

유전자 알고리즘에서 볼츠만 선택방법의 개선

윤기석, 김태형, 김유신

부산대학교 전자공학과

전화 : (051)510-2376 / 팩스 : (051)515-5190

The improvement of genetic algorithm using Boltzmann selection

Gi-Souk Yoon, Tae-Thyung Kim, Yoo-Shin Kim

Department of Electronics Engineering, Pusan University, South Korea

kimys@hyowon.cc.pusan.ac.kr

Abstract

In this paper, we propose a method to improve Genetic Algorithm using Boltzmann selection which Michael has suggested. But Michael uses temperature schedule(the initial temperature, the cooling rate), which can be applicable only to the limited range of problems. We propose a new method to find the critical temperature and the cooling rate as parameters of the temperature schedule. The critical temperature can be derived from the distribution of each individual's fitness. Through the application of the island model where each island has differing cooling rate, it is proved that it is unnecessary to find the optimal cooling rate. The simulation on the TSP's with various city sizes proves the proposed critical temperature correct.

1. 서론

최적화 문제를 해결하기 위하여 물리현상 또는 자연현상에 기초를 둔 방법이 많이 연구되어져 왔다. 대표적인 방법으로 GA(Genetic Algorithm, 유전자 알고리즘), SA(Simulated Annealing, 시뮬레이티드 어닐링)[10]등이 있다.

SA는 근본적으로 순차적인 처리구조를 가진다. 그래서 이의 병렬화에 대한 많은 연구가 행해졌다.

Hyeon-Joong[5]는 여러 해를 각각 다른 온도에서 annealing 시키면서, 목적함수의 값이 좋지 않은 해에 대하여 높은 온도를 가지게 하는 방법을 사용하여 해의 개선을 유도하는 SA의 병렬 처리 알고리즘을 제안하였고, James[7]는 그 알고리즘을 개선하였다.

Theodore[6]는 부개체군(subpopulation)에서 서로 개체들을 교환하게 하는 구조를 가진 DGA를 제안하여 62-processor NCUBE/six hypercube computer에 대하여 거의 선형에 가까운 성능 향상을 보여 주었다.

Saul[9]는 마이크로 프로세서의 디자인 과정에서 decoupling된 부분간에 각각 placement를 병렬로 처리함으로써 병렬처리가 가능하게 하는 SA 알고리즘을 제안하였다.

Krishna[3]는 NP-hard problem인 job shop scheduling를 해결하기 위하여 SA 병렬처리 알고리즘 3가지(temperature modifier, the locking edges and the modified locking edges algorithms)를 제안하였다.

그리고 SA와 GA와의 결합을 통하여 서로의 장점을 혼합하는 연구가 다음과 같이 진행되어왔다.

Hao Chen[8]은 GA와 SA의 결합모델인 GSA(Genetic Simulated Annealing)를 SIMD 구조에 적용하여 TSP와 Error Correction Code Design에 좋은 결과를 보여주었다.

Michael[15]는 Boltzmann distribution을 각 개체의 선택확률로 도입한 볼츠만 선택(Boltzmann selection)을 제안하였다. 그는 볼

츠만 선택이 translation invariance의 특성을 보이고, 온도라는 수렴 파라미터를 도입하였고, GA 진화과정이 가상 온도가 낮아지는 프로세스이므로 자연스러운 과정이 아니라는 것을 보였다. 그러나 그는 초기온도의 결정을 시행착오를 통한 값을 사용하여 여러 가지 문제에 적용함으로써 온도 스케줄을 정확히 구하지 못하였다.

본 연구에서는 SA와 GA를 결합한 모델인 볼츠만 선택에 대한 임계온도(critical temperature)를 개체군의 적합도의 분포로부터 구하고 서로 다른 cooling rate를 가지는 개체군을 사용한 island 모델을 적용시킨 모델을 제안한다.

2. GA using Boltzmann Selection

제안된 GA를 설명하기 전에 볼츠만 선택을 사용한 GA에 대하여 설명한다.[15]

2.1 Terms

1)개체군 (population)과 개체 (individual)

GA는 개체들의 집합인 개체군을 가지고 진화한다. 각각의 개체는 주어진 문제에 대한 한 가지의 해(solution)를 나타내는 염색체와 그 염색체의 값으로 계산되어지는 적합도를 가지고 있다.

2)염색체 (chromosome)와 부호화 (coding)

각각의 개체는 염색체 속에 해에 대한 정보를 부호화 하여 가지고 있다. 본 논문에서 사용할 TSP(Travel Salesman Problem, 순환 방문 판매원 문제)에서는 방문하는 도시의 번호(정수)를 차례대로 염색체로 사용한다. 10개의 도시의 예를 들면, 9 5 4 2 1 3 8 6 7 0 (도시의 번호는 0-9사이의 정수)과 같은 염색체는 9에서 5 그리고 4 그리고 2와 같은 차례로 도시를 방문한다는 것을 나타낸다.

3)적합도 (fitness)

개체가 나타내는 해의 우수함을 비교할 수 있는 척도이다. 일반적으로 목적함수(objective function, cost function)와 비슷한 역할을 한다. 적합도는 적응도라고 하기도 한다.

2.2 GA using Boltzmann

1.초기화 (initialize)

임의의 비트열로 개체를 주어진 개체 수만큼 만들고 그것을 초기 개체들로 사용한다.

2.적합도 계산 (Calculate fitness)

선택을 하기 전 각 개체에 대하여 적합도를 계산한다.

3.선택 (Selection)

볼츠만 선택에서 사용되는 선택 확률은 다음과 같다.[15]

$$P_j = \frac{e^{\frac{f_j}{T}}}{\sum_k e^{\frac{f_k}{T}}} \quad (5)$$

P_j : j 째 개체의 선택확률

f_j : j 번째 개체의 적합도

4. 교배 (Crossover)

선택 확률을 이용하여 두 개의 개체를 선택한 후 교배율 (Crossover Rate)의 확률로 교배를 행한다. 본 논문에서 사용할 TSP의 경우, 위와 같이 단순하게 유전자를 교환하여 교배를 행하면 같은 도시를 두 번 방문하는 부적합한 개체(invalid solution)가 생성될 수 있다.

따라서 TSP와 같은 문제에서는 적합한 개체를 유지하기 위해서 PMX의 교배 방법을 사용한다.

- PMX(Partially Matched Crossover) [14]

단순교배와는 달리 PMX의 경우 A,B 사이에 직접 유전자를 교환하는 게 아니라, 바꾸고자 하는 유전자를 자기 자신의 염색체 내의 유전자와 교환한다.

이와 같은 방법으로 교배하면 다음과 같이 적합한 개체가 생성된다.

A: 984|567|1320 A'': 984|230|1657
 B: 871|230|9546 B'': 801|567|9243

5. 돌연변이 (Mutation)

본 논문에서는 위와 같이 교배된 개체에 대하여 일정한 돌연변이율(Mutation Rate)로 임의의 두 점사이의 도시의 순서를 뒤집는다.

6. 수렴할 때까지 2~5 사이를 반복한다.

3. Temperature Schedule

3.1 Temperature

충분히 높은 온도에 대하여 $\frac{f_j}{T}$ 는 거의 0이 된다. 따라서

$P_j = \frac{1}{\text{개체수}}$ 이 되어, 모두 같은 확률로 개체가 선택되게 된다.

그러나 온도가 아주 낮아지면 적합도가 가장 큰 값의 개체만이 선택될 것이다. 제일 큰 적합도를 가지는 개체를 제외한 나머지 개체는 상대적으로 아주 작은 확률을 가지게 된다. 그리

고 적당한 온도에서는 적합도 값이 1의 차이가 날 때마다 $e^{\frac{1}{T}}$ 의 비율로 선택확률의 차이가 난다.

이렇듯이 온도는 선택의 정도를 결정하는 파라미터(수렴 제어 파라미터)의 역할을 한다.

높은 온도에서는 우수한(상대적으로 높은 적합도를 가지는) 개체에 대한 선택이 이루어지지 않으므로, 각 개체들이 자유롭게 탐색 공간을 탐색하게 되며, 낮은 온도에서는 우수한 개체에 대한 선택확률이 상대적으로 높아지게 되므로, 우수한 개체들이 살아남을 확률이 높아 지게 된다.

3.2 임계온도

본 논문에서는 일반적으로 많이 사용되는 다음과 같은 온도 스케줄을 사용한다.

$$T_i = (T_{cooling\ rate})^i T_0 \quad (1)$$

이러한 온도 스케줄은 초기온도와 cooling rate의 두 가지 파라미터를 가지게 된다. 이러한 파라미터를 구하는 것은 일반적으로 시행착오를 통하여 수하는 것이 일반적이다.

어느 온도에서부터 좋은 개체가 선택되는 확률이 뚜렷한 경향을 보이면서 돌연변이를 통하여 자유롭게 탐색하는 경향이 상대적으로 작아지게 된다. 그러면서 높은 적합도의 값들이 선택되기 시작한다. 이러한 온도가 임계온도라고 할 수 있다.

그러한 임계온도를 초기온도로 사용하면 고온에서의 필요없는 계산시간을 줄일 수 있다. 또한 저온에서 시작했을 때 발생할 수 있는 조기 수렴 현상도 해결할 수 있다. 그러나, Michael은 그러한 임계온도를 무시하고 여러 번의 실험을 통하여 우수한 개체가 구해지는 초기 온도를 선택하였다. 또한 한가지 문제를 사용한 실험에 대하여 결정된 초기온도를 여러 문제에 적용가능한 초기온도라 하였다 [15].

본 논문에서는 임계온도를 구하기 위해 선택도 α 를 다음과 같이 도입한다.

$$\alpha = \frac{P_{max}}{P_{min}} \quad (8)$$

α : 선택도

P_{max} : 가장 큰 적합도의 개체가 선택될 확률

P_{min} : 가장 작은 적합도의 개체가 선택될 확률

위 식과 같이 α 는 적합도가 가장 큰 개체와 작은 개체의 선택확률의 비이다. 위 (8)식을 다시 바꾸면

$$\alpha = \frac{P_{max}}{P_{min}} = \frac{e^{\frac{f_{max}}{T}}}{e^{\frac{f_{min}}{T}}} = e^{\frac{f_{max} - f_{min}}{T}} \quad (9)$$

따라서 현재 온도와 적합도의 최대값과 최소값의 차를 알면 (9) 식으로 선택도 α 를 구할 수 있게 된다.

이러한 선택도가 어느 정도 값에 이르면, 선택도에 의한 적합도의 값의 변화가 커지면서 임계온도에 다르게 된다. 따라서 임계온도에서의 α 값을 안다면 임계온도에 관한 식은 다음과 같이 (9)식을 간단히 변형하면 쉽게 구할 수 있다.

$$T_c = \frac{(f_{max} - f_{min})}{\ln \alpha_c} \quad (10)$$

T_c : 임계온도

$f_{max} - f_{min}$: 적합도의 최대, 최소의 차

α_c : 임계온도에서의 α

뒤에서 행한 시뮬레이션에서 여러 크기의 문제에 적용해 본 결과 α_c 값은 대략 e 의 값을 사용하면 적당하다. 이는 최대, 최소의 선택확률의 비가 1: e 가 될 때 임계온도에 도달한다는 것을 의미한다.

3.3 Island 모델 & Cooling Rate

Cooling rate는 1 세대에 대하여 온도가 작아지는 비율을 말한다. Cooling rate는 주로 0.9~1 사이의 값을 가지고, 이 cooling rate가 1에 가까우면 온도가 아주 천천히 떨어지게 되고 넓은 범위의 탐색공간을 충분히 검색하면서 annealing되어 최적해를 쉽게 찾을 수 있게 된다. 그러나 계산시간이 오래 걸리게 된다.

반대로 cooling rate가 작으면 온도가 급격히 떨어지게 되고 충분히 넓은 범위의 탐색공간을 탐색하지 못하고 국소해에 빠져서 나쁜 개체가 구해질 가능성이 높다. 그러나 빠른 시간에 해를 구할 수 있다.

적절한 범위의 cooling rate로 annealing 하는 island 모델을 적용함으로써 정확한 파라미터를 구하지 않고도 좋은 해를 구할 수 있다. Island 모델은 여러 개의 독립된 개체군(island)을 사용하여 일정한 비율로 개체군 간에 개체를 이동하는 GA에 적용가능한 진화 모델이다.[16]

본 논문에서는 cooling rate를 작은 값, 중간 정도값, 1에 아주 가까운 값으로(예를들면 0.9, 0.99, 0.999) 3개의 개체군에 각각 적용한다.

각각의 개체군에 대하여 한 세대당 1%의 개체를 더 낮은

cooling rate를 사용하는 개체군에 이동한다[6]. 가장 낮은 cooling rate에서는 개체를 이동하지 않는다.

위와 같이 하면 가장 빨리 온도가 줄어드는 개체군은 빨리 수렴하지만, 좋은 해로의 수렴은 보장하지 못하나, 그보다 높은 온도에서의 개체가 이동해오고, 이동해온 개체가 선택되어 교배하면서 국소 탐색 공간을 벗어날 수 있다. 그리고 천천히 떨어지는 개체군은 초기에는 자유롭게 탐색공간을 탐색하면서 낮은 온도에 국소 탐색 공간을 벗어날 수 있는 개체를 제공한다. 따라서 빨리 수렴하면서 우수한 개체를 찾을 수 있다.

3.4 제안된 알고리즘

제안된 온도 스케줄을 가지는 GA 알고리즘은 다음과 같다.

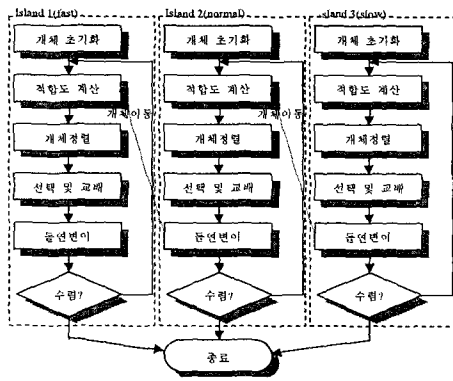


그림 1. 제안된 GA 알고리즘의 flowchart

fig. 1. The flowchart of the proposed GA algorithm

- 알고리즘의 프로세스

1. 각각의 개체군을 초기화 한다.
 - 1.1 초기온도를 (10)식을 통하여 구한다.
 - 1.2 임의의 개체를 생성
 - 1.3 그 외 돌연변이율, 교배율 등의 파라미터의 값을 정한다.
2. 각각의 개체군에 대하여 한 세대의 진화과정을 행한다.
 - 2.1 두 개의 개체를 불뜨만 선택을 통하여 선택한다.
 - 2.2 교배율의 확률로 교배를 행한다.
 - 2.3 다음 세대의 개체가 다 만들어질 때까지 2.2-2.3의 과정을 반복한다.
 - 2.4 만들어진 개체에 대하여 돌연변이율만큼 돌연변이를 발생시킨다.
- 3 각각의 개체군에 대하여
 $T_{cooling\ rate,\ fast}$, $T_{cooling\ rate,\ normal}$, $T_{cooling\ rate,\ slow}$ 의 cooling rate를 온도에 곱한다.
 4. 각각의 개체군에서 1%의 개체를 임의로 선택해서 그 보다 작은 cooling rate의 개체군으로 이동시킨다.
 5. 수렴할 때까지 2-4를 반복한다.

4. 시뮬레이션

4.1 Parameter

시뮬레이션에서 사용한 파라미터는 다음과 같다.

- 개체군(island)당 개체수 : 100 개
- 돌연변이율 : 0.1
- 교배율 : 0.6
- $T_{cooling\ rate,\ fast} : 0.9$,
- $T_{cooling\ rate,\ normal} : 0.99$

$$T_{cooling\ rate,\ slow} : 0.999$$

4.2 임계온도

임계온도가 올라감을 보이기 위하여 1 개의 개체군만을 사용하여 적합도의 그래프를 구하였다.

그림 4, 5, 6, 7, 8은 10, 50, 100, 200, 1000 개의 도시에 대하여 $T_{ini} = (f_{max} - f_{min}) \times 1000$ 에서 0.9995의 cooling rate를 가지고 50000 세대만큼 진화했을 때의 적합도 값의 최대, 최소, 평균값을 그래프로 그린 것이다. 세대의 진행방향은 온도가 줄어드는 방향이므로 오른쪽에서 왼쪽으로 진행한다.

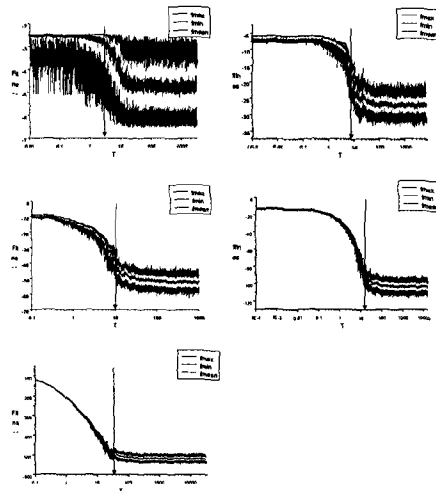


그림 2. 온도에 대한 적합도의 그래프 (도시 10, 50, 100, 200, 1000 개)

fig.2. The fitness graph versus temperature for 1000 city TSP

아래 표는 각각의 도시수에 대하여 첫 1000 회의 세대 동안의 $f_{max} - f_{min}$ 의 평균값을 구하였다.

표 1. 도시 수에 따른 $f_{max} - f_{min}$ 의 평균

Table 1. the mean values of $f_{max} - f_{min}$ with various city sizes

도시수	f_{max} 의 평균	f_{min} 의 평균	$f_{max} - f_{min}$ 의 평균
10	-3.17718	-6.09914	2.92196
50	-23.01106	-30.77153	7.76047
100	-47.12069	-57.45719	10.3365
200	-96.13731	-110.47499	14.33768
1000	-502.2578	-534.57958	32.32178

표 1에서 구한 $f_{max} - f_{min}$ 의 평균값을 온도-적합도 그래프에 화살표가 달린 세로 선으로 그렸다. 적합도 값의 평균값이 급격히 커지기 시작하는 온도를 임계 온도라 하면, 세로 선과 임계온도가 거의 일치함을 알 수 있다. 이 사실로 실험적인 α 값이 (10)식을 통해 e 임을 알 수 있다. 그러면 (10)식이 다음과 같이 간단히 된다.

$$T_c = f_{max} - f_{min} \quad (11)$$

도시 10 개에 대해서 임계온도는 조금 낮게 계산되어 진다. 이는 도시의 개수가 작은 경우 높은 온도에서 자유롭게 탐색하

는 개체 중에서 좋은 개체가 발생할 확률이 높으므로 계산되어진 임계온도보다 높은 온도에서 적합도의 값들이 커지기 시작하기 때문이다. 따라서 작은 수의 도시를 가지는 문제에서는 임계온도가 낮게 계산되어진다.

4.3 여러가지 온도 스케줄과 비교

표 2. 여러가지 온도 스케줄과 비교

Table 2. compare the proposed temperature schedule with other various schedules

적용 알고리즘	Cooling rate	초기 온도 100	10	1
일반적인 온도 스케줄	0.9	3289, - 8.5143	3512, - 8.57301	3231, - 8.54775
	0.99	3840, - 8.49933	4416, - 8.49171	3167, - 8.53157
	0.999	9904, - 8.91667	9686, - 8.41409	8768, - 8.49519
제안한 방법	제안된 cooling rate	5112, -8.36853 (초기온도는 제안된 (11)을 사용)		

위 표는 다양한 온도 스케줄과 제안한 스케줄을 사용하여 100 개의 도시에 대하여 10000 번의 세대동안 가장 좋았던 개체가 나타난 세대와 그 적합도를 15 번 구하여 평균을 내어 정리한 것이다.

표에서 보듯이 초기온도가 10 이고 cooling rate 가 0.999 일 때 가장 좋은 결과를 보여준다. 그러나 해를 구하는 데 걸리는 시간이 아주 길다. 그리고 초기온도가 10 이고 cooling rate 가 0.99 일때 해의 값은 두 번째로 좋은 해이나, 구하는 데 걸리는 시간이 앞서의 경우보다 약 2 배가량 짧다. 따라서 빠른 시간에 비교적 좋은 해를 구하려면 후자의 스케줄을, 시간이 다소 걸리더라도 좋은 해를 구하려면 전자의 스케줄을 사용하면 좋을 것이다.

그러나 본 논문에서 제안한 방법을 사용하여 구한 해는 온도 스케줄을 구하는 위와 같은 과정이 필요없이 가장 좋은 해를 구하였다. 그러나 가장 빠르게 해를 구하는 스케줄보다는 시간이 많이 걸렸다.

4.4 TSP 의 결과 예

제안한 알고리즘으로 10, 50, 100, 200 개의 도시를 가진 TSP 에 적용하여 10000 번의 세대동안 진화시킨 후의 시뮬레이션의 결과는 다음과 같다.

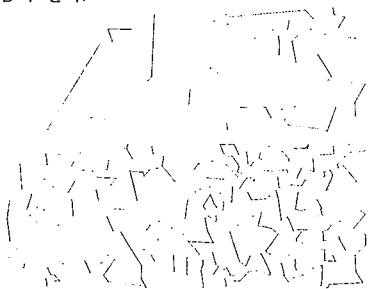


fig 3. Example of 10, 50, 100, 200 cities

5. Conclusion

본 논문에서는 GA 에 온도 개념을 도입한 볼츠만 선택에서의 온도 스케줄을 제안하였다. 제안된 수렴 스케줄은 개체군의 적합도의 분포에 따라 결정되어지는 임계온도를 초기온도로 사용하고 island 모델을 적용하여 각각의 개체군마다 서로 다른 cooling rate 를 사용하였다.

TSP 의 100 이상의 도시에 대해서 임계온도는 거의 정확했으며, 제안된 스케줄이 다른 여러 온도 스케줄보다 좋은 해를 구함을 시뮬레이션을 통하여 보였다.

5. 참고 문헌

- [1] Samuel Pierre and Gisèle Legault, "A Genetic Algorithm for Designing Distributed Computer Network Topologies," IEEE Transactions on Systems Man And Cybernetics, vol. 28, no. 2, pp249-258, April 1998
- [2] Shigenobu Kobayashi, Isao Ono, and Masayuki Yamamura, "An Efficient Genetic Algorithm for Job Shop Scheduling Problems," Proceedings of the sixth international conference on Genetic Algorithms 1995, pp.506-511, July 1995
- [3] K. Krishna, K. Ganeshan, and D. Janaki Ram, "Distributed Simulated Annealing Algorithm for Job Shop Scheduling," IEEE Transactions on System, Man, and Cybernetics, vol. 25, no.7, pp1102-1109, July 1995
- [4] J. Craig Potts, Terri D. Giddens, and Surya B. Yadav, "The Development and Evaluation of an Improved Genetic Algorithm Based on Migration and Artificial Selection," IEEE Transactions on Systems Man And Cybernetics, vol. 24, no. 1, pp73-86, January 1994
- [5] Hyeon-Joong Cho, Se-Young Oh, and Doo-Hyun Choi, "Population-oriented simulated annealing technique based on local temperature concept", Electronics Letters, vol. 34, No. 3, pp. 312-313, 1998
- [6] Theodore C. Belding, "The Distributed Genetic Algorithm Revisited," Proceedings of the sixth international conference on Genetic Algorithms 1995, pp114-121, July 1995
- [7] James M. Varanelli and James P. Cohoon, "Population-Oriented Simulated Annealing: A Genetic/Thermodynamic Hybrid Approach to Optimization," Proceedings of the sixth international conference on Genetic Algorithms 1995, pp174-181, July 1995
- [8] Hao Chen, Nicholas S. Flann, and Daniel W. Watson, "Parallel Genetic Simulated Annealing : A Massively Parallel SIMD Algorithm," IEEE Transactions on parallel and distributed Systems, vol. 9, no.2, pp126-136, February 1998
- [9] Saul A. Kravits, and Rob A. Rutenbar, "Placement by Simulated Annealing on a Multiprocessor," IEEE Transactions on computer-aided design, vol. CAD-6, no.4, pp534-549, July 1987
- [10] R. H. J. M. Otten, L. P. P. van Ginneken, "The Annealing Algorithm," Kluwer Academic Publishers, 1989
- [11] John Shawe-Taylor and Janez Šerovnik "Analysis of the Mean Field Annealing Algorithm for Graph Colouring," Journal of Artificial Neural Networks, vol. 2, No. 4, pp. 329-340, 1995
- [12] D. E. Van den bout, and T. K. Miller III, "Improving the Performance of the Hopfield-Tank Neural Network Through Normalization and Annealing," Biological Cybernetics, 62, pp. 129-139, 1989
- [13] Yoo-shin Kim, S. G. Cha, J. M. Kim, "A Generalized Model of Statistical Hopfield Neural Network to Solve TSP", ICICS(International Conference on Information, Communications and Signal Processing) '97, vol. 2, pp693-697, September 1997
- [14] D. E. Goldberg, "Genetic Algorithms in Search Optimization and Machine Learning," New York, Addison Wesley, 1991
- [15] Michael de la Maza, "The Boltzmann Selection Procedure," Practical Handbook of Genetic Algorithms : New Frontiers Volume II, vol. 2, chap.5, pp111-139, CRC Press, 1995
- [16] Conor, Ryan, "Niche And Species Formation in Genetic Algorithms," Practical Handbook of Genetic Algorithms : Applications Volume II, vol. 2, chap.2, pp 57-75 ,CRC Press, 1995
- [17] Alberto Leon-Garcia, "Probability and Random Processes for Electrical engineering," Addison Wesley, 1994