Jurnal TIMES , Vol. IV No 2 : 40-44 , 2015
ISSN : 2337 - 3601

INISIALISASI POPULASI PADA ALGORITMA GENETIKA MENGGUNAKAN SIMPLE HILL CLIMBING (SHC) UNTUK TRAVELING SALESMAN PROBLEM (TSP)

Delima Sitanggang
Program Studi Magister Teknik Informatika
Universitas Sumatera Utara
Jl. Dr Mansyur No.9, 20155, Indonesia
e-mail : djoshlimasitanggang@gmail.com

ABSTRACT

In classical genetic algorithm, the determination of the initial individu generated by random methods. In the study using a large individu, these methods often cause undesirable effects such as premature convergence in finding the optimal solution. In this study, algorithm Simple Hill Climbing (SHC) as the algorithm locally optimal analyzed its application to improve the performance of the genetic algorithm in order to avoid the genetic algorithm to the problem of convergence premature so expect to achieve optimal solutions in solving the Traveling Salesman Problem (TSP), In this research, three types of experiments by applying different parameters of Genetic Algorithm.In the first experiment, initial values obtained for the solution is 3596.6, Genetic Algorithm In the second experimental values obtained initial solution to SHC at 3494.1, and the best SHC for the best solution Genetic Algorithm In the third experiment obtained the initial value of 3330.9

Keywords : *Simple Hill Climbing, Genetic Algorithms, Travelling Salesman Problem*

1. Latar Belakang

Algoritma genetika adalah salah satu metode heuristik adaptif yang merupakan cabang dari *evolutionary algorithm*, yaitu suatu teknik untuk memecahkan masalah-masalah optimasi yang rumit dengan menirukan proses evolusi makhluk hidup. Algoritma genetika merupakan salah satu algoritma optimasi berbasis pencarian (*search*) yang didasarkan seleksi natural dan genetika natural (Golberg & Richardson, 1997).

Penyelesaian masalah optimasi menggunakan algoritma genetika tergolong sulit karena kualitas keputusan yang diselesaikan menggunakan algoritma genetika berhubungan dengan banyak karakteristik misalnya populasi awal, *crossover*, mutasi, probabilitas, fungsi *fitness*, faktor seleksi dan faktor yang lainnya. Keseluruhan faktor tersebut saling mempengaruhi satu sama lain sebagai sebuah sistem (Pedro & Gomez, 2007).

Algoritma genetika diawali dengan sebuah himpunan populasi awal (*initial population*). Hill (1999) menyatakan bahwa kualitas populasi awal sangat mempengaruhi hasil akhir yang dicapai oleh algoritma genetika. Populasi awal yang baik tentunya akan memfasilitasi algoritma genetika dalam mencapai konvergensi yang optimal dan sebaliknya populasi awal yang tidak baik (*poor*) akan menuntun kepada penemuan solusi yang tidak optimal.

Pada algoritma genetika klasik, penentuan populasi awal dibangkitkan dengan metode acak (*random*). Pada penelitian dengan menggunakan jumlah populasi yang besar, metode ini sering menimbulkan efek yang tidak baik berupa waktu eksekusi yang sangat lama dalam menemukan solusi optimal (Kumar *et al*. 2013). Penelitian serupa juga dikemukakan oleh Raja & Bhaskaran (2013) bahwa populasi awal yang tidak baik atau terlalu besar dapat mereduksi kecepatan konvergensi. Pedro *et al.* (2009) mengemukakan bahwa populasi yang terlalu besar dapat mengakibatkan konvergensi prematur. Melalui serangkaian proses yang iteratif , algoritma genetika akan membangkitkan populasi baru dan selanjutnya akan melakukan pencarian solusi terbaik.

Algoritma akan berhenti ketika menemukan so lusi yang optimal (Kumar, 2012). Penelitian terhadap metode pembangkitan populasi awal algoritma genetika merupakan bidang kajian riset yang tidak luput dari perhatian para peneliti akhir-akhir ini. Pendekatan ini dilakukan dalam rangka menghindari algoritma genetika mencapai konvergensi prematur. (Kuczapski *et al.* 2010) menerapkan metode *Weighted Sum of Priority Rules* (WSPR) untuk menentukan inisial populasi yang baik bagi *Genetic Algorithm* pada kasus *Job Shop Scheduling* (JSS), hasil penelitian menemukakan bahwa dengan metode WSPR diperoleh waktu yang lebih singkat sebesar 50 % dalam menyelesaikan JSS tersebut. Peneliti lain yaitu Deb *et al.*(2005) menerapkan metode *Non-dominate Sorting* (NS) dalam menghasilkan inisial populasi yang baik pada algoritma genetika. Hasil penelitian menunjukkan bahwa dengan metode tersebut diperoleh konvergensi yang mendekati tingkat *pareto-optimal* yaitu suatu kondisi dengan tingkat efisiensi yang paling tinggi.

Algoritma *Simple Hill Climbing* (SHC) merupakan salah satu algoritma heuristik yang bersifat optimum lokal (*local optimal*) dimana hasil pencarian yang ditemukan dengan metode tersebut bukan yang paling baik (optimum) dari semua solusi yang ada (Vaughan, 2000). Algoritma ini tergolong sederhana dan memerlukan waktu komputasi yang cepat. Algoritma *Simple Hill Climbing* (SHC) banyak diterapkan dalam penyelesaian masalah yang berkaitan dengan pencarian rute terpendek *(shortest path)*. Diantaranya adalah Thiang & Shih (2012) meyelesaikan pencarian rute terpendek pada aplikasi *mobile robot*, Novitasari *et al*. (2013) menyelesaikan masalah TSP (*Traveling Salesman Problem*) menggunakan metode SHC. Aawar & Bakri (2014) meneliti sekaligus mengimplementasikan *Simple Hill Climbing* (SHC) pada bidang *robotic automation control* 2D dan 3D. Pada penelitian tersebut *Simple Hill Climbing* (SHC) mampu memberikan pemetaan (*mapping*) yang baik bagi robot sehingga diperoleh jarak terpendek dan kemampuan yang lebih baik dalam menghindari tabrakan terhadap penghalang.

Dalam perkembangan penelitian selanjutnya *Simple Hill Climbing* (SHC) banyak dikombinasikan dengan algoritma yang lain untuk meningkatkan kinerja dari algoritma yang bersangkutan. Lim *et al*. (2006) mengkombinasikan *Simple Hill Climbing* (SHC) dengan *Ant Colony Optimization* (ACO) dalam meningkatkan *bandwidth* jaringan pada aplikasi GPS (*Global Positioning System*). Koschra &Joshi (2012) menerapkan *Simple Hill Climbing* (SHC) pada segmentasi citra, hasilnya di peroleh segmentasi yang baik.

Berdasarkan penelitian di atas penulis tertarik untuk meneliti metode atau algoritma yang lain untuk mencari solusi yang lebih variatif dalam rangka membangkitkan inisial populasi awal yang baik bagi algoritma genetika. Penulis meneliti penerapan algoritma *Simple Hill Climbing* (SHC) sebagai metode alternatif dalam rangka membangkitkan inisial populasi yang baik bagi algoritma genetika.

2. Algoritma Genetika

2.1. Pengantar

Algoritma genetika merupakan teknik pencarian yang didasarkan atas mekanisme seleksi dan genetika natural. Algoritma genetika berbeda dengan teknik pencarian konvensional, dimana pada algoritma genetika kondisi diawali dengan setting awal solusi acak yang disebut populasi. Tiap individu dalam populasi disebut kromosom, yang merepresentasikan suatu solusi atas permasalahan.

Kromosom adalah suatu string simbol, yang umumnya merupakan *string bit biner*. Kromosom berevolusi melalui iterasi berkelanjutan, yang disebut generasi. Pada setiap generasi, kromosom dievaluasi berdasarkan suatu fungsi evaluasi (Golberg & Richarson, 1997). Untuk menghasilkan generasi berikutnya, kromosom baru yang disebut offspring, dibentuk baik melalui penyatuan dua kromosom dari generasi awal menggunakan operator perkawinan silang (crossover) atau memodifikasi kromosom menggunakan operator mutasi (mutation). Suatu generasi baru dibentuk melalui proses seleksi beberapa induk (parents) dan anak (offspring), sesuai dengan nilai fitness, dan melalui eliminasi kromosom lainnya agar ukuran populasi tetap konstan. Kromosom yang sesuai memiliki kemungkinan tertinggi untuk dipilih. Setelah beberapa generasi, algoritma menghasilkan kromosom-kromosom terbaik yang diharapkan mewakili solusi optimal atau suboptimal atas permasalahan.

Hal-hal yang harus dilakukan dalam menggunakan algoritma genetika adalah:

1. Mendefinisikan individu, dimana individu menyatakan salah satu solusi (penyelesaian) yang mungkin dari permasalahan yang diangkat.
2. Mendefinisikan nilai *fitness*, yang merupakan ukuran baik-tidaknya sebuah individu atau baik-tidaknya solusi yang didapatkan.
3. Menentukan proses pembangkitan populasi awal. Hal ini biasanya dilakukan dengan menggunakan pembangkitan acak seperti *random-walk*.
4. Menentukan proses seleksi yang akan digunakan.
5. Menentukan proses perkawinan silang (*cross-over*).
6. Mutasi gen yang akan digunakan.

2.2. Simple Hill Climbing

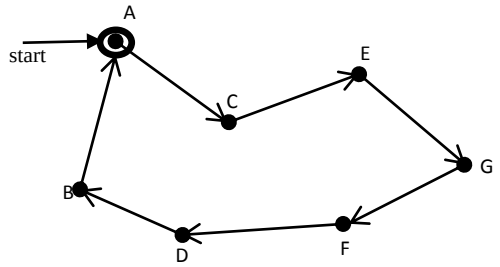
Metode *Simple Hill Climbing* adalah salah satu metode dari sekian banyak metode kecerdasan buatan untuk menyelesaikan permasalahan optimasi. Karena algoritmanya yang cukup sederhana, metode *Simple Hill Climbing* telah banyak diterapkan dalam berbagai aplikasi. Metode *Simple Hill Climbing* hampir sama dengan metode pembangkitan dan pengujian (*Generate and Test)*, hanya saja proses pengujian dilakukan menggunakan fungsi heuristik. Pembangkitan keadaan berikutnya (*next state*) sangat tergantung pada *feedback* dari prosedur pengujian, pengujian yang berupa fungsi heuristik ini akan menunjukkan seberapa besar nilai terkaan yang diambil terhadap keadaan lainnya yang mungkin.

2.3. Traveling Salesman Problem (TSP)

Traveling Salesman Problem (TSP) merupakan sebuah permasalahan optimasi yang dapat diterapkan pada berbagai kegiatan seperti *routing*. Masalah optimasi TSP terkenal dan telah menjadi standar untuk

mencoba algoritma yang komputational. Pokok permasalahan dari TSP adalah seorang salesman harus mengunjungi sejumlah kota yang diketahui jaraknya satu dengan yang lainnya dan kembali ke kota asal.

Traveling salesman problem dapat didefenisikan sebagai berikut : ada satu set kota {C1, C2, C3,...Cn} dan d {Ci, Cj} adalah jarak antar kota ke-i dan ke-j. Tujuannya adalah menemukan urutan π dari rumus berikut untuk mendapatkan nilai yang paling minimal (Gupta & Panwar 2013). Setelah jarak-jarak yang menghubungkan tiap kota diketahui, maka dicari rute terpendek dari jalur yang akan dilewati untuk kembali ke kota awal diperlihatkan pada gambar 1.



Gambar 1. Lintasan *(path)* dalam TSP

3. Metode Penelitian

Algoritma *Simple Hill Climbing* (SHC) sebagai algoritma yang bersifat lokal optimal diterapkan untuk memperbaiki kinerja dari algoritma genetika dalam rangka menghindari algoritma genetika pada masalah konvergensi prematur sehingga diharapkan dicapai solusi yang optimal dalam penyelesaian masalah *Traveling Salesman Problem* (TSP). *Data source* untuk penelitian ini adalah *data source* **Hasler Whithney TSPLIB95 A280**. *Data source* tersebut ditampilkan pada tabel 1. *Data source* tersebut terdapat 280 titik (*node*) yaitu dari C1 sampai C 280. Setiap titik (*node*) terdiri atas X (ordinat) dan Y (absis), misalnya kota 1 (C1) ada pada koordinat X = 288 dan Y=149.

Tabel 1. Data source Hasler Whitney

Data Source
[1, 8, 267, 101, 70, 215, 195, 205, 264, 102, 158, 208, 54, 78, 251, 36, 148, 266, 183, 224, 75, 172, 202, 96, 210, 212, 255, 171, 64, 203, 185, 220, 81, 24, 128, 126, 89, 241, 234, 263, 231, 196, 106, 52, 49, 142, 160, 26, 88, 105, 35, 190, 94, 62, 252, 194, 179, 229, 273, 134, 93, 127, 165, 123, 67, 269, 254, 44, 222, 22, 277, 246, 156, 237, 268, 265, 95, 186, 211, 119, 207, 57, 18, 178, 3, 168, 145, 114, 191, 272, 116, 161, 117, 45, 140, 227, 66, 16, 61, 32, 253, 46, 92, 238, 249, 275, 98, 250, 91, 138, 4, 47, 242, 232, 34, 150, 139, 6, 174, 198, 53, 173, 244, 274, 76, 270, 77, 245, 68, 17, 233, 262, 112, 107, 180, 258, 80, 236, 9, 69, 110, 199, 149, 41, 137, 87, 176, 217, 103, 230, 256, 223, 146, 121, 280, 167, 147, 239, 56, 235, 118, 131, 79, 155, 260, 82, 271, 37, 115, 132, 38, 100, 124, 72, 85, 84, 159, 111, 218, 133, 187, 23, 19, 136, 216, 163, 143, 276, 71, 10, 55, 151, 164, 43, 226, 166, 5, 141, 99, 181, 248, 30, 104, 214, 204, 193, 240, 261, 125, 59, 25, 73, 213, 169, 14, 200, 130, 201, 31, 108, 2, 197, 257, 162, 259, 86, 247, 120, 221, 50, 29, 39, 83, 228, 184, 74, 152, 33, 243, 225, 27, 63, 153, 206, 189, 135, 279, 20, 97, 278, 122, 11, 177, 90, 109, 175, 15, 12, 219, 13, 113, 170, 157, 21, 209, 48, 154, 129, 40, 7, 192, 58, 182, 144, 60, 188, 65, 51, 42, 28, 1]

4. Hasil dan Pembahasan

Penelitian ini menganalisis penerapan algoritma *Simple Hill Climbing* (SHC) dalam menentukan insial populasi untuk algoritma genetika. Inisial populasi yang baik diharapkan akan membantu algoritma genetika dalam pencarian solusi jarak terpendek dalam kasus TSP.

Pada Tabel 2, 3 dan 4 ditampilkan solusi awal (*initial solution)* yang dicapai menggunakan SHC. Pada lintasan tersebut titik pertama (*node 1*) dibuat sebagai titik awal (*start*) sekaligus menjadi titik akhir (*finish*). Pada percobaan pertama diperoleh nilai total jarak sebesar 3596.6, percobaan kedua diperoleh nilai solusi awal sebesar 3494.1 dan pada percobaan ketiga diperoleh nilai solusi awal sebesar 3330.9.

Tabel 2. Percobaan Pertama

Lintasan (*Path*)	Total Jarak
[1, 8, 267, 101, 70, 215, 195, 205, 264, 102, 158, 208, 54, 78, 251, 36, 148, 266, 183, 224, 75, 172, 202, 96, 210, 212, 255, 171, 64, 203, 185, 220, 81, 24, 128, 126, 89, 241, 234, 263, 231, 196, 106, 52, 49, 142, 160, 26, 88, 105, 35, 190, 94, 62, 252, 194, 179, 229, 273, 134, 93, 127, 165, 123, 67, 269, 254, 44, 222, 22, 277, 246, 156, 237, 268, 265, 95, 186, 211, 119, 207, 57, 18, 178, 3, 168, 145, 114, 191, 272, 116, 161, 117, 45, 140, 227, 66, 16, 61, 32, 253, 46, 92, 238, 249, 275, 98, 250, 91, 138, 4, 47, 242, 232, 34, 150, 139, 6, 174, 198, 53, 173, 244, 274, 76, 270, 77, 245, 68, 17, 233, 262, 112, 107, 180, 258, 80, 236, 9, 69, 110, 199, 149, 41, 137, 87, 176, 217, 103, 230, 256, 223, 146, 121, 280, 167, 147, 239, 56, 235, 118, 131, 79, 155, 260, 82, 271, 37, 115, 132, 38, 100, 124, 72, 85, 84, 159, 111, 218, 133, 187, 23, 19, 136, 216, 163, 143, 276, 71, 10, 55, 151, 164,	3596.6

43, 226, 166, 5, 141, 99, 181, 248, 30, 104, 214, 204, 193, 240, 261, 125, 59, 25, 73, 213, 169, 14, 200, 130, 201, 31, 108, 2, 197, 257, 162, 259, 86, 247, 120, 221, 50, 29, 39, 83, 228, 184, 74, 152, 33, 243, 225, 27, 63, 153, 206, 189, 135, 279, 20, 97, 278, 122, 11, 177, 90, 109, 175, 15, 12, 219, 13, 113, 170, 157, 21, 209, 48, 154, 129, 40, 7, 192, 182, 144, 60, 188, 65, 51, 42, 28, 1]	

Dengan menerapkan 20 iterasi maka diperoleh solusi SHC pada iterasi ke 10 dengan nilai total jarak **31618.0**

Tabel 3. Percobaan Kedua

Lintasan (*Path*)	Total Jarak
[1, 120, 44, 248, 189, 40, 103, 212, 35, 211, 19, 160, 223, 185, 232, 36, 100, 181, 55, 221, 132, 165, 59, 275, 169, 18, 49, 156, 149, 124, 131, 227, 17, 126, 150, 130, 262, 127, 148, 47, 201, 167, 98, 65, 57, 194, 164, 13, 260, 245, 83, 264, 111, 214, 8, 26, 85, 106, 62, 21, 271, 60, 256, 14, 141, 166, 243, 154, 15, 277, 66, 176, 224, 270, 258, 27, 88, 84, 196, 75, 219, 240, 76, 52, 182, 246, 222, 244, 20, 215, 168, 184, 50, 205, 99, 37, 89, 279, 259, 68, 250, 269, 230, 229, 170, 87, 144, 122, 56, 93, 46, 265, 117, 135, 152, 186, 23, 143, 118, 200, 109, 123, 67, 112, 272, 225, 121, 188, 274, 70, 280, 153, 190, 34, 237, 81, 110, 39, 10, 179, 140, 90, 136, 236, 92, 3, 267, 195, 203, 158, 162, 217, 4, 61, 261, 69, 146, 172, 142, 183, 38, 157, 138, 119, 48, 238, 77, 43, 151, 171, 79, 177, 95, 218, 63, 72, 145, 78, 105, 257, 30, 180, 25, 82, 73, 247, 263, 206, 242, 192, 96, 54, 94, 31, 129, 155, 208, 239, 113, 147, 133, 134, 234, 22, 231, 97, 253, 266, 33, 276, 11, 12, 6, 197, 175, 71, 161, 101, 107, 104, 16, 42, 159, 249, 202, 233, 24, 139, 254, 174, 191, 252, 268, 64, 251, 51, 220, 278, 91, 178, 114, 45, 209, 210, 137, 193, 116, 2, 125, 102, 5, 213, 226, 207, 53, 235, 108, 115, 9, 199, 80, 198, 255, 74, 7, 128, 41, 29, 32, 216, 241, 28, 86, 228, 163, 173, 204, 273, 187, 58, 1]	3494.1

Dengan menerapkan 40 iterasi maka diperoleh solusi SHC pada iterasi ke 25 dengan nilai total jarak **30609.0**

Tabel 4. Percobaan Ketiga

Lintasan (*Path*)	Total Jarak
[1, 55, 50, 67, 240, 155, 84, 235, 170, 163, 35, 78, 38, 87, 149, 130, 6, 238, 72, 111, 151, 157, 188, 178, 258, 65, 205, 247, 13, 230, 161, 80, 70, 220, 280, 74, 94, 119, 17, 24, 53, 97, 143, 189, 16, 242, 223, 39, 212, 121, 175, 271, 118, 162, 138, 14, 217, 266, 107, 147, 128, 131, 51, 40, 52, 69, 233, 255, 100, 145, 102, 172, 246, 177, 216, 15, 23, 10, 30, 237, 26, 254, 183, 185, 203, 25, 231, 259, 210, 244, 181, 225, 126, 154, 92, 7, 227, 232, 113, 264, 44, 211, 115, 89, 268, 79, 56, 265, 75, 108, 275, 253, 229, 98, 60, 249, 139, 276, 132, 106, 251, 73, 104, 190, 179, 95, 2, 86, 257, 252, 201, 129, 256, 96, 168, 133, 273, 47, 125, 12, 85, 214, 117, 46, 48, 99, 206, 270, 101, 83, 150, 186, 262, 5, 77, 9, 167, 234, 191, 228, 114, 169, 208, 61, 224, 263, 274, 152, 76, 241, 164, 116, 62, 159, 278, 174, 63, 165, 160, 171, 122, 277, 219, 176, 21, 204, 20, 110, 184, 34, 180, 81, 93, 120, 198, 28, 197, 182, 215, 105, 193, 58, 156, 45, 209, 144, 91, 236, 202, 221, 243, 22, 57, 59, 18, 36, 90, 218, 153, 123, 140, 136, 82, 54, 142, 199, 196, 112, 41, 134, 68, 3, 192, 29, 148, 245, 109, 207, 71, 127, 272, 279, 269, 124, 42, 27, 195, 194, 33, 19, 103, 4, 213, 267, 146, 222, 11, 173, 43, 88, 250, 8, 137, 200, 187, 64, 226, 32, 31, 166, 49, 37, 141, 260, 158, 66, 248, 239, 261, 135, 1]	3330.9

5. Kesimpulan dan Saran

Dari hasil penelitian dapat di ambil beberapa kesimpulan antaran lain :

1. Algoritma SHC dapat diterapkan untuk membangkitkan inisial solusi bagi algoritma genetika
2. Keberhasilan penerapan Algoritma Genetika pada penyelesian kasus komputasi TSP di pengaruhi oleh nilai parameter yang diterapkan
3. Besarnya parameter jumlah populasi yang diterapkan memiliki pengaruh yang signifikan terhadap pencapaian solusi akhir yang diharapkan semakin besar jumlah populasi maka di peroleh solusi yang semakin baik pula.

6. Daftar Pustaka

[1] Aawar, H.E. & Bakri, H. 2014. A simulated motion planning algorithm in 2d and 3d environment using hill climbing. *International Journal of Artificial intelligence and aplications*. 5(1) : 35-53.

[2] Deb, K., Prapta, A., Agarwal, S. & Meyarivan, T. 2005. *A fast and elitist multiobjetive genetic algorithm: NSGA-II*. IEEE-Evolutionary computation. 6(2) : 123-143.

[3] Ebrahimzadeh, R. 2013. Chaotic genetic algorithm based on lorenz chaotic system for optimization problems. *International Journal Intelligent Systems and Applications,* 13 (05) : 19-24

[4] Goldberg, D.E. & Richardson, J. (1987) Genetic algorithms with sharing for multimodal function optimization, *Proceedings of the Second International Conference on Genetic algorithms and their application* : pp. 41 -49.

[5] Hill, K. 1999. *Canonical genetic algorithm to optimize cut order plan solutionsinapparel manufacturing*. Canadian Institute of Actuaries : Canada

[6] Kühn, M., Severin, T. & Salzwedel, H. 2013. Variable mutation rate at genetic algorithms: introduction of chromosome fitness in connection with multi-chromosome representation. *International Journal of Computer Applications* 72(17) : 0975 – 8887

[7] Kumar, R. 2012. Novel encoding scheme in genetic algorithms.for better fitness. *International Journal of Engineering and Advanced Technology* (IJEAT). 1(6) : 2249 – 8958.

[8] Kochra, S. & Joshi, S. 2012. *S*tud*y* on hill climbing algorithm for image segmentation *International Journal of Engineering research and aplication.* 2(3) : 2171-2174

[9] Kuczapski, A., M., Micea, M., V. & Maniu., A., L. Efficient generation of optimal initial population to enhance genetic algorithm for job-shop scheduling. *Journal of information technology and control*. 39(1) : 32-38

[10] Kumar.V, Dutta, D., Roy, R. & Choudhury, K. 2013. An overview of methods maintaining diversity in genetic algorithms. *International Journal of Advanced Research in Computer Science and Software Engineering***.** 3(3) : 430-434.

[11] Lim, A., Lim, J., Rodrigues, B. & Xiao, F. 2006. Ant colony optimization with hill climbing for the bandwidth minimization problem. *Journal of applied soft computing*. 304(4) : 1-7

[12] Malhotra, R., Singh, N. & Singh.Y. 2011. Genetic algorithms concepts, design for optimization of process controllers. *Journal of Computer and Information Science*. 4(2): 39-5