

Optimasi Multi Travelling Salesman Problem (M-TSP) Menggunakan Algoritma Genetika

Wayan Firdaus Mahmudy
(wayanfm@ub.ac.id)

Program Studi Ilmu Komputer, Universitas Brawijaya, Malang, Indonesia

Abstrak. Masalah rute manakah yang memiliki biaya paling murah untuk dilalui n salesman ($n > 1$) yang harus mengunjungi sejumlah m daerah ($m > 1$), tiap daerah harus dikunjungi tepat satu kali kemudian kembali lagi ke tempat semula biasa disebut multi Travelling Salesman Problem (m-TSP). M-TSP merupakan masalah kombinatorial yang solusi optimumnya hanya bisa didapatkan dengan mencoba semua kemungkinan sehingga memerlukan waktu komputasi yang cukup tinggi dalam menyelesaikan masalah dengan nilai n dan m yang cukup besar. Makalah ini menjelaskan pengembangan metode heuristik menggunakan algoritma genetika untuk menyelesaikan m-TSP. Algoritma genetika yang dibangun menggunakan beberapa individu dan individu-individu ini terus berubah oleh proses mutasi dengan operator yang khusus didesain untuk m-TSP. Dengan arsitektur ini mekanisme seleksi individu dan tukar silang (crossover) tidak dilakukan sehingga iterasi ke generasi berikutnya berjalan sangat cepat. Evaluasi hasil dilakukan dengan cara membandingkan dengan solusi semua kemungkinan. Dari hasil ujicoba, algoritma genetika ini menghasilkan solusi yang sangat mendekati optimum. Pada data berukuran besar, algoritma genetika ini mempunyai waktu proses yang jauh lebih cepat dibanding dengan mencoba semua kemungkinan.

Kata kunci: m-TSP, algoritma genetika, mutasi, seleksi individu, tukar silang

1. Pendahuluan

Transportasi merupakan salah satu permasalahan yang sering dihadapi dalam kehidupan sehari-hari. Salah satu contoh yaitu rute manakah yang memiliki biaya paling murah untuk dilalui seorang salesman yang harus mengunjungi sejumlah daerah, tiap daerah harus dikunjungi tepat satu kali kemudian kembali lagi ke tempat semula. Permasalahan tersebut dikenal sebagai Travelling Salesman Problem (TSP). Jika terdapat lebih dari seorang salesman maka disebut multi Travelling Salesman Problem (m-TSP). Permasalahan TSP maupun m-TSP dimodelkan dalam bentuk graf lengkap. Pada kasus m-TSP graf lengkap tersebut memiliki n buah simpul yang menyatakan daerah-daerah yang harus dikunjungi oleh sejumlah m salesman. Bobot pada tiap garis pada graf tersebut menyatakan jarak tiap daerah [1]. Secara matematis m-TSP bisa diformulasikan dalam persamaan:

$$Z = \min \left\{ \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \right\} \quad (1)$$

dengan kendala

$$\sum_{i=1}^n x_{ij} = 1 \quad \text{untuk } j = 1, 2, 3, \dots, n-1 \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \text{untuk } i = 1, 2, 3, \dots, n-1 \quad (3)$$

$$\sum_{i=1}^n x_{i1} = m \quad (4)$$

$$\sum_{j=1}^n x_{1j} = m \quad (5)$$

$x_{ij} = 1$, apabila ada perjalanan salesman dari simpul i menuju simpul j

$x_{ij} = 0$, apabila tidak ada perjalanan salesman dari simpul i menuju simpul j

C_{ij} , menyatakan jarak dari simpul i menuju simpul j .

Persamaan (2) dan Persamaan(3) menjamin bahwa setiap simpul hanya dikunjungi sekali oleh salesman. Persamaan (4) dan Persamaan (5) menjamin bahwa sejumlah m salesman melakukan tur.

2. Algoritma Genetika

Pada pencarian solusi suatu masalah kadang-kadang dibutuhkan formulasi matematika yang kompleks untuk memberikan solusi yang pasti. Solusi optimum mungkin dapat diperoleh tetapi memerlukan proses perhitungan yang panjang dan tidak praktis. Untuk mengatasi kasus khusus seperti di atas dapat digunakan metode heuristik, yaitu suatu metode pencarian yang didasarkan atas intuisi atau aturan-aturan empiris untuk memperoleh solusi yang lebih baik daripada solusi yang telah dicapai sebelumnya [2]. Metode heuristik tidak selalu menghasilkan solusi terbaik tetapi jika dirancang dengan baik akan menghasilkan solusi yang mendekati optimum dalam waktu yang cepat. Algoritma genetika adalah salah satu cabang evolutionary algorithms, yaitu suatu teknik optimasi yang didasarkan pada genetika alami. Dalam algoritma genetika untuk menghasilkan suatu solusi optimal, proses pencarian dilakukan di antara sejumlah alternatif titik optimal berdasarkan fungsi probabilistik.

Apabila dibandingkan dengan prosedur pencarian dan optimasi biasa, algoritma genetika berbeda dalam beberapa hal sebagai berikut [3]:

- Manipulasi dilakukan terhadap kode dari himpunan parameter, tidak secara langsung terhadap parameternya sendiri.
- Proses pencarian dilakukan dari suatu titik populasi, tidak dari satu titik saja.
- Proses pencarian menggunakan informasi dari fungsi tujuan.
- Pencariannya menggunakan stochastic operators yang bersifat probabilistik, tidak menggunakan aturan deterministik.

Masalah utama pada algoritma genetika adalah bagaimana memetakan satu masalah menjadi satu string kromosom. Siklus perkembangan algoritma genetik diawali dengan pembuatan himpunan solusi baru (initialization) yang terdiri atas sejumlah string kromosom dan ditempatkan pada penampungan populasi. Kemudian dilakukan proses reproduksi dengan memilih individu-individu yang akan dikembangbiakkan. Penggunaan operator-operator genetik seperti pindah

silang (crossover) dan mutasi (mutation) terhadap individu-individu yang terpilih dalam penampungan individu akan menghasilkan keturunan atau generasi baru. Setelah proses evaluasi untuk perbaikan populasi, maka generasi-generasi baru ini akan menggantikan himpunan populasi asal. Siklus ini akan berlangsung berulang kali sampai tidak dihasilkan perbaikan keturunan, atau sampai kriteria optimum ditemukan.

Apabila $P(t)$ dan $C(t)$ merupakan *parents* dan *children* pada generasi ke- t , maka struktur umum algoritma genetika dapat dideskripsikan sebagai berikut:

```

procedure AlgoritmaGenetika
begin
   $t = 0$ 
  inialisasi  $P(t)$ 
  while (bukan kondisi berhenti) do
    evaluasi  $P(t)$ 
    seleksi  $P(t)$ 
    reproduksi  $C(t)$  dari  $P(t)$ 
    bentuk  $P(t+1)$  dari  $P(t)$  dan  $C(t)$  terbaik
     $t = t + 1$ 
  end while
end

```

Inisialisasi Populasi

Siklus perkembangan algoritma genetika diawali dengan pembuatan himpunan solusi baru (initialization) yang terdiri atas sejumlah string kromosom dan ditempatkan pada penampungan populasi. Populasi awal sebagai daerah pencarian solusi optimal dibangkitkan secara acak. Representasi kromosom diperlukan untuk menjelaskan setiap individu dalam populasi. Setiap individu atau kromosom tersusun atas urutan gen dari suatu alphabet. Suatu alfabet dapat terdiri dari digit biner (0 dan 1), floating point, integer, simbol-simbol (seperti A, B, C), matriks, dan lain sebagainya [4].

Pada makalah ini digunakan representasi kromosom bilangan bulat bertipe integer. Misalkan terdapat 9 daerah ($m=9$) dan 3 orang salesman ($n=3$) maka kromosom bisa digambarkan pada Gambar 1.

Posisi	1	2	3	4	5	6	7	8	9	10	11	12
Gen	5	7	6	1	8	4	9	3	2	2	4	3
	Bagian 1									Bagian 2		

Gambar 1. Representasi kromosom

Gen-gen pada bagian 1 (posisi 1 sampai 9) menunjukkan urutan daerah yang dikunjungi, bagian 2 (posisi 10 sampai 12) menunjukkan banyaknya daerah yang dikunjungi tiap salesman. Dari Gambar 1 dihasilkan rute tiap salesman sebagai berikut:

Saleman 1 : start → daerah 5 → daerah 7 → finish

Saleman 2 : start → daerah 6 → daerah 1 → daerah 8 → daerah 4 → finish

Saleman 3 : start → daerah 9 → daerah 3 → daerah 2 → finish

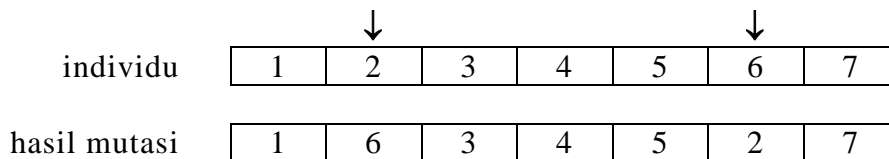
Evaluasi Individu

Hal penting dalam algoritma genetika adalah pemilihan individu / kromosom untuk menghasilkan keturunan berikutnya. Pemilihan dilakukan berdasarkan nilai kesesuaian (*fitness value*) setiap individu. Jika nilai Z pada Persamaan 1 dianggap sebagai biaya maka nilai *fitness* didefinisikan sebagai.

$$fitness = \frac{1}{biaya + 1} \quad (2)$$

Reproduksi

Individu baru terbentuk dari proses mutasi. Tukar silang (*crossover*) tidak dilakukan karena memerlukan waktu komputasi yang cukup besar dalam proses pemilihan induk dan perbaikan kromosom illegal [5]. Dalam makalah ini digunakan mutasi dengan 2 titik tukar pada kromosom bagian 1. Gambar 2 menunjukkan bagaimana mutasi dilakukan.



Gambar 2. Ilustrasi mutasi dengan 2 titik tukar

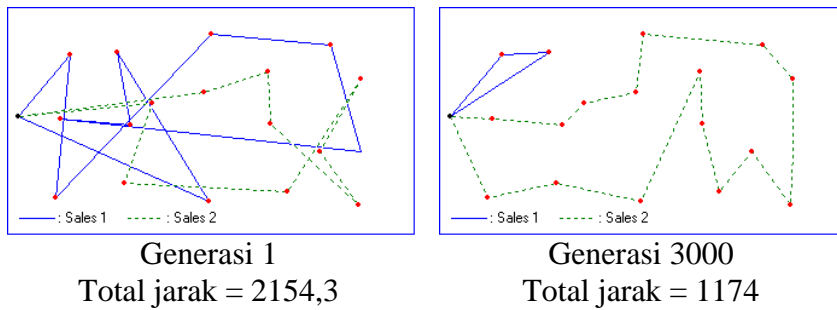
Mutasi pada kromosom bagian 2 dilakukan secara acak dengan menaikkan atau menurunkan sebesar satu. Setelah anak hasil mutasi dari induk terbentuk maka dihitung nilai *fitness*nya. Jika nilai *fitness* anak lebih besar dari nilai *fitness* induk maka anak dimasukkan ke populasi menggantikan induknya.

3. Metode

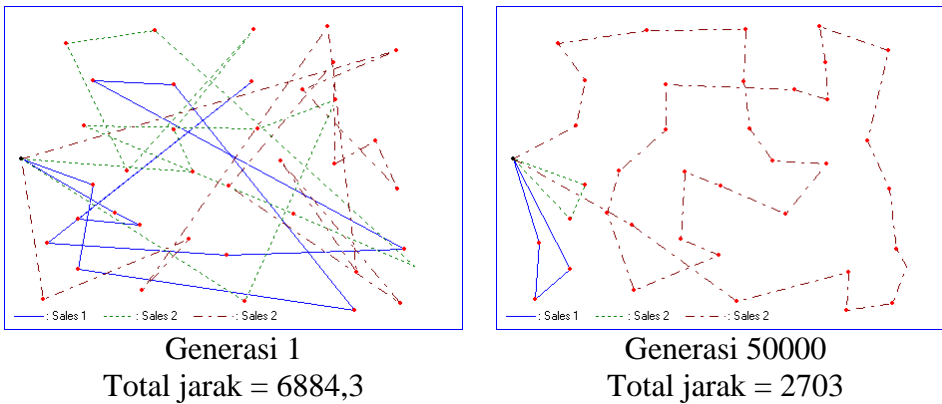
Untuk uji coba perangkat lunak yang dibangun berdasarkan algoritma genetika digunakan 4 buah data dalam bentuk citra yang masing-masing memuat 10, 20, 30, dan 40 daerah. Untuk tiap data diuji dengan menggunakan 2 dan 3 salesman. Data yang sudah tersedia dimasukkan ke dalam perangkat lunak. Perangkat lunak menghasilkan laporan akhir berupa rute perjalanan masing-masing salesman, total jarak tempuh dan waktu komputasi yang dibutuhkan. Evaluasi kehandalan algoritma genetika dilakukan dengan cara membandingkan dengan solusi mencari semua kemungkinan rute.

4. Hasil dan Pembahasan

Perangkat lunak dijalankan dengan ukuran populasi sebesar 10, iterasi berhenti jika dalam 10.000 generasi ke depan tidak ada perubahan *fitness*. Gambar 3 dan Gambar 4 menunjukkan perbaikan rute pada beberapa generasi.



Gambar 3. Hasil pada 20 daerah dengan 2 salesman



Gambar 4. Hasil pada 40 daerah dengan 3 salesman

Untuk mengetahui keoptimalan hasil algoritma genetika (AG), berikut ini diberikan tabel perbandingannya dengan hasil perhitungan pada semua kemungkinan jalur (SK) untuk data uji.

Tabel 1. Perbandingan Hasil Perhitungan

uji ke	daerah	salesman	total jarak (piksel)			waktu proses (detik)	
			AG	SK	Beda (%)	AG	SK
1	10	2	820	820	0	0,020	0,053
2	10	3	810	810	0	0,022	0,062
3	20	2	1174	1174	0	0,030	52,246
4	20	3	1162	1158	0,35	0,034	60,320
5	30	2	1687	1649	2,30	1,200	1506,20
6	30	3	1564	1495	4,62	1,320	2005,34
7	40	2	2690	2608	3,14	3,286	24452
8	40	3	2703	2587	4,49	4,027	28975

Pada data berukuran kecil AG menghasilkan solusi optimum. Semakin besar ukuran data maka terjadi penyimpangan hasil dari solusi optimum yang didapatkan dari pencarian semua

kemungkinan solusi. Semakin besar ukuran data maka penyelesaian dengan pencarian semua kemungkinan solusi tidak lagi bisa diterima dari waktu yang dibutuhkan. Pada uji ke-8 dibutuhkan waktu lebih dari 8 jam (28.975 detik) sementara algoritma genetika hanya membutuhkan waktu 4,027 detik dengan perbedaan hasil hanya 4,49%.

Penyelesaian dengan pencarian semua kemungkinan pada data berukuran besar membutuhkan waktu yang sangat lama. Jika ada n salesman yang harus mengunjungi sejumlah m daerah maka kombinasi solusi yang harus diperiksa adalah $m! \times n$.

5. Kesimpulan

Pencarian rute menggunakan algoritma genetika menghasilkan solusi optimum pada data berukuran kecil. Pada data berukuran besar, algoritma genetika menghasilkan solusi mendekati optimum dan waktu proses yang sangat cepat.

6. Daftar Pustaka

- [1] Setiyono, Budi. (2002), *Pembuatan Perangkat Lunak Penyelesaian Multi Travelling Salesman Problem (m-TSP)*. KAPPA, 3(2): 55-65.
- [2] Taha, H,A, (2002), *Operations Research- An Introduction, 6th ed*, Pearson Education Inc.
- [3] Michalewicz, Zbigniew. (1996), *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Heidelberg.
- [4] Houck, Christopher R, Jeffrey K dan Michael G Kay. (1999), *A Genetic Algorithms for Function Optimization: A Matlab Implementation*. http://www.dai.ed.ac.uk/groups/evalg/eag_local_copies_op_papers_body.html
- [5] Liliana, Dewi Yanti dan Mahmudy, Wayan Firdaus (2006), *Penerapan Algoritma Genetika pada Otomatisasi Penjadwalan Kuliah*, Laporan Penelitian DPP/SPP 2006, FMIPA Universitas Brawijaya, Malang.