Jurnal Mantik, 6 (1) (2022) 587-591

Published by:Institute of Computer Science (IOCS)

Jurnal Mantik

Journal homepage: www.iocscience.org/ejournal/index.php/mantik/index

ROOM OPTIMIZATION IN COURSE SCHEDULING USING THE GREEDY ALGORITHM

Jonas Franky Rudianto Panggabean[1], Kamson Sirait[2]

[12]AMIK Medicom, Jalan Darat No 74 Medan, 20153, Indonesia

E-mail: jonasfrankypanggabean@gmail.com

ARTICLE INFO

Article history:
Received: Mar 24, 2022
Revised: Aprl 26, 2022
Accepted: May 30, 2022

Keywords:
Greedy,
Optimization,
Schedule,
Room

ABSTRACT

Greedy algorithm is an algorithm that solves problems with systematic steps. In this research, the optimization used is the use of existing rooms on the AMIK Medicom Medan campus. The problem that often occurs on campus is the scheduling of courses with limited space. With these problems, it can hamper various other activities and the time needed is quite long to arrange a schedule because it is done manually. The purpose of this research is to test the greedy algorithm in optimizing empty spaces in the manufacture of the Course Scheduling System and to compare the algorithm testing with manuals. The application of the algorithm to the system will produce an effective course schedule without any problems in the use of the room so that there is no simultaneous use of the room in other courses.

1. Introduction

Scheduling courses is one of the problems that occur on the AMIK Medicom campus, because the making of the schedule system still uses the manual method that can be done for days and requires concentration and accuracy. There are many variables that must be considered in the preparation of the lecture schedule such as time, study groups, lecturers, classrooms and the number of courses. If the scheduler staff makes a mistake, there will be double use of the room, clashing class schedules and uncertain lecturer schedules (Y. Sari, 2019).

Therefore, it is necessary to create a system that can help create a course schedule system in room optimization, namely by implementing a greedy algorithm so that there will be no error in the course schedule.

The greedy algorithm is one of the most famous algorithms because it solves problems systematically, practically and easily in optimization problems. Based on previous studies in solving a problem. Application of Greedy Algorithm on Vending Machines (Alamsyah, 2014). The application of this algorithm is to form a step by step solution.

The problem of course schedule is very often done by the campus administration and it takes time in compiling the schedule. Based on these problems can be formulated in the formulation of the problem, namely how is the greedy algorithm implemented in terms of course schedules? And how to optimize the room using the greedy algorithm in the course schedule problem.

Problem limitation of the problem in this study are, the system manages lecture scheduling at the AMIK Medicom Medan campus. Scheduling of courses is carried out for semester lectures.
Comparison of the complexity of the application of the Greedy algorithm on several problems, including TSP (Traveling Salesman Problem), Minimum Spanning Tree, and Minimization of time on the system (Scheduling).

This Greedy algorithm forms a step solution, at each step of course the path will have many options and possibilities that can be exploration, with this algorithm the decision to take the next step is the most profitable in the current situation (Sabaruddin, 2016).This shortest route search has been applied in various fields to optimize the performance of a system, either to minimize costs or speed up the process (Insani, 2016).

Approach that the greedy algorithm do is making the choices that can give the best research ie by making the optimum choice at every step (V, 2005). Production scheduling is divided into several groups (J.E., 1987). Single-stage scheduling is scheduling each job (job) only through one work station to produce a product. If the

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

work is served by more than one work station, it is called single-stage in parallel machines, for example: service queues at bank tellers. According to (Angga, 2012), if the absolute best answer is not needed, then the greedy algorithm is often useful for producing a fairly good solution (approximation), rather than using a more complicated algorithm to produce the best solution.

Previous studies according to ((T. Sunarni, 2018)). Have tried to apply optimization methods to various cases such as university and school scheduling with various algorithms (K, 2007). From previous studies, optimal results can also be obtained, for example in research studies according to (T. Sunarni, 2018). However, there are also studies that are not able to produce optimal solutions because of the large amount of data according to (Lemos, 2019), so it requires a long computational time.

2. Methods

This study uses the prototyping method because the use of this method in developing a system is closer to the user according to Kendall, 2003, the method has stages as shown in Figure 1.

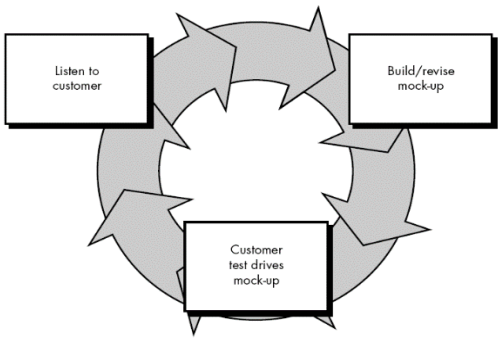

Figure. 1. Prototyping Method

2.1 Greedy Algorithm

Algorithms also have process loops (iteration), and also has a decision until the decision is complete (Maulana, 2017). One of the simplest Best First Search algorithms in calculating approximate costs, this algorithm always solves problems in steps. Choices are taken in steps without having to think about the effect of the overall settlement. Solution of the problem optimization is the set of variable values that satisfy the constraint, so that the objective function reaches the optimal value (Furqan, 2019**)**

Greedy algorithm is one of the popular methods in solving optimization (Juniar, 2015). There are 2 types of optimization, namely maximization and minimization. This study uses a greedy algorithm in the application of the maximization approach. The optimization problem in the greedy algorithm is structured using the following elements:

a. Candidate set, C. i.e. solution forming.
b. Set of Solutions, S. i.e. the solution to the problem.
c. Selection Function. namely the selection predicate function selects the most likely candidate in achieving the optimal solution.
d. Feasibility function, namely the predicate function is feasible. This feasibility function is a function to check that a selected candidate can provide a solution, namely the set of solutions that have been formed that do not violate existing constraints.
e. Objective function, namely the objective function to maximize the value of the solution or minimize the value of the solution.

Greedy algorithms do not always provide optimal solutions because of the following 2 things according to:

a. Greedy algorithms do not operate completely on all available alternative solutions as in the exhaustive search method.
b. There are several alternative selection functions. if you want the algorithm to work for to produce a solution that is close to optimal, it is necessary to choose the right selection function.

2.2 Scheduling

Scheduling is an activity of allocating existing resources in carrying out tasks at a certain time according to ((Baker, 1974). There are 2 limitations on the preparation of the course schedule, namely hard constraints and soft constraints. Hard constraint is a limitation that is made in the course schedule and carried out. Hard

constraints are set in the course schedule system on the basis of a greedy algorithm, namely:

a. One (1) lecturer may not be at the same time.
b. One (1) room cannot be used for two (2) or more courses at the same time.
c. One (1) class cannot have two (2) or more course schedules at the same time.
d. Theoretical courses must be in the general classroom and practical courses must be in the computer laboratory classroom.
e. Class days are Monday to Saturday and class hours start from 08:00 to 21:00 WIB.

Soft constraints are constraints that cannot be met in the scheduling process, but must have maximum results. Lecturers can make special requests within the teaching schedule. This request must not be violated and must be fulfilled. This limitation is a measure of the quality of course scheduling, so that the optimal course schedule can be made.

a. Requirements and definitions
   1) This stage is done to find out what are the requirements of the system to be built. The stages are:
   2) Interview
   3) Analysis
   4) Collecting data
b. System and software design
   1) This stage is to make a system design.
   2) The way the system works is designed using a flowchart.
   3) Determine the set of candidates.
   4) Selecting a set of candidates.
   5) Create an objective function to determine the best solution.
c. Implementation and Unit Testing
   1) This stage is the implementation of the design.
   2) Implementing and developing the system
   3) Conducting trials with comparisons.

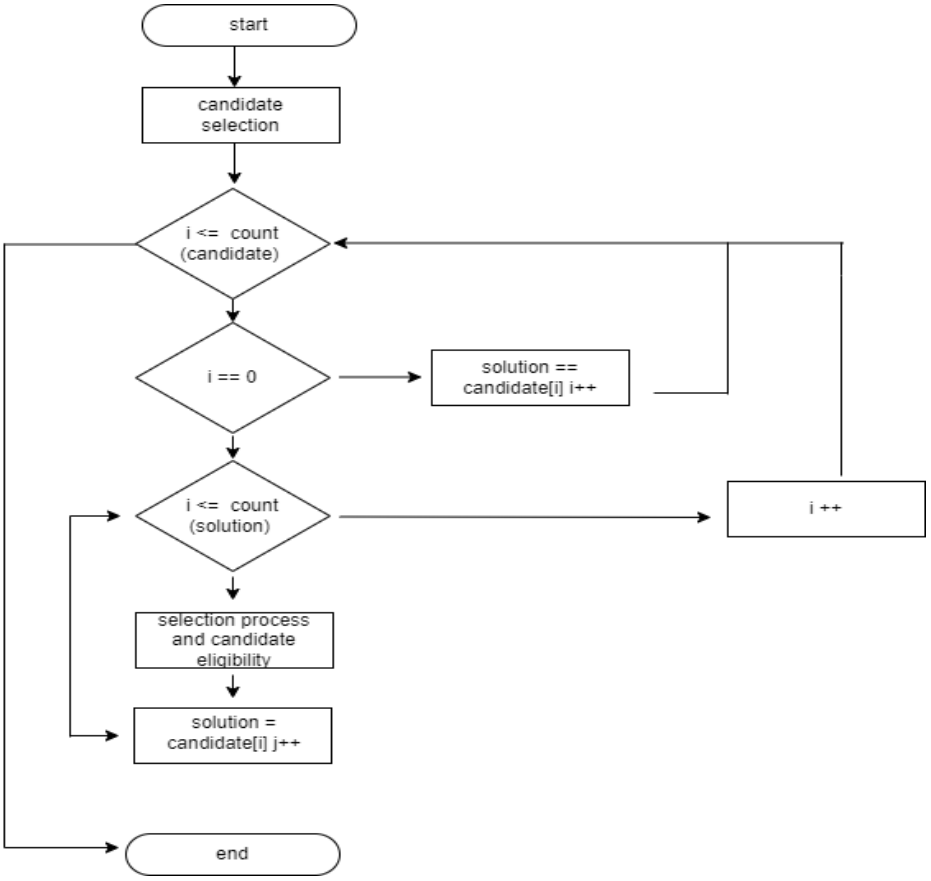

Figure 2. Scheduling Flowchart

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

d. System Testing

This stage includes testing the benefits of fact data which is carried out using the table test method by calculating the free time from the lecture schedule made by the system in a matter of days in one week.

TABLE 1
FACTS

No Data Quantity	No Data Quantity	No Data Quantity
1 Room 20	1 Room 20	1 Room 20
2 Courses 90	2 Courses 90	2 Courses 90
3 Classes / Study Group 30	3 Classes / Study Group 30	3 Classes / Study Group 30
4 Lecturers 50	4 Lecturers 50	4 Lecturers 50

1) The first test is to apply the old method based on the available space. This is a comparison to make the process easier.
2) The second test is to input data on available rooms, running courses, active learning classes/groups.
3) Testing the third step is to compare each room available every day with the manual method and the greedy algorithm method.
4) Testing the fourth step is to make conclusions in each classroom by counting the number of empty rooms with the formula
5) Testing the fifth step is to make conclusions from the entire room with a formula.

3. Results and Discussion

The results of data testing can be seen by testing the number of empty classrooms per day with the results of the greedy algorithm method. Because this is a guide to getting the second solution in space optimization, the more empty classrooms that day, the safer the schedule will be.

The results of the data tested can be seen in table 2, that the greedy algorithm is more effective and a solution in optimizing the use of classrooms compared to the old method used. Empty classrooms use the greedy algorithm more because of the stacking of rooms at the beginning, so that other rooms do not get a schedule. This can happen because the best candidate in the arrangement affects the room that will be used first.

TABLE 2
OVERALL TEST RESULT DATA

Academic Years	Semester	Day	Empty Room with manual method		Empty Room with greedy algorithm method	
2019/2020	Odd	Monday	13,8 %	3	25,8 %	8
		Tuesday	12,9 %	2	25,8 %	8
		Wednesday	13,8 %	3	28,2 %	10
		Thursday	15,1 %	4	30 %	12
		Friday	17,3 %	6	30 %	12
		Saturday	13,8 %	3	30%	12
2019/2020	Even	Monday	12,9 %	2	25,8 %	8
		Tuesday	15,1 %	4	50,3 %	16
		Wednesday	12,9 %	2	50,3 %	16
		Thursday	17,3 %	6	35 %	13
		Friday	25,8 %	8	35 %	13
		Saturday	50,3 %	16	40 %	14

Has a difference between the fact schedule with the manual method that is carried out. If you pay attention, there are more empty classrooms on the schedule with a greedy algorithm, so that in the process of determining the room, the course schedule becomes easier to make.

4. Conclusion

In this study it can be concluded that the greedy algorithm is very useful in optimizing the classroom. By applying this algorithm, the campus administration does not need to add rooms or change the course schedule. So that the empty classroom can be used for other learning activities. In the process, the greedy algorithm is very fast in processing room optimization so that the error rate of the class schedule can be minimized. But the greedy algorithm also has the disadvantage that it is not humane, where all component variables are calculated the same without any differences.

Accredited "Rank 4"(Sinta 4), DIKTI, No. 36/E/KPT/2019, December 13th 2019.

References

Alamsyah, I. T. (2014). *Penerapan Algoritma Greedy Pada Mesin Penjual Otomatis (Vending Machine).* blin: Fakultas Teknik. Unversitas Negeri .

Angga, C. &. (2012). Pengembangan Algoritma Greedy untuk Optimalisasi Penataan Peti Kemas Pada Kapal Pengangku. *Jurnal Sarjana Institut Teknologi Bandung bidang Teknik Elektro dan Informatika.*

Baker, K. (1974). Introduction To Sequencing and Scheduling . *New York: Jhon Willey and Sons*.

Furqan, M. (2019). Algoritma Hibrida Metode Heuristik Dan Eksak Untuk .

Insani, N. F. (2016). Perancangan Aplikasi Pencarian Hotel . *Kota Bandung Jurnal*.

J.E., B. D. (1987). *Integrated Production Control System.* New York: John Wiley & Sons.

Juniar, A. (2015). *Penerapan Algoritma Greedy pada Penjadwalan Produksi Single-Stage dengan Parallel Machine di Industri Konveksi.* Jakarta.

K, W. A. (2007). *Perbandingan Kompleksitas Penerapan Algoritma Greedy untuk Beberapa Masalah.* Institut Teknologi Bandung.

Lemos, F. S. (2019). Room usage optimization in timetabling: A case study at Universidade de Lisboa. *Oper. Res. Perspect*.

Maulana, G. G. (2017). Pembelajaran Dasar Algoritma Dan Pemrograman 75 Menggunakan El-Goritma Berbasis Web. *Jurnal Teknik Mesin*, 8.

Munir, R. (2009). *Diklat Kuliah: Strategi Algoritma.* Bandung: Penerbit ITB.

Sabaruddin, R. (2016). Solusi Optimum Minmax 0/1 Knapsack Menggunakan Algoritma Greedy. *Jurnal Evolusi*, 5–24.

T. Sunarni, R. B. (2018). Penerapan Teknik Pewarnaan Simpul Graf Pada Permasalahan Penjadwalan Kuliah. . *Pros. Ritektra*, 84-91.

V, S. (2005). KNAPSACK 0-1 PROBLEM. *ISBN : 3-540-40286-1*.

Y. Sari, M. A. (2019). Optimasi Penjadwalan Mata Kuliah Menggunakan Metode Algoritma Genetika dengan Teknik Tournament Selection. *J. Teknol. Inf. dan Ilmu Komput*, 85.