Asynchronous Island Model Genetic Algorithm for University Course Timetabling

Alfian Akbar Gozali

Telkom University

Tel.: +62-22-7564108

Fax.: +62-22-7562721

Email: alfian@telkomuniversity.ac.id

Jimmy Tirtawangsa

Telkom University

Tel.: +62-22-7564108

Fax.: +62-22-7562721

Email: tirtawangsa@yahoo.com

Thomas Anung Basuki

Telkom University

Tel.: +62-22-7564108

Fax.: +62-22-7562721

Email: tirtawangsa@yahoo.**com**

Abstract University course timetabling problem (UCTP) is similar to general timetabling problems with some additional unique parts. UCTP involves assigning lecture events to timeslots and rooms subject to a variety of hard and soft constraints. Telkom University has almost similar problem with its course timetabling. The current solution with Informed Genetic Algorithm for Telkom University UCTP still has the time consuming problem.

Island Model informed Genetic Algorithm was used in this research to solve this problem. The idea of this research is making distributed model exchanges an island's local best Individu with another island. Island model GA could create university course timetabling in reasonable time. This distributed model could run faster rather than single machine model decreasing constraint violations to reach optimum fitness. It could have less constraint violations because it could escape from stagnant local optimum easier. Island model GA could even produce great accuracy for Telkom University dataset (99.74%) and acceptable accuracy at 96.80% for Purdue dataset for student level timetabling.

Keywords *UCTP, informed genetic algorithm, island model genetic algorithm*

1 Introduction

University course timetabling problem (UCTP) is general timetabling problems with some additional unique parts [2]. One of the most recently studied for UCTP is the application of genetic algorithms (GAs), which are based on the theory of evolution [4], and that have proved to be efficient for problems of moderate and realistic size [5, 6, 7]. It is well known that fitness evaluation is the most time consuming part of the genetic programming (GAs) system. This limits the types of problems that may be addressed by GAs, as large numbers of fitness cases make GAs runs impractical.

Telkom University had almost similar problem with its course timetabling. University timetabling for Telkom University has been previously studied by Suyanto [9, 10] by implementing informed GA. The result was great, it can reduce student meeting violation down to 741 from 58,660 student meetings [10]. But it took until 3 days in practice.

Generic approach for university timetabling was done by Hana Rudová, et.al. [11]. They built a generic timetabling engine named UniTime [12]. However, it is not generic enough as several constraints required for Telkom University timetabling are not covered, for example “some special lecturers should be scheduled in their time constraints”, “lecturer meeting spread”, and “lecturers time preferences”. Moreover the result of UniTime may not be useful for Telkom University. When the Telkom University timetabling constraints are simplified to meet UniTime requirements, its resulting schedule still has high number of conflicts.

2 Telkom University Timetabling Problem

Telkom University has 6,570 students in 4 departments and 9 study programs. In one semester, there are 316 lecturers with 1,034 lecture meetings and 58,660 student meetings to schedule. It has 80 rooms categorized in 4 different capacities: extra-large (XL), large (L), medium (M) and small (S). There are 24 time slots per week, and have high occupancy of more than 78.11%.

Previously, there was a research attempt conducted by Suyanto [10]. He claimed that the most challenge in this case is that the courses are conducted in around 4 parallel classes in average and up to 27 parallel classes in maximum. It makes reducing student conflicts is complex.

3 Genetic Algorithm Model

Refer to the [10]; GA variant used in this research is Informed Genetic Algorithm (IGA). Directed mutation is also used for this research. This research used only mutation process but not mutation and crossover like Karol [16] did. Because crossover just scrambles the genes and does not make significant fitness improvement [9].

3.1 Fitness Function

Penalty determines the value of an interest. Higher penalty value means more important constraint. Fitness value can be calculated by the formula:

$$f = \sum_{i=1}^{N} (p_i * V_i) \quad (1)$$

Where N is the total number of events, p_i is declared value limits for penalty for all i and V_i is the number of violations that occurred on the i-th constraint. This fitness function is inverted fitness. Therefore, smaller fitness value shows better solution and bigger fitness value shows worse solution.

3.2 Hard and Soft Constraints

Hard constraints (HC) are constraints that must be met. While soft constraints (SC) have no restrictions to be complied with, but should be met in order to improve quality of the class schedule. This research used same constraints with [10] that have 12 constraints (5 HCs and 7 SCs) in total.

4 Island Model Genetic Algorithm

Architecture of island model GA used in this research consists of two types of islands: master and slave islands. Island model architecture used in this research described as below:

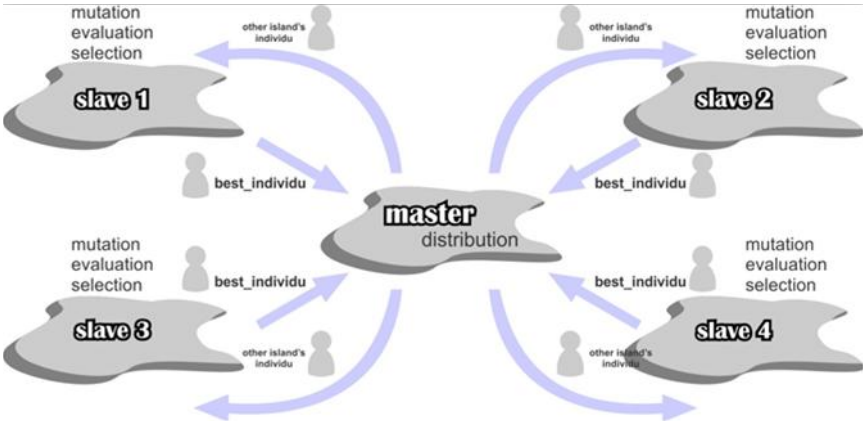

Figure 1. View of Asynchronous Island Model GA (Architecture Level)

Master Island works as a controller who distributes the given Individu to Slave Islands based on optimum fitness. Master Island can be attached to same computer with one of the slave islands because of its low resource consuming process. **Slave Islands** are the computational processor in the Island Model GA. It does mutation, fitness evaluation, and selection between iteration/generation processes. **Individual** or chromosome is a result representation of the GA process.

This Island Model will be run in asynchronous way. This means each island runs its own process independently. Process in one island is not directly depended on other island process. But at certain time, this island will take in another island's result to make better result. Figure 2 will explain the process of asynchronous island model used in this research.

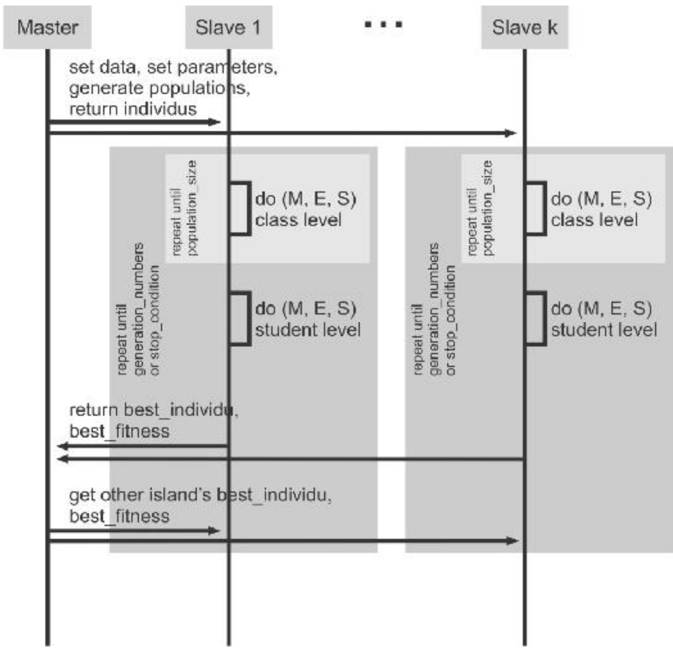

Figure 2. View of Asynchronous Island Model GA (Process Level)

5 Constraints Mapping

One of these research objectives is to compare result of the current world best-known solution [12] with proposed Island Model GA for university course timetabling problem. It must be defined and mapped the format and constraint mapping from UniTime to Telkom University data format and vice versa.

5.1 Constraint Mapping from Island Model GA into UniTime

The constraint mapping from Island Model GA into Unitime constraint is listed in Table 2. Can be seen that there are some Island Model GA constraints cannot be mapped into UniTime constraints directly.

Table 1. Constraint Mapping Island Model GA to Unitime

Island Model GA	**UniTime**
No lecturer conflict	No lecturer conflict
No class conflict	No class conflict
Lecture suitable capacity room	Lecture suitable capacity room, SAME_ROOM
Lecturers time departments cons	Lecturers time departments cons
Some special lecturers should be scheduled in their time constraints	Not supported directly
Lecturer meeting spread	Not supported directly
Class meeting spread	SPREAD
Lecturers time preferences	Not supported directly
Time constraints between meetings of the same lectures	NHB_GTE, NHB_LT, NHB
Time constraints between different lecture meetings in the same group	NHB_GTE, NHB_LT, NHB
Minimizing student conflicts	Minimizing student conflicts

5.2 Constraint Mapping from Unitime GA into Island Model

The constraint mapping from Island Model GA into Unitime constraint is listed in Table 3. Can be seen that there are some Island Model GA constraints cannot be mapped into UniTime constraints directly:

Table 2. Constraint Mapping from Unitime GA into Island Model

UniTime	**Island Model GA**
SAME_ROOM (same room)	Supported
SAME_TIME (same time)	Supported
SAME_START (same start time)	Supported
SAME_DAYS (same days)	Supported
BTB_TIME (back-to-back time)	Not supported
BTB (back-to-back)	Not supported
NHB_GTE(1)	Not supported
NHB_LT(6)	Not supported
NHB(x) (x hr(s) between)	Not supported
DIFF_TIME (different time)	Not supported
SPREAD (time spread)	Supported

6 Results

Population/sampling used in this research is Telkom University 2011-12 odd semester schedule and Purdue University 2007 fall lecture large room [12]. The result of system performance testing scenario for Telkom University course timetabling with Island Model GA are shown in Table 5.

Table 3. Fitness comparison in scenario 1

Island numbers		**Execution time**	**Optimum fitness**	**Number of Violations (lecturer/class)**
Suyanto's[10]	1	0h 41m 5s	90000	38/142
Proposed scheme	2	0h 44m 34s	85000	32/133
	3	0h 45m 43s	82500	29/133
	4	0h 46m 29s	83000	30/130
	5	0h 48m 1s	79000	24/130

Furthermore, figure 3 compares single and multiple island performance by its time consumption for reaching single island's optimum fitness. Compared to single island model, multiple islands can reach single island's optimum value in around half of the single island's time consumption because single machine model possibility to trap into a local optimum.

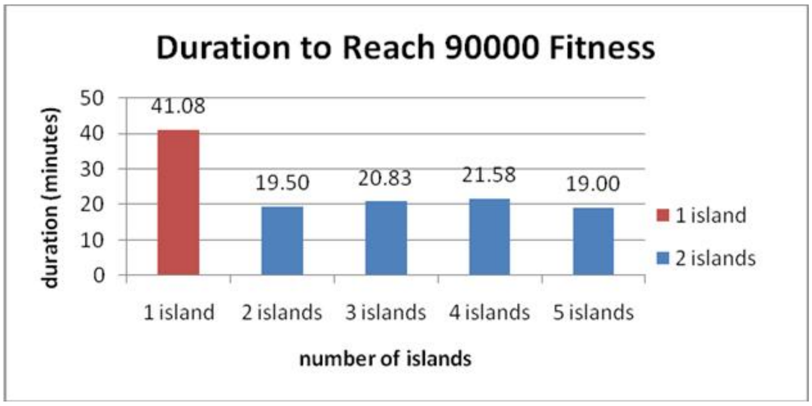

Figure 3. Duration Every Island to reach 90000 fitness

The result of system performance testing scenario for UniTime course timetabling with Island Model GA with one until five islands is shown in figure 4. There is a wide optimum (minimum) fitness gap between Purdue and Telkom University dataset. The reason behind this is because of both of them different characteristics.

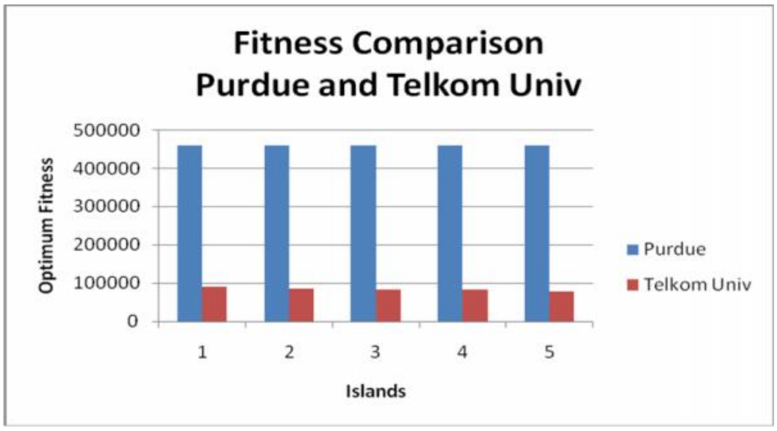

Figure 4. Fitness Comparison between Purdue and Telkom University Datasets

The comparison result of time consumption in same iteration (100 iterations) was shown in Table 9 below. UniTime and Island Model GA completes its running in just 16 minutes difference when applying Telkom University dataset. But when applying Purdue dataset, the difference can extremely widen the time gaps, more than 6 hours. Same with previous explanation, the reasons are in numbers of UniTime soft constraints and suitability of the engine.

Table 4. UniTime and Island Model GA Result Time Comparison

Engine	Dataset	Time	Accuracy
UniTime	Telkom University	0h 32m 1s	86.15%
Island GA	Telkom University	0h 48m 1s	99.74%
UniTime	Purdue	0h 33m 51s	80.43%
Island GA	Purdue	6h 38m 41s	96.80%

6 Conslusions

Island model GA could create university course timetabling in reasonable time. This distributed model could run faster rather than single machine model to decrease constraint violations to reach optimum fitness. It could have less constraint violations because it could escape from stagnant local optimum easier.

Island model GA could even solve another UCTP problem (Purdue University) but not quite well as Telkom University case. It produced great accuracy for Telkom University dataset (99.74%) and acceptable accuracy at 96.80% for Purdue dataset for student level timetabling.

Characteristics of datasets significantly influence the result of timetabling creating process. The main influencing characteristics are varieties and numbers of the soft constraints. And the most efficient number of island for Telkom University dataset is five islands.

References

[1] D. Johnson and M. Garey, Computers and intractability: A guide to the theory of NP-completeness, San Francisco: W.H. Freeman, 1979.

[2] R. Lewis, "A survey of metaheuristic-based techniques for university timetabling problems," *OR Spectrum*, vol. 30, no. 1, p. 167–190, 2007.

[3] S. Abdullah, E. K. Burke and B. Mccollum, "A Hybrid Evolutionary Approach to the University Course Timetabling Problem," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2007*, Singapore, 2007.

[4] D. E. Goldberg, Genetic Algorithm in Search, Optimization, and Machine Learning, Boston: Addison Wesley, 1989.

[5] J. Abela and D. Abramson, ""A Parallel Genetic Algorithm for Solving the School Timetabling Problem," CSIRO, Clayton South, 1991.

[6] E. K. Burke, J. P. Newall and R. F. Weare, "A Memetic Algorithm for University Exam Timetabling," *Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science,* vol. 1153, no. 1, pp. 241-250 , 1996.

[7] D. Corne, P. Ross and H. L. Fang, "Fast Practical Evolutionary Timetabling," in *AISB Workshop on Evolotionary Computation*, Leeds, 1994.

[8] R. Megasari, An Optimization of University Course Timetabling Using Case Based Reasoning and Graph Coloring (Master Thesis), Bandung: Institut Teknologi Telkom, 2011.

[9] Suyanto, Algoritma Optimasi: Deterministik atau Probabilitik, Yogyakarta: Graha Ilmu, 2010.

[10] Suyanto, "An Informed Genetic Algorithm for University Course and Student Timetabling Problems," *Artifical Intelligence and Soft Computing Lecture Notes in Computer Science,* vol. 6114, no. 1, pp. 229-236, 2010.

[11] H. Rudova, T. Muller and K. Murray, "Complex university course timetabling," *Journal of Scheduling,* vol. 14, no. 2, pp. 187-297, 2011.

[12] T. Muller, "unitime," UniTime LLC, 13 September 2012. [Online]. Available: www.unitime.org. [Accessed 26 December 2012].