

PENYELESAIAN MASALAH TRANSPORTASI MENGGUNAKAN ALGORITMA GENETIKA DENGAN INDIVIDU TUNGGAL

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

Jurusan Matematika, FMIPA
Universitas Brawijaya

ABSTRAK

Persoalan transportasi merupakan kasus khusus dari model pemrograman linier (*linear programming*). Dengan model matematis telah dikembangkan beberapa metode khusus untuk masalah transportasi yang secara umum disebut metode simplex transportasi (*transportation simplex method, TSM*). Dengan TSM, masalah transportasi bisa diselesaikan dengan akurat tetapi memerlukan waktu komputasi yang cukup tinggi dalam menyelesaikan masalah transportasi berukuran besar. Makalah ini menjelaskan pengembangan metode heuristik menggunakan algoritma genetika untuk menyelesaikan masalah transportasi. Algoritma genetika yang dibangun menggunakan satu individu dan individu ini terus berubah oleh proses mutasi. Dengan arsitektur ini mekanisme seleksi individu dan tukar silang (*crossover*) tidak dilakukan sehingga iterasi ke generasi berikutnya berjalan sangat cepat. 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 TSM.

1. PENDAHULUAN

Persoalan transportasi membahas pendistribusian suatu komoditas atau produk dari sejumlah sumber (*supply*) kepada sejumlah tujuan (*destination, demand*) dengan tujuan meminimumkan ongkos pengangkutan yang terjadi. Ciri-ciri khusus masalah ini adalah [8]:

- Terdapat sejumlah sumber dan sejumlah tujuan tertentu.
- Kuantitas komoditas atau barang yang didistribusikan dari setiap sumber dan yang diminta oleh setiap tujuan besarnya tertentu.
- Komoditas yang dikirim atau diangkut dari suatu sumber ke suatu tujuan besarnya sesuai dengan permintaan dan atau kapasitas sumber.
- Ongkos pengangkutan komoditas dari suatu sumber ke suatu tujuan besarnya tertentu.

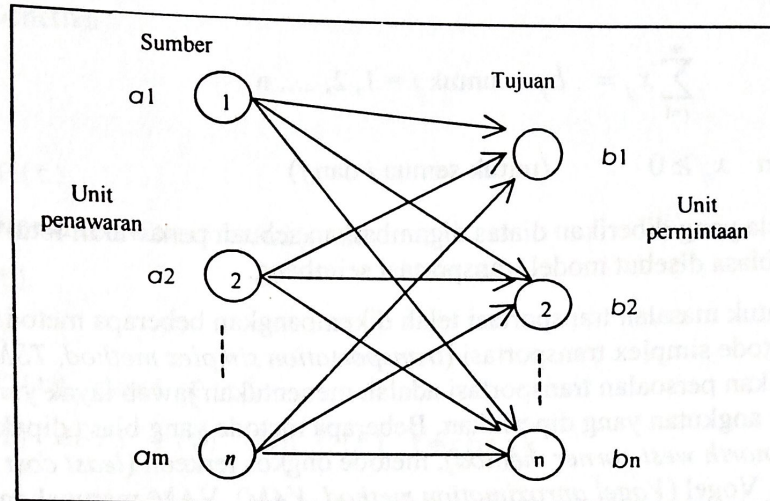
Persoalan transportasi merupakan kasus khusus dari model pemrograman linier (*linear programming*). Karakteristik khusus yang ada pada model khusus ini diantaranya ialah cenderung membutuhkan pembatas (*constrains*) dan variabel yang sangat banyak sehingga penggunaan komputer dalam penyelesaiannya dengan model matematis (misalnya dengan metode simplex) akan sangat mahal atau proses perhitungannya sangat panjang dan tidak praktis [7]. Untuk masalah transportasi telah dikembangkan beberapa metode khusus yang secara umum disebut metode simplex transportasi (*transportation simplex method, TSM*). Dengan TSM, masalah transportasi bisa diselesaikan dengan akurat. Tetapi metode ini memerlukan waktu komputasi yang cukup tinggi dalam menyelesaikan masalah transportasi berukuran besar. [9].

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 [8]. Metode heuristik yang bisa diterapkan pada masalah optimasi misalnya metode *hillclimbing*, algoritma *simulated annealing* dan algoritma genetika [3,6,11]. Algoritma genetika merupakan salah satu metode heuristik yang merupakan cabang dari *evolutionary algorithm*, yaitu suatu teknik untuk memecahkan masalah-masalah optimasi yang rumit dengan menirukan proses evolusi makhluk hidup. Algoritma genetika

berkembang seiring dengan perkembangan teknologi informasi yang sangat pesat. Algoritma ini banyak digunakan dalam bidang fisika, biologi, ekonomi, sosiologi dan lain-lain yang sering menghadapi masalah optimasi yang model matematikanya kompleks atau bahkan sulit dibangun [5].

1.1. Model Transportasi

Model transportasi sederhana dari sebuah jaringan dengan m sumber dan n tujuan digambarkan sebagai berikut [7]:



Gambar 1. Model transportasi

Dari gambar diatas, maka secara umum sumber i ($i = 1, 2, \dots, m$) mempunyai suplai a_i unit untuk didistribusikan ke tujuan-tujuan, dan tujuan j ($j = 1, 2, \dots, n$) mempunyai permintaan b_j unit untuk diterima dari sumber-sumber. Dari gambar model transportasi di atas, seluruh informasi yang ada dapat disimpan dalam sebuah tabel seperti di bawah ini:

Tabel 1. Tabel biaya dan kebutuhan transportasi

tujuan \ sumber	1	2	n	suplai
1	c_{11} x_{11}	c_{12} x_{12}	c_{1n} x_{1n}	a_1
2	c_{21} x_{21}	c_{22} x_{22}	c_{2n} x_{2n}	a_2
...
M	c_{31} x_{31}	c_{32} x_{32}	c_{3n} x_{3n}	a_m
Permintaan	b_1	b_2	b_n	

Variabel c_{ij} ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$) adalah biaya pendistribusian yang dibutuhkan dari sumber i ke tujuan j dan variabel x_{ij} ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$) adalah jumlah produk yang harus dikirimkan dari sumber i ke tujuan j .

Secara matematis, model penugasan dapat dinyatakan sebagai berikut:

$$\text{Minimumkan } Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (1)$$

Dengan kendala (*constraint*)

$$\sum_{j=1}^n x_{ij} = a_i \quad \text{untuk } i = 1, 2, \dots, m$$

$$\sum_{i=1}^m x_{ij} = b_j \quad \text{untuk } j = 1, 2, \dots, n$$

dan $x_{ij} \geq 0$ (untuk semua i dan j)

Dari kendala yang diberikan diatas digambarkan sebuah penawaran total sama dengan permintaan total yang biasa disebut model transportasi seimbang.

Untuk masalah transportasi telah dikembangkan beberapa metode khusus yang secara umum disebut metode simplex transportasi (*transportation simplex method, TSM*). Langkah pertama dalam menyelesaikan persoalan transportasi adalah menentukan jawab layak yang memenuhi semua kendala atau sistem angkutan yang diperlukan. Beberapa metode yang biasa dipakai adalah metode pojok barat-laut (*north west corner method*), metode ongkos terkecil (*least cost method*) dan metode pendekatan Vogel (*Vogel approximation method, VAM*). VAM merupakan metode terbaik karena seringkali menghasilkan solusi yang mendekati optimum [1].

Langkah kedua adalah untuk mencari pemecahan optimum. Metode yang biasa digunakan adalah metode batu loncatan (*stepping stone*), metode *multiplier* dan metode *modified distribution (MODI)* [8]. Ketiga metode tersebut akan menghasilkan solusi optimum yang sama. Pada makalah ini metode yang dibahas adalah metode batu loncatan. Langkah utama dalam metode ini adalah mencari *loop* tertutup untuk menentukan *entering variable* terbaik yang bisa mengurangi total biaya terbesar.

1.2. 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 [8]. 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 [2]:

- 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 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 Algoritma Genetika

begin

$t = 0$

inisialisasi $P(t)$

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

1.3. 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 alfabet. 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 penelitian ini digunakan populasi yang hanya berisi satu individu. Penggunaan satu individu ini akan menghilangkan proses seleksi induk yang digunakan pada proses reproduksi (*crossover*). Kedua proses ini membutuhkan waktu komputasi yang relatif besar [10].

Pada masalah transportasi misalkan terdapat 3 sumber dan 4 tujuan. Persediaan di sumber adalah 15, 25 dan 5. Permintaan di tujuan adalah 5, 15, 15 dan 10. Matriks biaya sebagai berikut:

10 0 20 11

12 7 9 20

0 14 16 18

Dengan algoritma Vogel didapatkan solusi awal sebagai berikut:

0 15 0 0 | 15

0 0 15 10 | 25

5 0 0 0 | 5

5 15 15 10 | 45

Cost = 335

Solusi awal dari algoritma vogel di atas dijadikan dasar pembentukan matriks kromosom sebagai berikut:

0	15	0	0
0	0	15	10
5	0	0	0

Gambar 2. Representasi individu (chromosome)

Angka yang terdapat pada Gambar 1 adalah alokasi barang yang dikirim dari sumber pada baris r ke tujuan pada kolom c .

1.4. Evaluasi Individu

Hal penting dalam algoritma genetika adalah pemilihan individu / kromosom untuk menghasilkan keturunan berikutnya. Pemilihan dilakukan berdasarkan nilai kesesuaian (*fitness value*) setiap individu. Karena pada penelitian ini hanya digunakan satu individu, maka proses yang dilakukan hanya menghitung nilai biaya (*cost*) dari chromosome menggunakan Persamaan 1.

1.5. Reproduksi

Individu baru terbentuk dari proses mutasi. Dalam penelitian ini digunakan mutasi dengan 4 dan 6 titik tukar. Gambar 2 di bawah ini menunjukkan bagaimana mutasi dilakukan.

Individu Asal

0	15	0	0
0	0	15	10
5	0	0	0

Hasil Mutasi

5	10	0	0
0	0	15	10
5	5	0	0

(a) 4 titik tukar

Individu Asal

0	15	0	0
0	0	15	10
5	0	0	0

Hasil Mutasi

5	10	0	0
0	5	10	10
0	0	5	0

(b) 6 titik tukar

Gambar 3. Ilustrasi mutasi

Pada Gambar 3, dipilih 4 atau 6 titik secara acak sehingga membentuk loop tertutup. Alokasi barang pada tiap sudut loop diubah sedemikian rupa sehingga total tiap baris dan total tiap kolom tidak berubah. Individu hasil mutasi dievaluasi dengan menghitung biaya perpindahan barang. Jika total biaya yang didapatkan lebih kecil maka individu hasil mutasi ini akan disimpan menggantikan individu sebelumnya. Proses ini diulang sampai tidak didapatkan individu yang lebih baik setelah 1000 iterasi..

2. METODE PENELITIAN

Pada penelitian ini digunakan 5 buah data berukuran 3x4 untuk mewakili data berukuran kecil, 5 buah data berukuran 20x20 untuk mewakili data berukuran sedang dan 5 buah data berukuran 40x40 untuk mewakili data berukuran besar. Tiap data masalah transportasi diselesaikan dengan TSM dan algoritma genetika. Untuk penentuan solusi awal digunakan metode vogel. Data yang sudah tersedia dimasukkan ke dalam perangkat lunak yang telah dibangun. Perangkat lunak ini menghasilkan laporan akhir berupa matriks alokasi, total biaya dan waktu yang dibutuhkan.

3. HASIL DAN PEMBAHASAN

Uji coba program dilakukan menggunakan komputer dengan prosesor Pentium II dan memori 256 kb. Berikut ini hasil ujicoba perangkat lunak penyelesaian masalah transportasi dengan TSM dan algoritma genetika (GA).

Tabel 2. Hasil Ujicoba Pada Data Berukuran Kecil (3x4)

Data	Biaya Solusi Awal	Solusi (total biaya)			Waktu Proses (detik)	
		TSM	GA	Perbedaan	TSM	GA
1	335	315	315	0 %	0,0002	0,0000
2	550	400	400	0 %	0,0002	0,0000
3	320	300	300	0 %	0,0002	0,0000
4	390	360	360	0 %	0,0001	0,0000
5	550	520	520	0 %	0,0001	0,0000

Tabel 3. Hasil Ujicoba Pada Data Berukuran Sedang (20x20)

Data	Biaya Solusi Awal	Solusi (total biaya)			Waktu Proses (detik)	
		TSM	GA	Perbedaan	TSM	GA
1	7.850	7.350	7.350	0,00 %	3,65	0,87
2	9.230	8.900	8.910	0,11 %	2,70	0,90
3	7.620	6.580	6.585	0,07 %	3,30	0,97
4	6.925	7.654	7.654	0,00 %	2,67	1,04
5	9.582	9.309	9.309	0,00 %	3,46	0,79

Tabel 4. Hasil Ujicoba Pada Data Berukuran Besar (40x40)

Data	Biaya Solusi Awal	Solusi (total biaya)			Waktu Proses (detik)	
		TSM	GA	Perbedaan	TSM	GA
1	88.740	85.800	85.910	0,12 %	26,31	2,71
2	53.230	49.640	49.895	0,51 %	29,76	3,12
3	78.710	75.583	75.860	0,37 %	28,62	2,15
4	87.430	85.658	86.102	0,52 %	23,45	3,46
5	91.620	89.863	90.320	0,51 %	28,07	3,47

Pada data berukuran kecil TSM dan algoritma genetika menghasilkan solusi yang sama. Waktu pemrosesan yang dibutuhkan keduanya sangat kecil dan tidak ada perbedaan yang nyata. Pada data berukuran sedang, algoritma genetika hanya menghasilkan solusi mendekati optimum, tetapi perbedaannya sangat kecil (semuanya di bawah 1%). Pada data berukuran sedang ini terdapat perbedaan waktu proses yang sangat nyata antara TSM dan algoritma genetika. Pada data berukuran besar algoritma genetika masih mampu untuk memberikan hasil yang mendekati optimum dengan waktu proses yang relatif cepat.

Pada data berukuran besar waktu yang dibutuhkan TSM akan naik secara drastis. Hal ini dikarenakan semakin banyaknya kombinasi *loop* berbentuk *polyhedron* yang harus diperiksa. Pada algoritma genetika hal ini tidak terjadi karena semua *loop* berbentuk segiempat atau segienam.

4. KESIMPULAN

Pada data berukuran kecil, algoritma genetika mampu memberikan solusi optimum. Pada data berukuran besar (20x20) algoritma genetika mampu memberikan solusi yang sangat mendekati optimum dalam waktu yang sangat cepat.

5. DAFTAR PUSTAKA

1. Siagian, P, (1987), *Penelitian Operasional, Teori dan Praktek*, UI Press, Jakarta.
2. Michalewicz, Zbigniew. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Heidelberg.
3. Gen, Mitsuo dan Cheng, Runwei (1997), *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York.
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. Rennard, JP. (2000). *Genetic Algorithms Viewer*. www.rennard.org/alife.
6. Nallamottu, U,B, Chambers, T,L, and Simon, W,E (2002), *Comparison of the Genetic Algorithm to Simulated Annealing Algorithm in Solving Transportation Location-allocation Problems With Euclidean Distances*, Proceedings of the 2002 ASEE Gulf-Southwest Annual Conference, The University of Louisiana at Lafayette, March 20 – 22, 2002,
7. Taha, H,A, (2002), *Operations Research- An Introduction*, 6th ed, Pearson Education Inc.
8. Dimiyati, T,T, dan Dimiyati, A, (2003), *Operations Research, Model-Model Pengambilan Keputusan*, Sinar Baru Algensindo, Bandung,
9. Kumar, T, (2004), *Comparison of Optimization Techniques In Large Scale Transportation Problems*, <http://www.mnsu.edu/research/URC/OnlinePublications/URC2004Articles/Kumar.pdf>, tanggal akses: 3 Januari 2007.
10. Liliana, Dewi Yanti dan Mahmudy, Wayan Firdaus (2006), *Penerapan Algoritma Genetika pada Otomatisasi Penjadwalan Kuliah*, Laporan Penelitian DPP/SPP 2006, FMIPA Universitas Brawijaya, Malang.
11. Mahmudy, Wayan Firdaus (2006), *Penerapan Algoritma Genetika pada Optimasi Model Penugasan*, Jurnal Natural, Vol 10 No 3 : 197-207, FMIPA Universitas Brawijaya, Malang.