

Integrasi Kromosom Buatan Dinamis untuk Memecahkan Masalah Konvergensi Prematur pada Algoritma Genetika untuk Traveling Salesman Problem

Muhammad Rikzam Kamal, Romi Satria Wahono dan Abdul Syukur
 Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
 rikzamrx@gmail.com, romi@brainmatics.com, abah.syukur@yahoo.com

Abstract: Algoritma genetika (Genetic Algorithm (GA)) adalah metode adaptif yang digunakan untuk memecahkan masalah pencarian dan optimasi. *Travelling Salesman Problem* (TSP) merupakan salah satu persoalan optimasi yang dipecahkan dengan GA, di mana rute terpendek merupakan solusi yang paling optimal. GA juga salah satu metode optimisasi global yang bekerja dengan baik dan efisien pada fungsi tujuan yang kompleks dalam hal nonlinear, tetapi GA mempunyai masalah yaitu konvergensi prematur. Konvergensi prematur merupakan suatu kondisi yang terjadi ketika populasi algoritma genetika mencapai keadaan suboptimal di mana operator genetika tidak dapat lagi menghasilkan keturunan dengan kinerja yang lebih baik dari *parents*. Untuk mengatasi masalah konvergensi prematur, maka pada penelitian ini diusulkan dynamic artificial chromosomes yang diintegrasikan ke dalam genetic algorithm yang disebut GA-DAC. Dynamic Artificial Chromosomes (DAC) digunakan untuk mengontrol keragaman populasi dan juga seleksi kromosom terbaik untuk memilih individu atau kromosom terbaik. Beberapa eksperimen dilakukan dengan GA-DAC, dimana *threshold* terbaik adalah 0,5, kemudian juga mendapatkan hasil perbaikan pada jarak terpendek yang dibandingkan dengan GA standar. Hasil pengujian untuk dataset KroA100 sebesar 12,60%, KroA150 sebesar 13,92% dan KroA200 sebesar 12,92%. Untuk keragaman populasi mendapatkan hasil pada KroA100 sebesar 24,97%, KroA150 sebesar 50,84% dan KroA200 sebesar 49,08%. Maka dapat disimpulkan bahwa GA-DAC bisa mendapatkan hasil lebih baik dibandingkan dengan GA standar, sehingga membuat GA dapat keluar dari konvergensi prematur.

Keywords: algoritma genetika, konvergensi prematur, dynamic artificial chromosomes, genetic algorithm dynamic artificial chromosomes, seleksi kromosom terbaik, *travelling salesman problem*.

1 PENDAHULUAN

Algoritma genetika (Genetic Algorithm (GA)) adalah bagian dari komputasi evolusioner yang berkembang pesat dalam bidang kecerdasan buatan (Siva Sathya & Radhika, 2013). GA adalah metode adaptif yang digunakan untuk memecahkan masalah pencarian dan optimasi. GA didasarkan pada proses genetik organisme biologis. Dengan meniru prinsip evolusi alam, yaitu "*survival of the fittest*", GA mampu mengembangkan solusi untuk masalah dunia nyata (De Giovanni & Pezzella, 2010). Sebelum GA dapat diterapkan, representasi atau pengkodean dari masalah harus dibuat terlebih dahulu. Inti dari GA adalah untuk mengkodekan satu set parameter (dikenal sebagai gen) dan gabungan dari gen-gen yang membentuk nilai tertentu dan menyatakan solusi yang

mungkin dari suatu permasalahan yang disebut sebagai kromosom (Y.-H. Chang, 2010). Fungsi *fitness* juga diperlukan untuk memberikan nilai yang diperoleh dari setiap solusi. Setiap individu tergantung pada kromosom dan dievaluasi oleh fungsi *fitness* (Pavez-Lazo & Soto-Cartes, 2011). Selama proses berjalan, orang tua harus dipilih untuk proses reproduksi dan digabungkan untuk menghasilkan keturunan. Orang tua secara acak dipilih dari populasi menggunakan skema yang menguntungkan individu. Setelah memilih orang tua, kemudian kromosom digabungkan, menggunakan mekanisme *crossover* dan mutasi. Solusi akan diperoleh ketika orang tua menghasilkan keturunan yang lebih baik. Proses iterasi ini terus berjalan sampai kriteria yang ditentukan telah tercapai.

GA sebagai metode pencarian dan optimisasi masalah sering digunakan dalam berbagai macam kasus seperti *job-shop scheduling problem* (De Giovanni & Pezzella, 2010), *timetabling* (Yang & Jat, 2011), *unit commitment problem* (Pavez-Lazo & Soto-Cartes, 2011), dan selain itu juga digunakan untuk menyelesaikan *travelling salesman problem* (TSP) (Liu & Zeng, 2009)(P.-C. Chang, Huang, & Ting, 2010). TSP merupakan salah satu masalah optimasi kombinatorial mendasar yang memiliki banyak aplikasi dalam penelitian operasional (Zhang, Tong, Xu, & Lin, 2015). Selain itu TSP juga termasuk dalam kategori masalah klasik, yaitu untuk menemukan rute terpendek melalui serangkaian poin dan kembali ke awal (Çavdar & Sokol, 2014).

GA juga dikombinasikan dengan berbagai metode lain untuk mengatasi masalah konvergensi prematur. Seperti Triangular Crossover (TC) (Elfeky, Sarker, & Essam, 2008), Unimodal Distribution Crossover (UNDX) (Ono, Kita, & Kobayashi, 2003), dan deterministic annular crossover (Pavez-Lazo & Soto-Cartes, 2011). Deterministic annular crossover menggunakan annular selection untuk menyeleksi individu atau orang tua dalam populasi yang akan mengalami proses deterministic crossover, dimana individu yang dipilih adalah individu dengan nilai *fitness* tertinggi yang dipasangkan dengan individu dengan nilai *fitness* terendah. UNDX menggunakan beberapa orang tua (*parents*) untuk menciptakan solusi keturunan (*offspring*) disekitar pusat massa dari orang tua, sementara probabilitas dengan nilai kecil ditugaskan untuk solusi terjauh dari pusat massa. Meskipun telah menunjukkan kinerja yang sangat baik untuk masalah yang sangat epistasis (ketika efek dari satu gen tergantung pada kehadiran satu atau lebih pengubah gen) (Ono et al., 2003). Tetapi UNDX tidak dapat menghasilkan keturunan dalam beberapa kasus seperti ketika ukuran populasi yang relatif terlalu kecil. UNDX juga memiliki kesulitan dalam menemukan solusi optimal pada ruang pencarian terdekat. TC menggunakan tiga orang tua untuk *constrained problems* (masalah yang dibatasi), satu orang tua tidak layak dan dua orang tua harus layak. Hal ini digunakan agar dapat menghasilkan satu dari tiga keturunan.

Kemudian dari setiap keturunan yang dihasilkan sebagai kombinasi linear dari tiga orang tua.

GA banyak digunakan untuk memecahkan masalah optimasi, walaupun pada kenyataannya juga memiliki kemampuan yang baik untuk masalah-masalah selain optimasi. Algoritma genetika terinspirasi oleh proses evolusi, yang diamati dari alam (Chen & Chien, 2011). Algoritma genetika adalah simulasi dari proses evolusi Darwin dan operasi genetika atas kromosom (S.N Sivanandam, 2008). GA juga salah satu metode optimisasi global yang bekerja dengan baik dan efisien pada fungsi tujuan yang kompleks dalam hal nonlinear, tetapi GA juga mempunyai masalah yaitu konvergensi prematur (P.-C. Chang et al., 2010) (Pandey, Chaudhary, & Mehrotra, 2014). Konvergensi prematur terjadi ketika populasi algoritma genetika mencapai keadaan suboptimal dimana operator genetika tidak dapat lagi menghasilkan keturunan dengan kinerja yang lebih baik dari orang tua (P.-C. Chang et al., 2010).

Beberapa peneliti telah mencoba melakukan uji coba menggunakan beberapa algoritma untuk menyelesaikan masalah konvergensi prematur di dalam GA, diantaranya yaitu dengan Parent Centric Crossover (PCX) (Elsayed, Sarker, & Essam, 2014), deterministic annular crossover (Pavez-Lazo & Soto-Cartes, 2011), dan Multi Parents Crossover (MPC) (Elsayed et al., 2014)(Elfeky et al., 2008). PCX memungkinkan menciptakan solusi terdekat setiap orang tua, bukan didekat pusat orang tua. Setiap keturunan satu orang tua dipilih dan dihitung perbedaan vektor antara orang tua dan orang tua yang terpilih. PCX menerapkan pendekatan adaptif diri dimana vektor solusi baru terus bergerak menuju optimum. Ketika PCX diterapkan dengan GA, dibutuhkan waktu yang lebih lama dibandingkan dengan operator *crossover* yang lain, dan menemukan kesulitan dalam memecahkan masalah multimodal. Deterministic annular crossover digunakan untuk memperkaya hasil keturunan (*offspring*) dari proses *crossover*, dengan operator seleksi deterministik. Keragaman yang lebih besar antara individu-individu dari populasi dapat diperoleh melalui informasi genetik dari individu terburuk dengan probabilitas yang sama. MPC menggunakan tiga orang tua dalam proses *crossover* untuk menghindari keturunan (*offspring*) yang kurang beragam dari orang tuanya (Elsayed et al., 2014).

Pada penelitian ini, keragaman didalam populasi dikontrol dengan keragaman operator agar lebih beragam dengan cara meningkatkan keragaman populasi tersebut ketika nilai keragamannya kurang dari *threshold* atau kurang beragam (P.-C. Chang et al., 2010). Keseimbangan yang tepat antara eksplorasi dan eksploitasi pencarian dapat dipertahankan dengan mengendalikan tingkat keragaman populasi. Mekanisme kontrol dapat dibangun ke dalam GA menggunakan Dynamic Artificial Chromosomes (DAC) yang dimasukkan ke dalam sistem sampai ukuran keragaman mencapai tingkat tertentu kemudian berhenti. Selain itu juga akan digunakan operator untuk memilih individu atau kromosom terbaik yang akan dipasangkan dalam proses *crossover* sehingga proses eksplorasi dan eksploitasi dalam mutasi juga akan lebih maksimal. Dengan menerapkan DAC pada GA diharapkan dapat meningkatkan tingkat keragaman rata-rata sehingga proses dapat keluar dari konvergensi prematur dan proses iterasi dapat lebih maksimal.

2 PENELITIAN TERKAIT

GA juga salah satu metode optimisasi global yang bekerja dengan baik dan efisien pada fungsi tujuan yang kompleks

dalam hal nonlinear, tetapi GA juga mempunyai masalah yaitu konvergensi prematur (P.-C. Chang et al., 2010) (Pandey et al., 2014). Konvergensi prematur terjadi ketika populasi algoritma genetika mencapai keadaan suboptimal dimana operator genetika tidak dapat lagi menghasilkan keturunan dengan kinerja yang lebih baik dari orang tua (P.-C. Chang et al., 2010).

Beberapa peneliti telah melakukan penelitian untuk memecahkan masalah konvergensi prematur pada GA, diantaranya yaitu GA dengan Multi-Parent Crossover yang disebut GAMPC, serta diusulkan juga *diversity operator* untuk lebih lanjut membuat variasi pada pembangkitan *offspring* (keturunan) (Elsayed et al., 2014) agar individu didalam populasi menjadi lebih beragam. Pada penelitian lain juga diusulkan Deterministic Annular Crossover Genetic Algorithm yang disebut DACGA (Pavez-Lazo & Soto-Cartes, 2011), dengan menggunakan dua buah metode, yaitu seleksi *deterministic* dan *annular crossover*. Seleksi *deterministic* digunakan untuk mencari nilai *fitness* (kecocokan) individu dengan *fitness* yang lebih tinggi, sedangkan *annular crossover* digunakan untuk melakukan proses pertukaran informasi genetik antara dua individu dengan operator *crossover* yang direpresentasikan dalam bentuk cincin. Chang et al. mengusulkan dynamic diversity control di dalam GA atau yang disebut sebagai DDC-GA untuk *mining unsearched solution space* di TSP (P.-C. Chang et al., 2010).

Pada penelitian ini digunakan pengkontrol keragaman pada populasi dengan menggunakan dynamic diversity control (P.-C. Chang et al., 2010), yang bekerja ketika tingkat keragaman pada sebuah populasi turun pada batas tertentu atau dibawah *threshold* yang sudah ditentukan. Sedangkan untuk meningkatkan keragaman populasi menggunakan Dynamic Artificial Chromosome (DAC) dan juga menggunakan seleksi kromosom terbaik untuk memilih kromosom yang terbaik yang akan diproses pada *crossover* sehingga ini bisa membuat GA keluar dari konvergensi prematur.

3 METODE YANG DIUSULKAN

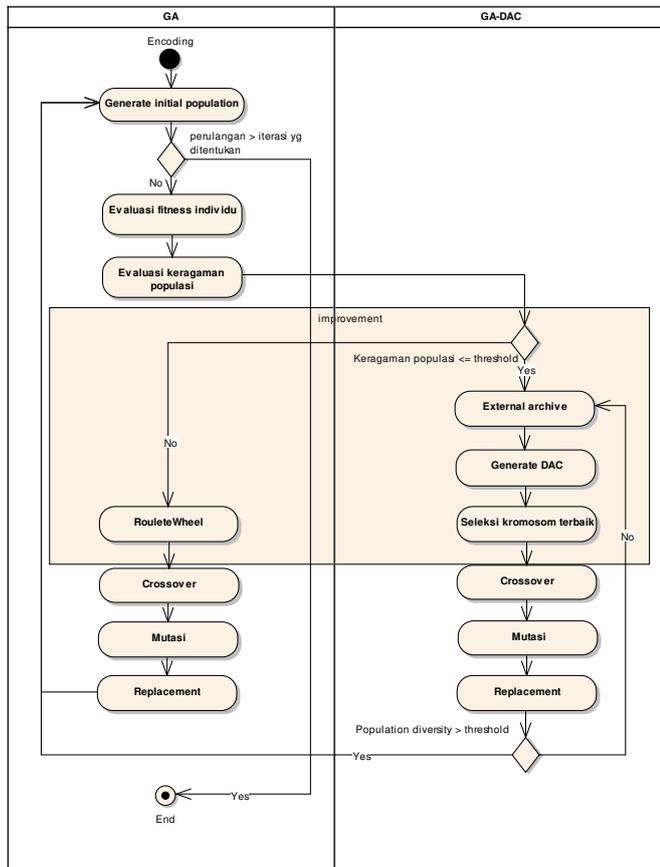
Pada penelitian ini diusulkan metode Dynamic Artificial Chromosome yang diintegrasikan kedalam Genetic Algorithm yang disebut GA-DAC dan juga seleksi kromosom terbaik seperti pada Gambar 1. Pada bagian kolom GA adalah struktur proses seperti pada GA standar pada umumnya, sedangkan pada bagian kolom GA-DAC adalah ketika pada proses evaluasi *fitness* dan evaluasi keragaman populasi (*population diversity*) diukur dengan menggunakan rumus *linear scale measure* (P.-C. Chang et al., 2010):

$$PD = \frac{\bar{d} - d_{\min}}{d_{\max} - d_{\min}}$$

PD	: keragaman individu atau kromosom
\bar{d}	: rata-rata keragaman
d_{\max}	: nilai maksimal dari keragaman
d_{\min}	: nilai minimal keragaman

Jika ternyata nilai keragamannya turun ke bawah, kurang dari atau sama dengan nilai dari *threshold*, maka kromosom buatan dinamis akan bekerja. Cara kerjanya dengan mengambil kromosom baru dari *external archive* dengan nilai *fitness* dan keragaman yang lebih baik. Fungsinya untuk menggantikan kromosom dengan nilai *fitness* paling terendah di dalam populasi, yang diharapkan dapat membuat populasi dengan nilai *fitness* dan keragaman terbaik yang akan diproses pada

tahap selanjutnya. Kemudian pada proses ini juga ditambahkan seleksi kromosom terbaik untuk memilih kromosom atau orang tua dengan nilai *fitness* paling terbaik yang akan mengalami proses *crossover*. Langkah ini akan terus dilakukan sampai dengan nilai keragaman pada populasi lebih dari nilai *threshold*. Sehingga diharapkan dengan metode ini dapat mengatasi masalah pada GA yaitu optimum lokal atau yang disebut juga sebagai konvergensi prematur.



Gambar 1. Metode yang Diusulkan GA-DAC

4 HASIL PENELITIAN

Pengujian hasil penelitian dilakukan menggunakan komputer dengan spesifikasi CPU Pentium (R) Dual-Core CPU 2.70 GHz, RAM 2GB, dan sistem operasi Microsoft Windows 7 Professional 64-bit. Aplikasi yang digunakan adalah NetBeans IDE 8.0.2. Data penelitian ini menggunakan TSP KroA100, KroA150, dan KroA200 (Wang, 2014) yang diperoleh dari situs <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp>.

Program GA dibuat sesuai dengan proses yang ada di dalam algoritma GA tersebut. GA-DAC dibuat berdasarkan program GA yang kemudian dikembangkan berdasarkan metode yang diusulkan seperti pada Gambar 1, yaitu untuk penentuan nilai keragaman populasi dan pengambilan kromosom baru (external archive) serta *generate* DAC.

Dalam penelitian ini GA-DAC menggunakan *threshold* dengan nilai yang terbaik sesuai dengan yang dilakukan Chang et al. (P.-C. Chang et al., 2010) yaitu 0,5, 0,6, dan 0,7.

Hasil dari uji pencarian rute terpendek GA dan juga GA-DAC ditunjukkan pada Tabel 1, 2, dan 3. Pada GA-DAC, *threshold* diatur ke nilai nilai 0,5 dan menghasilkan nilai perbaikan rute terpendek terbaik untuk KroA100 sebesar 12,60%, KroA150 sebesar 13,92%, dan KroA200 sebesar 12,92%.

Tabel 1. Hasil Pengujian Rute Terpendek GA dengan GA-DAC Menggunakan KroA100

Algo ritma	Thres hold	Rata-rata	Terbaik	STD	Perbaikan (%)
GA-DAC	0,5	139511,32	135746,05	3005,16	12,60
	0,6	147663,40	144981,55	1449,32	7,49
	0,7	155821,22	139145,86	6482,43	2,38
GA	-	159627,14	140623,90	9890,13	0

Tabel 2. Hasil Pengujian Rute Terpendek GA dengan GA-DAC Menggunakan KroA150

Algo ritma	Thres hold	Rata-rata	Terbaik	STD	Perbaikan (%)
GA-DAC	0,5	217836,81	209753,12	4350,82	13,92
	0,6	225226,64	217199	3633,59	11,00
	0,7	238986,27	235508,65	2475,26	5,56
GA	-	253068,79	246154,47	6657,12	0

Tabel 3. Hasil Pengujian Rute Terpendek GA dengan GA-DAC Menggunakan KroA200

Algo Ritma	Thres hold	Rata-rata	Terbaik	STD	Perbaikan (%)
GA-DAC	0,5	293804,89	288810,17	2893,44	12,92
	0,6	302040,42	297152,60	2921,81	10,48
	0,7	314607,09	308784,72	3304,86	6,75
GA	-	337406,73	326428,22	9334,52	0

Pada pengujian keragaman populasi yang dilakukan pada GA dan GA-DAC menggunakan dataset KroA100, KroA150 dan KroA200 dengan 60.000 iterasi, 10 kali *running* serta *threshold* 0,5, 0,6, dan 0,7.

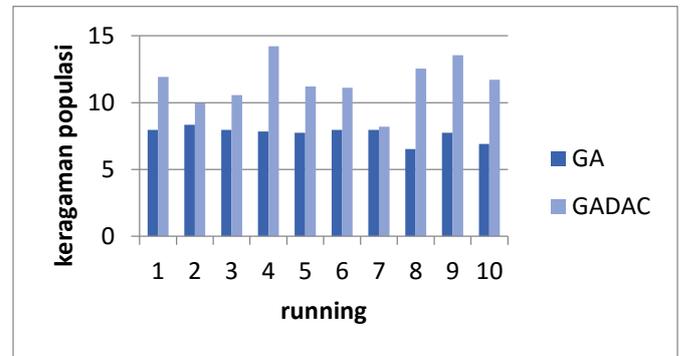
Hasil dari uji keragaman populasi dengan KroA100 ini pada Tabel 4 menunjukkan bahwa GA-DAC pada *threshold* 0,6 adalah nilai yang bisa mendapatkan keragaman terbaik yaitu sebesar 24,97%. Sedangkan untuk hasil keragaman pada setiap kali *running* terlihat pada Tabel 5 dan juga Gambar 2 yang menunjukkan bahwa dihampir setiap kali *running* GA-DAC mampu mendapatkan keragaman yang lebih baik dari GA.

Tabel 4. Hasil Pengujian Keragaman Populasi GA dengan GA-DAC Menggunakan KroA100

Algo Ritma	Thres hold	Rata-rata	Terbaik	STD	Perbaikan (%)
GA-DAC	0,5	9,2165	15,2805	2,5606	14,05
	0,6	10,55715	21,4500	4,4738	24,97
	0,7	8,7871	9,4575	0,6696	9,85
GA	-	7,9212	11,2506	1,6412	-

Tabel 5. Hasil Perbandingan Keragaman Populasi GA dan GA-DAC dengan KroA100

Run	GA-DAC	GA
1	7,033948	8,034521
2	15,28053	11,25059
3	8,761778	4,735462
4	9,498025	8,034521
5	7,834657	7,566016
6	7,837013	8,034521
7	8,411818	8,034521
8	9,300358	7,566016
9	6,583897	8,034521
10	11,6233	7,566016



Gambar 3. Hasil Perbandingan Keragaman Populasi GA dan GA-DAC dengan KroA150

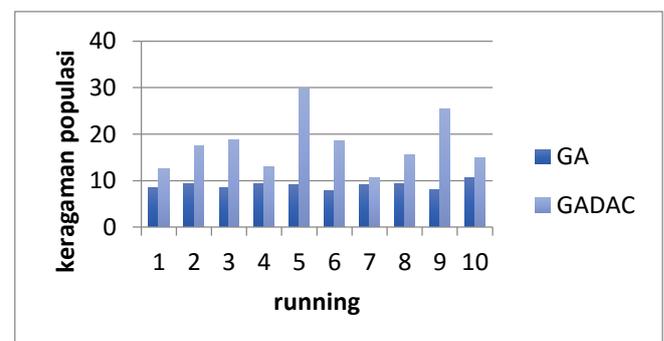
Hasil dari uji keragaman populasi dengan KroA200 ini pada Tabel 8 menunjukkan bahwa GA-DAC pada *threshold* 0,5 adalah nilai yang bisa mendapatkan keragaman terbaik yaitu sebesar 49,08%. Sedangkan untuk hasil keragaman pada setiap kali *running* terlihat pada Tabel 9 dan juga Gambar 4 yang menunjukkan bahwa setiap kali *running* GA-DAC mampu mendapatkan keragaman yang lebih baik dari GA

Tabel 8. Hasil Pengujian Keragaman Populasi GA dengan GA-DAC Menggunakan KroA200

Algo Ritma	Thres hold	Rata-rata	Terbaik	STD	Perbaikan (%)
GA-DAC	0,5	17,6395	29,6196	5,91173	49,08
	0,6	15,5960	26,2403	5,1432	42,41
	0,7	12,7379	15,6146	1,5683	29,48
GA	-	8,981565	10,5045	0,7489	-

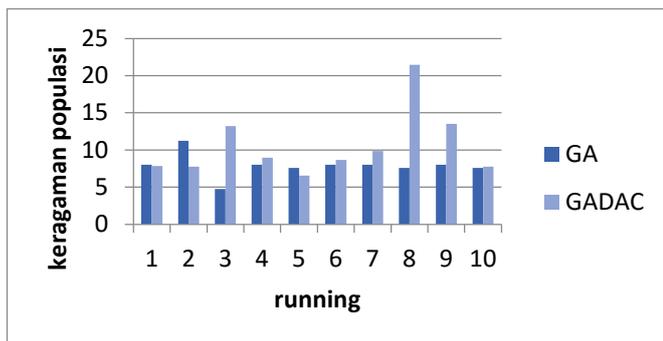
Tabel 9. Hasil Perbandingan Keragaman Populasi GA dan GA-DAC dengan KroA200

Run	GA	GADAC
1	8,5427707	12,601482
2	9,3446126	17,525871
3	8,5427707	18,729451
4	9,3446126	12,924827
5	9,0895829	29,619646
6	7,9008942	18,535542
7	9,0895829	10,511024
8	9,3446126	15,621517
9	8,1117358	25,348371
10	10,504478	14,977611



Gambar 4. Hasil Perbandingan Keragaman Populasi GA dan GA-DAC dengan KroA200

Selain membandingkan hasil penelitian ini dengan GA standar, penelitian ini juga dibandingkan dengan penelitian lain



Gambar 2. Hasil Perbandingan Keragaman Populasi GA dan GA-DAC dengan KroA100

Hasil dari uji keragaman populasi dengan KroA150 ini pada Tabel 6 menunjukkan bahwa GA-DAC pada *threshold* 0,6 adalah nilai yang bisa mendapatkan keragaman terbaik yaitu sebesar 50,84%. Sedangkan untuk hasil keragaman pada setiap kali *running* terlihat pada Tabel 7 dan juga Gambar 3 yang menunjukkan bahwa setiap kali *running* GA-DAC mampu mendapatkan keragaman yang lebih baik dari GA.

Tabel 6. Hasil Pengujian Keragaman Populasi GA dengan GA-DAC Menggunakan KroA150

Algo Ritma	Thres hold	Rata-rata	Terbaik	STD	Perbaikan (%)
GA-DAC	0,5	11,4977	14,2078	1,7401	49,43
	0,6	11,6063	17,7734	2,7588	50,84
	0,7	11,2672	15,4335	1,6739	46,43
GA	-	7,6945	8,3345	0,54405	-

Tabel 7. Hasil Perbandingan Keragaman Populasi GA dan GA-DAC dengan KroA150

Run	GA-DAC	GA
1	11,92446	7,958242
2	9,943509	8334489
3	10,55336	7,958242
4	14,20788	7,845096
5	11,21325	7,740081
6	11,11417	7,958242
7	8,208898	7,958242
8	12,55096	6,533787
9	13,55199	7,740081
10	11,70824	6,918578

yang sejenis, yang membahas tentang permasalahan yang sama, tentang konvergensi prematur dengan dataset TSP KroA100, KroA150 dan KroA200 yaitu pada metode DDCGA (P.-C. Chang et al., 2010). Perbandingan ini membandingkan hasil rute terpendek yang dihasilkan oleh GA-DAC dan DDC-GA, yang hasilnya bisa dilihat pada Tabel 10 yang menerangkan bahwa yang tercetak tebal menandakan hasil yang lebih unggul atau yang lebih baik. Ini membuktikan bahwa GA-DAC lebih unggul di hampir semua dataset yang digunakan dibandingkan dengan DDC-GA, kecuali untuk dataset KroA200 pada metode DDC-GA mendapatkan nilai lebih baik daripada GA-DAC.

Tabel 10. Perbandingan Hasil Rute Terbaik GA-DAC dengan DDCGA

Dataset	Algoritma	Hasil
KroA100	GA-DAC	12,60%
	DDC-GA	5,08%
KroA150	GA-DAC	13,92%
	DDC-GA	2,27%
KroA200	GA-DAC	12,92%
	DDC-GA	16,12%

5 KESIMPULAN

Dalam penelitian ini diusulkan metode kromosom buatan dinamis dan seleksi kromosom terbaik untuk mengatasi masalah konvergensi prematur didalam GA. Pada penelitian ini dilakukan beberapa pengujian untuk mencapai hasil perbaikan tertinggi rute terpendek dalam dataset KroA100 sebesar 12,60%, KroA150 sebesar 13,92% dan KroA200 sebesar 12,92%. Pada keragaman populasi GA-DAC dapat mencapai nilai perbaikan lebih baik dalam dataset KroA100 sebesar 24,97%, KroA150 sebesar 50,84% dan KroA200 sebesar 49,08% dibandingkan dengan GA.

Pada perbandingan hasil rute terbaik yang telah dilakukan GA-DAC dengan DDC-GA didapatkan hasil bahwa GA-DAC lebih unggul di beberapa dataset yaitu KroA100 dan KroA150 dibandingkan dengan DDCGA, tetapi pada dataset KroA200 DDC-GA lebih unggul dibandingkan dengan GA-DAC.

Dari hasil pengujian diatas maka bisa disimpulkan bahwa dengan menggunakan metode GA-DAC dan seleksi kromosom terbaik bisa menemukan rute terpendek dan membuat tingkat keragaman populasi menjadi lebih beragam, sehingga ini bisa membuat GA keluar dari optimum lokal (konvergensi prematur).

REFERENSI

- Çavdar, B., & Sokol, J. (2014). TSP Race: Minimizing completion time in time-sensitive applications. *European Journal of Operational Research*, 000, 1–8. doi:10.1016/j.ejor.2014.12.022
- Chang, P.-C., Huang, W.-H., & Ting, C.-J. (2010). Dynamic diversity control in genetic algorithm for mining unsearched solution space in TSP problems. *Expert Systems with Applications*, 37(3), 1863–1878. doi:10.1016/j.eswa.2009.07.066
- Chang, Y.-H. (2010). Adopting co-evolution and constraint-satisfaction concept on genetic algorithms to solve supply chain network design problems. *Expert Systems with Applications*, 37(10), 6919–6930. doi:10.1016/j.eswa.2010.03.030

- Chen, S.-M., & Chien, C.-Y. (2011). Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques. *Expert Systems with Applications*, 38(12), 14439–14450. doi:10.1016/j.eswa.2011.04.163
- De Giovanni, L., & Pezzella, F. (2010). An Improved Genetic Algorithm for the Distributed and Flexible Job-shop Scheduling problem. *European Journal of Operational Research*, 200(2), 395–408. doi:10.1016/j.ejor.2009.01.008
- Elfeky, E., Sarker, R., & Essam, D. (2008). Analyzing the simple ranking and selection process for constrained evolutionary optimization. *Journal of Computer Science and ...*, 23(1), 19–34. doi:10.1007/s11390-008-9109-z
- Elsayed, S. M., Sarker, R. a., & Essam, D. L. (2014). A new genetic algorithm for solving optimization problems. *Engineering Applications of Artificial Intelligence*, 27, 57–69. doi:10.1016/j.engappai.2013.09.013
- Liu, F., & Zeng, G. (2009). Study of genetic algorithm with reinforcement learning to solve the TSP. *Expert Systems with Applications*, 36(3), 6995–7001. doi:10.1016/j.eswa.2008.08.026
- Ono, I., Kita, H., & Kobayashi, S. (2003). A real-coded genetic algorithm using the unimodal normal distribution crossover. *Advances in Evolutionary Computing*. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-18965-4_8
- Pandey, H. M., Chaudhary, A., & Mehrotra, D. (2014). A comparative review of approaches to prevent premature convergence in GA. *Applied Soft Computing*, 24, 1047–1077. doi:10.1016/j.asoc.2014.08.025
- Pavez-Lazo, B., & Soto-Cartes, J. (2011). A deterministic annular crossover genetic algorithm optimisation for the unit commitment problem. *Expert Systems with Applications*, 38(6), 6523–6529. doi:10.1016/j.eswa.2010.11.089
- S.N Sivanandam, S. N. D. (2008). *Introduction to Genetic Algorithms*. (I. Integra Software Services Pvt. Ltd., Ed.) Vasa (p. 462). Berlin Heidelberg: Springer. doi:10.1007/978-3-540-73190-0_2
- Siva Sathya, S., & Radhika, M. V. (2013). Convergence of nomadic genetic algorithm on benchmark mathematical functions. *Applied Soft Computing*, 13(5), 2759–2766. doi:10.1016/j.asoc.2012.11.011
- Wang, Y. (2014). The hybrid genetic algorithm with two local optimization strategies for traveling salesman problem q. *COMPUTERS & INDUSTRIAL ENGINEERING*, 70, 124–133. doi:10.1016/j.cie.2014.01.015
- Yang, S., & Jat, S. N. (2011). Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(1), 93–106. doi:10.1109/TSMCC.2010.2049200
- Zhang, H., Tong, W., Xu, Y., & Lin, G. (2015). The Steiner Traveling Salesman Problem with online edge blockages. *European Journal of Operational Research*, 243(1), 30–40. doi:10.1016/j.ejor.2014.11.013

BIOGRAFI PENULIS



Muhammad Rizkam Kamal. Menyelesaikan pendidikan S1 Teknik Informatika di STMIK Widya Pratama Pekalongan, S2 Magister Teknik Informatika di Universitas Dian Nuswantoro Semarang. Saat ini menjadi Staf dan dosen STAI Ki Ageng Pekalongan. Minat penelitian saat ini adalah softcomputing.



Romi Satria Wahono. Memperoleh gelar B.Eng dan M.Eng pada bidang ilmu komputer di Saitama University, Japan, dan Ph.D pada bidang software engineering di Universiti Teknikal Malaysia Melaka. Menjadi pengajar dan peneliti di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro. Merupakan pendiri dan CEO PT Brainmatics, sebuah perusahaan yang

bergerak di bidang pengembangan software. Minat penelitian pada bidang software engineering dan machine learning. Profesional member dari asosiasi ilmiah ACM, PMI dan IEEE Computer Society.



Abdul Syukur. Menerima gelar sarjana di bidang Matematika dari Universitas Diponegoro Semarang, gelar master di bidang manajemen dari Universitas Atma Jaya Yogyakarta, dan gelar doktor di bidang ekonomi dari Universitas Merdeka Malang. Dia adalah dosen dan dekan di Fakultas Ilmu Komputer, Universitas Dian Nuswantoro,

Semarang, Indonesia. Minat penelitiannya saat ini meliputi decision support systems dan information management systems.