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | $$\tilde{C}\_{j}$$ | (3,5,2,2)  | (2,4,2,2) | $$\tilde{0}$$ | -$\tilde{M}$  | $$\tilde{0}$$ | $$B\_{i}$$ | $$R\_{i}$$ |
| $$\tilde{C}\_{j}$$ | $$\overbar{x}\_{i} x\_{j}$$ | $$x\_{1}$$ | $$x\_{2}$$ | $$s\_{1}$$ | $$d\_{1}$$ | $$s\_{2}$$ |
| $$\tilde{0}$$ | $$s\_{2}$$ | 0 | 3 | 1 | -1 | 1 | 2 |  |
| (3,5,2,2)  | $$x\_{1}$$ | 1 | 4 | 1 | 0 | 0 | 4 |  |
|  | $$\tilde{Z}\_{j}$$ | (3,5,2,2)  | (12,20,8,8)  | (3,5,2,2)  | $$\tilde{0}$$ | $$\tilde{0}$$ | (12,20,8,8)  |  |
|  | $$\tilde{Z}\_{j}-\tilde{C}\_{j}$$ | (-2,2,4,4) | (8,18,10,10) | (3,5,2,2)  | -$\tilde{M}$  | $$\tilde{0}$$ |  |  |
|  | $$R(\tilde{Z}\_{j}-\tilde{C}\_{j})$$ | 0 | 13 | 4 | $$\tilde{M}$$ | 0 |  |  |

Based on Table 3, the optimal solution has been found. It can be proved by all values $R\left(\tilde{Z}\_{j}-\tilde{C}\_{j}\right)\geq 0$. Hence the optimal solution is obtained, which is $\tilde{Z}=\left(12,20,8,8\right)$ for $x\_{1}=4$ and $x\_{2}=0$.

# Generalization Program of “DKN Fuzzy Simpleks” Application

* 1. *Generating Program*

Based on the generalized simplex fuzzy algorithm, the Big M Fuzzy Simplex application is compiled which is a program generalization of “DKN Fuzzy Simpleks” application. In general, a program consists of input, process, and output. The output application as in the following figure:



**Figure 1.**  Home Panel



**Figure 2.** Iteration Panel

In this application, the input data is transformed in the form of a matrix. Then the input data is calculated by pressing "HITUNG" button, while the core of the process that is executed on the button is presented in the following script:

function hitung\_Callback(hObject, eventdata, handles)

C1=str2num(get(handles.c1,'String'));

A11=str2num(get(handles.a1,'String'));

A22=str2num(get(handles.a2,'String'));

A77=str2num(get(handles.a7,'String'));

skrip1

skrip2

itrmaks=1;

batasitr=1;

while all (R>=0)==0

 if batasitr>=50

 warndlg ('Proses simpleks sudah mencapai iterasi ke-50 dan','Hasil belum belum optimal');

 break

 end

skrip3;

skrip2;

itrmaks=itrmaks+1;

end

%tampilan dipanel iterasi

displaytabel;

set (handles.iterasimaks,'String',num2str(itrmaks));

set (findall(handles.hasil, '-property', 'enable'), 'visible', 'on')

set (findall(handles.hasil, '-property', 'enable'), 'enable', 'on')

Based on the script above, if the "HITUNG" button is pressed it will read and retrieve the data that has been inputted. Then the data is processed with "skrip1" which is a collection of commands to transform data from input data into ready-to-process data such as transforming into a canonical form and selecting the *M* value. Basically, the *M* value is expressed as a very large number, because in MATLAB calculations, the *M* value must be in numeric, so the program will select the *M* value. The script for selecting the *M* value is as follows:

%tehnik Big-M

for c=1:cn

 matriksm(c,:)=rz(C1(c,:));

end

[m1,m2]= max(matriksm);

M=C1(m2,:)+[1 1 1 1];%nilai M

Furthermore, the data is processed according to the order of the command in the script. The iteration in the program will be repeated until the expression condition in the "*while-end*" command is not fulfilled, which means that all values ​​in $R\left(\tilde{Z}\_{j}-\tilde{C}\_{j}\right)\geq 0$. The iteration loop in the program is limited to 50 iterations, the program will stop processing if up to the 50th iteration an optimal solution has not been found from the simplex table. All the results of the simplex table calculation process are presented on the “ITERASI” panel by pressing the “ITERASI” button.

* 1. *Application Simulation*

The application will be tested by running the application to work on the trapezoidal FNLP problem in example 1. Writing the objective function in the application is done as follows: $[3 5 2 2; 2 4 2 2]$. The constraint function is separated between the inequality $\leq $ and $\geq $ and the constraint function of the equation. Writing on the application for the constraint function above is $[1 4 4]$ and $[1 1 2]$. The simulation in the program will be shown in Figure 3.



**Figure 3.** Home Layout

The application will seek the optimum solution using the generalized fuzzy simplex method. The optimal solution result will be displayed immediately after the calculation process is complete. Display of calculation results by the application as in Figure 4.



**Figure 4.** Display of Calculation Result

The optimum result by application is $\tilde{Z}=(12,20,2,2)$ for $x\_{1}=4$ and $x\_{2}=0$. This value is in accordance with the optimum value in Table 3.

The simplex table iteration is displayed on the iteration panel by pressing the "ITERASI" button. The maximum iteration of the resulting number of iterations will be displayed directly. If you want to display the previous iteration, enter the desired iteration number in "Tampilkan Iterasi Ke: ". The example shows the first iteration as in Figure 5.



**Figure 5.** Display of First Iteration

The maximum iteration display generated by the application calculation is shown in Figure 4.



**Figure 7.** Display of Maximum Iteration

Based on maximum iteration, the optimum solution has been found. Optimization test has been fulfilled with all values on $\tilde{Z}\_{j}-\tilde{C}\_{j}\geq 0$. The maximum value of the objective function obtained by the application is $\tilde{Z}=(12,20,2,2)$, the value obtained is in accordance with the results of manual calculations.

# Conclusion

The generalized simplex fuzzy algorithm is compiled by adopting the algorithm in the classical simplex method and the fuzzy simplex method. This method is used to solve the non-standard maximum trapezoidal fuzzy linear programming. The generalized fuzzy simplex method can be implemented in MATLAB-based computer programming so that a new application is produced which is a program generalization of the DKN Fuzzy Simplex application.

# Reference

1. Karyati, Wutsqa D U and Insani N 2016 *Journal of Innovative Technology and Education* Vol.3 pp 123-130
2. Karyati, Wutsqa D U and Insani N 2018 *Journal of Physics:* Conf. Series 983
3. Susanta, B 1994 *Program Linear* (Yogyakarta: Departemen Pendidikan dan Kebudayaan Direktorat Jenderal Pendidikan Tinggi)