SHM PUBLISHER
SURYA HIJAU MANFAAT

Journal of Student Research Exploration

https://shmpublisher.com/index.php/josre
p-ISSN 2964-1691| e-ISSN 2964-8246

Application of the greedy algorithm to maximize advantages of cutting steel bars in the factory construction

Rini Muzayanah[1], Endi Adika Tama[2]
[1, 2] Department of Computer Science, Universitas Negeri Semarang, Indonesia

Article Info

Article history:

Received Desember, 2022
Revised Desember, 2022
Accepted Desember, 2022

Keywords:

Greedy algorithm
Steel
Optimization
Dijkstra
Tender

ABSTRACT

Indonesia is one of the countries that is currently exuberant with development issues in order to balance the ongoing process of global modernization. In the infrastructure development process, the developer will enter into a contract with the contractor. This study aims to analyze the performance of the greedy algorithm in optimizing steel cutting with maximum profit to construction companies. The methods used include literature studies, program design, and program trials where the algorithm used is a greedy one. From the results obtained, it is evident that the Greedy algorithm can provide optimal steel cutting solutions because it works by calculating and deviating from all available separation settings, so there is no need to recalculate if the program performs that step.

1. Introduction

Indonesia is one of the countries currently exuberant on development issues to balance the ongoing process of global modernization. Infrastructure development is carried out by government and private parties. In the process of infrastructure development, the developer will enter a contract with the contractor. Commercial businesses in the construction sector are currently competing to provide the best service both during the process and after sales to attract customers/builders [1]. The competition between constructor companies to obtain a project is called a

[1] ***Corresponding Author:***
Rini Muzayanah,
Department of Computer Science,
Universitas Negeri Semarang,
Sekaran, Gunung Pati, Semarang, Central Java, 50229.
Email: rinimuzayanah0415@gmail.com
DOI: https://doi.org/10.52465/josre.v1i1.112

tender. One of the factors that influence the decision of the builder in choosing a construction company is the budget proposed by the construction company to carry out the project. The construction company that wins the tender optimizes the use of the budget received to achieve maximum profit. Optimization is done by minimizing the use of steel rods.

Steel bar is one of the basic construction materials which has an important role in establishing a building [2]. Contractors generally obtain reinforcing steel from steel construction manufacturers. From the perspective of a steel construction factory, this can be used to determine a strategy for reducing the size of steel so that the sales results generate the greatest profit so that the company's profits increase. Sales of steel bars to construction factories by applying the greedy algorithm and the algorithm of divinity and conquest is the object of the problem to be studied. This study aims to design and solve the problem of steel sales profits in construction plants.

There are many algorithms that can be applied to solve the problem above, all of which have their own characteristics, advantages and disadvantages. One of them is the Greedy algorithm which is commonly used to solve optimization problems. The Greedy Algorithm is a fairly popular method for solving optimization problem [3]. It is known that the characteristic of the greedy algorithm is determined by the underlying geometry of the problem setting [4]. The greedy algorithm works by finding a solution for each subproblem and cannot return it [5]. The greedy algorithm will take the best choice that can be obtained at that time without regard to the possible consequences [6].

In the implementation of the Greedy algorithm, what is done is to choose only the model that has the smallest residual and produces the longest steel reinforcement from a set of models that can still be applied [2]. Many options must be explored at each step of determining a solution so that at each step the best decision is taken to determine an option [7]. Optimization problems in the context of greedy algorithms are composed of several elements [8]. These elements include the following:

1. A set of candidates, C.

This set contains the elements that make up the solution. An example is a set of coins, a set of jobs to be done, a set of steel prices, and so on. At each step, one candidate is taken from the set.

2. A set of solution, S.

Contains candidates who were selected as a solution to the problem. The solution set is a subset of the candidate set that will form a solution to the problem.

3. Selection Function

The selection function is the function which at each step selects the candidate that will reach the optimal solution. Candidates who have been selected in one step are never considered again in the next step [9].

4. Qualification Function

Qualification function is a function that checks whether a candidate that has been selected can provide a feasible solution, that is, the candidate together with the set of solutions that have been formed do not violate the existing constraints. Eligible candidates are included in the solution set, while those that are not eligible are discarded and never considered again.

5. Objective function

The objective function is a function that maximizes or minimizes the value of the solution depending on the type of problem (maximization or minimization). Greedy does not operate thoroughly on all existing alternative solutions and some greedy problems do not always really give optimum but certainly provide solutions that are close to the optimum value [10].

In general, Greedy's algorithm can be written as follows :

```
function greedy(input C: himpunan_kandidat)
→ himpunan_kandidat
{
Mengembalikan solusi dari persoalan optimasi
dengan algoritma greedy
Masukan : himpunan kandidat C
Keluaran : himpunan solusi yang bertipe
himpunan kandidat
}

Deklarasi
   x : kandidat
   S : himpunan_kandidat

Algoritma:
   S ← {}    { inisialisasi S dengan kosong
   }
   while (not SOLUSI(S)) and (C ≠ {} ) do
       x ← SELEKSI(C)  { pilih sebuah
   kandidat dari C}
       C ← C - {x}  {  elemen  himpunan
   kandidat berkurang satu }
       if LAYAK(S ∪ {x}) then
           S ← S ∪ {x}
       endif
   endwhile {SOLUSI(S) or C = {} }

   if SOLUSI(S) then
         return S
   else
       write(' tidak ada solusi' )
endif
```

Figure 1. Greedy algorithm pseudocode

From the description above, an idea emerged to conduct research on optimizing steel bar cutting using a greedy algorithm. This study aims to analyze the performance of the greedy algorithm in optimizing steel cutting so that it can provide maximum profits to constructor companies.

2. Method

Systematically the steps of the research we carried out are contained in the form of a flowchart as shown below:

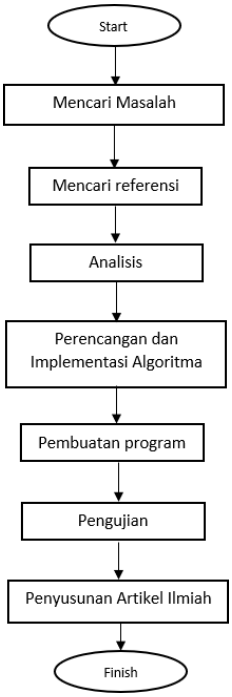

Figure 2. The flowchart of research method

A brief explanation of the above flowchart:

1. Looking for Trouble

The work begins by looking for problems around which an algorithm will be made. The problem will be solved by two suitable algorithms.

2. Looking for References

Determining the algorithm for the selected problem is done by collecting some materials as references such as books, journals, papers, or papers from various sources related to the topic of the problem.

3. Analysis

Analysis At this stage an analysis of the problems that have been determined is carried out.

4. Algorithm Design and Implementation

At this stage, an algorithm design is carried out which will later be implemented in an application program.

5. Programming

This stage starts implementing the algorithm into the selected programming language.

6. Testing

At this stage testing of the system that has been built, how it looks, functions run smoothly. As well as testing greedy and dynamic algorithms in maximizing profits in steel sales at construction factories.

7. Completion of Scientific Articles

This method is carried out by documenting the results of the analysis and implementation in writing.

Program Specifications:

The choice to use which algorithm, whether the greedy algorithm or the divide and conquer algorithm

Greedy algorithm flowchart:

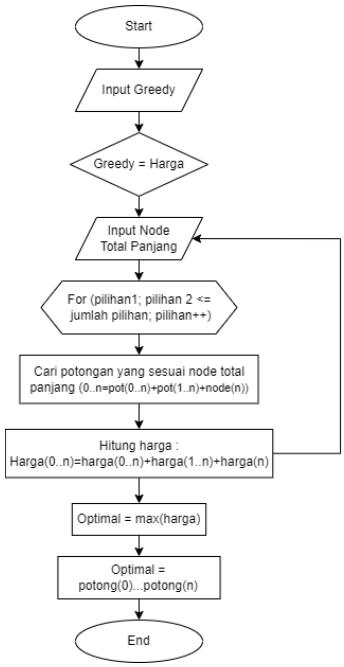

Figure 3. Greedy algorithm flowchart

In the flowchart above, the greedy algorithm process can be explained as follows:

Input: greedy

Process:

a) Greedy initialization
b) Price initialization
c) Input the total length node that you want to buy

d) Look for the right options to have maximum benefits
e) Look for pieces that match the desired length
f) The price will be according to the total length purchased
g) The price displayed is the price with the maximum profit
h) The profit is by the desired pieces

The program code of the Greedy Algorithm is as follows:

```
while(n){
    maksimal = 0;
    for(int i = 1;i <= n;i++){
        if( (n-i) >= 0 ){
            maksimal = price[i];
            indeks = i;
        }
    }
    n -= indeks;
    answer += maksimal;
}

return answer;
```

Figure 4. Display of greedy algorithm program source code

Divide and conquer algorithm flowchart:

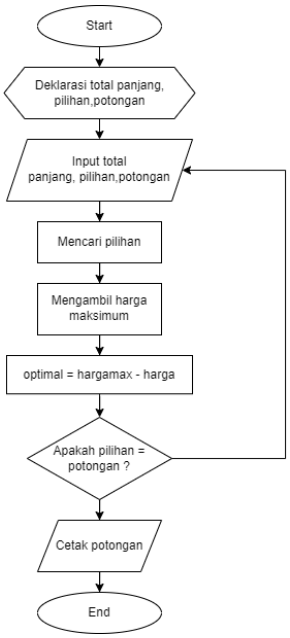

Figure 5. Flowchart of the divide and conquer algorithm

In the flowchart above, the process of Dijkstra's algorithm can be explained as follows:

Process:

a) Declare length, options, and chunks

b) Enter for length, options, and cut
c) Look for available options
d) Look for the maximum benefit from the existing options
e) Profits by reducing the original price to the maximum price
f) The choice depends on how many pieces of iron are desired
g) Display the option that has the maximum profit

3. Results

3.1.Interface Implementation

The program interface displays price options in scrap metal, done so you can find out the prices.

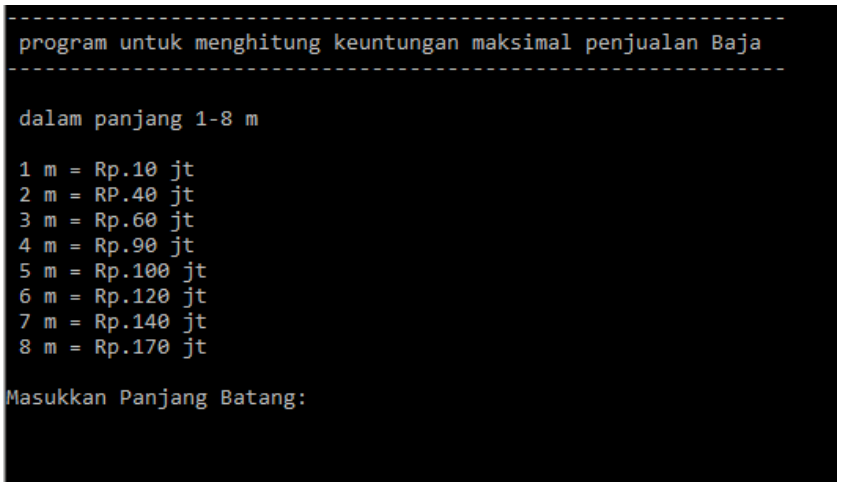

Figure 6. Interface display greedy algorithm to maximize advantages of cutting steel bars

In the calculation interface output, pieces in lengths from 1 to 8 meters are displayed with a predetermined price in millions.

3.2.Implementation in the Greedy Algorithm

Our simple definition is an example like:

in the algorithm .....

Long:	10	20	30	40	50	60	70	80m
Price:	10	40	60	90	100	120	140	170 (in million)

Tprice per meter of steel scrap, to use the greedy algorithm so that profits in sales can be maximized then,

For example, one buyer wants to buy 70 m

Option 1 = immediately given a price of 140 million

Option 2 = by providing 20 m and 50 m with a total price of 140 million

Option 3 = 30 m and 40 m with a total price of 150 million

To get the maximum profit, purchase 70 m iron by giving two pieces of 30 m and 40 m iron with a maximum profit of 150 million.

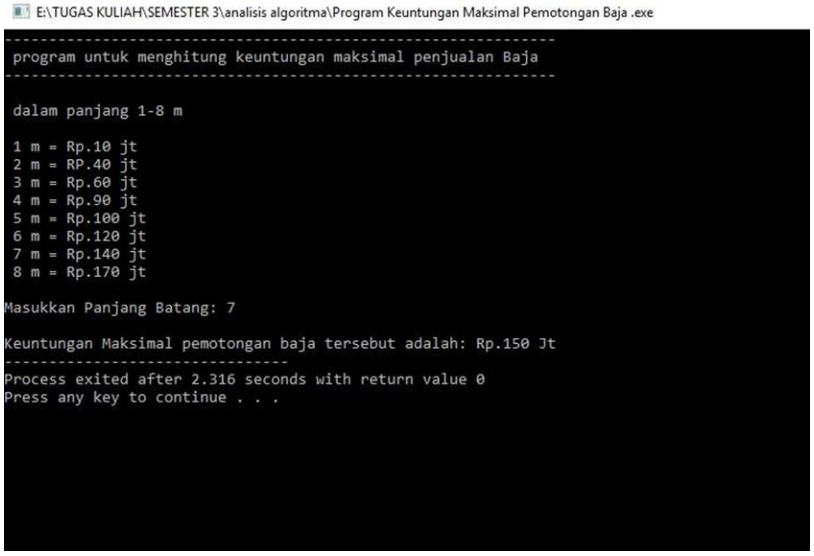

Figure 7. Output program using a greedy algorithm

3.2.1. Implementation in Dynamic Algorithms

For example:

Long:	10	20	30	40	50	60	70	80	(meter)
Price:	10	40	60	90	100	120	140	170	(in million)

Someone will cut steel 50 m long then the dynamic program will make several choices gradually starting from the first length factor then compared with all the prices listed then proceed with the first and second length factors then determine the price and so on.

After doing a dynamic program, we get a table:

Long 10+10+10+10+10 = Price 50

Long 10+10+10+20 = Price 70

Long 10+20+20 = Price 90

Long 10+10+30 = Price 80

Long 20+30 = Price 100

Then the dynamic program will take the most profit, namely the 5th long table in the form of taking a length of 20 and a length of 30 which will get the highest profit, namely 100 million.

4. Conclusion

Configuring an effective cut along with knowledge of market prices will provide the company with the greatest profit. Cutting configurations can be searched by various algorithms, one of which is greedy. The algorithms that our group uses optimize and maximize in solving a problem by solving each problem, thus the algorithms are suitable for our problem. But mostly in practice only a few factories use algorithms in maximizing their profits. Therefore, our group tries to use the algorithm, it turns out that the benefits provided are not much different from the profits that factories usually get through simple calculations in maximizing their profits.

REFERENCES

[1] V. Julianto, I. Saad, and S. Sukma, "Pengaruh Manajemen Waktu Penyelesaian Pekerjaan dan Mutu Pekerjaan Terhadap Kepuasan Pelanggan pada PT. Wijaya Karya (PERSERO), Tbk," *Jurnal Administrasi dan Management*, vol. 9, no. 1, pp. 21–28, 2019.

[2] A. Juniar, "Algoritma Optimasi Untuk Meminimalkan Sisa Pemotongan Bar Steel Pada Perusahaan Konstruksi," *Jurnal Teknologi dan Managemen*, vol. 11, no. 1, Feb. 2013.

[3] G. Hermawan, "Implementasi Algoritma Greedy Best First Search Pada Aplikasi Permainan Congklak Untuk Optimasi Pemilihan Lubang Dengan Pola Berfikir Dinamis," in *Seminar Nasional Teknologi Informasi dan Multimedia*, Jan. 2012, pp. 1–6.

[4] A. V. Dereventsov and V. N. Temlyakov, "Biorthogonal Greedy Algorithms in convex optimization," *Appl Comput Harmon Anal*, vol. 60, pp. 489–511, Sep. 2022, doi: 10.1016/j.acha.2022.05.001.

[5] C. Wang, P. Ge, L. Sun, and F. Wang, "Research on user-side flexible load scheduling method based on greedy algorithm," *Energy Reports*, vol. 8, pp. 192–201, Dec. 2022, doi: 10.1016/j.egyr.2022.10.352.

[6] A. Ambarwari and N. Yanto, *Penerapan Algoritma Greedy Pada Permasalahan Knapsack Untuk Optimasi Pengangkutan Peti Kemas*. 2016. doi: 10.13140/RG.2.1.2775.6569.

[7] H. Subakti and W. Gata, "Optimasi Minimum Pola Baju Khas Kain Tenun Sarung Samarinda Menggunakan Algoritma Greedy," *Inspiration: Jurnal Teknologi Informasi dan Komunikasi*, vol. 11, no. 1, p. 1, Jun. 2021, doi: 10.35585/inspir. v11i1.2602.

[8] E. N. Hayati and A. Yohanes, "Pencarian Rute Terpendek Menggunakan Algoritma Greedy," in *Prosiding Industrial Engineering National Conference (IENACO)*, 2014, pp. 391–397.

[9] H. Sunandar and Pristiwanto, "Optimalisasi Implementasi Algoritma Greedy dalam Fungsi Penukaran Mata Uang Rupiah," *Jurnal teknik Informatika UNIKA*, vol. 4, no. 2, pp. 193–201, 2019.

[10] H. Irwan, "Optimasi Penjadwalan Job Shop dengan Metode Algoritma Greedy," *PROFISIENSI: Jurnal Program Studi Teknik Industri*, vol. 8, no. 2, pp. 164–176, Dec. 2020, doi: 10.33373/profis. v8i2.2810.