

진화 알고리즘을 위한 가상 클러스터 기반 재조합 연산자 및 세대차 모델

Virtual Cluster based Recombination Operator and Generation Gap Model for Evolutionary Algorithm

최준석 · 서기성
Junseok Choi and Kisung Seo

서경대학교 전자공학과
E-mail: cjceous@hanmail.net, ksseo@skuniv.ac.kr

요약

본 논문은 실수 진화 알고리즘에 대해서 가상의 클러스터를 이용한 재조합 연산자 및 새로운 세대차 모델을 소개한다. 가상 클러스터의 자가 적응적인 크기 변화를 통해 자손의 생성범위를 적절히 조절하고, 선택과 대치를 포함한 진화방식을 개선하여 효율적인 세대차 크기를 구함으로써, 개체의 다양성 유지 및 탐색성능의 향상을 꾀하였다. 제안된 방법을 벤치마크 테스트 문제에 적용하여 G3 알고리즘과 CMA-ES 등과 성능을 비교하였다.

키워드 : Evolutionary Algorithm, Real Number Optimization, Virtual Cluster Generation Gap, PCX(Parent-centric crossover)

1. 서론

실수 최적화 문제 해결에 효과적인 ES(Evolution Strategy), DE(differential evolution) CMA-ES 등의 EA(Evolutionary Algorithm) 중에서 특히, 실수 파라미터 GA(Genetic Algorithm)이 특히 관심을 끌고 있다[1, 2]. 효과적인 실수 파라미터 GA 알고리즘 설계를 위하여 재조합 연산자에 대한 연구가 중점적으로 진행되고 있다.

실수 재조합 연산자인 UNDX(unimodal normal distribution crossover)[3], SPX(simplex crossover)[4] 그리고 SBX(simulated binary crossover)[5]에 대한 연구는 [2]에 잘 정리되어 있다. 이러한 실수 재조합 연산자들은 실수 탐색체간의 벡터 성분을 이용하여 자손을 생성한다. 전형적인 GA(genetic algorithm)와 달리 3개 이상의 부모를 연산하여 자손을 생성하는 방식이 우수한 성능을 보여 주고 있으며[4, 6, 7] 특히 다수의 부모를 사용하는 재조합 연산자 중 Deb[2]가 설계한 PCX(parent-centric recombination) 연산자는 기존의 다른 연산자보다 우수한 성능을 보여 주고 있다.

최적화 문제에서 빠르게 전역최적화로 접근하기 위하여 군집의 다양성을 고려한 세대차의 개념이 연구되었다. Satoh[8]가 제안한 진화 모형은 점진적인 진화 방식의 MGG(minimum generation gap)에서 UNDX 연산자를 사용하여 $\mu(=3)$ 개의 부모를 연산하여 $\lambda(=200)$ 개의 자손을 생성하고 그중 2개를 선택하여 다음 세대로 진화시키는 방식을 사용하였다[8]. 그 결과 MGG 모델에서 UNDX 연산자를 사용할 때, $\mu(=3)$ 의 부모를 이용하여 2개의 자손을 생성하는 것이 가장 우수한 결과를 보였다.

MGG의 연산속도를 향상시키기 위하여 G3(generalized generation gap)모델이 Deb에 의하여 제안 되었으며 G3 모델에서 다양한 연산자를 사용하여 자손과 부모의 수에 따른 성능의 관계를 실험을 통하여 증

명하였다. 그 결과 G3 모델은 PCX 연산자를 사용하였을 때 기존의 UNDX, SBX 그리고 SPX보다 우수한 성능을 보여 주었다[2].

그러나 기존의 PCX 연산자는 지역탐색에 치중한 나머지 전역탐색 능력이 부족하다. PCX 연산자의 우수한 성능을 유지하면서, 전역탐색과의 조화를 위하여 가상의 클러스터를 이용한 재조합 연산자 및 새로운 세대차 모델을 제안한다. 개체의 다양성 유지 및 탐색성능의 향상을 위하여, 가상 클러스터의 자가 적응적인 크기 변화를 통해 자손의 생성범위를 적절히 조절하고, 선택과 대치를 포함한 진화방식을 개선한다. 제안된 기법을 주요 벤치마크 문제에 적용하고, CMA-ES[9] 및 G3 모델의 결과와 비교 실험하였다.

2. 기존의 진화 알고리즘

2.1 PCX(Parent-centric recombination Crossover)

PCX(Parent-centric recombination Crossover) 연산자는 우수해를 중심으로 자손을 생성하는 연산자다(그림 1). 자손의 생성 수식은 식(1)과 같이 표현된다.

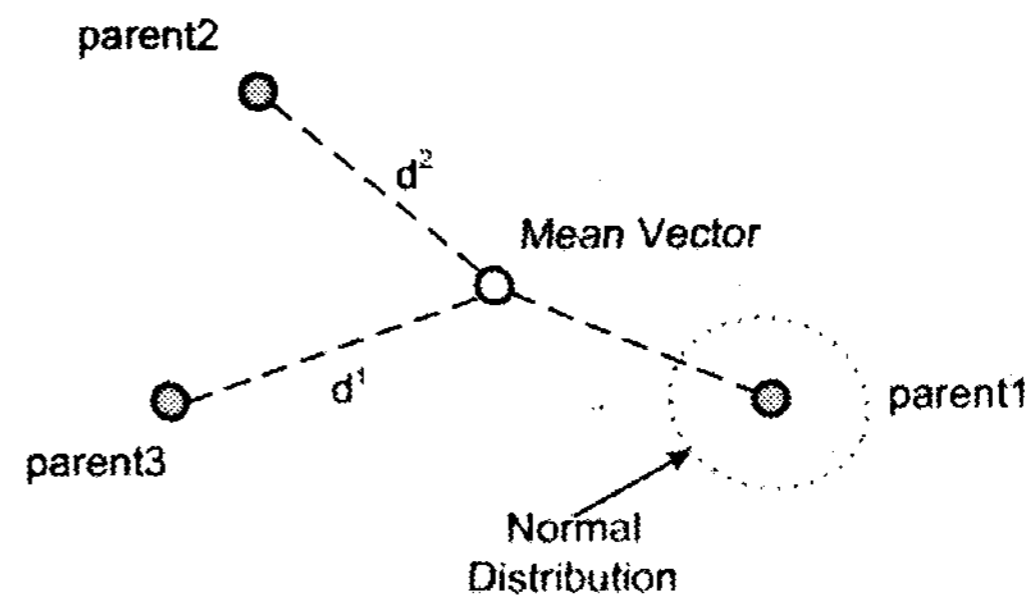


그림 1. Parent-centric recombination Crossover(PCX)

$$\vec{y} = \vec{x}^{(p)} + \omega_{\zeta} \vec{d}^{(p)} + \sum_{i=1, i \neq p}^u \omega_{\eta} \vec{De}^{(i)} \quad (1)$$

2.2 기존의 세대차 모델

먼저 Satoh가 제안한 MGG(minimal generation gap model)에 대하여 설명한다[8].

1. 전체 군집 P에서 μ 개의 임의의 부모를 선택한다.
2. μ 개의 부모를 재조합 연산자를 사용하여 λ 개의 자손을 생성한다.
3. 전체 군집 P에서 두 개의 부모를 임의로 선택 한다.
4. 두 개의 부모중, 하나는 생성된 λ 개의 자손중 가장 우수한 것으로 대체되고, 또 다른 하나는 λ 개의 자손과 선택된 두 개의 부모 가운데 룰렛선택 과정으로부터 얻어진 해로 대체된다.

두 번째로 설명하는 세대차 모델은 MGG 모델의 진화 속도를 향상시키기 위하여 룰렛선택을 대체한 G3 모델이다. G3(generalized generation gap) 모델은 PCX연산자를 사용할 때 기존의 점진적인 진화 모델 중 가장 우수한 결과를 보이고 있다[2].

1. 전체 군집 P에서 최우수 개체와 $\mu-1$ 개의 임의의 부모를 선택한다.
2. μ 개의 부모를 재조합 연산자를 사용하여 λ 개의 자손을 생성한다.
3. 전체 군집 P에서 두 개의 부모를 임의로 선택 한다.
4. 두 개의 부모와 λ 개의 자손의 합으로부터 두 개의 우수해를 선택하고, 이를 단계 3에서 선택된 두 부모와 대체시킨다.

3. 세대차 모델 및 가상 클러스터 기반 재조합 연산자의 설계

3.1 세대차의 분석과 새로운 자손의 대처방식

세대차는 일정한 세대에서 자손과 부모가 대체되는 양을 의미하며 세대차의 크기는 개체의 분포에 영향을 준다[10]. 특히 PCX 연산자를 사용할 때 세대차는 전체 개체들의 수렴과 분산에 많은 영향을 주게 된다. 따라서 자손과 부모가 대체되는 확률을 이용하여 최적화 문제에 적합한 세대차를 도출하고자 한다.

세대차에 따른 최적화 알고리즘의 성능을 분석하기 위하여 세대차 조절이 가능한 파라미터와 자손의 대처방식이 요구된다. 각각의 최적화 문제에서 일정한 세대차를 유지하기 위하여 생성된 자손을 다음세대로 선택할 때는 특정한 기준이 필요하다. 하지만 최적화 문제의 특성, 탐색영역 그리고 적합도의 변화를 고려해야 되기 때문에 각각의 세대에서 기준값을 설정하기 어렵다. 본 논문에서는 다음과 같이 세대차를 조절할 수 있는 대처방식을 설계하였다.

1. 전체군집 P에서 임의의 부모 1개를 선택한다.
2. 나머지 1개의 부모는 전체군집에서 토너먼트 선택을 이용하여 선택한다.
3. 전체 군집 P에서 선택된 부모와 생성된 2개의 자손과 비교하여 가장 우수한 2개의 후보해를 다음세대로

진화시킨다.

PCX 연산자는 2개 이하의 자손을 생성할 때 가장 우수한 결과가 나오는 것은 기존의 실험에서 증명되었다[2]. 본 논문의 세대차에 따른 실험에서도 2개의 자손을 생성할 때 우수한 결과가 나오는 실험으로 확인하였기 때문에 세대차 모델의 자손의 수를 2개로 제한하였다.

표 1. 세대차 조절을 위한 파라미터

w_{ζ}	Tournament size in proposed replacement scheme	Generation gap
$w_{\zeta} = 0.1$	2	$\approx 90\%$
$w_{\zeta} = 0.1$	4	$\approx 80\%$
$w_{\zeta} = 0.1$	12	$\approx 70\%$
$w_{\zeta} = 0.18$	12	$\approx 65\%$
$w_{\zeta} = 0.3$	12	$\approx 60\%$

아래의 그림 2는 Rosenbrock 함수에서 표 1의 파라미터를 사용하여 각각의 대체확률에 따라 성능을 실험한 것이다. $1e-20$ 이하의 전역 최소점을 찾을 때까지의 연산량을 표시하였다. 그림 2에서 나타나듯이 실험을 통하여 65%의 대체확률을 가질 때 가장 우수한 결과를 보였다.

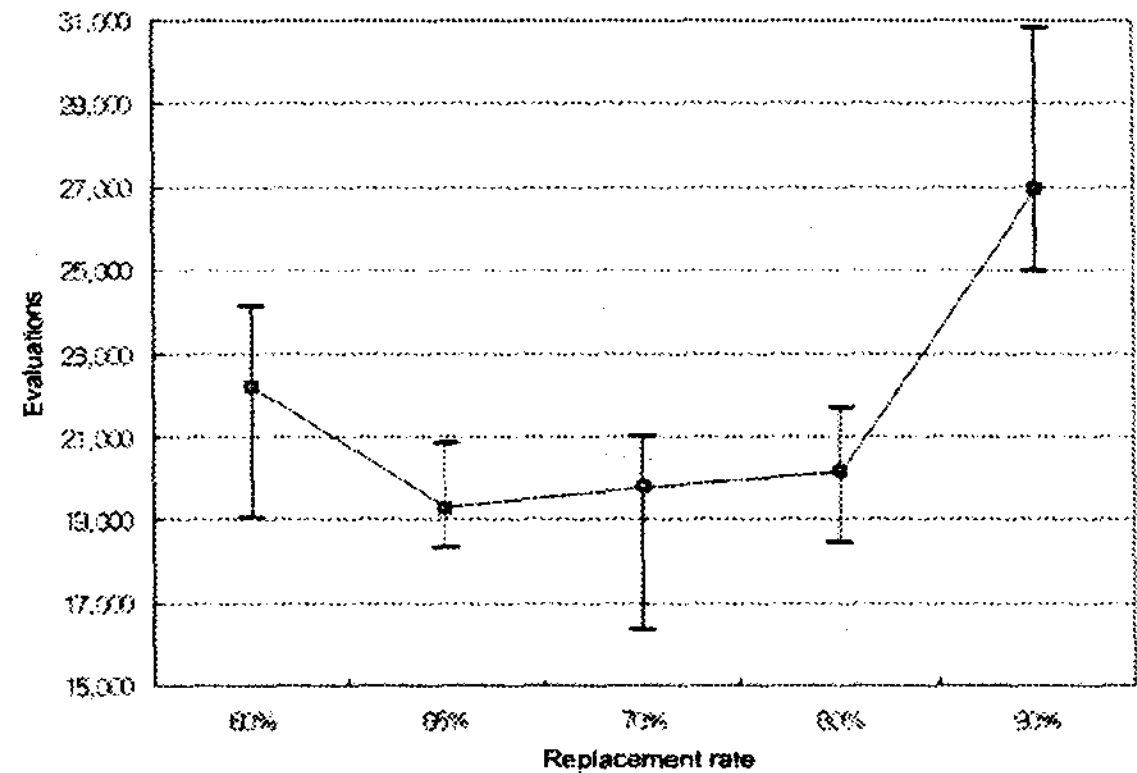


그림 2. Rosenbrock 함수에서 세대차에 따른 성능분석

3.2 가상 클러스터 기반 재조합 연산자

제안된 가상 클러스터 기반 재조합 연산자는 가상의 클러스터를 사용하여 자손의 생성범위를 조절함으로써, 군집내 개체의 다양화를 유지시키기 위해서 설계되었다.

즉, 그림 2처럼 자손의 생성범위를 제어하기 위하여, 각 부모가 속하는 영역 범위를 가상의 클러스터를 사용하여 조절한다.

재조합 연산자에 사용되는 부모의 수만큼 가상의 클러스터를 만들고 각각의 가상군집의 크기를 설정한다. 각각의 부모 클러스터의 크기를 이용하여 다른 부모와 일정한 거리를 유지시켜 주기 때문에 현재의 최우수 개체가 지역적으로 수렴하여도 적절한 탐색거리를 유지할 수 있다. 따라서, 기존 PCX 연산자의 단점인 지역해 수렴시에 이로부터 탈출하는데 많은 연산량이 소모되었던 점을 개선할 수 있다.

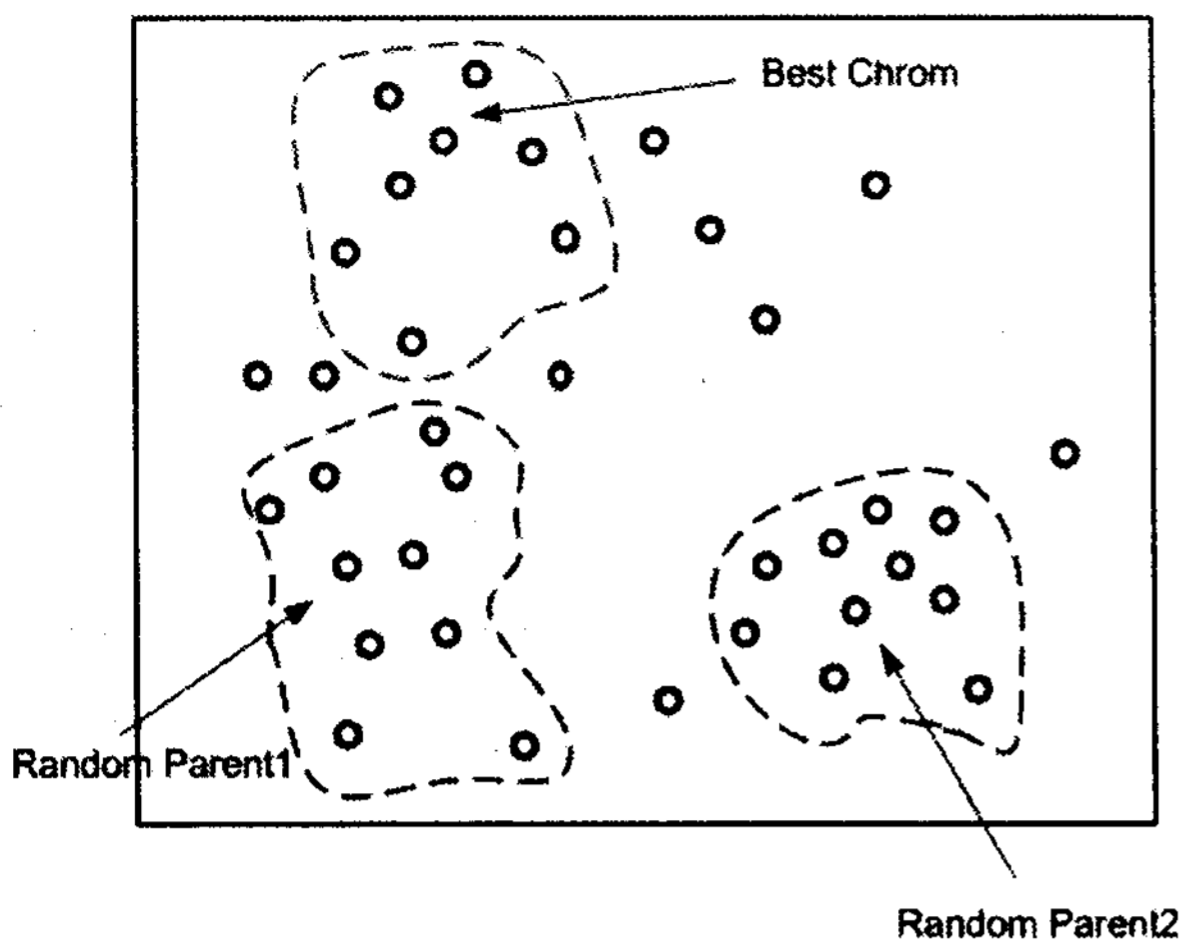


그림 3. 부모들의 가상 군집화

식 (2)와 식 (3)를 이용하여 탐색영역의 변화에 따라서 우수해 클러스터의 크기를 조절하게 하였고 임의의 부모의 클러스터의 크기는 25로 고정하였다.

$$Dist(g) = Dist(g_{s-1}) + \left(\sum_{j=1}^{\mu-1} \sum_{i=1}^n |X_{Best_parent}(i) - X_{parent(j)}(i)| \right) / (\mu-1) \quad (2)$$

$$Cluster_size = \left\lfloor \ln \left(\frac{Dist(g)}{Dist(g-S)} + 1 \right) \right\rfloor \quad (3)$$

initsize=15, S=10

3.3 VCGG(Virtual Cluster based Generation Gap)

새로운 세대차 방식과 가상 클러스터 기반 재조합 연산자를 통합하여 VCGG(Virtual Cluster based Generation Gap) 모델을 구성한다.

통합된 VCGG 모델은 아래와 같이 수행된다.

1. 군집 P, $\mu-1$ 개의 임의의 부모와 하나의 우수한 부모를 선택한다.
2. μ 개의 선택된 부모를 PCX 연산자를 이용하여 λ 개의 자손을 생성한다.
3. λ 개의 자손을 3.1의 대치방식을 사용하여 다음세대로 자손을 진화 시킨다.
4. 만일 수식(2)에서 $Dist(g) - Dist(g-S)$ 가 0.005 보다 크면 우수해의 클러스터의 크기를 10 세대마다 식(3)을 이용하여 제어한다. 그렇지 않으면 클러스터의 크기를 최초 설정한 15로 고정한다.

우수해 클러스터의 크기에 따라서 결과로 얻어지는 세대차는 조금씩 다르게 나오지만 평균 70%의 세대차를 가지고 있다. 이미 가상의 클러스터를 사용하여 자손의 생성범위를 조절하였기 때문에 추가적으로 ω_c 를 조절하면 오히려 성능이 저하되었다. 그러므로 VCGG 모델은 ω_c 를 0.1, 토너먼트 크기를 12로 고정하여 70%의 세대차에서 PCX 연산자의 탐색영역을 기준으로 우수해 군집의 크기를 조절하게 설계하였다.

4. 실험결과

4.1 실험 파라미터 설정

본 논문에서 제안한 VCGG(virtual cluster generation gap) 모델을 7개의 벤치마크 함수에 대하여 CMA-ES[9]와 PCX 연산자를 사용한 G3[2]모델과 비교한다. CMA-ES는 2004년 Hansen[9]의 Matlab 코드를 사용하였으며 G3 알고리즘은 Deb의 C++ 코드를 사용하였다.

G3 모델은 가장 우수한 조합인 $\mu=3, \lambda=1$ 개로 설정하였다. 그리고 PCX 연산자의 확률변수(ω_c, ω_η)는 모두 0.1로 고정하고 군집의 크기 P는 100으로 설정하였다[4].

VCGG 모델은 (ω_c, ω_η)는 모두 0.1, $\mu=3, \lambda=2$, 군집의 크기 P는 100 그리고 토너먼트 사이즈(T)는 12로 설정하였다. 나머지 파라미터는 식(2)와 식(3)의 파라미터를 사용하였다.

각각의 함수에서 문제의 크기(n)는 20이고, 변수 범위는 [-5.12, 5.12]로 설정하였다. 성능평가는 전역 최소값이 $1e-20$ 이 나올 때까지의 연산량으로 성능을 평가한다.

4.2 실험 함수

테스트 문제로 Deb[2], Hansen[9] 그리고 Raghuwanshi[11]가 제안한 다음 7개의 함수를 사용하였다. 비교 알고리즘에 대하여 같은 조건에서 실험 하였고, 각각의 문제에 대해서 20번의 실험을 반복하였다.

$$F_{sphere} = \sum_{i=1}^n x_i^2 \quad (4)$$

$$F_{elp} = \sum_{i=1}^n i x_i^2 \quad (5)$$

$$F_{Zak} = \sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5 i x_i^2 \right)^2 + \left(\sum_{i=1}^n 0.5 i x_i \right)^4 \quad (6)$$

$$F_{sch} = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2 \quad (7)$$

$$F_{ros} = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2) + (1 - x_i)^2) \quad (8)$$

$$F_{twoax} = \sum_{i=1}^{n/2} 10^6 x_i^2 + \sum_{i=n/2}^n x_i^2 \quad (9)$$

$$F_{ell} = \sum_{i=1}^n (10^6)^{\frac{i-1}{D-1}} x_i^2 - 450 \quad (10)$$

4.3 실험결과

본 논문에서 제안한 VCGG 알고리즘을 CMA-ES 그리고 G3와 비교한 결과와 표 2에 나와 있다. 그 결과 대부분의 실험 함수에서는 VCGG 모델이 가장 우수한 결과를 보였다. 먼저 기존의 두 알고리즘인 CMA-ES와 G3를 비교해 보면 F_{zak} 과 F_{ell} 을 제외한 나머지 5개 함수에 대해서 G3가 우수함을 보인다.

제안된 VCGG 기법의 경우, CMA-ES에 비해서는 7개 함수 중 6개, G3에 대해서도 마찬가지로 7개 함수 중 6개 함수에 대해서 우수함을 보였다. 참고적으로,

F_{ell} 함수에서는 CMA-ES가 가장 우수하였으며 F_{sphere} 함수에서는 G3모델이 가장 좋은 결과를 보여주고 있다.

제안된 방법이 대부분의 문제에서 최신의 연구 결과중 가장 경쟁력이 있는 두 알고리즘에 비해서 성능이 향상되었을 보여준다.

표 2. 실험함수 성능평가 비교

함수 \ EA	CMA-ES	G3	VC GG
F_{sphere}	5,621	3,670	4,027
F_{elp}	8,472	6,800	6,011
F_{zak}	14,057	16,254	12,884
F_{sch}	18,602	15,600	14,200
F_{ros}	33,049	21,452	18,662
F_{twoax}	35,021	27,938	23,599
F_{ell}	10,865	44,117	33,231

5. 결론

진화 알고리즘에서는 실수 연산자와 세대차 방법의 설계가 탐색성능에 매우 큰 영향을 미친다. 본 논문에서는 기존 PCX 연산자의 우수한 성능을 유지하면서, 전역탐색과의 조화를 위하여 가상의 클러스터를 이용한 재조합 연산자 및 새로운 세대차 모델을 제안하였다.

가상 클러스터의 자가 적응적인 크기 변화를 통해 자손의 생성범위를 적절히 조절하여 개체의 다양성을 유지하고, 선택과 대치를 포함한 세대차 진화방식을 개선하여 탐색성능을 향상 및 기존의 G3 모델보다 전역최적화로 다가가는 속도를 높였다.

제안된 기법을 주요 벤치마크 문제에 적용하고, CMA-ES[9] 및 G3 모델과 비교 실험하였다. 실험결과 대부분의 테스트 함수에서 G3와 CMA-ES 모델보다 향상된 성능을 보였다.

참 고 문 헌

[1] H.-P. Schwefel, *Evolution and Optimum Seeking*, John Wiley & Sons, Inc, 1995.
 [2] K. Deb, A. Anand and D. Joshi, "A Computationally Efficient Evolutionary Algorithm Conference for Real-Parameter Optimization." *Evolutionary Computation*, pp. 10(4): 371-395, 2002.
 [3] I. Ono and S. Kobayashi, "A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover." *In Proceedings of the Seventh International Conference on Genetic Algorithms(ICGA-7)*, pp. 246-253, 1997.
 [4] T. Higuchi, S. Tsutsui, and M. Yamamura, "Theoretical analysis of simplex crossover for real-coded genetic algorithms," *In Proceedings of Parallel Problem Solving from Nature (PPSN-VI)*, pp. 365-374, 2000.
 [5] K. Deb and R. B. Agrawal, "Simulated binary

crossover for continuous search space," *Complex Systems*, 9(2) pp. 115 - 148, 1995.

[6] H. Kita, I. Ono and S. Kobayashi, "Multi-parental Extension of the Unimodal Normal Distribution Crossover for Real-Code Genetic Algorithm," *In Proceedings of IEEE Conference on Evolutionary Computation, Vol. 2* pp. 1581-1587, 1999.

[7] O. Takahashi, H. Kite, S. Kobayashi, "A Distance Dependent Alternation on Real-coded Genetic Algorithms," *In Proceedings of IEEE Conference on Evolutionary Computation*, pp. 619 - 624 Vol .1, 1999.

[8] H. Satoh, M. Yamamura, and S. Kobayashi, "Minimal generation gap model for GAs considering both exploration and exploitation," *Methodologies for the Conception, Design, and Application of Intelligent Systems*, pp. 494 - 497, World Scientific, Singapore, 1996.

[9] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies," *In Proceedings of IEEE Conference on Evolutionary Computation*, pp. 312 - 317, 1996.

[10] Sarma. J and De Jong. K, *Generation gap methods*. Hand book of Evolutionary Computation, pages C2.7:1-C2.7:5, Oxford University Press, UK, 1997.

[11] M. M. Raghuwanshi and O. G. Kakde, "Distributed Quasi Steady-State Genetic Algorithm with Niches and Species," *International Journal of Computational Intelligence Research*, Vol. 3, No. 2, pp. 155-164, 2007