SMART TIMETABLING OPTIMIZATION USING BEES AND HARMONY SEARCH ALGORITHM

Muhamad Sauky[1], Galih Hermawan, S.Kom., M.T.[2]

[1,2] Universitas Komputer Indonesia
Jl. Dipatiukur No. 112-116, Bandung 40132
E-mail : ukkyms@gmail.com[1], galih.hermawan@ gmail.com[2]

Abstract

Timetabling can be defined as a process of placing resources or events into restricted timeslots in accordance with established constraints. This problem requires high computation time to search the solution, especially if the size of the problem increases with the increasing number of components and the specified constraints or conditions. For that, Bees and Harmony Search algorithm applied on this research, because the algorithm can be implemented into scheduling optimization.

The analyzed process are initialization of initial solution, local search and global search stage on Bees algorithm, Harmony Memory initialization, Harmony improvisation, and Harmony comparison stage on Harmony Search algorithm. Based on the research results, Bees and Harmony Search algorithms can be used to solve scheduling problems. Based on the test results, Bees algorithm parameters with n=20, m=5, e=4, nep=4, nsp=2, ngh=5 and Harmony Search parameters with HMCR=0.8, PAR=0.3 will optimize the search process. Based on the soft constraints tests results, the more number of not willing time slots are set the more average number of soft constraints violations occur. It can also be concluded that the safe limit on the number of not willing time slots is between 30 to 60 time slots and the number of not willing time slots have little effect on the average time spent on finding a solution.

Keywords : Timetabling, *Bees*, *Harmony Search*.

1. Introduction

Scheduling can be defined as a process for placing resources or events into limited timeslots in accordance with predefined constraints. Scheduling problems are a matter of optimization problems found in real situations. This problem requires high computation time to search for the solution, especially if the size of the problem increases with the increasing number of components and the specified constants or conditions [1]. During the optimization process, many aspects must be considered to obtain an optimal schedule, and often can not produce satisfying results because not all of their needs are met. Therefore, it is necessary to establish a limitation in the preparation of a schedule of hard constraints and soft constraints.

The Bees algorithm is an optimization algorithm that is inspired by the natural behavior of honeybees when finding it's food for the optimal solution [2]. The Bees algorithm is similar to the Artificial Bee Colony (ABC) algorithm in terms of having a local search process and global search. However there is a difference between the two algorithms during the local search process. The Artificial Bee Colony (ABC) algorithm uses a probabilistic approach, while the Bees algorithm does not use a probabilistic approach, but rather uses fitness value evaluations to drive its search [3]. The Harmony Search algorithm is an algorithm inspired by the process of improvising a music in which a group of musicians will mutually improvise their respective instrument pitches in search of perfect harmony according to aesthetic standards [4]. The Harmony Search algorithm handles intensification and diversification. Diversification will maintain a good local solution, explore a broader and more efficient global search space, and create new solutions that are pretty well approached to the best possible solution. While intensification will lead the search process to the search space a good solution.

According to Baris Y., Michael SP, Ernesto M., Duc TP, and Alfredo L. [3] The Bees algorithm has local and global search capabilities, can be implemented into several optimization problems, easy to use, can be combined with other algorithms, and has parameters Which can be set. According to Juliana W. [5] Harmony Search can be used to solve scheduling problems, but still needed improvement to improve the performance of the optimization, there are some issues to consider such as the memory consideration, and also the improvement process after improvisation. According to Hadwan M., Masri A., Nasser R. S., and Roug Q. [6] values on the parameters of high memory size harmony will contribute to the starting point of improved improvisation processes to obtain feasible solutions.

The combination of the Bees algorithm and the Harmony Search algorithm brings together the optimization capabilities of both. The Bees algorithm uses local search and global search, while the Harmony Search algorithm uses memory

consideration, random selection and pitch adjustment [7].

Based on the problems that have been described, the objectives to be achieved from this research is to determine the level of performance on the Bees and Harmony Search algorithms on the settlement of scheduling cases.

2. Research Content

2.1 Theoretical Basis

2.1.1 Timetabling

Timetabling is the allocation of resources to existing objects to available space in such a way that as far as possible meet certain conditions [8]. In the process of timetabling, existing resources must be allocated optimally, effectively and efficiently, and to the extent possible meet the existing needs. Fang [9] argues that timetabling is a challenging issue in the field of Artificial Intelligence, this is due to several things which include:

1. Timetabling is an NP-Complete (Non-deterministic Polynomial Complete) problem, which is a problem where with increasing variable linearly, the time required to solve the problem increases exponentially. This is often called Combinatorial Explosion, but the possible solution space for most real problems is too great for brute-force search or untargeted search.
2. High-level search techniques using multiple heuristics to trim the search space do not guarantee the creation of an optimal solution and it is very difficult to design an effective heuristic.
3. Timetabling problems are often complicated by special constraints.
4. Timetabling in the real world often involves constraints that can not be presented or even can not be properly expressed.

2.1.1.1 Timetabling Constraints

In timetabling there are known two types of constraints (requirements), namely hard constraints and soft constraints. Hard constraints are constraints that should not be violated at all whereas soft constraints are constraints to the extent possible. Violation of one hard constraint alone will cause the schedule to be useless whereas violations of soft constraints are still acceptable but will cause inconvenience to those who use the schedule [9].

2.1.2 Bees Algorithm

According to Pham [2] Bees algorithm is an algorithm included in the field of the Swarm Intelligence algorithm, the algorithm is still in touch with other bee-inspired algorithms such as Artificial Bee Colony algorithms, and other Swarm Intelligence algorithms such as Ant Colony Optimization and Particle Swarm Optimization. Bees's algorithm is inspired by the behavior of honeybees when feeding. Honeybees collect nectar from a large area around their nest. A colony sends a number of bees to collect nectar from the flower patch. Bees communicate with each other using waggle dance to inform the direction, distance, and quality assessment of food sources to other bees in the hive. The purpose of information processing of the Bees algorithm is to search and explore the best patch in the problem search space. Scout bee sent out to try trouble space at random and find best patch. Best patches are exploited using local search, where some of the best patches will be more explored than others. Best patch will continue to be exploited, although many scout bee sent out each iteration to find the best new patch.

The Bees algorithm is similar to the Artificial Bee Colony algorithm in terms of having a local search process and global search. However there is a difference between the two algorithms during the local search process. The Artificial Bee Colony algorithm uses a probabilistic approach during the local search stage, whereas the Bees algorithm does not use a probabilistic approach, but uses an evaluation of the fitness value to drive its search [3].

Table 1. Bees Parameter

Parameter	**Symbol**
Number of scouts on the selected patches	*n*
Number of best patches of selected patches	*m*
Number of elite patches on the best selected patches	*e*
Number of bees recruited to elite patches	*nep*
Number of bees recruited on the best non-elite patches	*nsp*
Search size for each patch	*ngh*

2.1.3 Harmony Search Algorithm

The Harmony Search algorithm is an algorithm inspired by the improvisation process of musicians, specifically when the musicians (who may have never played together before) quickly improvise to produce harmony that matches the esthetic estimate. Using the analogy, the Harmony Search algorithm performs the optimization process to get the best possible state by evaluating the fitness value. Just as the aesthetic estimate is determined using the set of pitches issued by the instrument, the fitness value of the Harmony Search algorithm is calculated using the set of values in each decision variable. Improved fitness values in the Harmony Search algorithm apply improvised improvisation from each iteration to iteration, just as improved aesthetic qualities are improved by doing exercise for exercise. If the new harmony created is better than the previous harmony, then this new harmony will be maintained.

Based on the concept, the outline of Harmony Search algorithm consists of five stages, namely:

1. Identify the problem and parameters of Harmony Search algorithm, with the required parameters are as follows.

Table 2. Harmony Search Parameter

Parameter	Symbol
Harmony Memory Size, is the number of vector solutions that are formed	HMS
Harmony Memory Consider Rate, is the probability of selecting components from an existing Harmony Memory, 0 ≤ HMCR ≤ 1. Generally ranging from 0.7 to 0.99	HMCR
Pitch Adjustment Rate, is the probability that a member of Harmony Memory for mutation is 0 ≤ PAR ≤ 1. Generally ranges from 0.1 to 0.5.	PAR

2. Harmony Memory Initialization, initial memory harmony is obtained from generating solutions randomly
3. Improvisation of New Harmony, every component of this solution is obtained under HMCR. HMCR is defined as the probability of selecting components from existing Harmony Memory. If x_1 ^ j is taken from Harmony Memory then randomly selected from the j-dimension of the Harmony Memory member, and can be mutated based on PAR. PAR is the probability of a member of Harmony Memory to be mutated. This improvisation process is similar to the offspring process on genetic algorithm with mutation and crossover process.
4. Updating Harmony Memory. Solutions from the third stage will be evaluated, if the fitness value of the new solution vector is better than the worst vector present in Harmony Memory, then a new solution vector will be used, otherwise if worse it will be eliminated.
5. Criterion stops, determines the process of generating new solutions and updating Harmony Memory. If the stop criterion is met then the iteration process will stop.

2.2 Problem Analysis

Problem analysis is an overview of issues raised in research on scheduling optimization using Bees and Harmony Search algorithms. The problem of college scheduling requires computation time is high enough to search the solution, this is due to the various components and limitations determined by the agency that will use the schedule. During the process, many aspects must be considered so as to produce an optimal and usable schedule. It is therefore necessary to set limits in the preparation of a hard constraints and should not be fulfilled but remain a reference in the soft constraints process.

Limits to be met (hard constraints) in the scheduling to be made are as follows.

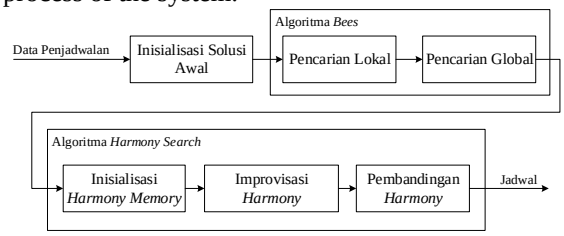
1. One space in a time slot can only take one course activity.
2. One lecturer at a time slot can only teach one course.
3. No lecture activities during Friday prayer time.

Limitations that do not have to be fulfilled (soft constraints) in the scheduling to be made are as follows.

1. The teaching time of the lecturer should be adjusted to the availability of the lecturer's time.

2.3 System Analysis

System analysis can be defined as the decomposition of a whole system into its component parts with a view to identifying and evaluating existing problems and constraints so that it can be proposed improvements in assisting the design process of the system.

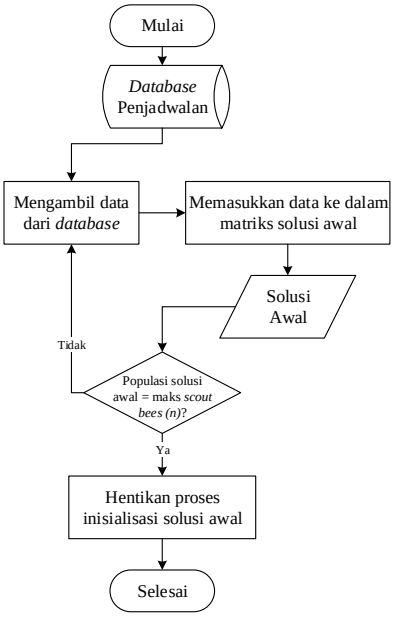


Picture 1. System Block Diagram

2.4 Process Analysis

2.4.1 Initial Solution Initialization

The initial solution initialization process is a process done to generate the initial solution which will be used as the basis for optimal solution search. The population number of initial generated solutions will be adjusted to the value of the population number parameter of scout bees (n) entered by the user. Each class data, clock data, day data, and space data are set into the initial solution matrix, where the class data is assigned sequentially while the clock data, day data, and space data are randomly assigned.



Picture 2. Initial Solution Initialization

2.4.1.1 Initial Solution Fitness Evaluation

As mentioned in the problem analysis, the scheduling under study has three hard constraints and one soft constraint, each constraint has a penalty value = + 1 to be evaluated using the following formula.

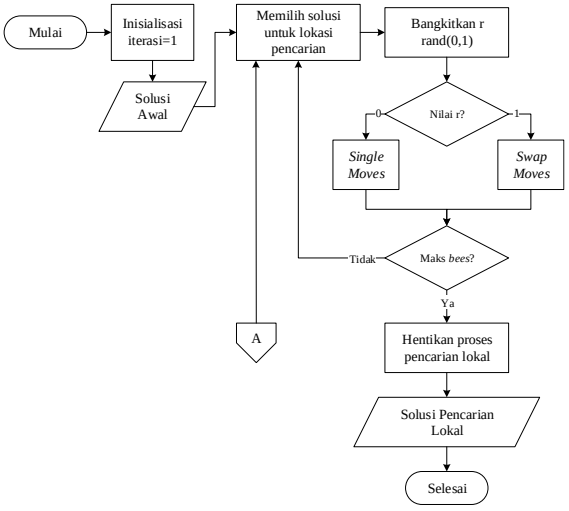
$$fitness = \frac{1}{(1+(penalty[1]+penalty[2]+penalty[3]+penalty[4]))} \quad (1)$$

With the limits analyzed are as follows.

1. Constraint 1 (penalty [1]), ie one space in a time slot can only take one course activity. This constraint can be subdivided into:
   a. When sks> = 1 and first hour = the first hour of another course on the same day and room.
   b. When sks> = 2 and second clock = the first hour of another course on the same day and room.
   c. When sks> = 3 and third hour = the first hour of another course on the same day and room.
   d. When sks> = 4 and the fourth hour = the first hour of another course on the same day and room.
2. Constraint 2 (penalty [2]), ie One lecturer at a time slot can only teach one course. This constraint can be subdivided into:
   a. When sks> = 1 and first hour = the first hour of another course on the same day and lecturer.
   b. When sks> = 2 and second clock = the first hour of another course on the same day and lecturer.
   c. When sks> = 3 and third hour = the first hour of another course on the same day and lecturer.
   d. When sks> = 4 and the fourth hour = the first hour of another course on the same day and lecturer.
3. Constraint 3 (penalty [3]), ie no lecture activity during Friday prayer time. This constraint can be subdivided into:
   a. When sks> = 1 and first hour = Friday prayer hours.
   b. When sks> = 2 and second clock = Friday prayer hour.
   c. When sks> = 3 and third hour = Friday prayer hour.
   d. When sks> = 4 and fourth hour = Friday prayer hour.
4. Constraint 4 (penalty [4]), ie teaching time of the lecturer should be adjusted to the lecturer's time availability.

2.4.2 Bees Local Search

In the Bees lgoritm phase, local searches are formed based on the number of bees on the best patch (m) by moving elements from the day, hour, or space columns to the selected solution as the search location, the moves can be either single moves or swap moves. This move is done at the selected location to find an alternative solution from a candidate for a course class that still has a constraint violation. The selection of single moves or swap moves is done by generating random numbers (r), if it is worth 0 then do single moves, while if it is worth 1 then do swap moves.



Picture 3. Local Search Process

2.4.2.1 Single Moves

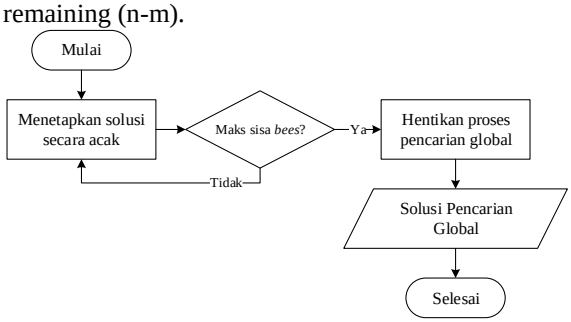
Movements made on single moves will change between day, hour or space of a candidate for a course class. Single moves are performed against the class of subjects who still have violations of constraints on existing solutions. First, select the single randomly-generated column, if the selected day's column, then change the value in the day column, if the clock column is selected, then change the value in the clock column, if the selected space column, Then make changes to the space column.

2.4.2.1 Swap Moves

Moves made on swap moves will change the day, hour, and space of a course class. Each swap move uses two components, the class of the course (ki) and the course class (kj). Firstly, for the course class (ki), select randomly the class of course kj that will be exchanged day, hour, and space with ki. Then, do a swap move between (ki, kj).

2.4.3 Bees Global Search

The global search is formed by randomly assigning solutions to explore new potential solutions based on the number of scouts of the remaining (n-m).

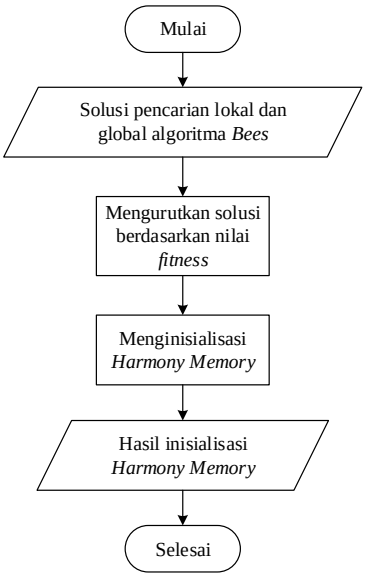


Picture 4. Global Search Process

The global search assigns a random solution from the search space, the process of determining this solution will continue until it reaches the maximum amount of scout bees left.

2.4.4 Harmony Memory Initialization

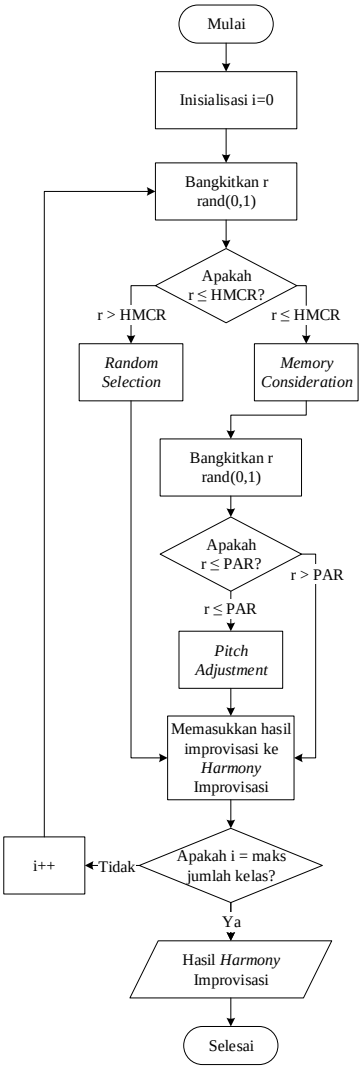
At this stage, the local and global search solutions of the Bees algorithm will be sorted by fitness values. Then Harmony Memory is initialized based on Harmony Memory Size (HMS).



Picture 5. Harmony Memory Initialization

2.4.5 Harmony Imptovization

In the Harmony Search algorithm, three operations steps will be applied to each improvisation of the new harmony, namely Memory Consideration, Random Selection, and Pitch Adjustment. The initial stage of harmony improvisation is generating random numbers r for comparison with HMCR parameters ie the probability of performing component selection from Harmony Memory. With HMCR range value is 0-1.
If r ≤ HMCR, then do Memory Consideration.
If r > HMCR, then pass Random Selection.
If r ≤ HMCR, then after Memory Consideration stage will raise random number r to be compared with PAR range value is 0-1.
If r ≤ PAR, then do Pitch Adjustment.
If r> PAR, then no Pitch Adjustment is performed.



Picture 6. Harmony Improvization

2.4.5.1 Memory Consideration

Operations in Memory Consideration will assign day, hour and space of each course class on improvised harmony based on solutions stored on Harmony Memory using the r value ratio (derived from the generation of new r with 0-1 random numbers) with HMCR.
If r ≤ HMCR, then do Memory Consideration.

If r > HMCR, then skip this step and do Random Selection.

2.4.5.2 Pitch Adjustment

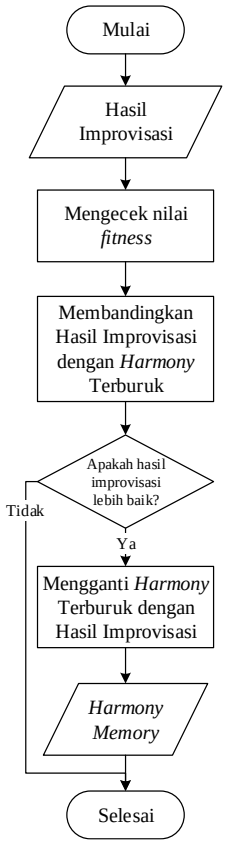
The Pitch Adjustment operation will be applied to the result of Memory Consideration operation by using the PAR value. The hours, days, and scores of the course classes set in improvised harmony will be taken from the best harmony in Harmony Memory by using the ratio of r value to PAR.
If r ≤ PAR, then do Pitch Adjustment.
If r > PAR, then no Pitch Adjustment is performed.

2.4.5.3 Random Selection

At this stage, the hours, days, and spaces of each course class on harmony improvisation will be randomly selected from the possible solutions.
If r ≥ HMCR, then do Random Selection.

2.4.6 Harmony Comparison

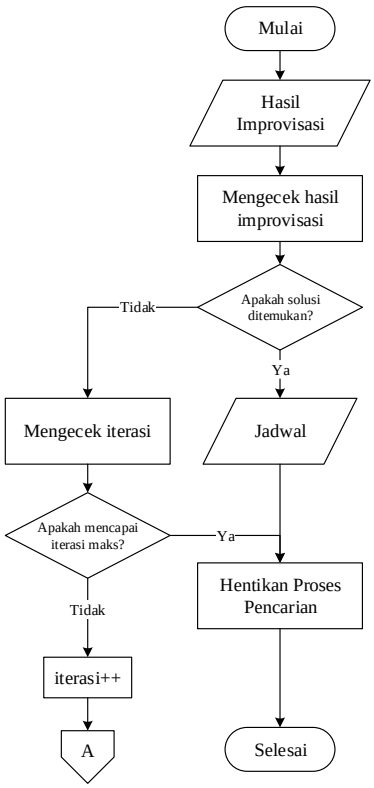
At this stage, improvised harmony will be compared with the lowest order harmony (the worst fitness value) in Harmony Memory. This comparison is based on fitness values. If the value of improvised fitness harmony is better than the worst harmony in Harmony Memory, then save harmony harmony to Harmony Memory and remove harmony harmony from Harmony Memory, and vice versa.



Picture 7. Harmony Comparison

2.4.6 ProcessTermination

The process stops if it meets the stop criteria, ie reaches the maximum number of iterations or finds the optimal solution, ie when the fitness value of the improvisation reaches 1 (one) which means that no constraints have been violated.



Picture 8. Termination Process Checking

2.5 Testing

Prior to the performance testing will be tested parameters with the aim of obtaining the most appropriate parameter values to be used in finding the solution of the scheduling studied. The most appropriate parameter values are obtained by observing the average number of iterations, times, and fitness values required to find a solution. Testing the value of each parameter is done to find the solution five times.

The parameter values of the Bees algorithm used in the test parameters are as follows.

Table 3. Tested Bees Parameter

Parameter	**Value**
n	[20, 30]
m	[4, 5]
e	[3, 4]
nep	[4, 5]
nsp	[2, 3]
ngh	[5, 10]

While the parameter values of the Harmony Search algorithm being tested are as follows.

Tabel 4. Tested Harmony Search Parameter

Parameter	Value
HMCR	[0.7, 0.8, 0.95]
PAR	[0.3, 0.4, 0,5]

For performance testing, it will be done by observing the number of violations against soft constraints ie availability of lecturers with normal priority, where constraints are soft constraints that can be violated and in case of violation the violation is still acceptable. This test uses the value of Bees and Harmony Search algorithm parameters based on the results of testing parameters that have been done before. The plan the number of time slots not willing to be used in testing is as follows.

Table 5. Tested not willing timeslot

The number of not willing time slots
30
60
90
120
150

2.5.1 Test Result

Based on the results of parameter testing, the optimal parameters obtained to resolve the scheduling problem under study are as follows.

Table 6. Optimal Parameter

Parameter	Value
n	20
m	5
e	4
nep	4
nsp	2
ngh	5
HMCR	0.8
PAR	0.3

Based on the results of performance testing of soft constraints that have been done, obtained the following results.

Table 7. Performance Testing Result

The number of not willing time slots	Average violation of soft constraints	Average time	Percentage of violations of soft constraints
30	0	3:11:14	0%
60	5	3:08:15	8%
90	21	3:14:35	23%
120	37	3:11:02	31%
150	88	3:11:22	59%

It can be concluded that the more number of time slots are not available, the more the average number of soft constraints violations occur.

Based on the test, it can be concluded also that the safe limit of the number of time slots is not willing is between 30 to 60 time slots, and the number of time slots is not willing not to have an effect on the average time spent on finding a solution.

3. Conclusion

Based on the results of research that has been done, it can be concluded that the Bees and Harmony Search algorithm can be used to solve the problem scheduling with optimal results (no hard constraints are violated). Based on the testing of parameters obtained that the value of Bees algorithm parameters with the value of n = 20, m = 5, e = 4, nep = 4, nsp = 2, ngh = 5 and Harmony Search with HMCR = 0.8, PAR = 0.3 will optimize the process Search for the final solution. Based on the results of testing of soft constraints that have been done, it can be concluded that the more the number of time slots are not willing to set the more the average number of violations of soft constraints that occur, the percentage of violations soft constraints also increased along with the increasing number of slots Time is not willing to set. The conclusion is also that the safe limit of the number of time slots is not willing is between 30 to 60 time slots, and the number of time slots is not willing not to have an effect on the average time spent on finding a solution.

3.1 Suggestion

Based on the results of research that has been done, still needed some further study. The suggestions for further research are as follows.

1. Improve the handling of violations of soft constraints so that the resulting schedule becomes more optimal.

References

[1] Suhartono E., 2015 “Optimasi Penjadwalan Mata Kuliah Dengan Algoritma Genetika (Studi Kasus di AMIK JTC Semarang).” Jurnal Ilmiah INFOKAM. No II, Th. XI Sept 15.

[2] Duc T. P., Ghanbarzadeh A., Koc E., Otri S., Rahim S., and Zaidi M., 2006 “The Bees Algorithm - A Novel Tool for Complex Optimisation Problems.” Proceedings of IPROMS 2006 Conference, pp. 454-461.

[3] Baris Y., Michael S. P., Ernesto M., Duc T. P., and Alfredo L. 2013 “Honey Bees Inspired Optimization Method: The Bees Algorithm.” Insects 4, no. 4: 646-662.

[4] Geem Z. W., Kim J. H., and Loganathan G. V., 2001 “A New Heuristic Optimization Algorithm: Harmony Search. Simulation.” SIMULATION: Transactions of The Society for

Modeling and Simulation International 76(2), pp. 60–68.
[5] Juliana W., and Naimah M. H., 2013 “Harmony Search Algorithm for Curriculum-Based Course Timetabling Problem.” International Journal of Soft Computing and Software Engineering, Vol. 3, No. 3, pp. 365-371.
[6] Hadwan M., Ayob M., Nasser R. S., and Roug Q., 2013. “A harmony search algorithm for nurse rostering problems.” Information Sciences Vol. 233, pp. 126–140.
[7] Khang N., Phuc N., and Nuong T., 2012 “A hybrid algorithm of Harmony Search and Bees Algorithm for a University Course Timetabling Problem.” International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1.
[8] Anthony W., 1996 “Scheduling, timetabling and rostering – A special relationship?” LNCS 1153, 46–75.
[9] Fang H. F., 1994 “Genetic Algorithms in Timetabling and Scheduling.” Ph.D. Thesis, Department of Articial Intelligence, University of Edinburgh.