INSTITUT TEKNOLOGI TELKOM

INSTITUT TEKNOLOGI TELKOM

SERTIFIKAT

Diberikan Kepada :

Ade Jamal

Sebagai

PEMAKALAH

dengan judul :

Evaluation Of Modified Scattered Hill-Climbing Method And Evolutionary Approach For Course Scheduling Problem

dalam Seminar Nasional Komputasi (Snakom) 2012 dengan tema
"Menjawab Tantangan Dunia dengan Ilmu Komputasi"
yang diselenggarakan pada tanggal 22 September 2012 di IT Telkom

Ketua Seminar Nasional Komputasi	Dekan Fakultas Sains Institut Teknologi Telkom
snakom seminar nasional komputasi	INSTITUT TEKNOLOGI TELKOM
Yuliant Sibaroni, M.T	Drs. Suwandi, M.Si

Telkom Indonesia — YAYASAN PENDIDIKAN TELKOM — GREEN CAMPUS

snakom
seminar nasional komputasi

Prosiding
Seminar Nasional Komputasi 2012

Menjawab Tantangan Dunia
dengan
Ilmu Komputasi

Bandung, 22 september 2012
Prodi Ilmu Komputasi
Institut Teknologi Telkom
snakom.ittelkom.ac.id

Penerbit
ISBN : 978-602-18892-0-6

INSTITUT TEKNOLOGI
TELKOM

Sambutan dekan Fakultas Sains IT Telkom

Ilmu Komputasi atau *Computational Science* bukanlah istilah yang asing di negara-negara maju. Saat ini bidang tersebut sudah menjadi bidang yang sangat bergengsi, termasuk tetangga dekat kita Singapura. Bagaimana di negara kita Indonesia? Sudah menjadi hal yang harus dimaklumi kalau istilah tersebut masih sangat asing. Rendahnya minat masyarakat terhadap bidang sains menjadi salah satu penyebabnya, selain itu juga rendahnya dukungan pemerintah terhadap sains merupakan penyebab yang lain.

Institut Teknologi Telkom (IT Telkom) adalah salah satu perguruan tinggi terkemuka Indonesia yang memiliki pandangan yang jauh ke depan. IT Telkom melihat bahwa harus ada sebuah bidang yang menjembatani rendahnya animo masyarakat terhadap Sains dan tingginya ketergantungan mereka terhadap teknologi. Dengan pertimbangan tersebut serta mengikuti trend di negara-negara maju serta hasil penelitian mereka, maka IT Telkom berketetapan hati untuk mendirikan Program Studi strata-1 Ilmu Komputasi. Ini adalah program studi strata-1 pertama di Indonesia, di perguruan tinggi lain merupakan program pasca sarjana ada juga berupa Kelompok Bidang Keakhlian (KBK). Saat ini bidang Ilmu Komputasi menjadi perhatian banyak bidang baik sains maupun teknologi, karena mereka membutuhkan Pemodelan dan Kemampuan Komputasi Tingkat Tinggi untuk memecahkan permasalahan sains dan teknologi. Kedua bidang tersebut menjadi perhatian utama Prodi Ilmu Komputasi IT Telkom.

Tentunya tidaklah mudah mengenalkan bidang Ilmu Komputasi ke masyarakat, prodi Ilmu Komputasi IT Telkom membutuhkan dukungan pihak lain termasuk Industri dan Lembaga Pemerintah. Sebagai langkah awal maka prodi Ilmu Komputasi IT Telkom mengajak masyarakat ilmiah untuk menghimpun diri bersama mengenalkan bidang tersebut melalui Seminar Nasional bidang Komputasi (SnaKom 2012). Tema yang diambil adalah "**Menjawab Tantangan Dunia dengan Ilmu Komputasi**". Para penggagas seminar ini bercita-cita yang sangat tinggi, mereka ingin suatu saat ajang ini tidak hanya menjadi *gawean* IT Telkom tapi mereka ingin menjadikannya milik bangsa Indonesia.

Semoga SnaKom 2012 menjadi tonggak berkembangnya Ilmu Komputasi di Indonesia, menjadi tempat diskusi, berbagi pengalaman antar akademisi, peneliti, industri, dan masyarakat umum yang memiliki minat tinggi terhadap sains.

Bandung, September 2012

Suwandi M.Si

Sekilas Tentang Snakom 2012

Dengan mengucapkan syukur Alhamdulillaahirobbil aa'lamiin, akhirnya seminar nasional komputasi 2012 (Snakom 2012) dapat diselenggarakan. Snakom 2012 ini adalah seminar nasional dalam bidang komputasi yang pertama diselenggarakan oleh program studi Ilmu Komputasi IT Telkom. Seminar ini diharapkan mampu menjadi wadah bagi akademisi, peneliti maupun praktisi baik yang berasal dari IT Telkom maupun institusi lain di seluruh Indonesia untuk berdiskusi dan menyampaikan hasil penelitian dalam bidang komputasi yang saat ini masih merupakan bidang yang tergolong baru di Indonesia. Dari acara ini, diharapkan akan memacu para peneliti untuk lebih banyak berkarya serta dapat memunculkan hasil-hasil penelitian baru dalam bidang komputasi yang dapat dimanfaatkan oleh bidang lain maupun dapat digunakan secara langsung dalam kehidupan nyata. Kami berharap, seminar ini tidak berhenti sampai disini tetapi dapat berlanjut dan menjadi semakin besar dalam tahun-tahun mendatang.

Secara umum, peserta Snakom 2012 ini datang dari berbagai kota dan provinsi di Indonesia seperti Riau, Samarinda, Yogyakarta dan lain-lain. Dalam pelaksanaan seminar tahun ini, terkumpul sejumlah 35 makalah hasil penelitian dimana mayoritas jumlah naskah masih berasal dari IT Telkom. Ibarat pepatah " tiada gading yang tak retak", kami menyadari banyak kekurangan yang dijumpai selama proses penyelenggaraan Snakom 2012 ini. Untuk itu kami selaku panitia mohon maaf sebesar-besarnya kepada peserta seminar atas ketidak sempurnaan tersebut. Kami juga memberikan apresiasi sebesar-besarnya kepada pimpinan IT Telkom atas dukungannya terhadap acara ini dan seluruh anggota panitia yang telah bekerja keras dalam mendukung kesuksesan pelaksanaan seminar selama ini.

Bandung, September 2012
Ketua Panitia

Yuliant Sibaroni S.Si, M.T

Susunan Kepanitiaan

Penyelenggara:
Prodi. Ilmu Komputasi IT Telkom

Pengarah :
Dekan Fak. Sains IT Telkom

Penanggung Jawab :
Wakil Dekan Fak. Sains IT Telkom
Ka.Prodi Ilmu Komputasi IT Telkom

Editorial Board :

Prof. Dr. Zaenal Hasibuan
Prof. Dr. Thee Houw Liong
Dr. Retantyo Wardoyo
Aly Muayyadi, PhD
Dr. Heroe Wijanto
Dr. Deni Saepudin
Dr. Maman Abdurohman
Dr. Ari M. Barmawi

Organizing Committee:

Ketua Pelaksana : Yuliant Sibaroni, S.Si., M.T.
Wakil Ketua : Dr.Adiwijaya

Kesekretariatan :
Fitriyani S.Si., MT
Rian Febrian Umbara M.Si
Dinda Belladini
Gentra Aditya Putra
Diwan Mukti

Acara Seminar Panel :
Jondri M.Si.
Putu Sudiksa

Acara Seminar Paralel :
Dr. Deni Saepudin
Dyas Puspandari, M.S.

Keuangan
Sri Utami

Sponsorship :
Mahmud Imrona, MT.

Daftar Isi

EVALUATION OF MODIFIED SCATTERED HILL-CLIMBING METHOD AND EVOLUTIONARY APPROACH FOR COURSE SCHEDULING PROBLEM

Ade Jamal

Informatics Engineering Department,
University Al-Azhar Indonesia, Jakarta
adja@uai.ac.id

Abstrak
Permasalah penjadualan kuliah termasuk kategori *NP-complete problem* yang sulit dan memerlukan waktu yang banyak untuk menyelesaikannya. Penjadualan kuliah adalah kegiatan menyusun sekumpulan aktifitas perkuliahan yang diajarkan oleh pengajar tertentu dalam sumberdaya waktu (jam kuliah) dan tempat (ruang kelas) tertentu dengan memperhatikan beberapa persyaratan. Masalah sejenis ini dapat diselesaikan dengan teknik pencarian dan metode optimalisasi. Jadual kuliah direpresentasikan dengan menggunakan model matriks permutasi. Dua buat metode pencarian dikaji yaitu metode pencarian *Hill-climbing* dan algoritma evolusi. Dua metode ini diuji dengan menggunakan dua konfigurasi jadual dengan jumlah ruang kelas yang berbeda untuk mendapatkan solusi yang "layak". Hasil kajian memperlihatkan bahwa metode *Hill-climbing* walaupun pada dasarnya pencarian lokal, dapat menemukan solusi global sebaik algoritma evolusi yang memang dibuat dengan strategi pencarian global.

Kata kunci : Penjadualan, Optimalisasi, Metode *Hill-climbing*, Algoritma Evolusi

Abstract
University course scheduling problem is considered as a NP-complete problem which is hard and time-consuming to solve. Constructing a course schedule involves assignment of a set of classes or events taught by certain lecturers into time and rooms resources, and subjected to some constraints. This problem can be solved by means of search and optimization techniques. Using a permutation based matrix model to represent the course schedule data, this problem has being studied by means of two methods, i.e. modified scattered hill-climbing search method and evolutionary algorithm. By using two test sets with different number of classrooms, these two methods are evaluated in searching for valid course schedule. The test result has shown that in spite of local nature of hill-climbing technique, this method can perform as good as evolutionary approach which is known as global optimization technique.

Keywords: Scheduling, Optimization, Hill-climbing method, Evolutionary Algorithm

1. Introduction

A course scheduling is a process that involves placement of many events in available time and space resources where some constraints must be satisfied. In this study, an event could be a teaching class or a laboratory class where a combination of lecturers or instructors, student-groups and a course was designated beforehand. Some scheduling problem involves also selection of which course should be taught by which lecturer. Another type of course scheduling, called post-enrolment course scheduling, takes into account the selections of individual student who enrols in a certain course.

There are two types of constraint that must be fulfilled in the course scheduling, namely hard- and soft constraints. A course schedule is said to be a valid one if it complies all the specified hard constraints completely. There are eight hard constraints considered in the on-going researches (Jamal, 2008, 2010, 2011), i.e.:

a. No room conflict, i.e. a room must not be used by more than one event at the same time
b. Available time-slot, i.e. a course must be given in the room's available time-slots
c. Contiguous course, i.e. a multi-hour course must be offered in same room contiguously
d. Room capacity, i.e. the room capacity must not be exceeded by number of attending student
e. Student conflict, i.e. a student's group from the same department and grade must not attend more than one event at the same time
f. Lecturer conflict, i.e. a lecturer must not teach more than one class at the same time
g. Lecturer inappropriate time, i.e. a lecture cannot teach any class in a time-slot which is inappropriate for him/her
h. Laboratory room; i.e. a laboratory class must be arranged in the suitable laboratory room

In the previous research (Jamal, 2010), there are soft constraints that are preferably fulfilled to obtain a so-called "nice" schedule. Some examples of these soft constraints are as follows:

a. A lecturer should be assigned in his/her preference time-slot

b. A classroom should not be half empty, i.e. the number of attending student less than a half of room capacity
c. No student should be scheduled to sit more than three events on the same day
d. Minimize scheduled events in the last time-slot of a day
e. Morning class-hours are preferred than afternoon class-hours
f. Minimize unoccupied room between two occupied time-slots

While the hard constraints are the common constraints in an optimization problem to set up a feasible or valid solution state, the soft-constraints can be the building elements for an optimization objective function. Constructing a valid course schedule is very complicated even for not an optimum schedule since the (hard) constraints are not small and frequently conflicting to each other's. In practice, having a valid schedule is often more demanding than having an optimum schedule where soft constraints are also considered. Hence, the presented article will not take into account the soft constraints. However, previous study has considered soft constraints introducing a three stage approach where the soft constraints become significant factors in the last stage (Jamal, 2010).

The main objective of the present article is to investigate reliable approaches to obtain a valid course schedule considering all hard constraints defined previously. We will begin with the construction of course schedule model, since the choice of the problem model will determine the computation complexity. The used optimization methods will be briefly presented before some testing results data will be discussed

2. Course Scheduling Solution Method

2.1 Course Scheduling Model

There are various types of model for representing course schedules. Some researchers (Al-Betar, 2008) and (Burke, 1994) used a two-dimensional matrix where a class-event conducted in the same room will be placed in the same row and while given in the same time will be found in the same column. Another approach model used by (Moody, 2008) is a tuple of <event, room, time-slot> as shown in Fig. 1. The matrix model has advantage related to the tuple model, namely the first hard constraint, i.e. room conflict is fulfilled by itself.

<class c> <room r> <day d, hour h>	<class c> <room r> <day d, hour h>	<class c> <room r> <day d, hour h>	<class c> <room r> <day d, hour h>

Fig. 1. A tuple model for representing a course schedule

We posed a three dimensional matrix for representing the course schedule as shown in Fig. 2 (Jamal, 2008, 2010, 2011). This is a deducted form from the two-dimensional matrix where one dimension time-slot column is exploded into two dimensions representing day and hour. In contrary to the work done by Burke et al (Burke, 1994) where the length of time-slots could vary, we fixed the size of these two dimensions of matrix for day and hour, hence this model always complies to the second hard constraint, namely a class-event is allocated in room's available time-slot.

Each matrix element will be filled by an identification number representing a particular class-event or will be blank (zero) if its resources (space and time) are not used. Using evolutionary terminology, one course schedule matrix represents a chromosome in which genes represented by matrix elements are the allocated class-events.

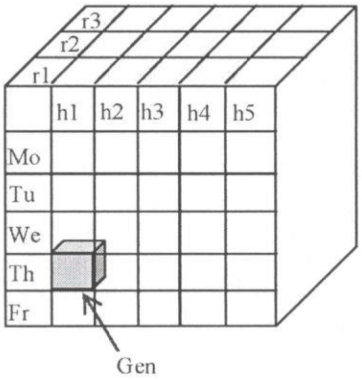

Fig. 2. A three dimensional course schedule matrix model where one gen shown representing a combination of particular course, lecturer and student group

A class-event can occupy one or more genes which depend on course duration (course hours). The multiple hours' class should be placed in the same room and on the same day contiguously as depicted in Fig. 3. This implies automatically the third hard-constraint, namely the contiguous courses.

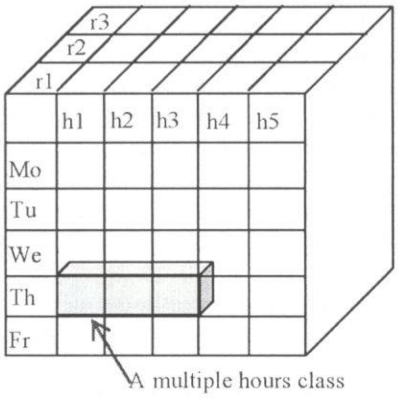

Fig. 3. A multiple hour class occupies multiple genes contiguously

2.2 Optimization Model

In computer science and management science, optimization is defined as the selection of the best solution with regards to some criteria from some set of available alternatives. An optimization problem can be considered as minimizing or maximizing an objective function f: A → R from some set of A to a real number R, where some equality or inequality constraints h(A) must be satisfied. The domain A of f is called search space, while A itself is a candidate solution state. Generally, there may be several local minima (or maxima) where all of the value of objective function on certain region around this local minima (or maxima), are greater (or less) than or equal to the value at that local minima (or maxima).

Optimization problems can be divided into two categories depending on whether the domain variables are continuous or discrete. The optimization problem with discrete variables is known as a combinatorial optimization problem. In combinatorial optimization problem, an object likes integer or permutation from a finite possibilities are sought.

The present course scheduling problem is a combinatorial optimization problem where the permutation based matrix represents a candidate solution state from a finite but very large set of alternatives. Each constraint is formulated such that if no violation occured within a candidate solution state then a maximum value, let say 100, will be assigned to a function. Its value will decrease by increasing number of violation to the lowest value (say zero) when all genes in the schedule matrix cause constraint violation. The sum of this function for all hard constraints makes up the objective function for a valid course schedule. In the present work, the first three of eight hard constraints defined previously, are already satisfied by the three-dimensional matrix model. Hence, the maximum value of objective function of the last five hard-constfratints for valid solution is 500. The sum of all constraint function including the soft constraints for an "optimum" course schedule will not be considered here.

To solve this type of optimization problem researchers often use heuristic methods that may provide approximate solutions in contrast to brute force exhaustive method that requires an extremely large number of objective function evaluations to find the global solution. Two methods of heuristic have been used in current research, namely, (modified) hill-climbing and evolutionary approach.

2.3 Hill-climbing search method

Hill-climbing search techniques is part of the local search algorithm family that search all possible solution in the neighbourhood of current state in the solution search space. The hill-climbing search algorithm is a simple algorithm that loops continually with increasing value of objective function- that is uphill move for maximizing problem. The algorithm is sometimes called greedy local search because it grabs a good successor from all possible neighbourhoods without thinking ahead about where to go next. The algorithm terminates when it reaches a "peak" where no neighbourhood has a higher objective function value. This leads to the famous disadvantageous of this algorithm, namely it often gets stuck for local maximum or plateau (a flat landscape). The simple hill-climbing algorithm is shown in Fig. 4.

```
function HILL-CLIMBING(problem)
inputs:            problem: a problem
local variables:   current, neighbor: a state
returns: a state that is local maximum
  current ← INITIAL-STATE(problem)
  loop do
    neighbour ← highest valued successor of current
    if VALUE(neighbor) ≤ VALUE(current) then
                      return current
    else current ← neighbor
```

Fig. 4. The simple Hill-Climbing Algorithm

Note that modifying the current state into all possible successors as in the statement:

"*neighbour*←a highest valued successor of *current*" involves looping over the whole possible neighbour which in our case will be polynomial to extend of the problem. To avoid the large number evaluation we modified the algorithm by choosing randomly a "small" amount the possible successors and then seek the highest valued successor from this small group (Jamal, 2008). Furthermore, we must let the iteration goes further for a while when the best found neighbour state has the same objective function value as the current state. These modification yields two advantageous, first the number of evaluation reduces significantly and second the possibility of getting stuck in the local maxima also reduces since this comprises a stochastic sense in the algorithm as shown in Fig. 5.

```
function MODIFIED-HILL-CLIMBING(problem)
inputs:            problem: a problem
local variables:   current, neighbor: a state
                   successors[], a state
returns: a state that is local maximum
  current ← INITIAL-STATE(problem)
  for iteration ←1 to ∞ do
    successors[]← SCATTERED-
                  SUCCESSORS(current)
    neighbor ← a highest valued of successors[]
    if VALUE(current) ≤ VALUE(neighbor) then
              current ← neighbor
    if VALUE(current) = OBJECTIVE-VALUE
    then return current
```

Fig. 5. The modified scattered Hill-Climbing Algorithm (Jamal, 2008)

A number of successor states are developed by modifying the current state looking for its neighbours. Here the use of term of neighbours is not strict that it means direct neighbour, as only small variation from current state such that single gen moves from the current position to a new position as depicted in Fig. 6.

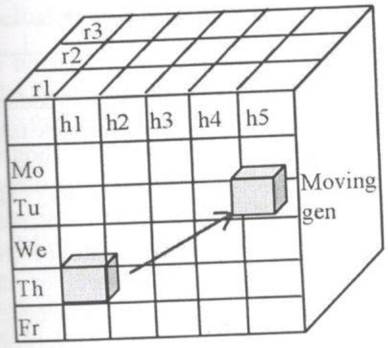

Fig. 6. Small variation of state modification

Noted that if the new position was already occupied by another gen then both genes will swap their positions. Another variation mechanism that also implemented is swapping two scheduled days as shown in Fig. 7.

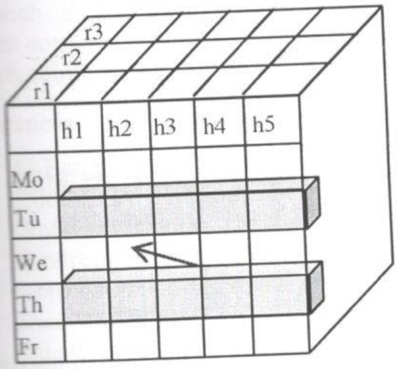

Fig. 7. Swapping two scheduled days

This modified scattered hill-climbing method is actually almost similar with the more popular stochastic hill-climbing method. The strategy was proposed to address the limitations of deterministic simple hill climbing techniques that were likely to get stuck in local optima due to their greedy acceptance of neighbouring moves (Brownlee, 2011).

2.4 Evolutionary Approach

A NP complete problem such as the current course scheduling problem is a type of optimization problem that needs non-deterministic approach or heuristic approach. Evolutionary algorithm is one of the population based heuristic optimization algorithm which is inspired by biological evolutions and natural selection. The first version of the evolutionary algorithm introduced by Rechenberg (1965) where he began with a parent and a mutated one, whichever is the fittest became a new parent (Haupt, 2004).

The evolution process involves a mutation and recombination mechanism such as cross-over between two predecessors. Mutation process takes place on a single individual where its gene or genes is changed (mutated) for instance by response of environment changing. Recombination process needs two individuals as parents where a successor inherits genes pattern from both by cross-over mechanism. The variation mechanism previously described in hill-climbing method can be considered as mutation process in the evolutionary terms.

Since chromosome in the course schedule model has a permutation scheme, the evolution mechanism must consider the permutation property. Duplication of same events and vanishing of events must be avoided during the evolution process. The recombination mechanism by cross-over process needs more attention to avoid this duplication and vanishing genes problem. Due to this problem this cross-over recombination is not taken into account in the present study.

The evolutionary algorithm used here is given in Fig. 8.

```
function EVOLUTIONARY-APPROACH(problem)
inputs:             problem: a problem
local variables:    population[],successors[]: states

  population[] ← INITIAL-POPULATION(problem)
  EVALUATE-SORT(population[])
  for iteration ←1 to ∞ do
    successors[]←R-WHEEL-SELECT(population[])
    MUTATE(successors[])
    population []←ELITISM(population[],successors[])
    FITNESS-EVALUATE-SORT(population[])
```

Fig. 8. The Evolutionary Algorithm

Initial population consist of a finite number of schedule matrices. This represents the size of population that is constant every iteration or usually called generation. The traditional roulette-wheel approach is used for selection process. Mutation will be performed using the same mechanism as in the Hill-Climbing; however it will be depend on some probabilistic parameters (Jamal, 2011). Note that the iteration is terminated when the objective is found or maximum iteration (generation) is reached. The objective function in the terminology of evolutionary theory is called fitness value deduced from the principle of the survival of the fittest. Elitism mechanism is also applied to keep the best individual from the last generation.

3. Discussion of Results

In the previously published works (Jamal, 2008, 2011), we had presented the results of these two methods for small test sets of schedule problem (i.e. 75 class-events for about 180 hours to be scheduled in 4 classrooms). The previous work (Jamal 2011) had shown that the evolutionary algorithm gave better success rate of convergence than the modified hill-climbing method. Looking deeper yields that this conclusion has a shortcoming since in the previous work we did not take into consideration the effect of the population size for the evolutionary algorithm and the size of scattered successors in the modified hill-climbing method. We used two different sizes i.e. 200 for population size in the evolutionary algorithm and 40 for scattered successors in the modified hill-climbing metod. This size reflects number of schedule matrices involve in the computation. It will be shown here that this parameter size strongly determines the success rate of convergence.

The modified hill-climbing has only one parameter to be considered, that is the scattered successor size. In the evolutionary algorithm, there are four parameters should be considered, i.e. the population size and three probabilistic parameters respectively for swapping two genes (class-events), exchanging two scheduled-days and elitism. The previous study (Jamal, 2011) has investigated the probabilistic parameters in more details and the deduced results will be reinvestigated here.

For this study two larger test sets will be used. The first set is the big one with 176 class-events of about 480 hours to be scheduled in 12 classrooms. The second set is smaller one with 96 class-events of about 250 hours to be scheduled in 6 classrooms. The space and time resources for these two sets are determines by number of days and hour available for schedule, namely 6 days per week and 8 hours per day, and by the number of classrooms. For the first set the space-time capacity is 6 x 8 x 12 = 576 class-events and for the second set, 6 x 8 x 6 = 288 class-events.

Time-complexity of the modified scattered hill-climbing method depends on the size of scattered successor linearly as evolutionary algorithm depends on population size respectively. If this size is equal then the computation time for each iteration of evolutionary approach is about two times slower than the modified scattered hill-climbing. This can be shown as complexity of evolutionary algorithm is about twice more complex than the hill-climbing algorithm.

Typical characteristic of heuristic method is that no guarantee of solution convergence but the good enough solution is well guaranteed. But in the case of hard-constraint problem, the global convergence solution is absolutely required. Thus, the number of failure of convergence to maximum objective function will be studied in this work. Table 1 shows success rate, i.e. percentage of success run that obtains a valid schedule from a number attempts for the both test sets.

Table 1. Success rate for for various size

Size	12 room set		6 room set	
	MHC	EA	MHC	EA
25	30%	0%	33%	15%
50	50%	0%	38%	20%
100	80%	40%	43%	45%
200	90%	80%	46%	55%
400	70%	90%	57%	75%
800	70%	90%	65%	75%
1600	-	-	45%	-

This has shown that both methods require sufficient size, i.e. number of schedules taken into account in the process, to get reasonable success rate. While the results of evolutionary approach shows that the success rate steadily increases by increasing the size, the success rate of modified scattered hill-climbing method starts to decrease after certain size. The logic explanation here is that increasing the successor's size causes the probability to steepest climb higher, hence the scattered hill-climbing behaves more like the original hill-climbing method.

When successful run is obtained, the number of iteration needed for converged solution in the evolutionary approach (EA) is significantly larger than the modified scattered hill-climbing (MHC) as can be depicted in Fig. 9 and Fig. 10.

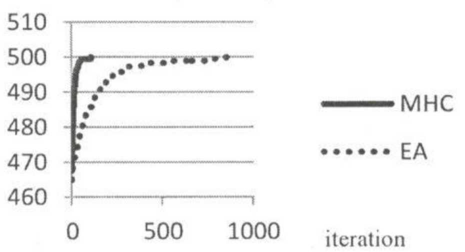

Fig. 9. Objective function as function of iteration

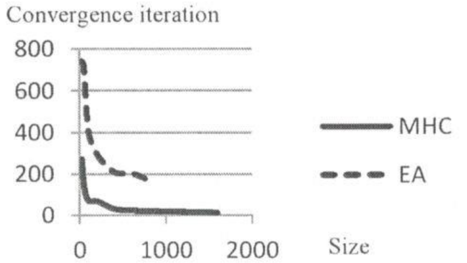

Fig. 10. Number of required iteration vs size

Fig. 10 depicts also the effect of size on the number of iteration required for convergence. For both methods, increasing size yields decreasing the required iteration number.

That hill-climbing method is much faster than the evolutionary approach can be understood since the hill-climbing is a direct local search method while the latter is a global optimization technique.

The result has shown that if number of scattered size was chosen wisely, the success rate of the modified hill-climbing method is as good as the evolutionary approach. Hence, the modified scattered hill-climbing method can be considered as global optimization with beter convergence-speed than the evolutionary algorithm.

4. Conclusion and Recommendation

We have presented two methods, namely the modified hill-climbing and the evolutionary algorithm for solving course scheduling problem. Both methods used same course schedule model and same variation mechanism, namely mutation process involving one gene only and swapping two scheduled days.

By evaluating two sets that differ in size, this study has shown the number of schedule matrices included in the process significantly determine the performance of searching for valid schedule. For both methods a sufficient number of schedule matrices are required to gain reasonable success rate of solution convergence.

In spite of the local search nature of hill-climbing method, the used modified scattered hill-climbing approach here has shown that it can avoid the local optimum as good as the global evolutionary approach if the size of scattered neighbours is chosen correctly. Thus we consider that the modified hill-climbing is a global optimization algorithm.

References:

1. Al-Betar, M.A., Khader, A.T. and Gani, T.A. (2008). "A Harmony Search Algorithm for University Course Timetabling". In: Burke E., Gendreau M. (eds.). *The Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling,* Montréal, Canada
2. Brownlee, J. (2011) *Clever Algorithms, Nature-Inspired Programming Recipes*, http://www.CleverAlgorithms.com
3. Burke, E.K., Elliman, D.G. and Weare, R.F. (1994). "A Genetic Algorithm based University Timetabling System", In *Proceedings of the 2nd East-West International Conference on Computer Technologies in Education*, Crimea, Ukraine, p. 35-40
4. Haupt, R.L. and Haupt, S.E. (2004). *Practical Genetic Algorithms*, Wiley-InterScience 2nd Edition
5. Jamal, A. (2008). "Solving University Course Scheduling Problem Using Improved Hill Climbing Approach", in *Proceeding of the International Joint Conference in Engineering*, Jakarta, Indonesia
6. Jamal, A. (2010). "A Three Stages Approach of Evolutionary Algorithm and Local-Search for Solving Hard- and Soft Constrained Couse Scheduling Problem", in *Proceeding of the 11th Seminar on Intelligent Technology and Its Application*, Surabaya, Indonesia, 2010, paper no.118, p.324-328
7. Jamal, A. (2011). "University Course Scheduling using the Evolutionary Algorithm", in *Proceeding of International Conference on Soft Computing, Intelligent System, and Information System*, Bali, Indonesia, 2010, republished in *Jurnal Al-Azhar Indonesia Indonesia, Seri Sains dan Teknologi*, Vol. I, No. 1 Maret 2011
8. Moody, D., Kendall, G. and Bar-Noy, A. (2008). "Constructing Initial Neighbourhoods to Identify Critical Constraints", in *The Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling,* Montréal, Canada