광역최적화 방법론의 비교 연구
Comparative study of some algorithms for global optimization

양승호, 이현주, 이재욱
(우편번호 : 790-784) 경상북도 포항시 남구 효자동 산 31번지
포항공과대학교 산업경영공학과
E-mail : grimaysh@postech.ac.kr

Abstract

Global optimization is a method for finding more reliable models in various fields, such as financial engineering, pattern recognition, process optimization. In this study, we compare and analyze the performance of the state-of-the-art global optimization techniques, which include Genetic Algorithm (DE,SCGA), Simulated Annealing (ASA, DSSA, SAHPS), Tabu & Direct Search (DTS, DIRECT), Deterministic (MCS, SNOBIT), and Trust-Region algorithm. The test functions for the experiments are Benchmark problems in Hedar & Fukushima (2004), which are evaluated with respect to efficiency and accuracy. Through the experiment, we analyse the computational complexity of the methods and finally discuss the pros and cons of them.

1. 연구배경

최적화는 제약된 조건 하에서 구하려는 목적함수의 최적값을 찾는 과정이다. 문제를 해결하기 위해 수식으로 모델화하고 이들 모델의 최적의 해를 구하는데 특히 광역 최적화는 여러 개의 국소 최적해가 있을 때 모델 전체의 최적해를 찾는 것이다. 현실의 많은 문제들은 "어떠한 목적을 가지고 행하는 과정" 이며 이는 최적화 문제로 모델링이 가능하다. 따라서 광역최적해를 찾는 알고리즘, 즉 좀 더 나은 해를 찾을 수 있는 방법론은 데이터 마이닝, 생물정보학(Bioinformatics), 무선통신, 금융공학(Financial Engineering) 등 다양한 분야로의 적용이 가능하다. 오늘날, 학계, 기업, 연구소 등, 다양한 분야에서 광역 최적화 알고리즘에 대한 수요가 크게 증가하고 있는 추세이다. 따라서 광역최적해를 구하는 방법론을 고찰해 보는 것은 큰 의의를 가진다.

기존의 비선형 최적화 알고리즘으로 도출된 해는 국소최적이라는 것만을 보장할 수 있기 때문에 광역최적해를 구하는 데에 한계가 있다. 이에 대한 대안으로 개발된 Tabu Search, 유전자 알고리즘, Simulated Annealing 등의 휴리스틱 방법을 통한 해는 광역 최적해 또는 그에 가까운 해라는 것을 보장할 수 있어 광역 최적기법으로 많이 사용되고 있다.

2. 알고리즘 종류

본 논문에서 사용된 알고리즘들에 대해 살펴보도록 하자. 우선 유전자 알고리즘은 자연 선택에 기반을 둔 생물학적 진화 과정을 모델링하여 최적화 문제를 푸는 방법으로 현재 Population에서 Individual들을 랜덤으로 선택하여 Parents로 정하고 Parents들이 다음 세대의 children를 생산하도록 한다. 연속적인 세대를 거친 후 Population은 최적해에 가까워 지도록 진화하게 된다.[1] 유전자 알고리즘에서 우리가 다루게 될 DE(Differential Evolution)는 Trial 모수벡터 만들어 내는 새로운 방법으로 가중치가 주어진 두 Population Member들 간의 차이를 제 3의 Member에 더하는 방법으로 모수벡터를 만들어낸다.[2] 유전자 알고리즘의 SCGA (Simplex Coding Genetic Method)는 유전자 알고리즘과 Nelder-Mead 방법을 결합하여 고안된 방법으로 새로운 Children이 생성될 때 Nelder-Mead 방법이 적용되어 초기단계와 중간단계의 Population을 개선하였고 선택 및 재생에는 Linear Ranking Selection, 교배 시 Multi-Parent Crossover 방법이 사용된다.[3]

다음으로 살펴볼 알고리즘은 Simulated Annealing으로 명칭과 Idea는 담금질에서 나왔으며 커다란 탐색공간에서 광역 최적화 문제에 대한 확률적 휴리스틱 접근방법이다. Steady

State로 수렴할 때까지 온도를 점차적으로 낮추어 가며 냉각 과정을 시뮬레이션한다.[4] Simulated Annealing의 ASA(Adaptive Simulated Annealing)에서는 담금질 시간 동안의 온도가 지수적으로 감소한다. 또한 재담금질을 도입하여 Multi-Dimensional Parameter Space에서 민감도의 변화에 적응하도록 하여 빠른 담금질 속도를 가진다.[5] DSSA(Direct Search Simulated Annealing)은 SSA(Simple Simulated Annealing)의 Cooling Schedule을 가속화시키고 SSA의 마지막 단계에서 전 단계에서 SSA로 얻은 Best Point에 Kelly' Modified Nelder-Mead Method를 적용하여 개선시키는 방법이다.[6] SA-HPS(Simulated Annealing Heuristics Pattern Search)은 Descent Direction을 생성하는 Add방법을 이용하여 Direction을 찾은 후 실제로 함수값을 개선시키는가를 Test하는 HPS 방법과 SA를 결합시킨 것이다.[7]

Tabu and DIRECT Algorithm에서 Tabu Search Algorithm의 DTS(Directed Tabu Search)는 Direct-Search에 기반을 둔 방법으로 Memory를 이용하여 목적함수 값이 개선되지 않더라도 이전에 탐색하였던 지역을 빠져나가 새로운 지역을 탐색하게 한다.[8] DIRECT (DIviding RECTangles) Algorithm은 탐색 구간 중앙의 목적함수 값을 샘플링함으로써 탐색을 시작하며 탐색 구간을 중앙을 기준으로 다시 부분 구간들로 분할하여 최적의 해를 찾는다.[9]

Deterministic 알고리즘은 주어진 Input에 대해 항상 똑같은 Output을 만들며, Underlying Machine은 항상 동일한 State들의 동일한 수열 하에서 작동하기 때문에 형태가 예측가능한 알고리즘이다. Deterministic의 SNOBFIT(Stable Noisy Optimization by Branch and Fit)은 추가적인 Soft(Hidden) 제약식을 가지는 Bound 내에서 변하는 연속 변수의 Noisy Optimization 문제를 Robust하고 빠르게 해결하는 알고리즘으로 Branching Strategy와 Quadratic Programming Method를 결합한다.[11] MCS(Multilevel Coordinate Search)는 N 차원의 탐색할 공간을 겹치는 부분이 없는 Hypercube(Box)들로 표현한 후 Box를 대표하는 점에서의 목적 함수 값과 Box의 크기를 참조하여 Axis Plane을 따라 Box들을 분할하여 최적 해를 찾아간다.[12]

Trust Region 알고리즘은 목적 함수의 특정 지역(Trust region)만을 이차형식으로 근사화하여 예측된 목적 함수값의 개선도와 실제 목적함수에서 관측되는 개선도의 비율을 이용하여 충분한 근사화가 이루어지면 지역을 확장하고, 실제 목적함수에 적합하지 않으면 지역을 축소해 가며 최적 해를 찾아간다.[13]

3. 실험 방법

본 논문에서는 실험하고자 하는 알고리즘을 대상으로 Hedar & Fukushima (2004)에서 제시된 벤치마크 문제를 사용하여 각 알고리즘들의 효율성 및 해의 정밀성을 평가한다.[8] 벤치마크 문제는 표1에서 제시하였다. 표1의 n은 입력 변수의 차원, Av F.E는 평균 함수계산 횟수, 그리고 S.R은 성공률이다. 평가를 위해 사용한 소프트웨어는 Matlab 7.04이며, 평가 항목은 각 문제들별로 모델의 최적해를 구하는데 걸린 함수계산 횟수의 최대값, 최소값 및 평균값, 실제 광역 최적해와의 평균 오차율 그리고 성공률로 이루어진다. 성공률 의 평가는 $|f^* - f| < \varepsilon_1 |f^*| + \varepsilon_2$ (f^*:광역 최적해, f: 모델을 한번 실험했을 때의 모델의 최적해, ε_1: 10^{-4}, ε_2:10^{-6})을 만족시키는 경우의 횟수를 더하여 총 시행횟수 100으로 나누어 계산하였다. 각 알고리즘 수행 시 사용한 Setting 조건들을 살펴보도록 하자. 기본적으로 각 알고리즘별로 제공하는 Setting 조건을 사용하였으며 추가적으로 바꾼 조건들에 대해서만 소개하도록 하겠다. 유전자 알고리즘의 DE은 Function evaluation의 최대값을 200으로 바꾸어서 실험하였다. Tabu & Direct Search의 DIRECT 알고리즘은 자체적으로 iteration을 돌면서 최적의 해를 찾기 때문에 알고리즘을 한번만 돌려 결과값을 측정하여 성공률을 구하였다. 따라서 성공률이 성공 혹은 실패의 두가지 경우만 존재하기 때문에 1 혹은 0의 결과로 나타난다. Trust-Region 알고리즘은 다른 알고리즘과의 형평성을 위하여 기울기정보 없이 실험하였다.

표1. 실험 결과표

No	Function Name	n	DE		ASA		DIRECT		SNOBFIT		Trust-Region	
			Av F.E	S.R	Av F.E	S.R	Av F.E	S.R	Av F.E	S.R	Av F.E	S.R
1	Branin RCOS	2	3000	100	1818.5	0	1295	1	92	98	26.67	100
2	Bohaschewvsky	2	1047	100	1570.5	100	1223	1	47	89	42.302	61
3	Easom	2	15	0	1001	0	59347	0	166	77	6.262	8
4	Goldstein&Price	2	358	11	1322	0	191	1	150	45	63.022	100
5	Shubert	2	15	0	1592	0	1811	0	124	79	20.58	1
6	Beale	2	935.10	100	1388	100	1143	1	197	6	58.196	100
7	Booth	2	867.15	100	1736.5	100	511	1	18	100	20.126	100
8	Matysd	2	710.4	100	1848	100	517	1	17	100	9.51	100
9	Hump	2	53.10	0	1903.5	100	511	1	200	0	38.216	100
10	Schwefel	2	49	0	1658.5	100	255	0	200	0	19.122	0
11	Rosenbrock	2	1583.10	100	1416.5	100	1913	1	200	0	106.59	100
12	Zakharov	2	725.40	100	1836.5	100	613	1	193	22	31.998	100
13	De Joung	3	1057.80	100	6396	100	16873	1	29	100	10.29	100
14	Hartmann	3	15	0	10151.5	0	4211	1	80	100	47.098	100
15	Colville	4	3000	0	9954.5	100	18261	1	.	..	204.35	100
16	Shekel	4	15	0	16500	0	255	0	510	63	58.826	97
17	Shekel	4	15	0	16894	0	1331	0	534	62	58.826	0
18	Shekel	4	15	0	16586	0	4939	0	508	60	58.826	0
19	Perm	4	3000	0	12031	100	20969	0	800	0	371.86	2
20	Perm	4	8000	84	11829.5	100	5203	1	738	24	139.27	74
21	Power Sum	4	3000	0	16725.5	100	20131	0	800	0	338.57	0
22	Hartmann	8	15	0	81150.5	0	303	0	607	92	137.83	0
23	Schwefel	8	88.80	0	108940	100	20443	0	1800	0	43.714	0
24	Trid	8	2062.2	100	58715.5	0	4897	1	577	100	61.814	100
25	Trid	10	18862	100	120761	0	40319	0	3943	40	144.81	100
26	Rastrigin	10	20000	0	108243	100	39891	0	5008	0	86.064	43
27	Griewank	10	1641	0	1001	100	22983	0	5008	0	365.62	84
28	Sum Squares	10	20000	0	135764	100	25903	0	129	100	154.29	100
29	Rosenbrock	10	20000	0	105137	100	30201	0	5008	0	723.035	100
30	Zakharov	10	20000	0	112277	100	20091	0	5008	0	225.43	100
31	Rastrigin	20	40000	0	81454	100	28127	0	.	.	163.42	43
32	Griewank	20	2822	8	1001	100	23535	0	.	.	242.25	100
33	Sum Squares	20	40000	0	109133	100	27621	0	375	100	454.62	100
34	Rosenbrock	20	40000	0	74512	100	41273	0	14524	100	1743.5	89
35	Zakharov	20	40000	0	102006	100	20067	0	.	.	539.28	100
36	Powell	24	48000	0	2000	0	30757	0	.	.	960.08	97
37	Dixon&Price	25	50000	0	102484	0	20943	0	27514	0	1863	67
38	Levy	30	60000	0	106798	100	41187	0	.	.	324.73	97
39	Sphere	30	60000	0	112639	100	20097	0	824	100	86.356	100
40	Ackley	30	60000	0	1001	100	33681	0	.	.	222.45	0

4. 실험 결과

각 알고리즘을 돌려본 결과는 표1과 같다. 본 논문에서는 유전자 알고리즘의 DE, Simulated Annealing의 ASA, Tabu & Direct Search의 DIRECT, Deterministic의 SNOBFIT, Trust-Region 알고리즘에 대한 결과를 정리하도록 하겠다.

5. 결과분석

이상으로 다양한 알고리즘을 대상으로 각 알고리즘들의 효율성 및 해의 정확성을 평가한다. 각 알고리즘들의 결과를 분석하면, 유전자 알고리즘의 DE의 경우 Large scale에 대한 결과값이 좋지 못하고 해를 찾는데 함수계산 횟수 및 걸리는 시간이 많음을 확인할 수 있었던 반면 n=2인 Small Scale에 대해서는 높은 정확도를 보임을 확인할 수 있다. Simulated annealing의 경우 Large 및 Small Scale의 모든 경우에 대해 높은 정확도를 보인 반면 Large Scale의 경우 함수계산 횟수가 많다.

Tabu & Direct Search의 DIRECT의 경우 역시 유전자 알고리즘의 DE의 경우와 같은 결과를 보인다. Deterministic의 SNOBFIT의 경우 대체적으로 최적의 해를 잘 찾았으나 못찾는 경우가 존재한다. 마지막으로 Trust-Region 알고리즘은 Small 및 Large Scale의 경우 모두 해를 잘 찾았으며, 함수계산 횟수가 적기 때문에 최적의 해를 찾는데 시간이 적게 걸림을 확인할 수 있다. 따라서 Trust-Region 알고리즘을 사용한 경우 광역 최적화 문제에 대해 가장 효율적임을 확인할 수 있다.

References

[1] Goldberg,David E, "Genetic Algorithms in Search", Optimization and Machine Learning, Addison Wesley, 1989.

[2] Rainer Stron and KennethPrice, " Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces", TR, 1995.

[3] A.Hedar and M.Fukushima, "Minimizing multimodal functions by simplex coding, genetic algorithm", Optimization Methods and Software, 18(2003)pp.265-282.

[4] S.Kirkpatrick and C.D.Gelatt and M.P.Vecchi, "Optimization by Simulated Annealing", Science, Vol 220, Number 4598, pp.671-680, 1983.

[5] L. Ingber, "Adaptive simulated annealing (ASA): Lessons learned", Control and Cybernetics, Vol. 25 No. 1,pp. 33-54, 1996.

[6] A.Hedar and M.Fukushima, "Hybrid simulated annealing and direct search method for nonlinear unconstrained global optimization", Optimization Methods and Software, 17(2002)pp.891-912.

[7] A.Hedar and M.Fukushima, "Heuristic pattern search and its hybridization with simulated annealing for nonlinear global optimization", Optimization Methods and Software, 19(2004)pp.291-308.

[8] A.Hedar and M.Fukushima, "Tabu Search directed by direct search methods for nonlinear global optimization", European Journal of Operation Research, 170 (2006) pp.329-349.

[9] M.Björkman and K. Holmström, "Global Optimization using the DIRECT Algorithm in MATLAB", 1999.

[11] W.Huyer and A.Neumaier, "SNOBFIT-Stable Noisy Optimization by Branch and Fit", ACM Transactions on Mathematical Software, 2000.

[12] W. Huyer and A. Neumaier, "Global optimization by multilevel coordinate search", Journal of Global Optimization 14 (1999), 331-355.

[13] Celis,M., J.E.Dennis and R.A. Tapia, "A trust region strategy for nonlinear equality constrined optimization", in Numerical Optimization 1994(P.Boggs, R.Byrd and R.Schnabel,eds), Philadelphia: SIAM 1985,pp.71-82.