

LAPORAN TUGAS PROGRAM 1

**Gabe Dimas Wicaksana**

**1301154245**

**IF 39-07**

SIMULATED ANNEALING

|  |
| --- |
| **CCH3F3**  **ARTIFICIAL INTELLIGENCE**  **SEMESTER GANJIL 2017/2018** |
| **S1 TEKNIK INFORMATIKA**  **FAKULTAS INFORMATIKA** |

## Deskripsi Masalah

**Simulated annealing** (SA) adalah salah satu [algoritma](https://id.wikipedia.org/wiki/Algoritma" \o "Algoritma) untuk untuk optimisasi yang bersifat generik. Berbasiskan [probabilitas](https://id.wikipedia.org/wiki/Probabilitas" \o "Probabilitas) dan [mekanika statistik](https://id.wikipedia.org/wiki/Mekanika_statistik" \o "Mekanika statistik), algoritma ini dapat digunakan untuk mencari pendekatan terhadap solusi optimum global dari suatu permasalahan. Masalah yang membutuhkan pendekatan SA adalah masalah-masalah optimisasi kombinatorial, di mana ruang pencarian solusi yang ada terlalu besar, sehingga hampir tidak mungkin ditemukan solusi eksak terhadap permasalahan itu.

## Rancangan Metode

Program ini akan mencari nilai paling minimum dari fungsi berikut dengan menggunakan algoritma Simulated Annealing :

Dengan batasan

Annealing adalah satu teknik yang dipakai dalam bidang metalurgi, khususnya dipakai dalam mempelajari proses pengkristalan pada suatu materi. Agar dapat membentuk susunan kristal yang sempurna, diperlukan pemanasan sampai suatu tingkat tertentu, kemudian dilanjutkan dengan pendinginan yang perlahan-lahan dan terkendali dari materi tersebut. Pemanasan materi pada proses awal annealing, memberikan kesempatan pada atom-atom dalam materi itu untuk bergerak secara bebas, mengingat tingkat energi dalam kondisi panas ini cukup tinggi. Proses pendinginan yang perlahan-lahan memungkinkan atom-atom yang tadinya bergerak bebas itu, pada akhirnya menemukan tempat yang optimum, di mana energi internal yang dibutuhkan atom itu untuk mempertahankan posisinya adalah minimum.

**Langkah-langkah Simulated Annealing :**

* Evaluasi *Initial State.* Jika state ini adalah *goal state,* maka tampilkan state tersebut sebagai solusi akhir dan *exit* dari program. Jika tidak, maka assign *initial state* ke *current state*.
* Inisialisasikan BEST\_SO\_FAR dengan *current state.*
* Inisialisasikan T (suhu) sesuai dengan *annealing schedule.*
* Lakukan perulangan sampai solusi didapat atau tidak ada aturan produksi lagi yang bisa diaplikasikan ke *current state.*
  + Pilih sebuah aturan produksi yang belum pernah diaplikasikan ke *current state* dan aplikasikan aturan produksi tersebut untuk menghasilkan *new state.*
  + Evaluasi nilai *new state*
    - Jika *new state* adalah *goal,* maka tampilkan *state* ini sebagai solusi lalu keluar dari program
    - Jika *new state* bukan *goal* tetapi lebih baik dari *current state,* maka assign *new state* dengan *current state* dan assign BEST\_SO\_FAR dengan new state.
    - Jika *new state* tidak lebih baik dari *current state*, maka assign *current state* dengan *new state* dengan rumus probabilitas berikut :

p(Delta E) = e-Delta E / T > R, dimana Delta E = *newstate* – *currentstate,* dan R adalah nilai random.

Langkah ini juga diimplementasikan dengan menggunakan nilai bilangan acak (R) dengan range dari 0 sampai 1. Tetapi, untuk mendapatkan nilai yang lebih baik, range bisa dicoba dari antara nilai 0.8 sampai dengan 0.99.

Jika nilai probabilitas tersebut memiliki nilai kurang dari R, maka assign *current state* dengan *new state.* Tetapi jika nilai probabilitas lebih dari nilai R, maka abaikan saja.

* Tampilkan nilai dari BEST\_SO\_FAR sebagai solusi dari fungsi diatas

## Hasil Program dan Screenshot

Untuk mengecek nilai akurasi dari program yang telah dibuat, maka akan digunakan perhitungan Akurasi Model sebagai berikut :

= nilai minimum yang didapatkan dari program kita.

= nilai minimum yang diasumsikan oleh Dosen

Dengan perhitungan diatas, akan didapatkan hasil program sebagai berikut :

Minimal answer : 0.02277282060414465

Accuracy : 198.48181195972367 %

