



PERANCANGAN PENJADWALAN PERKULIAHAN DENGAN MENGGUNAKAN PENDEKATAN *CONSTRAINT SATISFACTION PROBLEM (CSP) DAN ARTIFICIAL BEE COLONY ALGORITHM*

Danang Junaedi

Program Studi Teknik Informatika Universitas Widyatama
danangjunaedi@gmail.com

Abstrak

Penjadwalan perkuliahan merupakan suatu aktivitas yang sangat kompleks. Terdapat berbagai fakultas dan program studi yang masing-masing mempunyai kebutuhan sendiri bagaimana kegiatan perkuliahannya dilaksanakan. Selain itu juga penjadwalan harus mempertimbangkan sisi dosen, mahasiswa, ruangan dan waktu perkuliahan. Sampai saat ini telah dilakukan berbagai penelitian untuk mendapatkan cara atau metode yang mampu meningkatkan performansi dari penjadwalan perkuliahan. Paper ini mencoba untuk memaparkan rancangan penerapan penjadwalan perkuliahan dengan menggunakan pendekatan *Constraint Satisfaction Problem (CSP)* dan *Artificial Bee Colony Algorithm*.

Kata Kunci: Penjadwalan Perkuliahan *Constraint Satisfaction Problem (CSP)*, *Artificial Bee Colony Algorithm*.

1. Pendahuluan

Jadwal merupakan daftar kegiatan berdasarkan waktu, tempat dan pelaksana dari kegiatan tersebut. Jadwal harus bisa memenuhi beberapa hal yang disesuaikan dengan kebutuhan pelaksana dari kegiatan tersebut. Waktu pelaksanaan dari suatu kegiatan harus dibuat sedemikian rupa sehingga tidak ada lebih dari satu kegiatan pada waktu yang bersamaan.^[1]

Masalah penjadwalan merupakan masalah yang sangat kompleks. Ditinjau dari perspektif komputer sains, kebanyakan formulasi permasalahan penjadwalan termasuk ke dalam *NP-complete problems* yang berakibat pada tidak ada algoritma dengan waktu polinomial yang bisa digunakan untuk menyelesaikannya^[2]. Masalah penjadwalan sering dimodelkan sebagai *Combinatorial Optimisation Problems (COPs)*^[2]. Secara keseluruhan tujuan utama dari COPs adalah untuk menemukan nilai diskrit dari suatu variabel sehingga suatu solusi menjadi optimal untuk kriteria tertentu. Walaupun untuk kasus yang kompleks suatu algoritma tidak selalu menghasilkan solusi yang optimal, namun setidaknya bisa diperoleh solusi yang cukup baik untuk kebutuhan tertentu tergantung kriteria yang telah ditentukan. Selain itu masalah penjadwalan juga sangat kompleks, tergantung dari kebutuhan individu dan institusi yang terkait. Sehingga untuk kasus yang sama pada tempat

atau institusi yang berbeda suatu algoritma yang digunakan dan berhasil di suatu tempat belum tentu bisa digunakan untuk tempat yang lain.

Terdapat tren dalam komunitas sains untuk membuat model dan menyelesaikan optimasi untuk permasalahan yang kompleks berdasarkan kehidupan di alam. Hal ini diakibatkan karena ketidakefisienan algoritma optimasi klasik dalam memecahkan masalah kombinatorial dan/atau masalah non linear berskala besar. Dalam rangka mengatasi masalah ini diperlukan suatu algoritma yang lebih fleksibel dan mampu beradaptasi. Algoritma ini harus mudah dimodifikasi ke dalam suatu model yang semirip mungkin dengan permasalahan.

2. Identifikasi Permasalahan

Permasalahan yang sering muncul adalah :

1. Terdapat ketidaksesuaian antara fasilitas yang ada dalam suatu ruangan dengan kebutuhan mata kuliah tertentu.
2. Jika dosen terpaksa mengganti jadwal perkuliahannya karena sesuatu hal, sulit menentukan jadwal dan ruangan penggantinya.
3. Penempatan ruang kuliah sering tidak memperhatikan kondisi dosen atau mahasiswa peserta mata kuliah tersebut.
4. Ada waktu-waktu khusus yang sering dipilih oleh dosen, sehingga pada waktu-waktu tersebut sering

terjadi bentrok jadwal perkuliahan karena adanya kegiatan perkuliahan, namun ada juga waktu-waktu tertentu dimana banyak ruangan kosong yang tidak digunakan untuk kegiatan perkuliahan.

5. Pencarian ruangan kelas pengganti memerlukan waktu yang tidak sedikit, hal ini dikarenakan harus mempertimbangkan jadwal yang ada pada hari tersebut.

3. Studi Pustaka

3.1 Penjadwalan Perkuliahan

Penjadwalan perkuliahan dapat didefinisikan sebagai suatu pekerjaan untuk mendistribusikan sejumlah aktivitas yang melibatkan dosen, mata kuliah, dan seterusnya, ke dalam satuan waktu (bisa termasuk ruangan) yang terbatas dengan batasan-batasan tertentu. *Constraint* atau batasan yang biasa digunakan dalam kebanyakan masalah penjadwalan biasanya dibagi menjadi dua yaitu *hard constraints* dan *soft constraints*. *Hard constraints* memiliki prioritas lebih tinggi dibanding *soft constraint*, dan akan menjadi pertimbangan utama untuk bisa dipenuhi. Kenyataannya, penjadwalan biasanya hanya akan mempertimbangkan sebagian atau keseluruhan *hard constraints* dari suatu masalah yang telah dipenuhi. Sedangkan *soft constraints*, merupakan *constraint* atau pertimbangan yang bisa dilaksanakan jika memungkinkan dan biasanya menjelaskan apa saja yang bisa dipertimbangkan disesuaikan dengan kebijakan universitas, atau berdasarkan pengalaman dari orang yang biasa melaksanakannya.^[2]

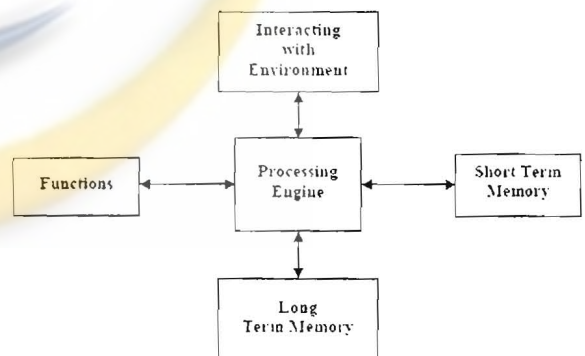
Romero mengidentifikasi 3 stakeholder utama dalam proses penjadwalan perkuliahan, yaitu^[5]:

1. Bagian Akademik, membuat standar minimum yang harus dipenuhi oleh penjadwalan. Misalnya jam perkuliahan, satu sks setara dengan berapa menit.
2. Program studi, lebih mementingkan pelaksanaan dari penjadwalan tersebut. Program studi menginginkan penjadwalan sesuai dengan kebutuhan dari mata kuliah tertentu.
3. Mahasiswa, merupakan stakeholder yang paling dipengaruhi oleh penjadwalan perkuliahan dan cukup sulit menentukan kebutuhan khusus untuk mahasiswa dikarenakan jumlah mahasiswa biasanya cukup banyak.

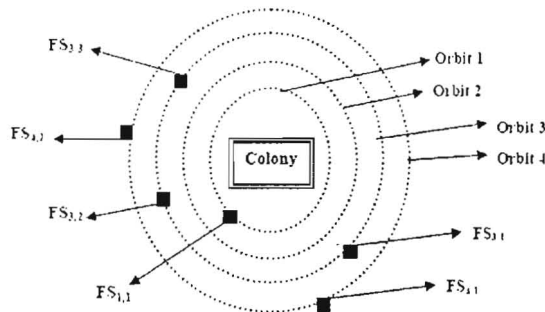
3.2 Artificial Bee Colony^[3]

Artificial Bee Colony (ABC) adalah algoritma yang dikenalkan oleh Dervis Karaboga pada tahun 2005, dimotivasi oleh *intelligent behavior of honey bees*. Algoritma ini sederhana algoritma *Particle*

Swarm Optimization (PSO) dan algoritma *Differential Evolution* (DE), dan hanya menggunakan parameter kontrol yang biasa seperti ukuran koloni dan jumlah maksimum *cycle*. ABC sebagai alat optimisasi, menyediakan prosedur pencarian berbasis populasi dimana individu yang disebut penunjuk makanan dimodifikasi dengan *artificial bees with time* dan tujuan lebah adalah untuk menemukan tempat dimana terdapat sumber makanan dengan jumlah nektar yang banyak dan akhirnya dengan jumlah nektar paling banyak. Dalam sistem ABC, lebah terbang berkeliling dalam multi dimensional *search space* dan beberapa lebah (pekerja dan *onlooker bees*) memilih sumber makanan tergantung dari pengalamannya dan pasangannya, dan kemudian menentukan posisinya. Beberapa terbang dan memilih sumber makanan secara acak tanpa menggunakan pengalaman apapun. Jika jumlah nektar dari suatu sumber lebih dari sumber sebelumnya (berdasarkan ingatan lebah tersebut), maka posisi baru tersebut akan disimpan di memori menimpa lokasi yang lama. Sistem ABC mengkombinasikan metode pencarian lokal, yang dilakukan oleh pekerja dan *onlooker bees*, dengan metode pencarian global, dikelola oleh *onlookers* dan *scouts*, berdasarkan keseimbangan dan proses eksplorasi. Gambar 1 berikut adalah arsitektur dari sistem *artificial bee colony* dan Gambar 2 adalah area dari *bee colony*.



Gambar 1 Arsitektur Sistem *Artificial Bee Colony*^[6]



Gambar 2 Bee Colony Area [6]

3.3 Constraint satisfaction Problem [5]

Constraint satisfaction problem (CSP) didefinisikan sebagai sekumpulan variabel X_1, X_2, \dots, X_n , dan sekumpulan *constraint* C_1, C_2, \dots, C_m . Setiap variabel X_i memiliki *non empty domain* D_i dari nilai yang mungkin. Setiap *constraint* C_i melibatkan suatu *subset* dari variabel dan menunjukan kombinasi yang diijinkan dari nilai untuk *subset* tersebut. *State* dari masalah didefinisikan sebagai penugasan dari suatu nilai ke suatu atau semua variabel $\{X_i = v_i, X_j = v_j, \dots\}$. Penugasan yang tidak mengganggu *constraint* lain disebut *consistent* atau *legal assignment*. Penugasan lengkap jika sudah melingkupi atau memenuhi semua variabel, dan solusi dari CSP adalah penugasan lengkap yang telah memenuhi semua *constraint*. Beberapa CSP juga membutuhkan solusi yang merupakan solusi maksimum.

CSP bisa diformulasikan dalam tahapan sebagai berikut :

1. *Initial state* : *empty assignment* $\{\}$ dimana semua variabel belum terhubung.
2. *Successor function* : nilai yang bisa dihubungkan dengan variabel manapun yang belum terhubung, namun tidak mengakibatkan konflik dengan variabel yang telah terhubung sebelumnya.
3. *Goal test* : Penugasan telah lengkap.
4. *Path cost* : nilai konstan (misalnya 1) untuk setiap tahap.

Semua solusi harus merupakan penugasan yang telah lengkap dan bagaimanapun akan muncul dengan ke dalaman n untuk n buah variabel. Untuk alasan inilah algoritma *depth first search* sangat populer untuk kasus CSP. ada kemungkinan kasus dimana *path* yang dijadikan solusi tidak relevan, untuk itu kita juga menggunakan *complete-state formulation*, yang digunakan untuk setiap *state* yang merupakan penugasan lengkap yang akan atau tidak akan

memenuhi *constraint*. *Local search* bekerja cukup baik untuk formula tersebut.

4. Pembahasan

4.1 Tujuan

1. Mengetahui kebutuhan penerapan penjadwalan perkuliahan dengan menggunakan pendekatan constant satisfaction problem dan artificial bee colony algorithm
2. Mendapatkan gambaran mengenai model penjadwalan perkuliahan dengan menggunakan pendekatan *Constraint Satisfaction Problem* (CSP) dan *Artificial Bee Colony Algorithm*

4.2 Model Penjadwalan

Sebelum membuat penjadwalan terlebih dahulu harus ditentukan *constraint* dari penjadwalan tersebut. Dalam penelitian ini *constraint* yang akan menjadi pertimbangan adalah:

1. *Hard Constraint*, yang terdiri dari
 - a. Dalam satu waktu seorang dosen hanya boleh mengajar satu kelas
 - b. Mahasiswa dalam satu waktu hanya boleh terdaftar dalam satu kelas
 - c. Ruangan dalam satu waktu hanya bisa digunakan oleh satu kelas dan jumlah peserta dalam kelas tersebut tidak melebihi kapasitas ruangan
 - d. Untuk perkuliahan satu SKS setara dengan 50 menit, untuk praktikum satu SKS setara dengan 60 menit
2. *Soft Constraint*, yang terdiri dari:
 - a. Pelaksanaan perkuliahan untuk mata kuliah harus ditempatkan pada ruangan sesuai kebutuhan dari mata kuliah tersebut (misalnya membutuhkan infokus, komputer dan seterusnya),
 - b. Jarak ruangan dari suatu mata kuliah ke mata kuliah yang lain tidak terlalu jauh (untuk pergantian kelas pada jam yang berdekatan)
 - c. Dosen dan/atau mahasiswa ditempatkan pada tempat sesuai dengan kondisi fisik dari yang bersangkutan (misalnya dosen yang sudah berusia lanjut ditempatkan di ruangan yang dekat dengan ruang tunggu dosen)
 - d. Untuk mata kuliah yang dianggap sulit (dapat dilihat dari tingkat kelulusan mahasiswa untuk mata kuliah tersebut pada semester sebelumnya) dilaksanakan pada awal-awal minggu dan jam yang lebih pagi.

Yang menjadi variabel adalah dosen, ruangan, mata kuliah dan waktu. Dimana masing-masing variabel tersebut memiliki atribut-atribut tertentu sesuai dengan kebutuhan atau kesepakatan di universitas tersebut. Berdasarkan variabel dan constraint di atas dibuat kombinasi-kombinasi berupa pemetaan antar variabel yang memenuhi *constraint* di atas dan hasil dari pemetaan ini nantinya berupa jadwal perkuliahan.

Algoritma yang digunakan untuk melakukan pemetaan akan dicoba menggunakan *artificial bee colony system* seperti yang dibuat oleh Saif Mahmood Saab^[2] sebagai berikut :

1) **Initialization:**

- b) **X**: number of scouts.
- c) **Y**: number of foragers.
- d) **N**: number of bees in the colony population.
- e) **I**: number of orbits.
- f) **CM**: colony moral.
- g) **FSij**: the *j*th food source at the *i*th orbit.
- h) **M**: number of maximum iterations.
- i) **FSQij**: the quality of the food source, **FSij**.
- j) **Dij**: the direct distance between colony and **FSij** (lihat Gambar 2).

2) **Assumptions:**

- a) The food sources have been previously defined.
- b) Every scout bee, **Sx**, visit only one food source, **FSij**, per each trip.

3) **REPEAT X times** (i.e. all scouts, **X**, finish one trip):

- {
- a) Scout should choose one food source, **FSij**, find its distance, **Dij**, quality **FSQij**, and net gain of energy **FSGij**:
$$FSGij = FSQij / Dij$$
- b) Update **JST** table.
- c) Update **FTT** table.
- }

REPEAT UNTIL the **CM** goal is achieved or reached the maximum number of iterations, **M**:

- {
- REPEAT Y times** (i.e. all foragers, **Y**, complete one trip):
- {
- (a) Generate a random number, **RN**.
- (b) Calculate food source total quality
$$FSQTij = FSGij + SRx$$
- (c) Foragers, **FRy**, choose which food source, **FSij**, to forage by comparing the random number, **RN**, with **FSQTij** on the **JS** table.
- }
- }

If **FSQTij** > **RN** Then

Choose **FSij** and go to (h)

Else

Repeat (g)

End If

(d) Update **JST**.

(e) Update the **CM**.

}

5. Kesimpulan

Model pendekatan yang digunakan untuk membuat penjadwalan perkuliahan adalah *Constraint Satisfaction Problem (CSP)* dan *Artificial Bee Colony Algorithm*. Algoritma yang digunakan untuk melakukan pemetaan akan dicoba menggunakan *artificial bee colony system*^[2]. Diharapkan dengan menggunakan model ini penjadwalan perkuliahan dapat memenuhi kebutuhan dari pihak-pihak yang terkait sehingga pemanfaatan ruangan dan waktu dapat lebih optimal dari yang sebelumnya. Tahap berikutnya adalah melakukan implementasi dari model rancangan pada paper ini dan kemudian melakukan pengukuran apakah jadwal yang diperoleh lebih optimal dari yang sebelumnya. Selain itu perlu dipertimbangkan teknik pengujian yang sebaiknya digunakan untuk mengukur kehandalan dari model yang telah dibuat sebelumnya.

Referensi

- [1] Willemen, Robertus J. 2002. "School timetable construction : algorithms and complexity". Technische Universiteit Eindhoven.
- [2] Rhydian Marc Rhys Lewis. 2006. "Metaheuristics for University Course Timetabling". Napier University
- [3] -. "Artificial Bee Colony". <http://mf.erciyes.edu.tr/abc/>, Tanggal Akses : 9 Mei 2009
- [4] E. Burke, et al .1997. "Automated University Timetabling: The State of the Art". The Computer Journal 40(9):565-571. <http://com.jnl.oxfordjournals.org/cgi/content/short/40/9/565> Tanggal Akses : 12 Juli 2010

- [5] Stuart Russell and Peter Norvig. 2002. *"Artificial Intelligence: A Modern Approach"*. <http://aima.cs.berkeley.edu/>, Tanggal Akses : 9 Mei 2009
- [6] Saif Mahmood Saab, et al. 2010. " *Developing Optimization Algorithm Using Artificial Bee Colony System*". http://www.ubicc.org/files/pdf/Developing_Optimization_Algorithms_Using_Artificial_Bee_Colony_System.doc_391.pdf Tanggal Akses: 5 Mei 2010
- [7] Adil Baykasoğlu, et al. 2007. *"Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem, Swarm Intelligence: Focus on Ant and Particle Swarm Optimization"*, Book edited by: Felix T. S. Chan and Manoj Kumar Tiwari, ISBN 978-3-902613-09-7, pp. 532, December 2007, Itech Education and Publishing, Vienna, Austria
- [8] Hiroshi MORITA and Naoki SHIO. 2004. *"Hybrid Branch and Bound Method with Genetic Algorithm for Flexible Flowshop Scheduling Problem"*. Department of Information Physics and Sciences. Graduate School of Information Science and Technology, Osaka University
- [9] Chin Soon Chong, et al. 2006. *"A Bee Colony Optimization Algorithm To Job Shop Scheduling"*. Proceedings of the 2006 Winter Simulation Conference
- [10] Muhammad Aria. *"Aplikasi Algoritma Genetik Untuk Optimasi Penjadwalan Mata Kuliah"*, Majalah Ilmiah Unikom, Vol.6, hlm. 17-25