

Fitness sharing Application For diversity Control with Evolutionary Algorithm To Resolve Travelling Salesman problem (TSP)

Feri susilawati¹, Taufik A. Gani², Yuwaldi Away³

^{1,2}Magister Teknik Elektro, Teknik Elektro, ³Universitas Syah Kuala, Banda Aceh 23111
feri@politeknikaceh.ac.id, topgan@unsyiah.ac.id, yuwaldi@unsyiah.ac.id

ABSTRACT

To send a messages a courier (salesman) Cost much time and money. Thus it become a problem to optimizing in searching the shortest route that called as Traveling Salesman's Problem (TSP).which a salesman should delivering letters to many places, each place must be visited in single time and then back to the starting place. The optimizing problem that should be reached is the shorter route and the minimum cost. In this research, Fitness Sharing with Evolutionary Algorithm (EA) proposed to minimize the route and individual convergent (diversity) in the population of each generation, before reaching the optimum point. In the result it will produce a computation time with the minimum route.

Keywords : Travelling Salesman Problem (TSP), Evolutionary Algorithm, Fitness Sharing, Diversity

ABSTRAK

Untuk mengantarkan surat seorang kurir (salesman) membutuhkan banyak waktu dan biaya. Sehingga hal tersebut menjadi persoalan optimasi dalam mencari rute terpendek yang disebut sebagai Traveling Salemans Problem (TSP). Dimana seorang salesman harus mengantar surat kesejumlah daerah, tiap daerah harus dikunjungi tepat satu kali kemudian kembali lagi ke daerah asal. Persoalan optimasi yang ingin dicapai ialah rute yang dilalui dan biaya yang digunakan paling minimum. Pada penelitian ini Fitness Sharing dengan Evolutionary Algorithm (EA) diusulkan untuk meminimalkan rute dan kekonvergenan individu (diversity) dalam populasi pada setiap generasi, sebelum mencapai titik optimum. Sehingga dapat menghasilkan waktu kumputasi dengan rute minimum.

Kata kunci : Travelling Salesman Problem (TSP), Evolutionary Algorithm, Fitness Sharing, Diversity

I. PENDAHULUAN

TSP dikenal sebagai salah satu masalah *optimasi* yang banyak menarik perhatian para peneliti sejak beberapa dekade terdahulu, karena alternatif atau kombinasi rute perjalanan yang banyak sekali, maka permasalahan TSP termasuk ke dalam persoalan yang sangat sulit diselesaikan dan membutuhkan waktu yang sangat lama untuk jumlah kota yang besar, dimana dalam penyelesaian TSP

rute terpendek dan biaya yang digunakan paling minimum merupakan solusi yang paling optimal. Diantara permasalahan yang dapat direpresentasikan dengan TSP adalah rute pengiriman barang atau surat, transportasi, rute pengisian uang pada ATM, rute penjadwalan mesin-mesin produksi, dan lain-lain. Tipe permasalahan TSP juga banyak muncul di beberapa persoalan aplikasi teknik, hal ini dapat diselesaikan dengan metode-metode yang

dikembangkan dan dapat efisien untuk penyelesaian TSP.

Salah satu metode yang digunakan untuk menyelesaikan masalah TSP adalah EA. Algoritma ini dikembangkan berdasarkan proses genetik makhluk hidup melalui perkawinan alami. Perkawinan ini ditambah dengan faktor alam lainnya dan menyebabkan spesies mengalami evolusi secara terus menerus untuk menyesuaikan diri dengan lingkungan hidupnya. Hanya individu yang kuat yang mampu bertahan, sehingga dalam tahapan proses evolusi akan diperoleh individu yang terbaik. Prinsip ini diambil dari teori Darwin “ *survivals of the strongest* “ yaitu dimana yang kuatlah yang akan bertahan ke tahap evolusi selanjutnya. [1]

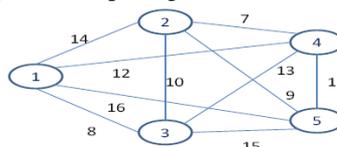
Kelemahan EAhanya menghasilkan solusi optimal pada tingkat *local*. Kelemahan tersebut dapat diperbaiki dengan selalu menjaga keberagaman individu dalam populasi. Ada banyak cara yang dapat dilakukan dalam penjagaan *diversity* ini. Salah satunya metode *fitness sharing* yang digunakan untuk menjaga agar resiko dari hasil *mating* dalam menghasilkan *individu* baru tidak terjadi cacat data atau proses perkawinan yang dihasilkan tidak sempurna.

fitness sharing adalah melakukan pemilihan induk yang memiliki kemiripan (kesamaan) antara satu sama lain. Setiap induk yang memiliki kesamaan akan membagi nilai *fitness*. Sementara induk yang tidak memiliki kemiripan antara satu dengan yang lain, tetap akan mempertahankan nilai *fitness* yang mereka dapatkan[2].

Pada jurnal ini melakukan pengujian pada permasalahan karakteristik dari komponen algoritma berevolusi pada *fitness sharing*, yang berguna untuk menjaga nilai *deversity* dan untuk mendapatkan solusi optimal.

II. METODE PENELITIAN

Persoalan Traveling Salesman Problem merupakan salah satu persoalan kombinatorial. Banyak permasalahan yang dapat direpresentasikan dalam bentuk TSP. Persoalan ini sendiri menggunakan representasi *graf* untuk memodelkan persoalan yang diwakili, sehingga lebih memudahkan penyelesaiannya. Diantara permasalahan yang dapat direpresentasikan dengan TSP ialah masalah transportasi, efisiensi pengiriman surat atau barang, perancangan pemasangan pipa saluran, proses pembuatan PCB (*Printed Circuit Board*) dan lain-lain. Persoalan yang muncul ialah bagaimana cara mengunjungi simpul (*node*) pada *graf* dari titik awal ke setiap titik-titik lainnya dengan bobot minimum. Bobot ini sendiri dapat mewakili berbagai hal, seperti biaya, jarak, bahan bakar, waktu, dan lainnya [4]. Gambar sederhana TSP dapat dilihat pada gambar 1



Gambar 1. Posisi Kota-Kota Yang Akan Dilewati[5]

Untuk mencari nilai minimum dengan menggunakan rumus jarak kartesian antara kota A dan kota B dimana koordinat kota A (x, y) dan mempunyai koordinat kota B (x, y). Sehingga jarak antar kedua kota dapat dihitung dengan rumus [6].

$$|A - B| = \sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2}$$

Algoritma Evolusi (*evolutionary algorithms*, EAs) merupakan teknik optimasi yang meniru proses evolusi biologi. Menurut teori evolusi terdapat sejumlah individu dalam populasi. Dari generasi ke generasi, individu-individu ini berperan sebagai induk (*parent*) yang melakukan reproduksi

menghasilkan keturunan (*offspring*). Individu-individu ini (beserta *offspring*) berevolusi dan individu-individu yang lebih baik (mampu beradaptasi dengan lingkungannya) mempunyai peluang lebih besar untuk melewati seleksi alam (*natural selection*) dan bertahan hidup. Individu yang lebih baik juga cenderung (tidak selalu tapi mempunyai kemungkinan lebih besar) menghasilkan keturunan yang lebih baik sehingga dari generasi ke generasi akan terbentuk populasi yang lebih baik.[7]

EAs dapat membangkitkan populasi awal secara random. Proses yang dilewati dalam EAs adalah seleksi, crossover, dan mutasi. Proses ini dilakukan berulang-ulang sehingga didapatkan jumlah kromosom yang cukup untuk membentuk generasi baru, dimana generasi baru ini merupakan representasi dari solusi baru.

Kromosom-kromosom berevolusi dalam suatu proses iterasi yang berkelanjutan yang disebut generasi. Pada setiap generasi, kromosom dievaluasi berdasarkan suatu fungsi evaluasi sampai didapatkan suatu individu yang dianggap optimal.[1]

fitness sharing adalah membagi suatu populasi individu berdasarkan letaknya didalam search space. Bila letaknya di dalam *search space* lebih dekat dengan *crowding*, maka nilai *fitness*-nya akan dikurangi, sebaliknya jika individu tersebut letaknya berjauhan dengan individu yang lain di sekitarnya, maka nilai fitnessnya akan ditambah. Hal ini dilakukan sampai terpenuhi kriteria berhenti, bila kriteria berhenti belum terpenuhi maka generasi baru akan terus dibentuk. Beberapa kriteria berhenti yang sering digunakan antara lain: Berhenti pada generasi tertentu, berhenti jika dalam beberapa generasi berturut-turut didapatkan nilai fitnessnya tidak berubah,

berhenti bila dalam n generasi berikut tidak didapatkan nilai *fitness* yang lebih tinggi.[8]

Diversity dinyatakan sebagai nilai keanekaragaman dari populasi, dimana Setiap kromosom dalam populasi dibandingkan dan dihitung rasio kemiripannya. Nilai keragaman didapatkan dengan mengurangi 1 dengan rata-rata rasio kemiripan[5]. Fungsi dari *diversity* adalah untuk menjaga perbedaan antar individu di dalam sebuah kelompok populasi. Dimana proses penjagaan tersebut akan berpengaruh pada hasil eksekusi algoritma berevolusi agar tidak cepat terperangkap dalam ruang lingkup yang sama dari sebuah kelompok individu [9].

Tabel 1: Data Set TSPLIB95

No	DataSet	kota	Edge Type
1	eil51	51	EUC_2D
2	pr76	76	EUC_2D
3	d198	198	EUC_2D
4	kroA100	100	EUC_2D
5	kroB100	100	EUC_2D
6	kroC100	100	EUC_2D
7	kroD100	100	EUC_2D
8	kroE100	100	EUC_2D
9	rd100	100	EUC_2D
10	eil101	101	EUC_2D
11	lin105	105	EUC_2D
12	ch130_	130	EUC_2D
13	ch150	150	EUC_2D
14	kroA150	150	EUC_2D
15	kroB150	150	EUC_2D

Data Pengujian menggunakan data file . tsp dengan data TSP simetri yang hanya mendukung tipe *EDGE_WEIGHT_TYPE* : EUC_2D yaitu koordinat posisi dengan format *Euclidian* 2 dimensi dan menggunakan dataset SPLib yang didownload [10] dengan parameter yang diuji adalah jarak antara satu kota dengan kota lain nya. TSPLIB adalah perpustakaan

contoh sampel untuk TSP dari berbagai sumber dan berbagai jenis [10]. Adapun data yang digunakan pada penelitian ini dapat dilihat pada tabel 1.

Beberapa Tahapan dalam Pengembangan Simulasi adalah sebagai berikut :

1. Memahami perangkat lunak atau system yang akan digunakan
2. Mengembangkan sistem yang sudah ada untuk disimulasikan
3. Mengembangkan model matematika dari system
4. Mengembangkan model matematika untuk simulasi
5. Membuat program (*software*) komputer
6. Menguji, memverifikasi dan memvalidasi keluaran simulasi
7. Mengeksekusi program simulasi untuk tujuan yang diharapkan

1. RANCANGAN ALGORITMA

Beberapa tahapan yang dilewati dalam pembuatan program berevolusi ini, di antaranya:

Inisialisasi Populasi

Dilakukan untuk menentukan ukuran populasi yakni melakukan penentuan nilai awal, Bagian penentuan nilai awal ini merupakan input yang dilakukan oleh pengguna sendiri. Input-input yang diperlukan dalam algoritma berevolusi pada penelitian

ini meliputi. Penentuan banyak generasi yang akan dilakukan (*Generation Number=100*). Penentuan banyaknya node dalam setiap kromosom (*OffspringSize = 80*. Penentuan besar populasi dalam satu generasi (*population Size=20*). Jumlah gen dalam setiap kromosom sama dengan jumlah node. Node awal (*DataposisiX*) dan tujuan (*DataposisiY*).

Evaluasi kromosom

Proses ini akan menghitung nilai *fitness* dari setiap kromosom yang telah dibangkitkan secara *random* pada tahap inisialisasi populasi. Nilai *fitness* dari setiap kromosom dihitung berdasarkan panjang jalur *linier* yang dihasilkan dari jumlah jarak keseluruhan dari urutan node-node yang dilalui. Dimana kromosom yang bernilai *fitness* tinggi yang akan bertahan hidup atau yang akan terpilih. Nilai *fitness* yang dicari adalah kromosom yang memiliki panjang jalur yang pendek.

Mencari nilai *cost function* yang lebih minimum dengan menggunakan rumus jarak dari *node* satu ke *node* (antara kota A dan kota B).

$$|A - B| = \sqrt{(X_A - X_B)^2 + (Y_A - Y_B)^2}$$

Dimana (X_A, Y_A) menyatakan posisi koordinat kota A dan (X_B, Y_B) menyatakan posisi koordinat kota B.

Tahap evaluasi kromosom dilakukan berulang sebanyak *Pop Size*, sehingga didapat nilai *fitness* dari semua kromosom dalam satu populasi. Nilai *fitness* suatu kromosom ini kemudian akan dibandingkan dengan *fitness-fitness* kromosom yang lainnya yang ada pada semua generasi. Dimana nilai *fitness* paling tinggi yang akan terpilih.

Seleksi kromosom

Seleksi kromosom yang berfungsi memilih individu-individu yang akan dijadikan induk pada proses rekombinasi ataupun yang akan tetap bertahan pada generasi berikutnya. Metode yang digunakan pada penelitian ini adalah metode *roulette wheel*[11]

Fitness Sharing

Fitness sharing menurunkan *fitness* dari tiap-tiap anggota populasi dengan jumlah yang hampir sama dengan banyaknya

individu yang memiliki kemiripan dalam populasi

$Fitness f'_i$ yang dibagi dari suatu individu i dengan $fitness f_i$ secara sederhana dapat ditulis :

$$f'_i = \frac{f_i}{m_i}$$

Dimana m_i adalah hitungan *nich* (kedudukan yang sesuai) untuk mengukur jumlah jarak yang berdekatan dari individu – individu dengan siapa $fitness f_i$ dibagi. Hitungan *niche* dihitung dengan menjumlahkan suatu fungsi *sharing* semua anggota populasi.[8]

Crossover

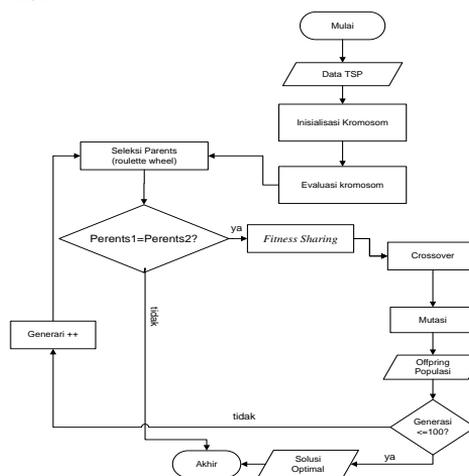
Menyilangkan dua kromosom sehingga membentuk kromosom baru yang harapannya lebih baik dari pada kromosom sebelumnya. Tidak semua kromosom pada suatu himpunan akan mengalami proses rekombinasi. Kemungkinan suatu kromosom akan mengalami proses pindah silang didasarkan pada probabilitas pindah silang yang telah ditemukan terlebih dahulu. Probabilitas crossover dapat menyatakan banyaknya kromosom yang akan dilakukan *crossover*.[11]

Mutasi

Proses mutasi atau pertukaran titik dilakukan setelah proses rekombinasi atau *crossover* dengan cara memilih kromosom yang akan ditukar secara acak kemudian menentukan titik pertukaran pada kromosom tersebut. Banyaknya gen yang akan mengalami pertukaran dihitung berdasarkan probabilitas mutasi yang telah ditentukan terlebih dahulu. Apabila probabilitas mutasi adalah 100% maka semua gen yang ada pada himpunan tersebut akan mengalami pertukaran. Sebaliknya, jika probabilitas pertukaran yang digunakan adalah 0% maka tidak ada gen yang mengalami pertukaran.

Populasi Baru

Dari beberapa proses yang dilewati yaitu inialisasi *kromosom*, *evaluasi kromosom*, *fitness sharing*, *crossover* dan *mutasi* terbentuklah populasi baru dengan nilai fitness yang lebih dan dapat menjadi solusi dalam penyelesaian permasalahan TSP.



Gambar 1. Flowchart Prosedur EA pada FS

Langkah-langkah penyelesaian EA dalam permasalahan TSP dengan penerapan fitness sharing.

1. Penentuan banyak generasi yang akan dilakukan (Generation Number=100)
2. Penentuan banyaknya node dalam setiap kromosom (OffspringSize = 80)
3. Penentuan besar populasi dalam satu generasi (populationSize=20)
4. Jumlah gen dalam setiap kromosom sama dengan jumlah node.
5. Node awal (Dataposisi X) dan tujuan (Dataposisi Y)
6. Masukkan generasi pertama, kemudian jalankan sampai 100 generasi dengan mengulang sebanyak 15 kali pengujian.
7. Lakukan langkah 1 sampai dengan langkah 5 untuk mendapatkan hasil dari semua dataset yang diuji.

III. HASIL DAN PEMBAHASAN

Setelah dilakukan komputasi algoritma pada setiap dataset. selanjutnya diambil nilai minimum jarak, maximum jarak, rata-rata jarak dan nilai diversity, yang didapatkan dari setiap output algoritma untuk dimasukkan kedalam tabel.

Tabel 2 Hasil *Fitness Sharing* dengan AE

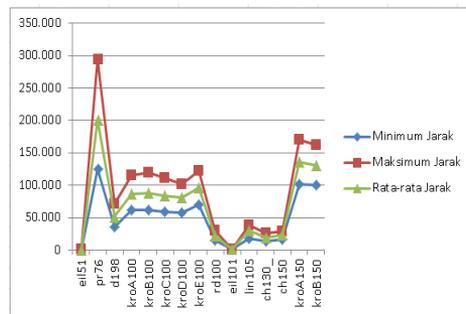
No	Dat aset	Minim um Jarak	Maksim um Jarak	Rata-rata Jarak	Diver sity
1	eil51	351,00	961,00	646,10	64,12
2	pr76	125303,00	294667,00	201114,20	76,25
3	d198	35313,00	72096,00	51622,30	83,46
4	kro A100	61756,00	115384,00	87104,80	76,95
5	kroB100	62571,00	119902,00	88642,70	79,70
6	kroC100	58878,00	111576,00	84292,80	79,00
7	kro D100	57920,00	102311,00	80628,20	77,40
8	kroE100	69540,00	122605,00	95594,00	80,60
9	rd100	15428,00	29927,00	22482,60	79,25
10	eil101	955,00	1824,00	1355,20	79,90
11	lin105	18256,00	39196,00	29710,40	78,67
12	ch130_	13222,00	25658,00	18984,70	81,77
13	ch150	16673,00	29388,00	22921,30	83,30
14	kro A150	101805,00	170378,00	136717,10	82,50
15	kroB150	99776,00	162032,00	130008,70	81,87
Rata-rata		49.183	93.194	70.122	78,98

Pada Tabel 2 dibawah ini merupakan hasil dari komputasi algoritma *fitness sharing* dengan *Algorithm Evolutionary* dalam permasalahan TSP.

Tabel diatas menghasilkan populasi solusi dan standar deviasi dari *fitness sharing* dengan melakukan simulasi pada 15 dataset, setiap dataset diuji sebanyak 5 kali. Hasil yang diperoleh kemudian diolah menggunakan *software excel 2010* dan *IBM Statistic 23*.

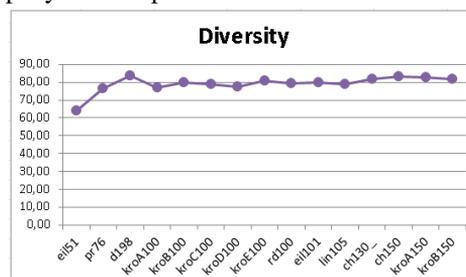
Nilai minimum, maksimum, diversity dan rata-rata jarak didapatkan dari hasil proses seleksi, crossover dan mutasi

Generasi yang menjadi output pada tabel adalah generasi ke-100, dengan *offspring size* 80 dan Besarnya populasi yang diujia dalam satu generasi adalah 20 (*population size*).



Gambar 2. Grafik *minimum jarak*

Berdasarkan Gambar 2 dapat dijelaskan bahwa semakin kecil nilai minimum yang diperoleh maka akan semakin cepat proses untuk mencapai titik optimum dalam penyelesaian permasalahan TSP.



Gambar 3. Grafik *Diversity*

Nilai diversity yang dihasilkan adalah nilai pada semua individu saat proses AE berlangsung yaitu proses sebelum

tercapainya titik optimum. Hal ini dilakukan untuk menghasilkan individu-individu yang bervariasi dalam TSP.

Jika nilai *fitness* sudah mendekati nilai optimum dimana rata-rata *diversity* sudah diatas 50% dan hasil ini menunjukkan hampir semua generasi sudah tidak memiliki *konvergen* atau sudah tidak ada kemiripan. Sehingga algoritma yang digunakan dapat bekerja dengan baik dalam menyelesaikan permasalahan TSP.

Pada percobaan ini melakukan pengujian dengan data yang diuji, mulai dari node terbesar, sedang, dan kecil. Hasil *Diversity* pada grafik diatas menghasilkan output yang sudah sangat aman dan berkualitas. Dimana setiap *diversity* dari dataset yang diuji sudah diatas 50%. Hal ini sangat baik dalam menjaga keragaman individu dalam populasi. sehingga dapat dikatakan hasil yang didapatkan dapat menghasilkan individu yang lebih bervariasi.

IV. KESIMPULAN DAN SARAN

1. Hasil dari pengujian yang sudah dilakukan pada AE dengan menggunakan *fitness sharing* menghasilkan solusi dalam menyelesaikan permasalahan TSP dalam menjaga *diversity*, sudah mendekati solusi optimum.
2. *fitness sharing* dapat bekerja dengan baik untuk menyelesaikan permasalahan TSP dalam mencapai nilai optimum. *fitness sharing* dapat meminimalkan kemungkinan terjadinya kekonvergenan data secara dini. Artinya, nilai *diversity* yang dihasilkan sudah diatas 50%.

DAFTAR PUSTAKA

- [1] J. Liu, Z.-x.Cai, and J.-q. Liu, "A novel genetic algorithm preventing premature convergence by chaos operator," *Journal of Central South University of Technology*, vol. 7, pp. 100–103, 2000.
- [2] Nurmaulidar, "The Application Of *Fitness Sharing Method In Evolutionary Algorithm To*

Optimizing The Travelling Salesman Problem (TSP)," Jurusan Matematika, FMIPA Universitas Syiah Kuala, Darussalam, Banda Aceh. *Jurnal Natural* Vol. 14, No. 2, pp. 23-32, September 2014.

- [3] Yong Wang, "The hibrida genetic algorithm with two local optimization strategies for traveling salesman problem, *Computers & Industrial Engineering*". Volume 70, April 2014.
- [4] M. I. Faradian, "Perbandingan Penggunaan Algoritma Genetika Dengan Algoritma Konvensional Pada Traveling Salesman Problem," Program Studi Teknik Informatika, Institut Teknologi Bandung, 2007.
- [5] W. F. Mahmudy, *Algoritma Evolusi*, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang 2013.
- [6] W. F. Mahmudy, "Optimasi Penjadwalan *Two-Stage Assembly Flowshop* Menggunakan Algoritma Genetika Yang Dimodifikasi," *Konferensi Nasional Sistem Informasi (KNSI)*, STMIK Dipanegara, Makassar, Maret 2014.
- [7] M. F. Tasgetiren, O. Bulut, Q. K. Pan and P. N. Suganthan, "A differential evolution algorithm for the median cycle problem," *Differential Evolution (SDE), 2011 IEEE Symposium on*, Paris, 2011, pp. 1-7.
- [8] J. Husna, "Pengaruh *Fitness Sharing* Dalam Algoritma Berevolusi Untuk Mengoptimalkan *Traveling Salesmen Problem (TSP)*," Skripsi Fak. Mipa Usyiah, Banda Aceh, 2010.
- [9] D.E. Goldberg, "Genetic Algorithms in Search Optimisation and Machine Learning," Addison Wesley, 1989.
- [10] Dataset TSPLIB : <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/>
- [11] P. I. Vitra, "Perbandingan Metode-Metode dalam Algoritma Genetika untuk *Travelling Salesman Problem*," Universitas Islam Indonesia, Seminar Nasional, Aplikasi Teknologi Informasi, Yogyakarta, 19 Juni 2004.