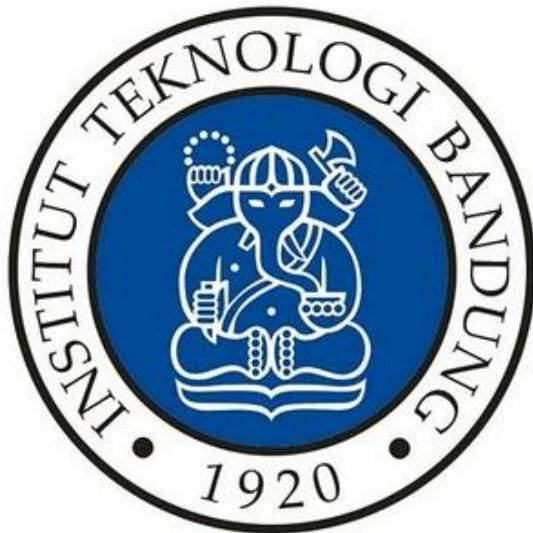


Laporan Tugas Besar 1

IF3054 Inteligencia Buatan

Penjadwalan Menggunakan Local Search



Kelompok : Altakata

Anggota Kelompok :

- 1. Amir – 13514017**
- 2. Jeremia Jason Lasiman – 13514021**
- 3. Bervianto Leo Pratama – 13514047**
- 4. Richard Wellianto – 13514051**
- 5. M. Az-zahid Adhitya Silparensi – 13514095**

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika


Institut Teknologi Bandung

Daftar Isi

| | |
|-------------------------------------|----------|
| Dokumentasi | 3 |
| Halaman Awal | 3 |
| Halaman Tentang | 3 |
| Halaman Hasil | 4 |
| Dasar Teori..... | 5 |
| Hasil Eksekusi Uji Coba..... | 8 |
| Referensi | 9 |

Dokumentasi

Halaman Awal

 **SCHEDULER OP**

Home About Us

Input Your File To Bottom

Select a file:

Browse... No file selected.

Select Method:

☐ Hill Climbing


☐ Simulated Annealing

☐ Genetic Algorithm

Upload

© 2013 ShapeBootstrap. All Rights Reserved.


Halaman Tentang

 **SCHEDULER OP**





Home About Us


Meet the Team

This web application created by :








Amir
13514017










Jeremia Jason Lasiman
13514021










Bervianto Leo Pratama
13514047






Richard Wellianto
13514051





M. Az-zahid Adhitya Silparensi
13514095

Halaman Hasil

 SCHEDULER OP

[Home](#) [About Us](#)

7602 7603 7610 Labdas2

Jadwal Matakuliah di Ruangan 7602

| Time | Monday | Tuesday | Wednesday | Thursday | Friday |
|-------|--------|---------|-----------|----------|--------|
| 7:00 | IF3110 | IF2110 | | IF3170 | |
| 8:00 | IF3110 | IF2110 | | IF3170 | |
| 9:00 | | IF2110 | | IF3170 | |
| 10:00 | IF2150 | IF2110 | | IF3111 | |
| 11:00 | IF2150 | | | IF3111 | |
| 12:00 | IF2150 | | | IF3111 | |
| 13:00 | | | | | |
| 14:00 | | | | | |
| 15:00 | | | | | |
| 16:00 | | | | | |
| 17:00 | | | | | |

Jumlah Konflik : 0

Hasil Detail :
IF2110 7 10 2 7602
IF2130 11 14 3 7603
IF2150 10 12 1 7602
IF2170 9 12 2 7610
IF3110 7 8 1 7602
IF3130 10 12 4 Labdas2
IF3170 7 9 4 7602
IF3111 10 12 4 7602

© 2013 ShapeBootstrap. All Rights Reserved

Dasar Teori

- **CSP (Constraint Satisfaction Problem)**

CSP adalah sebuah masalah matematis dimana kita harus menemukan variabel dan domain yang memenuhi constraint pada masalah.

- **Variabel**

Variabel adalah sesuatu yang nilainya tidak tetap, bisa berubah. Nilai variabel menentukan banyak konflik yang terjadi dalam Constraint Satisfaction Problem.

- **Domain**

Domain adalah seluruh nilai yang bisa dimasukkan ke variabel.

- **Constraint**

Constraint adalah aturan atau rule yang harus dipenuhi untuk mendapatkan hasil akhir.

- **Local Search**

- **Hill-Climbing**

Algoritma *Hill-Climbing* termasuk kategori pencarian heuristik. Metode ini awalnya menginisiasi solusi acak dari suatu masalah, kemudian berulang-ulang mengganti nilai tiap variabel yang menyalahi *constraint* yang ada, sehingga menuju solusi yang lebih baik dari solusi sebelumnya. Terdapat beberapa variasi *Hill-Climbing* seperti *Stochastic Hill-Climbing* dan *Random-Restart Hill-Climbing*. Karena metodenya yang mirip algoritma *Greedy*, Algoritma *Hill-Climbing* belum pasti mendapatkan solusi optimum global, hanya solusi optimum lokal.

- **Simulated Annealing**

Algoritma *Simulated Annealing* adalah algoritma pencarian heuristik. Metode ini mirip seperti metode *Hill-Climbing*, tetapi di awal dapat dilakukan langkah yang tidak menuju solusi optimum lokal. Ini ditentukan dari suatu "temperatur" awal yang ditetapkan terlebih dahulu, yang merupakan pembanding banyak langkah yang boleh tidak menuju solusi optimum lokal dengan banyak langkah yang menuju solusi optimum lokal. Seiring berjalannya program, temperatur akan terus berkurang dengan rasio tertentu sehingga ketika temperatur menjadi atau mendekati nol, langkah-langkah yang diambil akan sama dengan langkah yang diambil ketika menggunakan algoritma *Hill-Climbing*.

- **Genetic Algorithm**

Algoritma *Genetic Algorithm* adalah algoritma yang menggunakan representasi genetik untuk memecahkan masalah. Pertama program akan membuat populasi calon solusi yang berjumlah k-buah. Dari populasi tersebut akan dipilih 2 solusi yang akan menghasilkan solusi baru(selection). Selection dipilih dengan fitness function. Kemudian solusi baru didapat dari crossover dan mutasi kedua solusi yang dipilih dari selection. Crossover adalah pendapatan solusi baru dari pertukaran kromosom solusi yang dipilih. Mutasi adalah dari pengubahan solusi baru dalam skala kecil dengan kemungkinan tertentu. Genetic algorithm juga mempunyai terminasi yang digunakan saat solusi mempunyai fitness function yang telah memenuhi standar tertentu.

Implementasi ke Program (Dokumentasi Teori)

- **CSP**

- **Variabel**

Pada program ini kami memilih course sebagai variabel. Karena, pertama seluruh course harus di assign untuk mengetahui apakah program sudah selesai atau belum, kedua jika slot schedule sebagai variabel diisi maka akan ada kemungkinan slot yang mempunyai dua isi.

- **Domain**

Domain variabel kami inialisasi dari unary constraint yang didapat pada file teks eksternal.

- **Constraint**

Constraint kami bedakan menjadi dua yaitu unary constraint dan binary constraint. Implementasi unary dilakukan dengan pengecekan kedalam class constraint yang kami buat. Sedangkan binary constraint diimplementasikan saat local-search dimulai. Implementasi binary constraint dilakukan dengan pengecekan dua variabel jika tabrakan maka constraint belum terpenuhi.

- **Local Search**

- **Hill-Climbing**

Algoritma "*Hill-Climbing*" kami implementasikan dengan cara, pertama kami inialisasi seluruh variabel dan inialisasi pasti memenuhi unary constraint. Kemudian kami mengecek berapa konflik yang didapat dari inialisasi tersebut. Jika 0 maka tidak ada tindakan yang dilakukan. Sebaliknya jika ada konflik maka komputer akan mengambil seluruh action possible yang memenuhi unary constraint. Seluruh action tersebut akan dilihat menghasilkan berapa konflik. Dan program kami mengambil action yang konfliknya paling kecil. Action tersebut dilakukan kemudian diperiksa kembali jika konflik 0 maka program akan selesai dan mengembalikan hasilnya. Jika tidak maka program akan mencoba kembali sampai konflik 0 atau stepnya sudah terlalu banyak.

- **Simulated Annealing**

Algoritma "*Simulated Annealing*" kami implementasikan dengan cara sebagai berikut. Pertama, dilakukan inialisasi variabel sama seperti hill climbing. Selanjutnya akan dicari setiap aksi yang dapat dilakukan kemudian dipertimbangkan apakah setelah menjalankan aksi tersebut lebih baik dari aksi sebelumnya dengan melihat *state* sebelum dan sesudah serta membandingkan jumlah konfliknya. Jika *state* sesudah memiliki konflik lebih sedikit maka aksi itu akan dilakukan dan mengubah isi variabel yang ada. Tetapi, jika *state* sesudah memiliki konflik lebih banyak, maka akan dilakukan random apakah *state* tersebut akan dilakukan atau tidak menggunakan fungsi penerimaan kemungkinan. Program akan berjalan terus hingga mencapai konflik 0 atau melebihi jumlah move yang diberikan.

- **Genetic Algorithm**

"*Genetic Algorithm*" kami implementasikan dengan cara sebagai berikut. Definisi string yang kami buat adalah kumpulan variabel dan fitness function merupakan jumlah variabel yang tidak memiliki konflik. Pertama akan melakukan inisialisasi seluruh string (jumlah default string yang dipakai adalah 4). Kemudian akan dipilih aksi yang akan dilakukan, apakah *mutasi* atau *cross-over*. Kemudian akan diulang sampai konflik mencapai 0 atau melebihi jumlah move yang diperbolehkan.

Hasil Eksekusi Uji Coba

Jumlah Konflik : 0

Hasil Detail :

- IF2110 7 10 1 7602
- IF2130 10 12 3 7602
- IF2150 11 12 1 7602
- IF2170 9 11 1 7610
- IF3110 7 8 4 7602
- IF3130 7 8 3 7602
- IF3170 7 8 2 7602
- IF3111 7 8 1 7603

Jadwal Matakuliah di Ruang 7610

| Time | Monday | Tuesday | Wednesday | Thursday | Friday |
|-------|--------|---------|-----------|----------|--------|
| 7:00 | | | | | |
| 8:00 | | | | | |
| 9:00 | IF2170 | | | | |
| 10:00 | IF3170 | | | | |
| 11:00 | IF2170 | | | | |
| 12:00 | | | | | |
| 13:00 | | | | | |
| 14:00 | | | | | |
| 15:00 | | | | | |
| 16:00 | | | | | |
| 17:00 | | | | | |

Testcase.txt (-/PycharmProj)

Open Save

Testcase.txt x Testcase2.txt x

Ruangan

7602;07.00;14.00;1,3,4,5

7603;07.00;14.00;1,3,5

7610;09.00;12.00;1,2,3,4,5

Labdas2;10.00;14.00;2,4

Jadwal

IF2110;7602;07.00;12.00;4;1,2,3

Plain Text Tab Width: 8

Testcase 1

Jumlah Konflik : 0

Hasil Detail :

- IF2110 7 10 1 7602
- IF2130 11 13 3 7602
- IF2150 9 10 3 7610
- IF2170 9 11 4 7610
- IF3110 7 8 5 7602
- IF3130 10 11 4 7610
- IF3170 7 8 3 7602
- IF3111 10 11 2 7602
- IF2121 11 13 4 7602
- IF3150 7 9 2 7604
- IF3150 7 9 4 Multimedia
- IF3151 11 12 5 7603
- IF3170 8 10 4 7602
- IF2110 8 10 5 7603
- IF2130 9 10 5 7604
- IF2150 9 10 3 7602
- IF2110 9 10 5 7610

Jadwal Matakuliah di Ruang 7602

| Time | Monday | Tuesday | Wednesday | Thursday | Friday |
|-------|--------|---------|-----------|----------|--------|
| 7:00 | IF3110 | | IF3170 | | IF3110 |
| 8:00 | IF3110 | | IF3170 | IF3170 | IF3110 |
| 9:00 | IF3110 | | IF2150 | IF3170 | |
| 10:00 | IF3110 | | IF2150 | IF3170 | |
| 11:00 | | | IF2130 | IF2121 | |
| 12:00 | | | IF2130 | IF2121 | |
| 13:00 | | | IF2130 | IF2121 | |
| 14:00 | | | | | |
| 15:00 | | | | | |
| 16:00 | | | | | |
| 17:00 | | | | | |

Testcase2.txt (-/PycharmProj)

Open Save

Testcase.txt x Testcase2.txt x

Ruangan

7602;07.00;14.00;1,3,4,5

7603;07.00;14.00;1,3,5

7610;09.00;12.00;1,2,3,4,5

Labdas2;10.00;14.00;2,4

7604;07.00;13.00;2,5

Multimedia;07.00;14.00;4,5

Plain Text Tab Width: 8

Testcase 2

Jumlah Konflik : 0

Hasil Detail :

- IF2110 7 9 2 7602
- IF2130 9 11 1 7602

Jadwal Matakuliah di Ruang 7602

| Time | Monday | Tuesday | Wednesday | Thursday | Friday |
|-------|--------|---------|-----------|----------|--------|
| 7:00 | | | | | |
| 8:00 | | IF2130 | | | |
| 9:00 | IF2130 | IF2130 | | | |
| 10:00 | IF2130 | | | | |
| 11:00 | IF2130 | | | | |
| 12:00 | | | | | |
| 13:00 | | | | | |
| 14:00 | | | | | |
| 15:00 | | | | | |
| 16:00 | | | | | |
| 17:00 | | | | | |

Testcase3.txt (-/PycharmProjects/scheduler-op/doc) - gedit

Open Save Undo Redo

Testcase.txt x Testcase2.txt x Testcase3.txt x

Ruangan

7602;07.00;12.00;1,2

Jadwal

IF2110;-;07.00;10.00;3;1,2

IF2130;-;09.00;12.00;3;1,2

Plain Text Tab Width: 8 Ln 5, Col 27 INS

Testcase3

Referensi

Template Web - <http://www.free-css.com/free-css-templates/page202/flat-theme>