2003년도 대한전기학회 하계학술대회 논문집 2003.7.21-23

다중 최적화 문제에서 파레토 방법들 비교 연구

고영상
성균관 대학교 대학원 전기전자 및 컴퓨터 공학부

A study on Comparison of the Pareto Methods for Multi-objective optimization ptoblem

Young Sang Ko
Department of Electrical and computer engineering Sungkyunkwan University

Abstract - 유전자 알고리즘은 다윈의 자연선택설과 유전자의 진화 개념을 이용한 적응 탐색 알고리즘으로 적용하고자 하는 문제의 매개 변수를 유전자와 비슷한 데이터 구조로 부호화하고, 유전 연산자를 이용하여 문제의 해답을 찾는 알고리즘이다. 최근 유전자 알고리즘은 이러한 복수개의 목적 함수를 최적화 하기 위한 다중 최적화 문제를 위한 최적화 기술로서의 관심이 크게 다루어지고 있으며 전송 문제, 생산 공정 문제 계획 등과 같은 다목적 함수를 다루는 많은 응용 부분에 대해 적용되고 있다. 본 논문에서는 기본적인 다중 목적 함수용 예와 Gen과 Kim이 제안한 네트워크 신뢰도를 고려한 연결 비용과 메시지 지연을 고려한 이중 구속 통신망 설계 문제를 가지고 가중치 합과 여러 가지 파레토 방법들을 비교하고 연구 검토 하고자 한다.

1. 서 론

Holland에 의해서 소개된 유전자 알고리즘은 다윈의 자연선택설(natural selection)과 유전자의 진화(evolution) 개념을 이용한 적응 탐색 알고리즘(adaptive search algorithm)[1]으로 적용하고자 하는 문제의 매개 변수를 유전자와 비슷한 데이터 구조로 부호화(encoding)하고, 유전 연산자(genetic operater)를 이용하여 문제의 해답을 찾는 알고리즘이다.[2] 이러한 유전 알고리즘은 병렬 처리 구조와 탐색 능력이 뛰어난 장점을 가지고 있으나, 적용하고자 하는 문제의 특성에 따라 부호화 방법을 변경시켜주어야 하며 부호화 방법에 따라 적합한 유전 연산자(genetic operator)를 개발해야 한다.

초기의 유전 알고리즘은 모두 단 한 개의 목적 함수를 가진 경우를 가정하였다. 대부분의 경우는 이 범주에 속한다. 그러나 실제 세계에서는 복수 개의 목적 함수를 동시에 만족시켜야하는 경우가 있다. 만일 이 복수 함수를 독립적으로 최적화 할 수 있다면 별 논란거리가 되지 않는다. 그렇지만 A라는 목적에 잘 맞는 해가 B라는 목적에 반드시 잘 맞지 않는다면 이들을 모두 고려하는 것은 쉽지 않은 일이 된다.

최근 유전자 알고리즘은 이러한 복수개의 목적 함수를 최적화 하기 위한 다중 최적화 문제(Multi-objective optimization problem)를 위한 최적화 기술로서의 관심이 크게 다루어지고 있으며 전송 문제[3],최소 생성 트리 문제[4], 생산 공정 계획 문제[5]등과 같은 다목적 함수를 다루는 많은 응용 부분에 대해 적용되고 있다.

다중 최적화 문제를 위해 기존에 주로 사용한 방법은 각 목적함수를 하나로 통합시키는 가중치 합(weight-sum) 방법이다. 그러나 최근에는 보다 뛰어난 파레토 기반의 방법들을 사용하는 경우가 늘고 있다.

본 논문에서는 기본적인 다중 목적 함수용 예와 Gen과 Kim이 제안한 네트워크 신뢰도를 고려한 연결 비용과 메시지 지연을 고려한 이중 구속 통신망 설계 문제를 가지고 가중치 합과 여러 가지 파레토 방법들을 비교하고 연구 검토 하고자 한다.

논문의 구성은 다음과 같다. 2.1 에서는 사용된 알고리즘의 구성과 특성을 설명하였고 2.2에서는 시뮬레이션에 사용될 환경과 결과에 대하여 설명하였다.

2. 본 론

2.1 파레토 방법들 [7],[8]

지금까지의 유전 알고리즘은 모두 단 한 개의 목적 함수를 가진 경우를 가정하였다. 대부분의 경우는 이 범주에 속한다. 그러나 가끔 복수 개의 목적 함수를 동시에 만족시켜야하는 경우가 있다. 만일 이 복수 함수를 독립적으로 최적화 할 수 있다면 별 논란거리가 되지 않는다. 그렇지만 A라는 목적에 잘 맞는 해가 B라는 목적에 반드시 잘 맞지 않는다면 이들을 모두 고려하는 것은 쉽지 않은 일이 된다. 이것은 OR 분야에서는 오래 전부터 문제가 되어 왔던 주제로 모든 목적 함수에 대해 최적인 해는 근본적으로 존재하지 않는 경우가 많다.

이를 위해 해들을 크게 열성해(dominated solution)와 비열성해(non - dominated solution)의 두 범주로 나눌 수 있다. 우성해가 아니고 비열성해라는 용어를 사용하는 점에 유의해야 한다. 목적 함수가 k 개 ($f_1, \ldots f_k$) 있고 이들을 가능한 한 최대화하는 문제를 가정하자. 임의의 해 x 에 대해 모든 목적 함수 f_i(i=1,...k)에 대해 $f_i(x) < f_i(y)$인 해 y가 존재하지 않으면 x는 비열성해 또는 파레토 최적해(Pareto-optimal solution)라 부른다. 모든 파레토 최적해는 고려 가치가 있다.

복수 개의 목적 함수를 다루는 가장 전형적인 방법은 각 목적 함수에 대해 가중치를 주어 하나의 목적 함수로 통일하는 방법이다. 즉, 다음과 같이 만드는 방법이다.

$$F(x) = \sum_{i=1}^{k} w_i f_i(x), \qquad w_i \in [0,1], \ \sum_{i=1}^{k} w_i = 1$$

복수개의 목적 함수를 갖는 문제를 위한 최초의 유전 알고리즘은 Schaffer(1984)에 의한 것이다. 그는 q 개의 목적함수가 있다면 해집단을 q 개의 부분 해집단으로 나누어 각 해집단이 한 함수에 대해 적합도를 갖도록 하여 각 해집단에서 이를 근거하여 선택을 하게 한 다음, 교차와 변이는 일반적인 유전자 알고리즘과 동일하게 하는 방법을 사용하였다.

2.1.1 Goldberg 파레토 랭킹 방법

랭킹에 근거한 파레토 적합도 할당 방법은 Goldberg에 의해서 처음으로 제안 되었는데 Baker의 랭킹 선택 방법과 유사하다.

Baker의 랭킹 선택 방법은 개체군의 개체들을 각 개체들의 적합도의 크기에 따라 순서대로 정렬하고 순위에 따라 정해진 적합도를 부여하는 방법이다. 그러나 이 파레토 랭킹 방법은 그 각 개체의 적합도가 아니고 파레토 최적해들을 근거로 순위가 결정된다.

랭킹 과정은 다음과 같다 :
① 파레토 최적해들에게 랭킹1 을 부여한다.
② 랭킹 1 의 개체들을 제거한다.
③ 남은 개체의 파레토 최적해들을 찾는다.
④ 파레토 최적해들에게 랭킹2 을 부여한다.
⑤ 이것을 전체 모집단이 전부 제거 될때까지 반복한다.
⑥ 랭킹에 따라서 개체들에 선택 확률들을 지정해 준다.

그림 1는 두 개의 목적함수를 갖는 경우에 해의 순위을 Goldberg의 랭킹 방법에 의해 할당한 예이다.

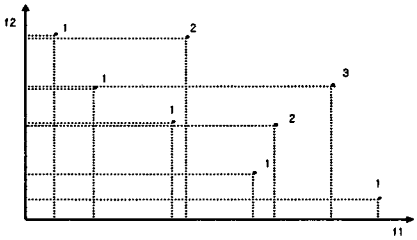

그림 1. Goldberg의 랭킹
Fig. 1. Goldberg's ranking

2.1.2 Fonseca와 Fleming's 파레토 랭킹 방법

Fonseca와 Fleming은 파레토 랭킹의 약간 다른 방법을 제안하였다.
그들은 복수개의 목적함수를 갖는 문제에 대하여 파레토 최적성에 근거하여 순위를 매긴 다음, 이에 근거하여 적합도를 배정하는 방법을 제안하였다.
이 방법에서는 개체의 순위를 그것의 지배 개체들의 수에 1을 더하여 랭킹을 할당한다. 이렇게 하면 모든 파레토 최적해의 랭킹은 1이 된다. 그림 2는 두 개의 목적함수를 갖는 경우에 해의 순위을 위의 방법에 의해 할당한 예이다.

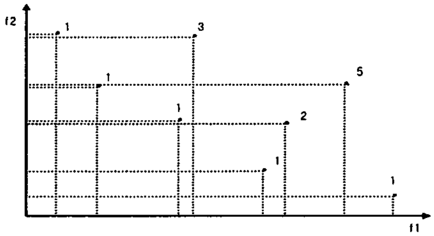

그림 2. Fonseca와 Fleming의 랭킹
Fig. 2. Fonseca and Fleming's ranking

2.1.3 파레토 토너먼트 방법

Horn, Nafpliotis, 그리고 Goldberg(1994)는 복수개의 목적 함수를 갖는 경우를 위한 토너먼트 선택을 제안 하였다.
그 랭킹 절차는 다음과 같습니다 :
① 선택을 위해 2개의 후보자가 모 집단으로부터 랜덤으로 선택한다.
② 개체들의 비교를 위한 비교군을 개체군으로부터 랜덤으로 선발한다.
③ 각 후보자는 비교군 안의 각 개체에 대해서 비교된다.
이 때 다음의 2가지 결과가 나올 수 있다.

첫째. 1개의 후보자가 비교군안의 개체들에 의해서 열성해인데 다른 개체가 비열성해라면, 그 비열성해 후보자가 재생을 위해 선택된다.

둘째. 양쪽 후보자가 모두 열성해이거나 비열성해라면 분배 방법을 사용하여 승자를 선택한다.

Horn과 다른 사람들이 제안한 분배 방법는 최소 적소 계산과 그 선택된 객체를 승자로써 선택하는 방법이다.
그 적소 계산은 후보자의 특정한 거리의 안쪽에 모집단에 개체들의 수를 세는 것에 의해서 계산된다.
다음의 그림 11은 두 개의 비열성해 사이의 분배 방법 사용의 예를 보여준다.
선택을 위한 두 후보자는 파레토 해들안에 있고 비교군에 의해 지배되지 않는다.

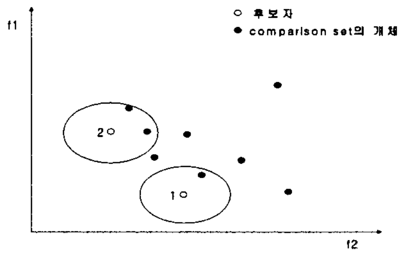

그림 3 후보자 선택 예
Fig. 3. example of choose the winner

개체군의 다양성을 유지하기를 위해서는 적은 적소 계산을 가지는 후보자를 최선으로 선택해야 한다.
위의 예에서 각 거리내의 개체 수를 보면 후보자 1은 1개 후보자 2는 2개이므로 후보자 1이 선택된다.

알고리즘의 성공은 개체군의 비교군의 크기의 파라메터에 좌우된다.
만약 작으면 선택된 해가 전체 개체군으로 보면 파레토 최적해가 되지 않을 확률이 높으므로 개체군안의 적은 수의 비지배해만이 결과로 나올 것이다.
만약 크면 대부분의 선택이 초기의 파레토 최적해가 될 가능성이 높으므로 급속 수렴하게 될 것이다.

2.2 시뮬레이션

본 논문에서는 기존의 연구에 사용되었던 테스트용 문제인 SRN문제와 이중 구속(Bicriteria) 통신망에 가중치 합과 Goldberg의 파레토 랭킹방법, Fonseca와 Fleming의 파레토 랭킹방법, 파레토 토너먼트 방법들을 비교 연구하고자 한다.

시뮬레이션(1)

SRN 문제 [9]

변수 범위가 $x_1 \in [-20, 20]$
$x_2 \in [-20, 20]$

목적 함수가 $f_1(X) = (x_1-2)^2 + (x_2-1)^2 + 2$
$f_2(X) = 9x_1 - (x_2-1)^2$

구속 조건이 $g_1(X) = x_1^2 + x_2^2 \le 225$
$g_2(X) = x_1 + 3x_2 \le -10$

집단 크기 100

최대 세대 수 500
교배 확률 0.7
돌연변이 확률 0.3

표 1. 알고리즘에 대한 비교
Table1. Comparison of algorithms

	연산시간	최저 f1값	최저 f2값	최적파레토
가중치합	5.20s	29.7462	-259.3673	27
Goldberg의 파레토 랭킹	8.98s	28.3353	-244.8951	31
Fonseca와 Fleming의 파레토 랭킹	8.66s	28.3353	-268.9314	29
파레토 토너먼트	17.31s	28.3353	-263.2171	34

위의 표에서 연산시간은 펜티엄866, 램 256메가의 환경에서 10회 평균값이다.
위의 결과를 가중치 합은 연산 시간은 가장 짧으나 파레토의 개수, 최저 값들이 다른 방법에 비해 떨어지는 것을 볼 수 있다. 파레토 랭킹방법들은 연산시간과 파레토의 개수가 다른 방법의 중간이고, 파레토 토너먼트 방법은 최저값이나 파레토의 개수는 다른 방법에 비해 좋으나 연산 시간이 가장 긴 것을 알 수 있다.

시뮬레이션(2)

이중 구속 통신망 설계에 대한 파라메터는 기존 방법[10]과 비교하기 위해 같은 값을 사용 하였다.

센터(n) 6
사용자(m) 30
센터별 최대 사용자 할당 수 (g_1) 10
센터별 연결 비용(w_{1ij}) [100-250]
센터와 사용자간 연결 비용(w_{2ij}) [100-250]
센터 최대 메시지 처리 용량(C_1) 300
제한 신뢰도($R_{\min}$) 0.9
목적함수:
딜레이 타임

$$\min \frac{1}{T}\left[\sum_i^n \frac{F_i(X)}{C_i - F_i(X)} + \sum_{i=1}^{n-1}\sum_{j=1}^{n} \beta_{ij} \cdot f_{ij}(X)\right]$$

코스트

$$\min \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w_{1ij}\cdot x_{1ij} + \sum_{i=1}^{n}\sum_{j=1}^{m} w_{2ij}\cdot x_{2ij}$$

구속함수 $s.t. R(x) > R_{\min}$

$$\sum_{j=1}^{m} x_{2ij} \le g_i,\ i=1,2,\cdots,n$$

$$\sum_{j=1}^{n} x_{2ij} = 1,\ j=1,2,\cdots,n$$

$$F_i(X) < C_i,\ i=1,2,\cdots,n$$

집단 크기 20
최대 세대 수 500
교배 확률 0.7
돌연변이 확률 0.3

위의 표에서 연산시간은 펜티엄866, 램 256메가의 환경에서 10회 평균값이다.

표 2. 알고리즘에 대한 비교
Table2. Comparison of algorithms

	연산시간	최저비용	최저지연시간	최적파레토
가중치합	44.15s	1499	0.005513	27
Goldberg의 파레토 랭킹	48.11s	1248	0.008267	31
Fonseca와 Fleming의 파레토 랭킹	46.7s	1298	0.00697	29
파레토 토너먼트	46.63s	1236	0.000845	34

위의 결과를 보면 가중치 합은 연산 시간은 가장 짧으나 파레토의 개수, 최저 값들이 다른 방법에 비해 떨어지는 것을 볼 수 있다. Goldberg의 파레토 랭킹방법은 가중치 합보다는 파레토의 개수, 최저 값들이 좋으나 다른 파레토 방법보다는 떨어지며 연산시간이 가장 길다. Fonseca와 Fleming의 파레토 랭킹방법은 Goldberg의 파레토 랭킹방법보다 모든 면에서 약간씩 좋은 결과를 보여주고 있다. 파레토 토너먼트 방법은 최저값이나 연산시간은 Fonseca와 Fleming의 파레토 랭킹방법과 비슷하나 최적 파레토의 개수가 가장 많다.

3. 결 론

본 논문에서는 기본적인 다중 목적 함수용 예와 Gen과 Kim이 제안한 네트워크 신뢰도를 고려한 연결 비용과 메시지 지연을 고려한 이중 구속(Bicriteria) 통신망 설계 문제를 사용하여가지고 가중치 합과 여러 가지 파레토 방법들을 비교하였다.
파레토 토너먼트 방법은 보다 복잡한 문제에서 더 효율적이고 가중치 합 방식은 연산속도는 빠르나 결과가 그리 좋지 못하며 파레토 랭킹 방식들은 그 중간 정도라는 것을 알 수 있다.

[참 고 문 헌]

[1] John R.Koza. GENETIC PROGRAMING. The MIT Press.
[2] I. De Falco, R. Del Balio, E. Tarantino, R. Vaccaro. Improving Search by Incorporating Evolution Principles in Parallel Tabu Search. IEEE International Conference of Evolutionary Computation(ICEC'94).Vol.2.1994
[3] Gen, M. and Y. Z. Li, "Spanning tree based genetic algorithm for bicriteria transportation problem", Proceedings of Japan China Joint Interantional Workshops on Information Systems, pp.123-134, 1997.
[4] Zhou, G., and M. Gen, "Approach to degree constrained minimum spanning tree problem using genetic algorithm", Engineering Design&Automation,Vol.3, No2, pp.157-165, 1997.
[5] Zhou, G., and M. Gen, "Evolutionary computation on multicriteria production process planning problem", Proceedings of IEEE International conference on Evolutionary Computation pp.419-424,1997.
[6] Zbigniew Michalewicz,"Genetic Algorithms + Data Structure = Evolution Programs", Springer-Verlag, New York. Third Edition. 1995.
[7] 문병로 " 유전알고리즘" 다성출판사 pp104-107. 2001
[8] Mitsuo Gen, Runwei Cheng " Genetic Algorithms and Enginnering Optimization" A Wiley-Interscience Publication pp118-123. 2000
[9] N.Srinivas and K. Deb, "Multiobjective function optimization using nondominated sorting genetic algorithms," Evol, Comput, vol.2, no 3 pp. 221-246, Fall 1995
[10] Kumar, A. , R., M. Pathak, and Y. P. Gupta, "Genetic - algorithm - based reliability optimization for computer network expansion",IEEE Transactions on Reliability, Vol. 44, No. 1, pp. 63-72, 1995