Simulated Annealing

Simulated annealing merupakan salah satu algoritma *local search* yang menganalogikan proses *annealing,* yaitu salah satu teknik yang dikenal dalam bidang metalurgi untuk membentuk Kristal dari suatu materi. Pada proses *annealing,* materi yang berbentuk cairan akan berangsur-angsur menjadi lebih padat seiring dengan penurunan suhu. Hal itu dikarenakan pada suhu yang tinggi, partikel-partikel materi akan lebih bebas bergerak dan juga sebaliknya. Seperti pada proses *annealing*, pada saat variabel suhu bernilai tinggi, algoritma dapat melakukan langkah yang menghasilkan *state* yang tidak lebih baik dari sebelumnya. Namun semakin nilai suhu menurun, kemungkinan untuk menerima *state* yang tidak lebih baik akan menurun dan pada suhu nol derajat atau dibawah nol derajat, algoritma tidak akan menerima *state* yang lebih buruk dari sebelumnya (menyerupai proses *hill-climbing*). Berbeda dengan proses *hill-climbing*  yang hanya melakukan langkah terbaik, langkah yang dilakukan *simulated annealing* dipilih secara *random* atau acak. Proses *simulated annealing* bertujuan agar proses atau persoalan yang akan diselesaikan tidak berhenti di titik minimum lokal, sehingga peluang untuk menemukan solusi persoalan menjadi lebih tinggi. Namun sama seperti *local search* lainnya, algoritma ini tidak menjamin untuk menemukan solusi, namun melakukan aproksimasi yang baik dalam waktu yang singkat dengan berangsur-angsur memperbaiki *state*.

Proses *simulated annealing* diawali dengan inisiasi *state* secara acak dari suatu persoalan. Pada tahap tersebut, *state* tidak harus mendekati solusi atau sebaliknya (dilakukan secara acak). Proses yang dilakukan setelah itu adalah membangkitkan solusi persoalan secara acak. Bila *state* yang dihasilkan lebih baik dibandingkan sebelumnya, *state* tersebut akan diterima dan *state* persoalan berubah menjadi *state* tersebut. Namun, bila *state* yang dihasilkan lebih buruk dibandingkan sebelumnya, *state* dapat diterima maupun ditolak, Kemungkinan (*P*) bahwa *state* diterima dihitung dengan metode faktor probabilitas Boltzmann:

P = exp (-(Et – Ei)/ T

Pada persamaan tersebut, *E* merupakan biaya atau nilai evaluasi dari solusi dan *T* merupakan suhu pada saat itu. *State* solusi dipastikan berpindah bila biaya pada *state* solusi acak (Et) lebih rendah dari biaya *state* sebelumnya (*Ei*). Setelah perpindahan *state* atau pengembalian ke *state* sebelumnya bila *state* baru tidak diterima, algortima akan kembali membangkitkan solusi tetangga secara acak dan mengulangi proses penentuan perpindahan *state*. Solusi ‘tetangga’ memilikik makna bahwa hanya ada satu hal yang membedakan solusi baru dengan solusi sebelumnya. Proses tersebut akan dilakukan secara berulang-ulang sampai pada batas tertentu, misalkan batas jumlah langkah yang dapat dilakukan, dengan nilai suhu yang terus menurun pada setiap langkah.

IMPLEMENTASI

Langkah dasar yang dilakukan oleh algoritma *simulated annealing* pada program adalah sabagai berikut:

1. Membangkitkan solusi acak

2. Memilih secara acak mata kuliah yang akan dipindahkan

3. Menempati mata kuliah tersebut secara acak pada slot yang diperbolehkan

3. Menghitung nilai evaluasi solusi baru dan lama

4. Bila nilai evaluasi solusi baru lebih kecil, maka pemindahan pasti dilakukan. Namun bila nilai evaluasi solusi baru lebih besar, pemindahan dilakukan dengan kemungkinan yang dihitung dengan menggunakan rumus Boltzmann

Pada algoritma ini, nilai evaluasi berisi jumlah slot pada jadwal yang mengandung jadwal yang bertabrakan.

Referensi:

Geltmann, Katrina Ellison. 2014. “The Simulated Annealing Algorithm”. <http://katrinaeg.com/simulated-annealing.html>. (diakses pada 26 September 2016)

Russel, Stuart, Peter Norvig. 2010. Artificial Intelligence A Modern Approach – Third Edition. Pearson Educatiom: New Jersey

Setiawan, Wahyudi. 2009. “Simulated Annealing”. https://wahyudisetiawan.wordpress.com/2009/12/22/simulated-annealing/ . (diakses pada 26 September 2016)

Soto, Francisco. 2012. “Simulated Annealing and the 8-Queen Problem”. <http://ebobby.org/2012/10/22/SimulatedAnnealingEightQueenProblem.html> (diakses pada 26 Septermber 2016)