

Using Genetic Algorithm to Generate Pareto-Front in Multi-Objective Problem

Ir. M. Dachyar, PhD
M.Sc
Industrial Engineering Dept.,
University of Indonesia

Farizal, PhD
Industrial Engineering Dept.,
University of Indonesia

Celine Kurniajaya
S.T., Industrial Engineering
Dept.,
University of Indonesia

ABSTRACT

Business must minimized time and cost for their operational activity. By doing this, customer can increase their satisfaction. In order to fulfill every order, one of core activity was material order process. Material order process optimization was multi-objective problem because there is a need to minimize both cost and time used for this process. Genetic Algorithm was used to solve this problem. This paper aims to provide pareto-front solutions for material order process multi-objective problem.

General Terms

Multi-Objective Genetic Algorithm, Pareto-Front, Multiple Variables.

Keywords

Pareto-front, genetic algorithm, multi-objective problem, objective function, constrained optimization.

1. INTRODUCTION

With so many competition lurking around same market target, business needed to increase their competitive advantage. Some business decided to minimize time and cost for their operational activity. By doing this, customer can increase their satisfaction. In every business, there are several distinguished activity for one of core activity was material order process. Material order process optimization was multi-objective problem because there is a need to minimize both cost and time used for this process. Genetic Algorithm was used to find out how many resource (in this case, workers) that must be available to fulfill every order from customer. This paper aims to provide pareto-front solutions for material order process multi-objective problem.

2. GENETIC ALGORITHM

Genetic Algorithms are stochastic methods that can be used to solve a very broad class of optimization problems. They are known to solve problems in a heuristic way under consideration of the problem's environment. Therefore, it is useful to apply Genetic Algorithms to improve and manage allocation problem [1]. GAs is a robust general-purpose search program based on the mechanism of natural selection and natural genetics [2]. Genes and chromosomes are the fundamental elements in GAs. A chromosome is a string of genes. In a real problem, genes are the variables that are considered influential in controlling the process being optimized, and a chromosome is a solution to the problem. Genetic Algorithms search for the optimal solution from populations of chromosomes. The representation chosen for the genome is pivotal to the performance of GA [3].

In tandem with this, more sophisticated approaches to directing the evolution of a GA population are aimed at

improving performance on classes of problem known to be difficult for GAs. [6].

3. MULTI-OBJECTIVE GENETIC ALGORITHM

Multi-objective optimization deals with solving optimization problem which involve multiple objectives. Most real-world search and optimization problems involve multiple objectives (such as minimizing both cost and time) and should be ideally formulated and solved as a multi-objective optimization problem. However, the task of multi-objective optimization is different from that of single-objective optimization in that in multi-objective optimization, there is usually no single solution which is optimum with respect to all objectives [4].

3.1 Problem Definition and Modelling

There are two objectives in this problem: to minimize time used for one activity in material order process and also to minimize labor cost.

```
function y = mypaper1 (w)
y(1) = (140 * w(1) * w(2)) ;
y(2) = w(1) * w(2) ;
```

Fig 1: objective function(s) mathematical model

As can be seen at Figure 1, there are two objective(s), which is to minimize both cost function (y (1)) and time (y (2)). In this optimization process, decision variables indicated by worker's working hours that was denoted by notation w(1) and w(2) remarked number of workers.

As constraint(s), there is government-policy about labor work time. Labor's work time must be under 8 hours for every worker in one work-day (Monday until Friday).

$$w(1) \leq 8 \text{ hours} \dots\dots\dots (1)$$

$$w(1), w(2) > 0 \dots\dots\dots (2)$$

3.2 Matlab Software Tool

In this study, MATLAB was used to get a result by using optimization tool applications embedded in Matlab Software. This applications used user-interface to input information needed to get the result. MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment. A proprietary programming language developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python. [5].

Optimization Tool
File Help

Problem Setup and Results

Solver:

Problem

Fitness function:

Number of variables:

Constraints:

Linear inequalities: A: b:

Linear equalities: Aeq: beq:

Bounds: Lower: Upper:

Nonlinear constraint function:

Fig 2: user interface for Matlab optimization application

Options

Population

Population type:

Population size: Use default: 50 for five or fewer variables, otherwise
 Specify:

Creation function:

Initial population: Use default: []
 Specify:

Initial scores: Use default: []
 Specify:

Initial range: Use default: [-10;10]
 Specify:

Selection

Selection function:

Tournament size: Use default: 2
 Specify:

Reproduction

Crossover fraction: Use default: 0.8
 Specify:

Mutation

Mutation function:

Fig 3: Matlab optimization application and option

The image shows two overlapping windows of the Matlab optimization application's options menu. The top window displays settings for Crossover, Migration, Multiobjective problem settings, and Hybrid function. The bottom window displays settings for Migration, Multiobjective problem settings, Hybrid function, Stopping criteria, Plot functions, Output function, Display to command window, and User function evaluation.

Options

Crossover

Crossover function: Intermediate

Ratio: Use default: 1.0
 Specify: 1.0

Migration

Direction: Forward

Fraction: Use default: 0.2
 Specify: [Slider]

Interval: Use default: 20
 Specify: [Slider]

Multiobjective problem settings

Distance measure function: Use default: @distancecrowding
 Specify: [Slider]

Pareto front population fraction: Use default: 0.35
 Specify: [Slider]

Hybrid function

Hybrid function: None

Options

Fraction: Use default: 0.2
 Specify: [Slider]

Interval: Use default: 20
 Specify: [Slider]

Multiobjective problem settings

Distance measure function: Use default: @distancecrowding
 Specify: [Slider]

Pareto front population fraction: Use default: 0.35
 Specify: [Slider]

Hybrid function

Stopping criteria

Plot functions

Plot interval: 1

Distance Genealogy Score diversity
 Selection Stopping Pareto front
 Average Pareto distance Rank histogram Average Pareto spread
 Custom function: [Text Box]

Output function

Custom function: [Text Box]

Display to command window

Level of display: iterative

User function evaluation

Evaluate fitness functions: in serial

Fig 4: user interface for Matlab optimization application (options)

4. RESULT

After input relevant information at Matlab user-interface and choose the right solver applications, Matlab will generate a pareto-front (Fig 5). Pareto-front population fraction value was different for every different result. Based on these outputs (pareto-front), solutions list (Table 1) provided to management and the management will take decision based on their business target and resource.

For this study, five different population size value and three pareto-front population fraction value. For each value, a process will running about ten replication to get the minimal value of decision variables. After all running process, the list will be arranged to make management easier for choosing which solutions that they will take.

5. CONCLUSIONS

While Pareto-Front Solution was already been generated from Genetic Algorithm, the management still needed to take a decision between number of workers or their working hours. This paper can be used to help management take decision

regarding number of workers and their working hours needed to fulfill order from customer. There are several limitations regarding study for this paper. This paper didn't consider overtime. While using application from Matlab Software make the coding process faster, this can limit the number of constraints even though the constraints needed to make the result can really reflect the real conditions.

Table 1. Solutions List

Objective 1 (cost)	Objective 2 (time)	Number of workers	Number of hours
234.600	1675	20	81
270.300	1930	16	115
304.834	2177	19	109
415.204	2965	34	86
534.580	3818	32	117

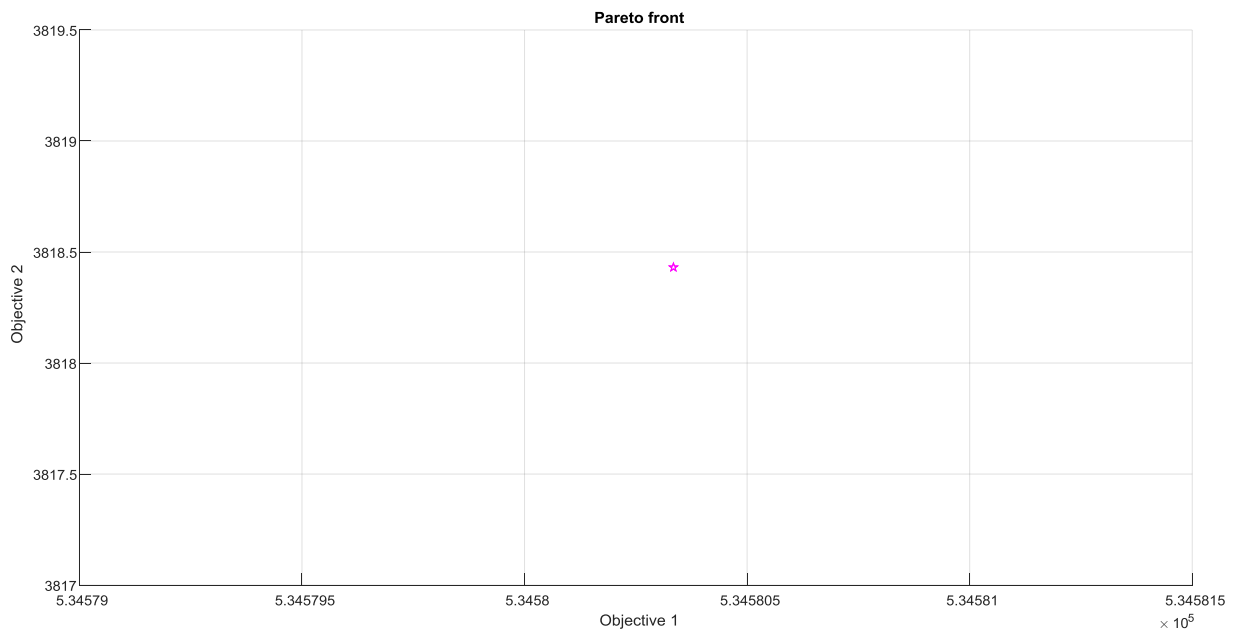


Fig 5: pareto-front result (1) from Matlab software (solution 5)

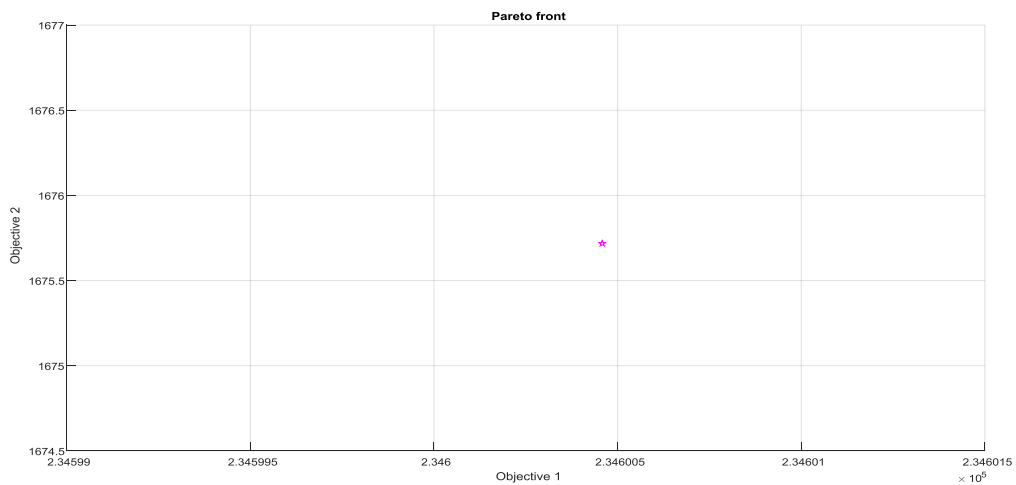


Fig 6: pareto-front result (2) from Matlab software (solution 1)

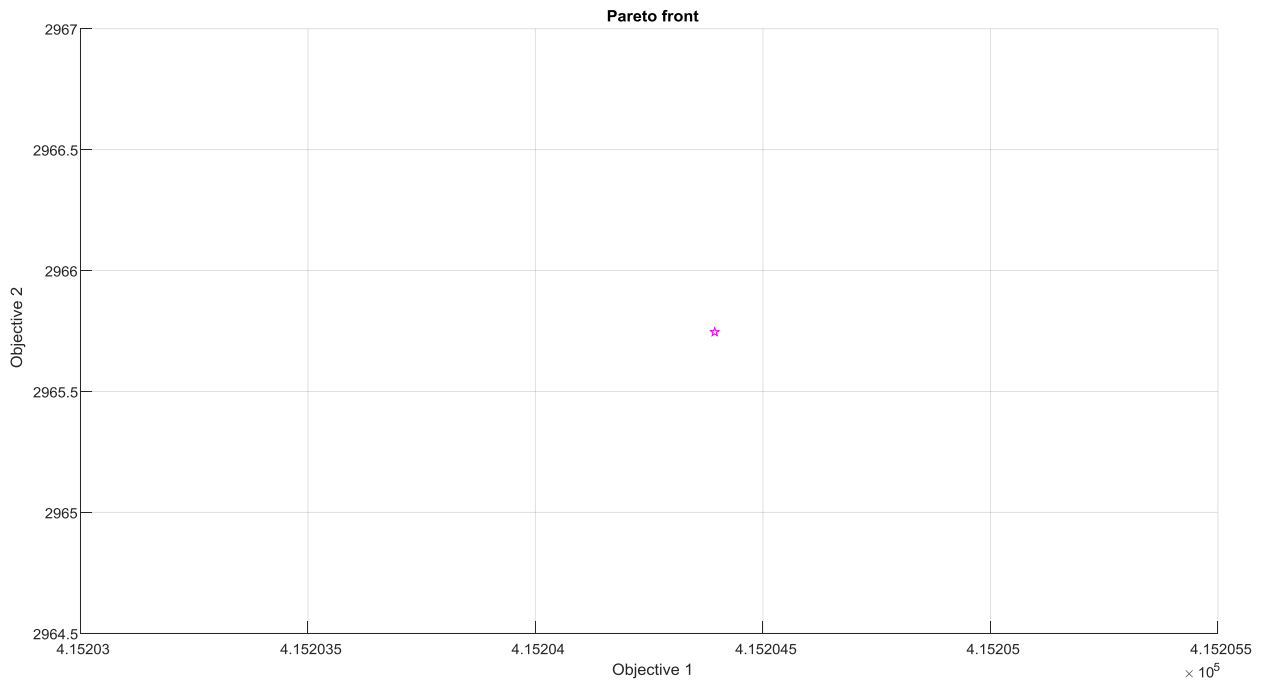


Fig 7: pareto-front result (3) from Matlab software (solution 4)

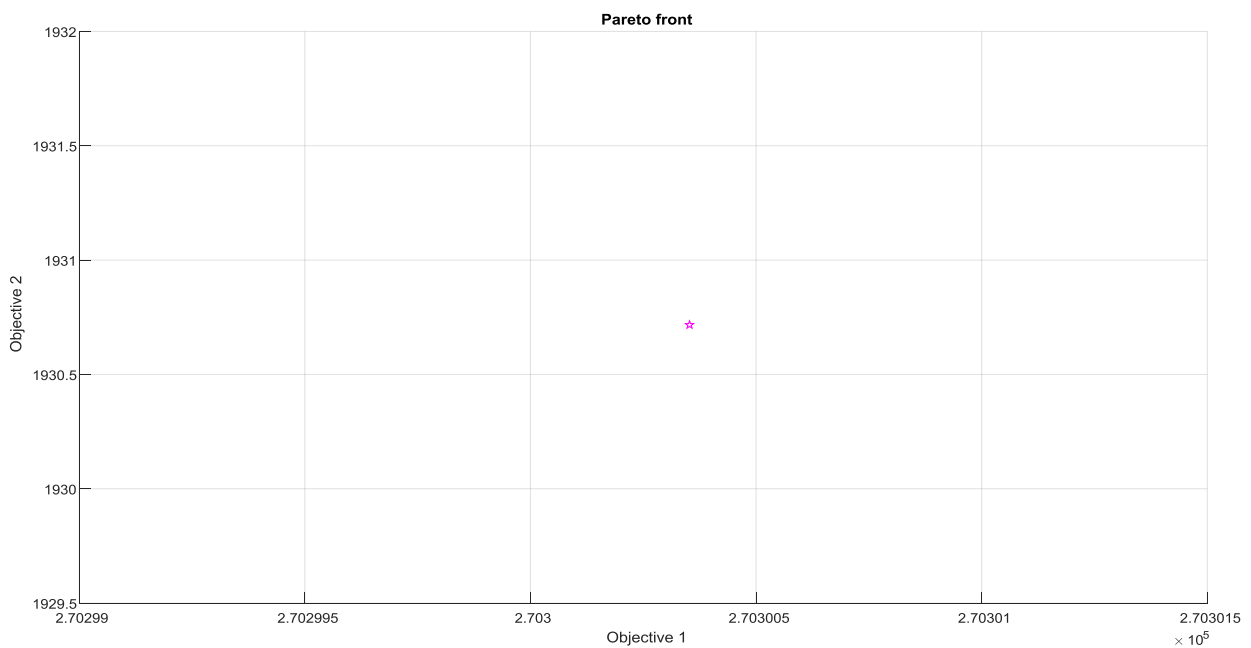


Fig 8: pareto-front result (4) from Matlab software (solution 2)

6. REFERENCES

- [1] Milena Karova, Julka Petkova, Vassil Smarkov. 2008. A Genetic Algorithm for Project Planning Problem. In Proceedings of the International Scientific Conference Computer Science.
- [2] Holland J. 1975 Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor.
- [3] D.A. Coley. 1999 An Introduction to Genetic Algorithms for Scientists and Engineers. New Jersey.
- [4] E. Zitzler, K. Deb, L. Thiele. 2000. Comparison of multi-objective evolutionary algorithms: empirical results. Evolutionary Computation 8 (2000), 125–148.
- [5] ww2.mathworks.cn (Accessed on May 20th, 2018 at 1 pm)
- [6] Engelbrecht A.P. 2002. Computational Intelligence: an Introduction. Wiley, New York.
- [7] J.H. Holland. 1975. Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor, MI
- [8] Mitchell. M. 1998. An Introduction to Genetic Algorithms. MIT Press, Cambridge, MA