OPTIMASI PENJADWALAN MENGGUNAKAN ALGORITMA ARTIFICIAL BEE COLONY

Harits Wahyu Fahlevi[1], Galih Hermawan, S.Kom., M.T.[2]

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

ABSTRAK

Penjadwalan kegiatan perkuliahan dalam perguruan tinggi merupakan hal yang kompleks. Hal ini disebabkan karena ada beberapa hal yang harus diperhatikan dalam menyusun jadwal. Algoritma *artificial bee colony* bersifat *problem independent* sehingga algoritma ini dapat diterapkan pada permasalahan apapun dengan sedikit penyesuaian serta memiliki kemampuan dalam pencarian lokal maupun global di ruang pencarian yang besar menggunakan lebih sedikit parameter jika dibandingkan dengan pendekatan lain seperti *particle swarm optimization* atau algoritma genetika.

Penelitian sebelumnya pada tahap lebah pencarinya menggunakan konstanta bernilai antara 0 dan 1 serta nilai *fit* solusinya berasal dari nilai minat mengajar dosen terhadap kondisi tertentu yang kurang mendekati realitas yang terjadi sehari-hari. Oleh karena itu penelitian ini menggunakan tahap lebah pencari yang berbeda dan nilai solusinya berdasarkan banyaknya pelanggaran kandidat solusi terhadap batasan atau aturan yang ada di kehidupan sehari – hari.

Hasil pengujian dalam penelitian ini menunjukkan bahwa algoritma ini mampu menyelesaikan masalah penjadwalan kegiatan perkuliahan dengan menggunakan nilai parameter jumlah kandidat solusi 10, *limit* 25000 dan jumlah iterasi maksimal 50000. Selain itu informasi lain yang didapat dari penelitian ini adalah nilai *fit* kandidat solusi pada tahap inisialiasi berpengaruh terhadap jumlah iterasi yang harus dilakukan untuk mencapai solusi optimal serta penanganan soft constraint yang masih kurang maksimal.

Kata kunci : optimasi, penjadwalan perkuliahan, *artificial bee colony*, *metaheuristic.*

1. PENDAHULUAN

1.1 LATAR BELAKANG MASALAH

Penjadwalan kegiatan perkuliahan dalam perguruan tinggi merupakan hal yang kompleks. Hal ini disebabkan karena ada beberapa hal yang harus diperhatikan dalam menyusun jadwal seperti mata kuliah, ketersediaan ruangan, waktu pelaksanaan perkuliahan, keberadaan serta ketersediaan dosen dan lain sebagainya. Semakin banyak jumlah variabel yang mempengaruhi jadwal maka akan meningkatkan tingkat kompleksitasnya karena memperbesar peluang terjadinya kesalahan dalam penyusunan jadwal. Belum lagi adanya sejumlah mata kuliah yang harus dialokasikan di ruangan tertentu yang memiliki kriteria tersendiri.

Teknik-teknik yang bisa dilakukan untuk mengatasi permasalahan ini telah dikenalkan dalam kurun waktu 4 dekade terakhir oleh para peneliti di bidang riset operasional dan kecerdasan buatan yang di antara lainnya adalah pendekatan heuristik seperti *graph colouring heuristic*, teknik *constraint-based*, *case-based reasoning* dan pendekatan metaheuristik seperti *local search-based metaheuristic* dan *population-based metaheuristic* [1]. Hal ini dikarenakan masalah penjadwalan bersifat Nondeterministic Polynomial-Hard (NP-Hard) yang menyebabkan penyelesaian secara eksak sulit untuk dilakukan sementara pendekatan (heuristic) lebih logis untuk dilakukan.

Metode pendekatan terdiri dari dua jenis yaitu heuristik yang bersifat *problem dependent* atau hanya bisa dipakai di permasalahan tertentu dan metaheuristik yang sifatnya *problem independent* atau bisa dipakai untuk berbagai jenis permasalahan. Algoritma *artificial bee colony* adalah salah satu algoritma *population-based metaheuristic* yang *problem independent* serta memiliki kemampuan dalam pencarian lokal maupun global di ruang pencarian yang besar menggunakan lebih sedikit jumlah parameter jika dibandingkan dengan pendekatan lain seperti *particle swarm optimization* atau algoritma genetika [2].

Pada penelitian sebelumnya yang dilakukan Deeptimanta Ojha, Rajesh Kumar Sahoo dan Satyabrata Das pada tahap lebah pencarinya menggunakan sebuah konstanta yang nilainya diantara 0 dan 1 dan jumlah nektar kandidat solusi yang berasal dari nilai minat mengajar tiap kandidat solusi dari dosen [3]. Pada penelitian ini konstanta tersebut diganti dengan perhitungan probabilitas kandidat solusi dan jumlah nektar kandidat solusinya berasal dari berapa banyak kandidat solusi tersebut melanggar hard constraint.

Dengan dilakukannya penelitian ini harapannya adalah menghasilkan informasi bagaimana performa algoritma *artificial bee colony*

dalam menyelesaikan masalah penjadwalan kegiatan perkuliahan.

1.2 TUJUAN PENELITIAN

Berdasarkan permasalahan yang diteliti, maka tujuan yang ingin dicapai dari penelitian ini adalah menguji performa dari algoritma *artificial bee colony* dalam optimasi masalah penjadwalan kegiatan perkuliahan.

1.3 BATASAN PENELITIAN

Studi kasus di Prodi Teknik Informatika UNIKOM Bandung. Dengan data set yang diambil ialah data semester genap tahun ajaran 2015/2016 sebanyak 325 data (tidak termasuk mata kuliah kewirausahaan).

1.4 SWARM INTELLIGENCE

Swarm Intelligence adalah salah satu teknik kecerdasan buatan yang berbasis pada perilaku kolektif sistem terdesentralisasi yang dapat mengatur dirinya sendiri. Pengimplementasian *swarm intelligence* biasanya berupa sebuah populasi yang beranggotakan agen-agen sederhana yang berinteraksi secara lokal dengan sesama anggota populasi dan juga berinteraksi secara global dengan lingkungan.

Walaupun biasanya tidak memiliki struktur kendali terpusat yang memberi komando bagaimana masing-masing agen harus bertindak, interaksi secara lokal antar anggota populasi cenderung menuju pada suatu pembentukan perilaku global [4]. Dalam hal ini *swarm inteligence*-nya terinspirasi dari perilaku kawanan lebah madu dalam mencari makan.

1.5 PERILAKU KAWANAN LEBAH MADU

Model sederhana dari perilaku kawanan lebah madu dalam pencarian makanan mengarah pada munculnya kecerdasan secara kolektif dari kawanan lebah madu yang terdiri dari tiga komponen penting : sumber makanan, lebah pekerja, lebah yang mengganggur dan dua model perilaku : pengerahan ke sumber makanan dan ditinggalkannya sumber makanan.

Nilai dari sumber makanan bergantung pada banyak faktor seperti kedekatannya dengan sarang, kekayaan atau konsentrasi energinya dan kemudahan dalam mengambil energinya. Keuntungan dari sebuah sumber makanan bisa direpresentasikan dengan sebuah angka. Lebah pekerja terasosiasi dengan sumber makanan tertentu yang sedang mereka eksplorasi. Mereka juga membawa informasi dari sumber makanan tersebut, jarak dan arahnya dari sarang, keuntungan dari sumber makanan tersebut dan membagikan informasinya dengan probabilitas tertentu.

Lebah yang menganggur terus menerus mencari sumber makanan lain untuk dieksplorasi. Ada 2 tipe lebah yang mengganggur : lebah pencari, mencari di lingkungan sekitar sarang untuk sumber makanan baru dan lebah penonton yang menunggu di sarang dan memilih sebuah sumber makanan dari informasi yang dibagikan lebah pekerja. Jumlah rata-rata lebah pencari dengan kondisi normal adalah 5-10% [5]. Model perilaku ini lah yang menginspirasi algoritma artificial bee colony.

1.6 ALGORITMA ARTIFICIAL BEE COLONY

Algoritma artificial bee colony dikemukakan oleh Karaboga dan Basturk pada tahun 2007 dan Karaboga pada tahun 2005. Ide dasar dari algoritma ini adalah meniru perilaku mencari makan (seperti eksplorasi, eksploitasi, pengerahan, dan meninggalkan) dari lebah madu. Dalam algoritma ini diasumsikan bahwa setiap posisi sumber makanan adalah solusi yang mungkin dan jumlah nektar dari sumber makanan adalah kualitas dari solusi tersebut. Langkah-langkah utama dari algoritma ini menurut [6] adalah :

a. inisialisasi : populasi awal dapat didefinisikan sebagai P (G = 0) dari SN solusi (posisi sumber makanan). Dimana SN menunjukkan ukuran dari lebah pekerja atau lebah pencari. Selain itu, setiap solusi x_{ij}(i = 1, 2,..., SN; j = 1,2,…,D) adalah vektor D-dimensional. Disini, D adalah jumlah dari parameter optimasi.

b. Tempatkan lebah pekerja pada sumber makanan di memori dan memperbaharui sumber makanan yang layak. Untuk menghasilkan posisi kandidat sumber makanan dari yang lama (x_{ij}) di memori, memori lebah pekerja diperbaharui dengan persamaan berikut ini.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), \quad j \in \{1,2,\ldots,D\}, k \in \{1,2,\ldots,SN\} \wedge k \neq i, \tag{1}$$

Dimana v_{ij} adalah nilai dimensi sumber makanan layak baru yang dimodifikasi dari nilai sumber makanan sebelumnya (x_{ij}) berdasarkan perbandingan dengan nilai sumber makanan tetangga yang terpilih secara acak dan $\emptyset_{ij}$ adalah angka acak diantara |-1, 1| untuk menyesuaikan produksi dari sumber makanan tetangga disekeliling x_{ij} dan merepresentasikan perbandingan dua posisi sumber makanan secara visual.

c. Tempatkan lebah pencari pada sumber makanan di memori. Lebah pencari memilih nilai probabilitas dari sumber makanan tersebut (p_i) dengan persamaan berikut ini.

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n}, \tag{2}$$

Dimana fit_i adalah nilai *fitness* dari solusi i yang sebanding dengan jumlah nektar dari sumber makanan di posisi tersebut, SN menunjukkan ukuran dari lebah pekerja atau lebah pencari. Jadi, semakin

tinggi nilai fit_i semakin besar probabilitas x_{ij} untuk terpilih.

d. Perbaharui sumber makanan yang layak dengan lebah pencari menggunakan persamaan di langkah 2. Jika nilai fit_i tidak meningkat setelah sejumlah percobaan yang banyaknya sudah ditentukan sebelumnya, biasanya dinamakan *limit*. Sekarang, lebah pekerja atau lebah penonton berubah menjadi lebah pencari.

e. Kirim lebah pencari untuk mencari sumber makanan baru. Proses lebah pencari menjelajahi sumber makanan baru dapat didefinisikan dengan persamaan berikut ini.

$$x_i^j = x_{\min}^j + rand[0,1]\left(x_{\max}^j - x_{\min}^j\right), \qquad (3)$$

Dimana x_{min} dan x_{max} adalah batas bawah dan batas atas masing-masing dari lingkup pencarian di tiap dimensi. Disini nilai dari setiap komponen dalam semua vektor x_i harus diantara rentang | x_{min}, x_{max} | untuk mengurangi kemungkinan lebah pencari meninggalkan ruang pencarian.

f. Simpan sumber makanan terbaik yang ditemukan sejauh ini.

g. Periksa apakah kriteria untuk berhenti terpenuhi atau tidak contohnya jumlah iterasi maksimal atau MCN (*Maximum Cycle Number*), jika terpenuhi, maka prosesnya sudah selesai. Jika tidak ulangi algoritmanya kembali.

2. ISI PENELITIAN

Tahap ini akan membahas analisis yang dilakukan terhadap masalah yang diangkat dalam penelitian ini. Analisis yang dibahas adalah analisis masalah, analisis sistem yang akan dibangun dan analisis metode yang digunakan untuk menyelesaikan masalah dalam penelitian ini.

2.1 Analisis Masalah

Masalah yang dibahas dalam penelitian ini adalah masalah penjadwalan kegiatan perkuliahan yang dimana dalam penyusunan jadwal kegiatan perkuliahan tersebut ada batasan yang harus diperhatikan agar jadwal yang dihasilkan optimal dan bisa digunakan. Batasan tersebut terbagi 2 menjadi *hard constraint* yaitu batasan yang harus terpenuhi dan menjadi syarat jadwal bisa dikatakan optimal serta *soft constraint* yaitu batasan tidak harus terpenuhi yang menjadi ukuran kualitas dari jadwal yang dihasilkan.

Hard constraint dalam penelitian ini adalah sebagai berikut :

a. Seorang dosen hanya bisa mengajar 1 kelas mata kuliah dalam 1 slot waktu
b. Satu ruangan hanya bisa digunakan oleh 1 kelas mata kuliah dalam 1 slot waktu
c. Tidak boleh ada perkuliahan pada saat pelaksanaan shalat jumat

Sementara untuk *soft constaint* yang digunakan dalam penelitian ini adalah setiap dosen memiliki minimal 1 hari libur dari kegiatan mengajar.

Jadi, masalah utama dalam penelitian ini adalah sebagai berikut :

a. Bagaimana menerapkan algoritma *artificial bee colony* pada masalah penjadwalan kegiatan perkuliahan
b. Bagaimana menghasilkan jadwal kegiatan perkuliahan yang optimal menggunakan algoritma *artificial bee colony*

2.2 Analisis Data Masukan

Data set yang digunakan dalam penelitian ini terdiri dari data dosen sebanyak 76 data, data kelas mata kuliah sebanyak 325 data, data mata kuliah sebanyak 25 data, data ruangan perkuliahan sebanyak 18 data dan data slot waktu sebanyak 120 data.

2.3 Analisis Metode

Sebelum melakukan analisis terhadap metode yang digunakan berikut ini istilah – istilah yang digunakan dalam algoritma *artificial bee colony* beserta padu padannya dengan istilah yang biasa digunakan pada optimasi [7]

Tabel 1. Tabel Istilah Dalam *Artificial Bee Colony*

Istilah dalam *Artificial Bee Colony*	Istilah Optimasi
sumber makanan	solusi
jumlah nektar	nilai *fit* solusi
cycle	iterasi
posisi sumber makanan	posisi solusi
sumber makanan terbaik	solusi terbaik

2.3.1 Inisialisasi

Penentuan nilai parameter–parameter algoritma seperti ukuran populasi sumber makanan (SN), jumlah maksimum iterasi (MCN) dan *limit*. Jika sumber makanan terbaik memiliki jumlah nektar 1 yang artinya tidak ada hard constraint yang dilanggar sebelum nilai parameter MCN tercapai maka proses optimasi akan dihentikan. Dan yang terakhir *limit* merupakan ukuran maksimal percobaan untuk mengeksploitasi sumber makanan.

Buat populasi awal sumber makanan x_{ij}(i = 1,2…,SN; j = 1,2…,D) dimana x_{ij} adalah sumber makanan, SN adalah ukuran populasinya, i adalah indeks sumber makanan, D adalah jumlah dari parameter optimasi dan j adalah indeks parameter optimasinya. Setiap sumber makanan kemudian dihitung jumlah nektarnya dengan rumus 1/(1+B) dimana B adalah banyaknya pelanggaran *hard constraint.*

2.3.2 Fase Lebah Pekerja

Tempatkan lebah pekerja pada sumber makanan di memori dan hasilkan sumber makanan disekitar

sumber makanan yang dipilih menggunakan persamaan (1) dimana v_{ij} adalah sumber makanan baru hasil modifikasi dari sumber makanan sebelumnya (x_{ij}) berdasarkan perbandingan dengan nilai sumber makanan tetangga yang dipilih secara acak. $\emptyset_{ij}$ adalah bilangan acak diantara |-1,1| untuk mengendalikan produksi sumber makanan baru.

Penerapan persamaan tersebut dalam kasus ini adalah dengan menukarkan salah satu parameter optimasi setiap sumber makanan dengan sumber makanan lainnya yang dipilih secara acak kemudian dihitung jumlah nektarnya. Jika setelah ditukar jumlah nektarnya menjadi lebih tinggi maka sumber makanan hasil pertukaran yang digunakan namun jika sebaliknya maka sumber makanan lama yang digunakan.

2.3.3 Fase Lebah Penonton

Tempatkan lebah penonton pada sumber makanan di memori. Lebah penonton menghitung probabilitas masing-masing sumber makanan dengan persamaan (2) dimana i adalah nilai indeks sumber makanan dan *fit* adalah nilai jumlah nektarnya.

Sumber makanan dengan jumlah nektar tertinggi dilakukan pertukaran seperti yang dilakukan pada tahap lebah pekerja. Jika setelah ditukar jumlah nektarnya menjadi lebih tinggi maka sumber makanan hasil pertukaran yang digunakan namun jika sebaliknya maka sumber makanan lama yang digunakan.

2.3.4 Fase Lebah Pencari

Jika jumlah nektar sumber makanan tidak mengalami peningkatan setelah melakukan *cycle* sebanyak nilai parameter *limit*, sumber makanan baru dibuat menggunakan persamaan (3). Dimana i adalah nilai indeks sumber makanan, j adalah nilai indeks parameter optimasi, x_{min} dan x_{max} adalah nilai batas atas dan batas bawah dari setiap parameter optimasi.

Jika nilai *cycle* masih 1 maka tidak akan dilakukan perbandingan namun jumlah nektar sumber makanan lama sebagai pembanding apakah ada peningkatan nilai atau tidak akan diinisialisasi menggunakan jumlah nektar sumber makanan saat ini.

Jika nilai *cycle* lebih dari 1 lakukan perbandingan antara jumlah nektar lama dengan jumlah nektar saat ini, jika jumlah nektar saat ini tidak lebih tinggi dari jumlah nektar lama maka nilai limit untuk sumber makanan tersebut berkurang 1 dan jika lebih tinggi maka nilai limit akan tetap.

kemudian pilih sumber makanan dengan jumlah nektar tertinggi untuk dibandingkan dengan jumlah nektar sumber makanan terbaik, jika lebih tinggi maka sumber makanan terbaik digantikan oleh sumber makanan dengan jumlah nektar tertinggi tersebut jika tidak maka tidak ada perubahan terhadap sumber makanan terbaik.

2.4 Hasil Pengujian

Pengujian dilakukan secara bertahap dimulai dengan mata kuliah 1 semester lalu ditambahkan dengan mata kuliah semester berikutnya untuk pengujian berikutnya. Tujuan pengujian ini untuk mengetahui sejauh mana performa algoritma ini dalam masalah penjadwalan kegiatan perkuliahan dilihat dari banyaknya iterasi yang dilakukan, berapa pelanggaran soft constraintnya dan berapa nilai fit kandidat solusi terbaik setelah proses optimasi selesai.

2.4.1 Pengujian Dengan Mata Kuliah 1 Semester

Setelah beberapa kali percobaan hasil optimal tercapai dengan nilai parameter jumlah kandidat solusi = 5, limit = 3500, jumlah iterasi maksimal 8000.

Tabel 2. Tabel Pengujian dengan mata kuliah 1 semester

Percobaan ke -	Banyak iterasi	Pelanggaran soft constraint	Nilai fit kandidat solusi terbaik
1	8000	12	0.142
2	8000	6	0.083
3	619	18	1
4	1097	4	1
5	6209	11	1
6	8000	8	0.125
7	3502	7	1
8	1442	13	1
9	8000	19	1
10	4954	7	1

Hasil pengujian terbaik adalah percobaan ke – 4 dengan banyak iterasi dan pelanggaran soft constraint paling minimal.

2.4.2 Pengujian Dengan Mata Kuliah 2 Semester

Setelah beberapa kali percobaan hasil optimal tercapai dengan nilai parameter jumlah kandidat solusi = 10, limit = 10000, jumlah iterasi maksimal 15000.

Tabel 3. Tabel Pengujian dengan mata kuliah 2 semester

Percobaan ke -	Banyak iterasi	Pelanggaran soft constraint	Nilai fit kandidat solusi terbaik
1	15000	8	0.03
2	5885	17	1
3	9053	5	1
4	15000	34	0.0156
5	12466	7	1
6	10655	23	1
7	7894	20	1
8	15000	11	0.006

Percobaan ke -	Banyak iterasi	Pelanggaran soft constraint	Nilai fit kandidat solusi terbaik
9	8036	12	1
10	13895	18	1

Hasil pengujian terbaik adalah percobaan ke – 3 dengan banyak iterasi dan pelanggaran soft constraint paling minimal.

2.4.3 Pengujian Dengan Mata Kuliah 3 Semester

Setelah beberapa kali percobaan hasil optimal tercapai dengan nilai parameter jumlah kandidat solusi = 10, limit = 15000, jumlah iterasi maksimal 30000.

Tabel 4. Tabel Pengujian dengan mata kuliah 3 semester

Percobaan ke -	Banyak iterasi	Pelanggaran soft constraint	Nilai fit kandidat solusi terbaik
1	8977	8	1
2	23438	10	1
3	30000	9	0.333
4	7106	14	1
5	14615	19	1
6	30000	15	0.5
7	30000	7	0.167
8	30000	11	0.2
9	19328	7	1
10	30000	23	0.25

Hasil pengujian terbaik adalah percobaan ke – 1 dengan banyak iterasi dan pelanggaran soft constraint paling minimal.

2.4.4 Pengujian Dengan Mata Kuliah 4 Semester

Setelah beberapa kali percobaan hasil optimal tercapai dengan nilai parameter jumlah kandidat solusi = 10, limit = 25000, jumlah iterasi maksimal 50000.

Tabel 5. Tabel Pengujian dengan mata kuliah 4 semester

Percobaan ke -	Banyak iterasi	Pelanggaran soft constraint	Nilai fit kandidat solusi terbaik
1	50000	32	0.333
2	50000	17	0.1
3	42035	33	1
4	38185	19	1
5	50000	25	0.125
6	50000	28	0.083
7	50000	14	0.25
8	28364	9	1
9	50000	23	0.2
10	50000	15	0.167

Hasil pengujian terbaik adalah percobaan ke – 8 dengan banyak iterasi dan pelanggaran soft constraint paling minimal.

2.5 Kesimpulan Pengujian

Dari pengujian yang telah dilakukan dapat diambil kesimpulan sebagai berikut :

a. Nilai fit kandidat solusi pada tahap inisialisasi menjadi salah satu faktor yang berpengaruh besar dalam menentukan banyaknya iterasi yang harus dilakukan untuk mencapai nilai solusi yang optimal.
b. Penerapan soft constraint masih menghasilkan tingkat pelanggaran yang cukup tinggi.
c. Hasil pengujian terbaik dengan menggunakan semua data kelas mata kuliah adalah jumlah pelanggaran soft constraint 9 dan nilai fit kandidat solusi terbaiknya 1 yang artinya tidak ada hard constraint yang dilanggar. Untuk parameter yang digunakan adalah jumlah kandidat solusi 10, limit 25000 dan jumlah iterasi maksimal 50000.

3. PENUTUP

3.1 Kesimpulan

Berdasarkan hasil yang didapat pada penelitian ini, diperoleh informasi algoritma *artificial bee colony* mampu menyelesaikan permasalahan penjadwalan kegiatan perkuliahan dengan hasil pengujian terbaik adalah jumlah pelanggaran *soft constraint* 9 dan nilai *fit* kandidat solusi terbaiknya 1 yang artinya tidak ada *hard constraint* yang dilanggar. Untuk parameter yang digunakan adalah jumlah kandidat solusi 10, *limit* 25000 dan jumlah iterasi maksimal 50000.

Informasi – informasi lain yang didapat pada penelitian ini diantara lainnya nilai *fit* kandidat solusi pada tahap inisialiasi berpengaruh terhadap jumlah iterasi yang harus dilakukan untuk mencapai solusi optimal serta tingkat pelanggaran terhadap *soft constraint* yang diterapkan juga masih tinggi**.**

3.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, masih perlu dilakukan beberapa kajian lebih lanjut. Adapun saran untuk penelitian lebih lanjut yaitu :

a. Pengoptimalan proses pada tahap inisialisasi agar nilai *fit* kandidat solusi awal lebih baik.
b. Penerapan *soft constraint* yang lebih baik, agar jadwal yang dihasilkan lebih minim tingkat pelanggaran *soft constraint*nya.

DAFTAR PUSTAKA

[1] A. L. Bolaji, A. T. Khader, M. A. Al-Betar and M. A. Awadallah, "A Modified Artificial Bee Colony Algorithm for Pos-Enrolment Course Timetabling," *Advances In Swarm Intelligence*, vol. 1, pp. 377-386, 2013.

[2] S. Agahian, H. Pehlivan and R. Dekharghani, "Adaptation and Use of Artificial Bee Colony Algorithm to Solve Curriculum-based Course Time-Tabling Problem" *Fifth International Conference on Intelligent Systems, Modelling and Simulation*, pp. 77-82, 2014.

[3] D. Ojha, R. K. Sahoo and S. Das, "Automated Timetable Generation using Bee Colony Optimization," *International Journal of Applied. Information Systems*, vol. 10, no. 9, pp. 38-43, 2016

[4] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*, San Francisco, Morgan Kaufmann Publishers Inc, 2001

[5] D. Karaboga, "An Idea Based on Honey Bee Swarm For Numerical Optimization," *Technical Report-tr06*, vol. 200, 2005.

[6] B. Xing and W.-J. Gao, *Innovative Computer Intelligence : A Rough Guide to 134 Clever Algorithms*, Cham, Springer International Publishing, 2013.

[7] A. L. Bolaji, A. T. Khader, M. A. Al-Betar and M. A. Awadallah, "Tackling University Course Timetabling Problem Using Artificial Bee Colony Algorithm," *MASAUM Network*, vol. 12, pp. 87-96, 2012.