

개선된 Bacteria Chemotaxis 알고리즘을 이용한 전역적 탐색 기법

박중원⁰ 박정은 오경환

삼성전자 디지털 미디어 총괄⁰, 서강대학교 컴퓨터학과
kindjjong@hanmail.net⁰, {jepark, kwoh}@sogang.ac.kr

Global Search Strategy using Enhanced Bacteria Chemotaxis algorithm

Park Jong Won.⁰, Park J.E., Oh K.W.

Digital Media Business, Samsung Electronics⁰, Dept. of Computer Science, Sogang University

요 약

함수 최적화는 주어진 자원의 한도 내에서 최대의 이익 혹은 최소의 손실을 내는 최선의 결정을 내리는 것을 목표로 한다. 본 논문은 Müller의 연구를 바탕으로 박테리아의 주화성을 형상화한 'Chemical Sensing Bacteria Chemotaxis'라는 알고리즘을 제안한다. 이 알고리즘은 multimodal 환경에서의 전역 탐색을 목표로 한다 또한 실험을 통해, 제안 알고리즘의 타당성을 분석하고, 결과적으로 제안 알고리즘이 기존의 자연계 기반의 알고리즘에 비해 경쟁력이 있음을 입증하였다.

1. 서 론

주어진 함수의 최소값을 찾는 문제는 오랜 시간 동안 연구되어 온 주제 중의 하나이다. 많은 연구자들이 이 문제를 연구한 결과, 성능이 우수한 여러 결정적(deterministic) 알고리즘들이 설계되었다. (Dennis and Schnabel, 1989) 하지만, 일반적으로 이러한 결정적 알고리즘들은 주어진 환경에 따라 그 성능에 한계를 갖는다. 특별히, 주어진 함수가 multimodal 이고, 미분 불가능하며, 불연속적인 경우, 일반적인 결정적 알고리즘들은 전역 최적값이 아닌 지역 최적값에 수렴하는 경향이 있다. 이러한 문제를 해결하기 위하여 많은 확률적 알고리즘들이 연구되고 있다. 특별히 최근에는 자연계 기반의 많은 확률적 알고리즘이 연구되고 있다.[1]

자연계 기반 알고리즘은 곤충의 먹이 찾는 모습이나, 우수한 개체의 생존율이 더 높은 자연계의 생존 법칙과 같은 자연계의 여러 모습들을 알고리즘에 응용한 것이다. 대표적인 알고리즘으로는 유전자 알고리즘(Genetic Algorithm)이 있으며, 시뮬레이티드 어닐링(Simulated Annealing), 개미 시스템 알고리즘(The Ant System), 집단 지능 알고리즘(Particle Swarm Optimization) 등도 이에 포함된다.

최근에는 Müller에 의해 새로운 Bacteria Chemotaxis(BC, Müller et al, 2002) 알고리즘이 제안되었다. BC 알고리즘은 박테리아의 주화성(chemotaxis)을 형상화 한 것으로 Dahlquist 및 Brown, Berg의 생물학적 연구에 기반을 두고 있다. (Dahlquist 1976 ; Brown and Berg, 1974)[2][3]

본 논문은 Müller의 Bacteria Chemotaxis 알고리즘을 개선하여 보다 효율적이고 우수한 성능을 갖는 전역적 탐색 기법인 Chemical Sensing Bacteria Chemotaxis(CSBC) 알고리즘을 제안하고 있다. 또한 실험을 통하여, 제시한 알고리즘의 타당성을 분석하였다.

2절에서는 본 논문에서 제안 알고리즘을 2차원 모델에서 n차원 모델까지 확대해가며, 그 탐색 기법을 소개하고, 3절에서는 실험을 통해 CSBC 알고리즘이 우수한 성능을 가짐을 기존 알고리즘들과의 비교를 통하여 보였다. 끝으로 4절에서는 위의 내용을 바탕으로 결론 및 향후 연구 과제에 대해 기술하였다.

2. Chemical Sensing Bacteria Chemotaxis 알고리즘

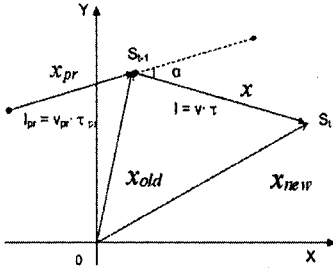
제안된 알고리즘은 Dahlquist의 연구(Dahlquist, F. W. et al., 1976)와 Müller 및 그의 동료들의 논문(Müller, S. D. et al., 2002)을 기반으로 하고 있으며, multimodal 환경에서의 미분불가능하고, 불연속인 함수의 전역 최소값을 찾는 것을 그 목표로 하고 있다. 제안 알고리즘은 기존의 chemotaxis algorithm(Bremermann, H. J., 1974)이나 Müller의 연구에 대해 다음과 같은 점에서 차이가 있다.[2][3][4]

- 화학감각기관이라는 개념을 추가함으로써 주화성(chemotaxis)를 좀더 효과적으로 표현했다.
- 박테리아의 속도 v 를 정의함으로써 박테리아가 지역적 최소값에서 벗어나 전역적 최소값을 찾도록 하였다.
- 박테리아의 특정 방향에 대한 의존성을 낮추었다. 생명체는 정확한 방향과 정확한 거리까지는 아니어도, 자신을 자극하는 자극체의 대략적 위치는 파악할 수 있다. 제안된 알고리즘은 화학감각기를 가진 박테리아가 자신을 양성으로 자극시키는 화학물질이 있는 곳으로 이동하는 습성을 이용하여 기존의 알고리즘들 보다 더 빠르고 정확하게 전역 최소값을 찾아낸다. 이번 장은 Dahlquist에 의해 제안된 박테리아의 습성에 대한 2차원 모델에 대해 설명한 후에 화학감각기의 정보를 이용한 탐색기법을 제시하도록 하겠다.[3]

2.1 박테리아의 2차원 모델

박테리아의 2차원 모델은 다음과 같이 정의된다.

- 박테리아의 움직임은 직선의 자취 벡터 형태로 표현되며, 이를 위해 필요한 인자들은 속도, 방향, 지속도이다.
- 속도는 물농도 탐색의 범위를 결정한다.
- 박테리아가 경로를 변경할 때는 주변환경의 물농도에 영향을 받아 결정되며, 그 방향은 확률밀도 함수에 의해 결정된다.
- 지속도는 지수적 감소(exponential decaying) 확률밀도 함수에 따라 변한다.
- 각 인자의 각각의 실행 확률밀도 함수는 독립적이다.



s_{t-1}, s_t : time t-1과 t에서의 박테리아 탐색지점
 x_{old}, x_{new} : time t-1과 t에서의 박테리아 탐색지점을 나타내는 위치벡터
 x_{pr} : time t-2와 t-1 사이의 박테리아 이동지위를 나타내는 벡터
 x : time t-1과 t사이의 박테리아 이동지위를 나타내는 벡터(실제 time t에서 생성되는 벡터)
 그림 1. 박테리아의 2차원 모델

2.2 박테리아의 탐색 전략

박테리아는 기본적으로 주화성에 의해 run 과 twiddling의 상태를 반복하며 최종 목적지를 찾아간다. (Biondi, S. A. et al., 1998) 이것으로부터 박테리아의 상태(status)는 swimming, seeking, perturbation 의 3가지로 정의된다. swimming 상태는 박테리아가 탐색공간 전체를 run과 twiddling을 반복하며 탐색하는 것이고, seeking은 높은 물농도가 집중되어 있는 일정한 지점을 세밀하게 탐색하는 상태를 의미한다(exploitation). 또한 perturbation은 지역적 한계값을 벗어나기 위해 보다 먼 곳으로의 탐색을 시도하는 상태이다(exploration).[5]

2.3 CSBC 알고리즘

제안된 알고리즘은 박테리아의 지속도와 움직임의 속도 그리고 이동 방향을 매 단계 갱신하는 것을 골자로 한다. 지속도 τ 는 박테리아의 자취 길이에 영향을 미치며 다음의 확률밀도함수를 따른다.

$$P(X = \tau) = \frac{1}{T} e^{-\tau/T}$$

여기서,

$$\mu = E(X) = T, \quad \sigma^2 = Var(X) = T^2$$

$$T = \begin{cases} T_0, & \text{for } \frac{f_{pr}}{l_{pr}} \geq 0 \\ T_0 \left(1 + b \left| \frac{f_{pr}}{l_{pr}} \right| \right), & \text{for } \frac{f_{pr}}{l_{pr}} < 0 \end{cases}$$

b : gradient coefficient
 $f(x, y)$: function to minimize
 $f_{pr} = f(x_i, y_i) - f(x_{i-1}, y_{i-1})$
 $l_{pr} = |x_{pr}|$

또한 속도 v 는 박테리아의 활동성을 표현하며, 화학감각기의 한계 인지 범위를 뜻한다. 이 값은 지속도 τ 의 상수배로 표현되며 박테리아의 탐색상태에 따라 값이 달라진다.

$$v = k \cdot \tau, \quad k = \begin{cases} k = 1, & \text{if (status = run)} \\ k < 1, & \text{if (status = swim)} \\ k > 1, & \text{if (status = perturbation)} \end{cases}$$

끝으로, 탐색 방향 α 는 박테리아의 이전 움직임의 방향과 새로운 진행 방향 사이의 각도이다. 박테리아의 진행방향은 현재의 위치에서 박테리아의 화학 감각기를 가장 크게 자극하는 쪽이 선택이 되며, 실제적인 값은 Gaussian 확률 분포에 의해 결정된다.

$$P(X = |\alpha|, v = \mu) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left[-\frac{(\alpha - v)^2}{2\sigma^2} \right]$$

위의 세가지 인자가 결정된 후에는 벡터의 합 연산에 의하여 새로운 위치 벡터 x_{new} 가 결정된다. (Müller et al.)

$$\begin{aligned} l &= v\tau \\ r & \quad r \\ x &= n_u l \\ r & \quad r \quad r \\ x_{new} &= x_{old} + n_u l \end{aligned}$$

2.4 지역적 최소값을 벗어나기 위한 전략

박테리아는 스스로 자신의 상태(status)를 변화시키며 지역적 최소값에서 벗어난다. 처음 박테리아가 탐색을 시작하였을 때의 상태(status)는, swimming이다. 이 상태에서 탐색 중, 박테리아가 지역(region) δ 에서 최소값을 갱신한 후, 다시 지역(region) δ 에서 탐색을 시도하면, 박테리아의 상태는 seeking으로 바뀐다. Seeking 상태에서는 박테리아의 활동속도 v 를 낮춤으로써 박테리아의 활동거리 l 과 감각기관의 민감도 η 를 낮춘다. 이것은 결과적으로 지역 탐색(local search)으로 이어진다.

한편, 특정지역 δ 에서 특정 시간 κ 동안 최소값을 변경시키지 못하게 되면, 상태는 perturbation이 된다. 이 상태에서는 v 를 증가시켜 박테리아의 활동성을 높여주고, 감각기관의 민감도 η 역시 증가시킴으로 박테리아가 지역적 한계를 벗어나 전역적 탐색이 가능하도록 해준다.

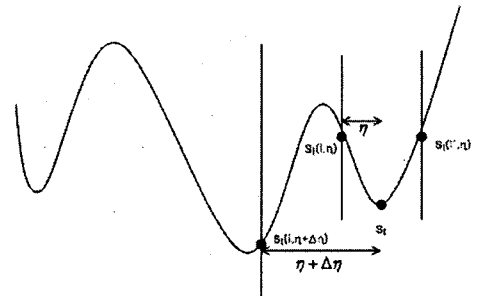


그림 2. Perturbation 상태에서의 감각기관의 민감도 η 증가

$$\eta = \begin{cases} \eta + V\eta, & \text{if } f < \min_{i \in \{1, \dots, \omega\}} f_i \\ \eta, & \text{if } f \geq \min_{i \in \{1, \dots, \omega\}} f_i \end{cases}$$

ω : 감각기관의 탐색 범위각의 수

f : s_i 에서의 물농도

f_i : α_i 에서의 물농도

$\Delta \eta$: 민감도의 증분

3. 실험 및 결과

제안된 알고리즘의 성능을 평가하기 위하여 기존에 알려진 유전자 알고리즘과 시뮬레이티드 어닐링 그리고 박테리아 알고리즘이 사용되었다.

처음에 사용된 평가함수는 $F_1 = (x-1)^2 + (x-1)^2$ 이다. 이 함수는 sphere 2-D 함수로서 (0,0)에서 최소값을 갖는다. 이 경우 GA와 BC 알고리즘은 대략의 최소값을 찾아낸 반면, SA와 CSBC는 매우 정확한 값을 찾아내는 것을 알 수 있다. 더욱이 CSBC의 경우 SA보다 평균적으로 약 5.88×10^4 배의 적은 평가 함수 호출만으로도 약 9.49×10^5 배의 높은 정밀도를 나타내고 있다.(표 1)

표 1. F_1 에 대한 실험 결과

| trial | GA | | SA | |
|--------|---------------------------|----------------|---------------------------|----------------|
| | # of Function Evaluations | Function value | # of Function Evaluations | Function value |
| 1 | 2000000 | 0.011 | 99043000 | 0.0000001 |
| 2 | 2000000 | 0.01 | 96671000 | 7.20E-07 |
| 3 | 2000000 | 0.019 | 99186000 | 1.77E-07 |
| 4 | 2000000 | 0.000456 | 9854000 | 4.91E-07 |
| 5 | 2000000 | 0.000457 | 89792000 | 1.39E-07 |
| Mean | 2000000 | 0.0081826 | 96769200 | 3.26E-07 |
| St. D. | 0 | 0.007037 | 3590992.14 | 2.41E-07 |

| trial | BC | | CSBC | |
|--------|---------------------------|----------------|---------------------------|----------------|
| | # of Function Evaluations | Function value | # of Function Evaluations | Function value |
| 1 | 1145 | 0.0461 | 21544 | 1.20E-14 |
| 2 | 1292 | 0.05978 | 17818 | 1.04E-12 |
| 3 | 1349 | 0.099603 | 11933 | 1.20E-13 |
| 4 | 1395 | 0.059302 | 7730 | 3.24E-13 |
| 5 | 1505 | 0.01498 | 14992 | 2.17E-13 |
| Mean | 1337.2 | 0.055953 | 14803.4 | 3.43E-13 |
| St. D. | 118.845 | 0.0272 | 4750.39423 | 3.64E-13 |

두 번째 실험에 사용된 평가함수는 최적화 알고리즘의 성능 척도로 많이 이용되는 Ackley 5차 함수이다.

$$F_2(x) = -c_1 \cdot \exp\left(-c_2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(c_3 \cdot x_i)\right) + c_1 + e$$

여기서 $n=5$, $c_1 = 20$; $c_2 = 0.2$; $c_3 = 2\pi$

$$\vec{x} = (0, \dots, 0)^T; f_j^* = 0; n = 30; -20 \leq x_i \leq 30$$

이 경우, GA와 SA 모두 적절한 값을 찾지 못함을 알 수 있다. 오히려 SA의 경우 지역적 최소값에 수렴한 경향을 보이고 있다. 반면 CSBC의 경우, 정밀도는 앞선 실험보다는 떨어지지만 여전히 비교 알고리즘들보다는 더 좋은 함수 값을 찾아내었다. (표 2)

표 2. F_2 에서의 실험 결과, $n=5$

| trial | GA | | SA | | CSBC | |
|--------|--------------------------|----------------|--------------------------|----------------|--------------------------|----------------|
| | # of function evaluation | Function value | # of function evaluation | Function value | # of function evaluation | Function value |
| 1 | 5000000 | 2.43317 | 471198 | 4.22475 | 32858 | 0.06910 |
| 2 | 5000000 | 6.29450 | 38126441 | 4.21582 | 31670 | 0.03467 |
| 3 | 5000000 | 3.45545 | 50001373 | 4.21494 | 87902 | 0.05809 |
| 4 | 5000000 | 1.66341 | 79320568 | 4.22026 | 38999 | 0.02982 |
| 5 | 5000000 | 3.61226 | 30738001 | 4.53483 | 57899 | 0.05866 |
| mean | 5000000 | 3.42176 | 39731516.2 | 4.22212 | 49865.6 | 0.05007 |
| St. D. | 0 | 1.57065 | 25681102.6 | 0.14640 | 21211.9049 | 0.01515 |

표 2와 그림 3에서 알 수 있듯이 CSBC 알고리즘이 기존의 유전자 알고리즘이나 시뮬레이티드 어닐링 기법보다 안정적이고 정확함을 알 수 있다.

이상의 실험으로 볼 때, CSBC 알고리즘이 기존의 다른 알고리즘들에 비하여 호출 횟수에서도 우위가 있을 뿐 아니라 정확도 면에서도 더 뛰어난 것을 알 수 있다.

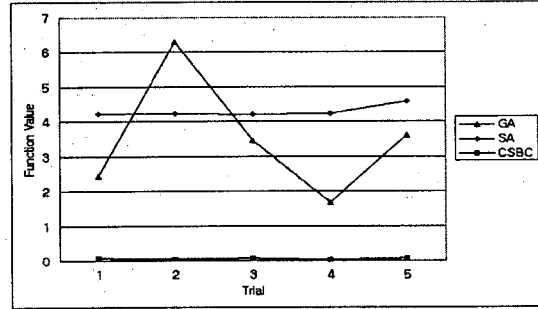


그림 3. F_2 에서의 실험결과, $n=5$

4. 결론 및 향후 연구 과제

현실에서 접하게 되는 많은 문제는 최적화의 문제와 연관되어 있다. 하지만 이러한 대부분의 문제들은 multimodal 함수이다. 이러한 환경에서 전역적 최소값을 찾는 일은 쉬운 일이 아니다. 본 논문에서 제안하고 있는 CSBC 알고리즘은 이러한 환경에서 전역적 최소값을 찾기 위해 고안되었다. 또한 여러 실험들을 통하여 그 성능의 우수함을 보였으며 더불어 기존의 알고리즘들보다 효율적임을 보였다. 다만 CSBC 알고리즘 역시 확률기반의 알고리즘이기 때문에 다른 결정적 알고리즘에 비하여 시간이 더 소요된다는 점과 성능이 일정하지 못하다는 단점이 있다. 이러한 점들은 향후 연구를 통하여 보완 되어져야 할 것으로 기대된다.

감사의 글

이 연구는 산업자원부 지원으로 수행하는 21세기 프론티어 연구개발사업(인간기능 생활지원 지능로봇 기술개발사업)의 일환으로 수행되었습니다.

참고문헌

- [1] Dennis, J. E. and Schenabel, R. B. (1989), "A View of Unconstrained Optimization," in Optimization (Handbooks in OR and MS, vol. 1, G. G. Nemhauser et al., eds.), Elsevier, New York, pp.1-72
- [2] Müller, S. D., Marchetto, J., Airaghi, S. and Koumoutaskos, P., "Optimization Based on Bacterial Chemotaxis," IEEE Transactions on Evolutionary Computation, vol.6, pp.16-29, 2002
- [3] Dahlquist, F. W., Elwell, R. A., Lovely, P. S. "Studies of bacterial chemotaxis in defined concentration gradients - a model for chemotaxis toward l-serine," J. Supramolecular Structure, Vol. 4, pp. 329(289)-342(302), 1976.
- [4] Bremermann, H. J. (1974), "Chemotaxis and optimization," J. Franklin Inst., vol. 297, pp. 397-404
- [5] Biondi, S. A, Quinn, J. A., and Goldfine, H. (1998), "Random Motility of Swimming Bacteria in Restricted Geometries," AIChE Journal, 44 (8) 1923-1929