

## PENERAPAN ALGORITMA GENETIKA PADA PERMASALAHAN MATEMATIKA

**Awang Andhyka**

Universitas Nahdlatul Ulama Sidoarjo

E-mail : awang85.si@unusida.ac.id

### *Abstrak*

*Algoritma genetika merupakan salah satu cara mengoptimasi dalam mencari hasil dari suatu permasalahan. Algoritma terdiri dari fungsi yang harus diselesaikan dengan nilai random awal, yang kemudian dilakukan pertukaran yang disebut mutasi dan crossover. Dalam penyelesaiannya, algoritma genetika mempunyai fungsi rate pada random dan rate kawin silang sehingga tidak semua individu awal hasil random dilakukan pertukaran atau persilangan, serta dapat untuk menyelesaikan permasalahan dalam matematika seperti kuadrat, liner, dan lain lain. Banyak sekali cara atau tool yang tersedia, akan tetapi yang digunakan pada paper ini adalah menggunakan matlab dengan dilakukan secara manual, dalam arti tidak menggunakan tools yang ada sehingga hasil crossover dan mutation serta gen dalam persilangan dapat di ketahui. Perbedaan nilai dalam rate mutasi dan rate crossover sangat berpengaruh dalam penyelesaian pada algoritma genetika, jika nilai rate lebih dari 1, maka dapat menyebabkan banyak individu awal hilang yang berakibat tidak adanya solusi yang ditemukan. Begitu juga dengan popsize, yang merupakan nilai awal yang dihasilkan secara random acak, semakin banyak popsize, maka peluang untuk menyelesaikan sebuah permasalahan menjadi lebih besar.*

**Kata Kunci:** *Algoritma genetika, crossover, linier, mutasi, popsize.*

### *Abstract*

*Genetic algorithms to optimize in finding results from a problem. The algorithm consists of functions that must be completed with an initial random value, which is then carried out by an exchange called mutation and crossover. In its completion, the genetic algorithm has a rate function at random and crossbreeding rates so that not all initial individuals are randomly exchanged or crossed, and solved problems in mathematics such as math, liners, many tools available, but what is used in this paper is to use Matlab by doing it manually, in the sense of not use tools so that crossover and mutation results and genes in crosses can be known. The difference in the value in the mutation and crossover rate is very influential for completion genetic algorithm, if the rate is more than 1, it can cause many initial individuals to disappear which used to a solution found. Likewise with popsize, which is the initial value generated randomly, the more popsize, the greater the chance of completing a problem.*

**Keywords:** *Genetic algorithm, crossover, linear, mutation, popsize.*

## 1. PENDAHULUAN

Dalam bidang era saat ini seperti teknologi informasi banyak permasalahan yang didapati, sehingga memunculkan ide supaya hal tersebut dapat diselesaikan. Permasalahan tersebut juga berhubungan dengan kehidupan sehari-hari. Sebagai contoh dalam lingkungan pendidikan, yaitu permasalahan dalam matematika, yang sampai hari ini mejadi momok pada anak sekolah.

Pemahaman tentang matematika dan penyelesaiannya membutuhkan waktu proses yang lama. Dari hal tersebut, ada beberapa cara penyelesaian permasalahan pada matematika, salah satunya dengan menggunakan suatu metode yaitu algoritma genetika[1]. Sebagai contoh apabila digunakan untuk mencari permasalahan linier kuadrat, garis ataupun kuadrat

memudahkan dalam mencari penyelesaian. Algoritma genetika sendiri didasarkan pada kehidupan gen manusia yang berkembang setiap saat, dalam arti kata bawah penyelesaian menggunakan metode algoritma genetika yaitu membutuhkan nilai awal yang disebut individu, dengan pengertian lain semakin banyak individu awal yang digunakan semakin banyak punya kesempatan untuk menyelesaikan permasalahan. Dalam teori algoritma genetika, dibatasi denganya adanya batasan-batasan seperti pada kehidupan sehari-hari, dimana batasan tersebut dapat digunakan sebagai nilai atau data yang membatasi supaya individu awal tidak melebihi batas[2]. Algoritma genetika disarankan pada kehidupan manusia yang berkembang biak dengan diselingi adanya gangguan pada kehidupan sehingga menjadi lebih baik, dimana gangguan tersebut disebut dalam algoritma dengan kata sederhana

yaitu mutasi diterapkan di metode algoritma genetika disebut dengan mutasi[3]. Algoritma genetika menghasilkan populasi lengkap dari titik penjawab. Setiap titik diuji secara mandiri dan untuk menetapkan populasi baru, termasuk poin yang dimodifikasi, poin poin yang ada dapat diuji[4]. Algoritma genetika mempertimbangkan banyak titik secara bersamaan dan karakteristik ini menyesuaikan dengan lebih paralel prosesor pada penyelesaian suatu masalah, karena mempertimbangkan setiap titik membutuhkan beberapa perhitungan termasuk fungsi target diferensiasi, dll[5]. Dalam algoritma ini, operan dan mekanisme yang berbeda di implementasikan, yang dijelaskan di sini untuk penemuan algoritma genetika sebagai algoritma mengoptimalkan terutama berdasarkan simulasi evolusi alami dan mereka belum didasarkan pada teori matematika yang kuat[6]. Teknik ini menggunakan genetika alami, metode statistik dengan mencoba menghasilkan suatu kesimpulan untuk mengoptimalkan fungsi pada algoritma genetika[7].

Saat ini, ada beberapa metode umum untuk mengoptimalkan pencarian yang meliputi: analitis, enumerasi dan acak [8]. Teknik analisis telah dipelajari dengan ekstensif. Mereka terbagi menjadi kategori langsung dan tidak langsung[9]. Metode tidak langsung mendapatkan yang dioptimalkan poin dengan memecahkan satu set persamaan non-linear yang dicapai dengan menempatkan fungsi target diferensiasi sama ke nol. Metode langsung dengan mencari titik yang dioptimalkan oleh fungsi target dan bergerak masuk menuju arah gradien. Kedua teknik ini dioptimalkan dan dikembangkan tetapi ada beberapa bukti dijelaskan kemudian menunjukkan bahwa mereka tidak mampu mengoptimalkan dengan benar[10].

Kedua teknik memiliki titik optimal yang berbeda. Titik optimal yang dicari teknik ini adalah titik terbaik lingkungan dari titik sekarang. Sebagai contoh, jika kita memiliki fungsi dengan dua puncak, salah satunya adalah salah dan salah satunya nyata, kemudian mencapai puncak yang lebih kecil menyebabkan kita terhalang dari titik yang sebenarnya. Karena itu, sebaiknya kita gunakan metode lain, seperti restart acak dengan loop untuk mendapatkan respons terbaik[11]. Teknik analisis bergantung dengan adanya diferensiasi. Bahkan jika kita menggunakan angka perkiraan diferensiasi dalam perhitungan, akan menghasilkan solusi optimal dengan kekurangan yang intensif. Oleh sebab itu jelaslah bahwa teknik berdasarkan batas kelanjutan dan turunan tidak cocok untuk setiap kasus.

Meskipun algoritma pencarian acak memiliki lebih banyak opsi dan melakukan lebih

baik daripada dua metode lainnya, tetapi hasil yang di inginkan tidak sesuai, ketika domain responsif besar. Tetapi genetika memiliki semua algoritma keuntungan dari metode yang disebutkan, serta tidak memiliki kekurangan, dan menggunakan kreativitas, yang sebagian besar terbukti dalam sikap manusia. Algoritma genetika juga mendapat bantuan fungsi probabilitas sebagai alatnya. Karenanya perbedaan teknik lain dengan algoritma genetika dibagi menjadi beberapa kategori[12], yaitu algoritma genetika dengan satu set pengkodean, algoritma genetika dengan sekumpulan titik, yang memulai pencarian tidak hanya dari satu titik, algoritma genetika menggunakan informasi akhir (fungsi target), bukan dari turunan atau perangkat tambahan lainnya. Algoritma genetika dalam menyelesaikan menggunakan hukum probabilitas dan bukan hukum tetap.

Dalam penelitian sebelumnya penyelesaian banyak menggunakan pemrograman linier yang merupakan satu model pemrograman matematika, yang dianggap sebagai alat yang diakui yang menyebabkan jutaan dolar tabungan untuk unit perdagangan besar di kota-kota maju di dunia dan penggunaannya dengan cepat maju di bagian lain dari komunitas solusi terbaik dan optimal[13]. Dengan maksud lain, dalam kasus melakukan kegiatan terkait dengan memanfaatkan sumber terbatas, biasanya diperlukan untuk mereka, sumber daya alokasi dan karenanya penentuan tingkat aktivasi harus dipertimbangkan [14]. Memang, dapat dikatakan bahwa ada puluhan buku dan artikel yang diterbitkan membahas aplikasi dan sekitar  $\frac{1}{4}$  dari perhitungan praktis total untuk komputer terkait dengan pemrograman linier dan derivasi. Singkatnya, pemrograman linear berhubungan dengan masalah pembatasan sumber daya antara kegiatan kompetitif untuk menemukan Model matematis digunakan dalam pemrograman linier untuk menjelaskan masalah yang dipertimbangkan.

Kata "linear" menunjukkan bahwa semua relasi permodelan matematis perlu menjadi fungsi linear. Hal tersebut mempunyai opini bahwa pemrograman linier digunakan mengoptimalkan supaya mempunyai hasil lebih baik daripada hasil alternatif. Spesifikasi pada model pemrograman linier adalah bahwa fungsi target dan kendala di dalamnya bersifat linier dan linearitas dari beberapa model dapat dibenarkan sesuai dengan karakteristik fisik. Seperti disebutkan, teknik ini menggunakan gradien fungsi untuk mencapai target dan solusi optimal, yang pada gilirannya membutuhkan turunan dan kontinuitas fungsi di semua pengoptimalan, karena jika fungsi tidak memiliki karakteristik itu, kita akan menghadapi kesulitan dalam memperoleh gradien, sedangkan

teknik algoritma genetika tidak memiliki ketergantungan terhadap derivabilitas dan melanjutkan fungsi dan mulai dengan memilih beberapa titik di permasalahan, serta mencari jawaban optimal dengan menggunakan poin utama. Selain itu, teknik ini tidak perlu perhitungan tambahan, seperti diferensiasi dan memperoleh gradien dan juga mengganti nilai dalam fungsi yang rumit[15]. Sementara itu solusi masalah yang berbeda oleh kedua metode, Secara empiris, menunjukkan bahwa jawabannya diperoleh dalam waktu yang lebih singkat oleh algoritma genetika.

## 2. METODE PENELITIAN

Algoritma genetika yang akan dilakukan dalam penelitian ini untuk mencari penyelesaian pada permasalahan matematika. Hasil dari optimasi penyelesaian tersebut berupa solusi dengan bermacam - macam nilai crossover dan mutasi. Ada beberapa langkah harus diperhatikan didalam melakukan proses algoritma genetika untuk menghasilkan sebuah solusi yang paling optimum, diantaranya:

1. Menentukan persamaan fungsi.
2. Menentukan fitness.
3. Mendefinisikan bentuk kromosom
4. Menentukan seleksi.
5. Menentukan crossover.
6. Menentukan mutasi.

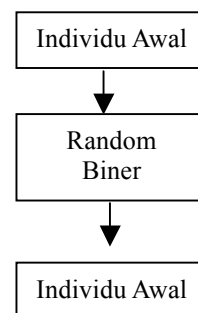
### 2.1 Persamaan Fungsi dan Fitness

Langkah pertama penelitian dengan membangkitkan populasi awal dengan membangkitkan beberapa individu yang bangkitkan secara acak yang memiliki prosedur tertentu. Ukuran pada populasi individu dilihat pada permasalahan yang dituju dengan beberapa jenis kromosom pada algoritma genetika. Setelah ukuran populasi selesai dihasilkan, kemudian dilakukan inisialisasi terhadap beberapa kromosom didalam populasi tersebut. Setelah itu yang dilakukan dalam pemecahan masalah dengan algoritma genetika adalah menentukan fungsi, dimana merupakan contoh dalam permasalahan matematika. Setelah fungsi ditentukan maka selanjutnya yaitu menentukan fitness. Penentuan fitness sangat penting pada proses algoritma genetika. Nilai fitness adalah suatu ukuran dalam nilai optimum yang baik dan merupakan solusi pada satu individu, atau dengan arti lain nilai fitness dapat digunakan dalam menyatakan nilai dari fungsi untuk mencapai tujuan. Untuk menentukan nilai fitness maka penelitian ini, menggunakan persamaan dari fungsi, dalam arti

kata jika kita menghendaki nilai minimum maka nilai terkecil yang diambil, sedangkan jika menghendaki nilai maksimum, maka nilai terbesar yang diambil.

### 2.2 Mendefinisikan Bentuk kromosom

Menentukan individu pada metode algoritma genetika dengan menentukan kromosom, kromosom yang digunakan adalah kromosom biner 8 bit. Model kromosom 8bit merupakan kromosom standart dari algoritma genetika (yang terdiri dari bilangan biner yang disusun dan gen 0 dan 1. Skema pengkodean ditentukan oleh algoritma genetika dengan diinisialisasi pada sebuah populasi dengan beberapa kromosom.



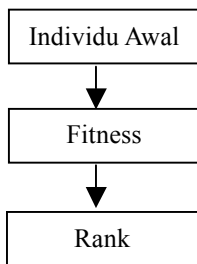
Gambar 1. Proses Kromosom

Gen dari algoritma genetika akan mengisi pada masing-masing kromosom dan dibangkitkan secara random. Masing- masing kromosom mempunyai pengkodean dari individu dengan memiliki nilai fitness yang ditentukan, dan kemudian menghasilkan populasi baru hasil pembentukan dengan menggunakan mekanisme pada seleksi secara alamiah, yaitu dengan random individu - individu awal dengan proporsional terhadap nilai fitness yang telah ditentukan, dan genetika secara alamiah dengan arti lain yaitu pindah silang atau dengan kata lain yaitu crossover dan mutasi. Pada umumnya algoritma genetika mempunyai metode yaitu adalah pergantian skema pada populasi dan disebut dengan generational replacement, dengan kata lain, kromosom pada suatu generasi dengan digantikan oleh pindah silang serta mutasi. Kromosom ditentukan dengan dibatasi oleh nilai individu awal dan individu akhir dari suatu random.

### 2.3 Menentukan Seleksi

Seleksi dilakukan setelah masing-masing fitness dari kromosom didapatkan. Fitness yang memiliki nilai paling besar memiliki probabilitas

paling besar untuk menjadi individu terbaik. Hasil individu dari seleksi digunakan untuk menghitung probabilitas individu mana yang keluar sebagai individu crossover. Seleksi menggunakan roulette wheel. Metode roulette wheel adalah metode yang sederhana serta paling sering digunakan di algoritma genetika, serta banyak juga dikenal mempunyai nama stochastic dengan sampling. Pada metode roulette wheel, individu-individu awal dipetakan di suatu segmen garis dan beraturan sehingga tiap segmen pada individu awal mempunyai ukuran yang sama dengan ukuran fitnessnya.



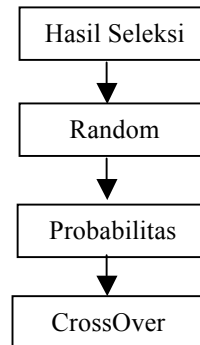
Gambar 2. Flowcart Seleksi

Sebuah bilangan pada random individu awal akan seleksi dan dibangkitkan serta individu awal yang memiliki fitness dalam kawasan yang ditentukan pada bilangan random akan diseleksi. Proses seleksi ini diulang sehingga memperoleh sejumlah individu seperti yang diharapkan sesuai fitness. Beberapa langkah dalam melakukan seleksi yaitu hitung peluang probabilitas untuk tiap individu, bangkitkan data random sebanyak individu dan yang ketiga, tukar posisi individu dengan nilai random terdekat. Seleksi yang digunakan menggunakan persentasi terbanyak individu yang keluar pada data.

**2.4 Menentukan CrossOver**

Hasil individu dari kromosom seleksi akan dilakukan cross-over, dimana proses yang akan dilakukan adalah mengkombinasikan dua individu hasil seleksi, yang diharapkan dari hasil tersebut mendapatkan individu yang mempunyai fitness lebih tinggi dari sebelumnya. Tidak semua pasangan individu hasil seleksi mengalami proses crossover, jumlah pasangan individu hasil seleksi yang akan diproses melalui crossover ditentukan pada parameter probabilitas crossover. Proses crossover membutuhkan dari dua induk individu hasil seleksi untuk membentuk kromosom sebagai individu baru. Pindah silang dari hasil crossover menghasilkan individu, yang nanti akan diteruskan pada proses mutasi di algoritma genetika. Probabilitas crossover menentukan pada

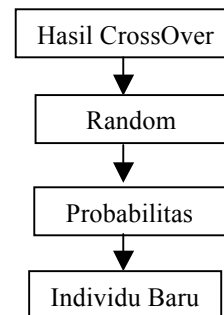
proses persilangan, yang dengan kata lain apabila nilai probabilitas lebih dari 1, maka akan menyebabkan hilangnya individu baru yang mempunyai gen lebih baik.



Gambar 3. Flowcart CrossOver

**2.5 Menentukan Mutasi**

Langkah selanjutnya di algoritma genetika yaitu proses mutasi. Dimana mutasi sangat berperan untuk menggantikan nilai dari hasil individu dari crossover dengan nilai yang dibangkitkan dari nilai random, tidak semua individu hasil crossover terkena mutasi, letak rotasi sudut yang akan dilakukan mutasi dipilih secara acak dan ditentukan oleh nilai dari probabilitas mutasi. Nilai mutasi dapat diisi sesuai yang diinginkan. Mutasi adalah bagian proses digunakan untuk menggantikan isi nilai pada satu individu atau beberapa dari gen pada suatu kromosom individu hasil crossover. Optimasi proses mutasi yang akan dilakukan di kromosom memiliki tujuan untuk mendapatkan kromosom-kromosom baru dari individu crossivovervsebagai kandidat individu baru sebagai solusi pada generasi individu baru dengan fitness yang mempunyai individu lebih baik, dan proses optimasi sehingga memiliki solusi optimum seperti yang ditentukan. Akan tetapi, untuk mendapatkan hasil ini, penekanan selektif memiliki peranan penting.



Gambar 4. Flowcart Mutasi

### 3. HASIL DAN PEMBAHASAN

Penelitian pertama dilakukan pada contoh soal matematika yang memiliki persamaan linier seperti persamaan 1.

$$f(x) = 5x_1 + 4x_2 + 8x_3 - 100 \quad (1)$$

$f(x)$  adalah fitness yang dicari dengan variable yang berbeda. Individu awal yang dibangkitkan secara acak. Dengan menggunakan popsize 3000, rate mutasi 0,6 dan rate crossover 0,7 dengan pengulangan gen sebanyak 7 kali, didapatkan hasil seperti pada tabel 2.1:

Tabel 2.1 Hasil dengan Popsiize 3000

$f(x)$	$x_1$	$x_2$	$x_3$	Individu ke	Gen ke
...	...	...	...	...	...
100	0	9	8	11827	4
219	15	22	7	12808	5
100	12	4	3	12809	5
775	27	30	65	12810	5
463	43	22	20	12811	5
1131	71	64	65	12812	5
439	43	36	10	12813	5
616	56	26	29	12814	5
679	95	51	0	12815	5
...	...	...	...	...	...

Dari hasil pada tabel 2.1,  $f(x)$  adalah hasil yang di inginkan dengan nilai 100, sedangkan jumlah individu awal berjumlah 3000 dengan pengulangan sebanyak 7 kali, maka individu yang akan dihasilkan untuk menyelesaikan problem di matematika tersebut sebanyak 21000 dengan random yang berbeda. Apabila individu ke 12809 disubstitusikan ke persamaan, maka  $5(12) + 4(4) + 3(3) = 100$ . Hasil tersebut membuktikan bahwa dengan algoritma genetika problem matematika dapat diselesaikan dan juga dapat memberikan solusi lebih dari satu. Proses penyelesaian tergantung pada besarnya individu awal yang disebut popsize dan probabilitas gen crossover dan gen mutasi. Semakin banyak popsize maka peluang untuk penyelesaian juga semakin banyak, dan juga probabilitas pada gen crossover dan gen mutasi tidak boleh lebih dari satu, apabila semakin banyak probabilitas, maka individu awal hasil seleksi, yang merupakan individu yang baik akan hilang, yang menyebabkan penyelesaian tidak dapat ditemukan. Berikut beberapa soal dari penyelesaian algoritma

genetika dengan berbagai macam popsize, dengan probabilitas yang berbeda-beda seperti pada tabel

Tabel 2.2 Penyelesaian fungsi linier

popsiize	crossover	mutasi	a	b	c
3000	0,6	0,7	12	8	1
			4	12	4
			3	3	6
			4	10	5
2000	0,4	0,5	8	11	2
			4	18	1
			0	25	0
			12	0	5
			0	3	11
			4	16	2

Beberapa hasil percobaan yang dilakukan seperti pada tabel 2.2 menjelaskan bahwa perbedaan popsize mempengaruhi hasil yang ingin diselesaikan. Peluang untuk mendapatkan hasil lebih besar jika random individu awal lebih banyak, rate mutasi dan rate crossover juga mempengaruhi hasil yang di inginkan. Penelitian kedua dilakukan pada contoh soal matematika yang memiliki persamaan kuadrat seperti persamaan 2.

$$f(x) = x_1^2 + 4x_2 + 3x_3 - 100 \quad (2)$$

Tabel 2.3 Penyelesaian fungsi kuadrat

popsiize	crossover	mutasi	$x_1$	$x_2$	$x_3$
3000	0,5	0,7	4	9	16
			3	7	21
			2	12	16
			2	3	28
2000	0,4	0,5	7	12	1
			7	0	17
			6	4	16
			3	10	17
			1	18	9
			5	12	9

Algoritma genetika pun dapat digunakan untuk menyelesaikan fungsi yang lebih kompleks seperti berikut tabel 2.2. Semakin banyak popsize maka semakin banyak pula individu awal yang dirandom, sehingga menyebabkan hasil dari penyelesaian fungsi mempunyai peluang lebih banyak. Begitu pula dengan crossover dan mutasi, nilai dari rate kedua fungsi tidak boleh lebih dari 1, agar dari pertukaran individu tidak semua yang mengalami proses crossover dan mutasi yang menyebabkan hilangnya individu baik. Proses pengulangan pun dapat

mempengaruhi dari hasil yang diinginkan, sebagai contoh popsize 3000 dengan popsize 2000, hasil individu mempunyai popsize lebih sedikit cenderung menghasilkan angka nol pada penyelesaiannya.

#### 4. KESIMPULAN

Algoritma genetika yang telah dilakukan dalam penelitian ini untuk mencari penyelesaian pada permasalahan matematika. Hasil dari optimasi penyelesaian tersebut berupa solusi dengan bermacam - macam nilai crossover dan mutasi. Perbedaan nilai dalam rate mutasi dan rate crossover sangat berpengaruh dalam penyelesaian pada algoritma genetika, jika nilai rate lebih dari 1, maka dapat menyebabkan banyak individu awal hilang yang berakibat tidak adanya solusi yang ditemukan. Penelitian selanjutnya diharapkan menambahkan fungsi elitism pada algoritma genetika atau membandingkan genetika algoritma dengan NSGA.

#### DAFTAR PUSTAKA

- [1] Andleeb, L., Munshi, A., H., and Dar, A., R., “*Genetic Diversity in Salix viminalis in the Kashmir Valley, India*”. American Journal of Biochemistry and Molecular Biology, 1: 178-184. 2011.
- [2] Asfaw, T., D., and Saiedi, S. “*Optimal Short-term Cascade Reservoirs Operation using Genetic Algorithm*”. Asian Journal of Applied Sciences, 4: 297-305. 2011.
- [3] Baptiste, J. “*Optimisation et analyse convexe: exercices et problemes corriges, avec rappels de cours, L'Editeur : EDP Sciences*”. 2009.
- [4] D'Ambra, A. and Sarnacchiaro, P. “*Some Data Reduction Methods to Analyze the Dependence with Highly Collinear Variables: A Simulation Study*”. Asian Journal of Mathematics & Statistics, 3: 69-81. 2010.
- [5] Enrique, A. “*Optimization techniques for solving complex problems.*, John Wiley and Sons. 2009.
- [6] Faigle ,U., Kern , G. “*Algorithmic Principles of Mathematical Programming (Texts in the Mathematical Sciences)*”, December 7. 2010.
- [7] Goldberg, D, and Deb, K. “*A Comparative Analysis of Selection Schemes Used in Genetic Algorithms*”, In Foundations of Genetic Algorithms, B. Spatz, Ed. Morgan Kaufmann Publishers, Inc., pp:69–93. 1991.
- [8] Goldberg, D.E, ”*Genetic Algorithms in Search*”, Optimization, and Machine Learning AddisonWesley. 1989.
- [9] Groover, M. “*Automation Production System and Computer Integrated Manufacturing*”, PracticeHall Englewood., NJ. 1987.
- [10] Kusaik, A. “*EXgt-S: A knowledge based system for group Technology*”, International Journal of Production Research, Vol. 26(5), pp: 887-904. 1989.
- [11] Murtagh, B., A., Saunders, M., A. “*Large-Scale Linearly Constrained Optimization Mathematical Programming*”, Volume 14, Number 1, 41-72, May 16. 2005.
- [12] Pant, M., Sharma, P., Radha, T., Sangwan R., S. and Roy, U. “*Nonlinear Optimization of Enzyme Kinetic Parameters*”. Journal of Biological Sciences, 8: 1322-1327. 2008.
- [13] Rajni, V., Patel, F., Shadpey. “*Control of redundant robot manipulators: theory and experiments*”, Springer. 2005.
- [14] Russell, C. “*Introduction to Mathematical Programming*”, Pearson Learning Solutions, 3 edition, ISBN-10: 0558859143, November 18. 2010.
- [15] Sivanandam, Deepa, S., N. “*Introduction to genetic algorithms, Springer*”. 2007.