현대의 유전 알고리즘

DOI: 10.3938/PhiT.27.002

Modern Genetic Algorithms

In-Ho LEE

Genetic algorithms are widely used to generate high-quality solutions to search problems and optimization by relying on bio-inspired concepts such as selection, mutation, crossover, replacement, and chromosomes. Genetic algorithms appear to have considerable potential for use as the basis for designing new algorithms that can search complex spaces and solve hard problems. Increased diversity in candidate solutions can be accomplished in a number of ways. One of the advantages of genetic algorithms is their ability to be easily parallelized.

들어가는 글

유전 알고리즘은 컴퓨터 과학 분야에서 널리 사용되는 광역 최적화(global optimization) 알고리즘이다.^[1] 다윈(C. Darwin)의 자연선택설과 유전학에서 아이디어를 따온 목적함수(objective function) 최적화 알고리즘으로 잘 알려져 있다. 유전 알고리즘은 비선형, 미분 불가능한 함수의 최적화에 사용된다. 이학·공학 문제풀이에서 많은 수로 구성된 독립변수들의 함수로 목적함수를 정의할 수 있다. 여기서 최적화는 최대화 또는 최소화를 지칭한다. 목적함수에 “-1”을 곱하면 최소화는 최대화 문제로 변환되기 때문이다. 결국, 여기서 최대화나 최소화는 최적화라는 말로 통일된다.

저자약력

이인호 박사는 한국과학기술원 (물리학과) 이학박사로서 Univ. of Illinois at Urbana-Champaign (Beckman Institute for Advanced Science and Technology, Materials Research Laboratory) 박사 후 과정 그리고 고등과학원 (물리학과) 리서치-펠로우를 거쳐 한국표준과학연구원 선임 연구원 (2001-2005), 책임 연구원(2005-)으로 재직 중이다. 고등과학원 인실리코 단백질과학 연구단 서브그룹리더(2009-2017) 그리고 과학기술연합대학원대학교(UST) 나노계측학과 겸임교수(2013-현재)로서 활동하고 있다. (ihlee@kriss.re.kr)

이 인 호

Fig. 1. Bio-inspired concepts such as selection, crossover, mutation, replacement, and chromosome are shown. Repeated objective function evaluation for complex problems is often the most prohibitive and limiting segment of genetic algorithms.

전공분야마다 특정 문제를 효율적으로 푸는 방법이 존재할 수 있다. 상황이 조금 다른 경우를 생각해 보자. 유전 알고리즘은 특정 문제를 해결하려고 하는데 특별한 방법이 없을 때 사용하는 방법이다. 모든 가능성에 대해서 테스트하는 것이 거의 불가능한 경우의 최적화 문제풀이를 발견법(heuristic)이라고 부른다.

유전 알고리즘은 결정론적 문제풀이 방식이 아니기 때문에 항상 정답을 보장하지 못한다. 하지만, 범용적으로 사용될 수 있는 특징을 가진 계산 방법이다. 특별히, 유전 알고리즘은 유전학과 자연선택의 개념을 활용한 비결정론적 상위 발견법(metaheuristic method)의 한 종류에 해당한다.

그림 1에 표시된 것처럼 유전 알고리즘은 선택(selection), 변이(mutation), 교차(crossover), 대체(replacement) 등의 네 가지 요소들로 이루어져 있다. 돌연변이는 유전된다. 자손의 유전자는 부모 유전자 조합으로부터 얻어진다. 전자는 변이를 지칭하고 후자는 교차를 지칭한다. 목적함수 값 기준으로 보다 더 최적화된 개체의 경우 부모가 될 확률을 더 높게 해준다. 이것이 선택이다. 후보해(candidate solution)로서의 자손이 목적함수 값 기준으로 더 최적화된 경우 기존의 부모들 중 하나를 대

REFERENCES

- [1] J. Holland, *Adaptation in Natural and Artificial Systems* (Cambridge, MA, MIT Press, 1992).

신하여 그 자리를 차지한다. 이것이 대치이다. 즉, 기존의 후보해 중 하나를 버리고 새로운 후보해를 도입하는 것이다.

본 해설에서는 전통적인 유전 알고리즘의 기본 요소들과 특징을 소개하고 응용 문제풀이에 최적화된 유전 알고리즘을 소개하고자 한다.

후보해의 일차원 표현방식

유전 알고리즘의 구현에서 가장 중요한 항목 중 하나가 후보해를 표현하는 방법을 확립하는 것이다. 후보해를 표현하는 것은 교차 작업 그리고 변이 작업들과 밀접하게 연관되어 있다. 후보해를 표현하는 방식은 일반적으로 일차원 방식이면 충분하다. 개체를 대표하는 염색체(chromosome)는 유전자(gene)들의 일차원 나열이다. 개체가 후천적으로 획득한 개체변이가 유전된다는 라마르크(J.-B. Lamarck)의 용불용설(theory of use and disuse) 그리고 학습능력이 유전된다고 보는 볼드윈(J. M. Baldwin) 효과는 유전 알고리즘에서 고려하지 않는다. 개체(후보해)를 결정짓는 것은 염색체이다. 후보해, 벡터 $\mathbf{x}$ 는 변수들의 나열이다. 각 성분들은 임의의 물리량이 될 수 있다.

특정 최적화 문제풀이에서 후보해로서 만족해야 하는 전제 조건들이 있다. 교차 또는 변이 작업을 수행한 다음 전제 조건들이 만족되는지 체크해야 한다. 전제 조건을 만족하지 못하면 새로운 형식의 후보해를 만들어 낸다.

예를 들어, 후보해의 각 성분들은 연속된 변수이면서 최저값·최고값 등으로 변수의 범위가 한정될 수 있다. 그 이외의 변수 영역에서는 목적함수 최적화를 진행하지 않을 수도 있다. 목적함수에 이러한 부분을 벌점함수(penalty function) 형식으로 넣어두면 아주 편리하다.

다수의 후보해들과 선택

유전 알고리즘을 활용한 문제풀이에서는 여러 개의 후보해들을 동시에 고려한다. 후보해는 유전학에서 말하는 개체를 지칭하는 것이다. 이러한 후보해 무리는 한 세대를 만든다. 새로운 세대는 기존의 세대로부터 만들어진다. 여러 개의 후보해들을 동시에 취급하기 때문에 유전 알고리즘은 아주 좋은 병렬 알고리즘의 예가 된다.

유전 알고리즘의 기본 요소 중 하나인 선택 작업은 변이 또는 교차를 위해서 필요한 것이다. 선택된 개체들 또는 개체는 교차 또는 변이를 통해서 새로운 후보해가 된다. 물론, 새로운 후보해가 모두 다 최적화되어서 기존의 후보해들 중 하나를 대체할 수는 없을 것이다.

선택 작업은 여러 가지 방식이 가능하다. 통상 하나의 방식

을 활용한다. 유전 알고리즘에서 변이와 교차를 위해서는 각각 한 개, 두 개(부모)의 개체가 필요하다. 부모가 될 두 개체를 뽑을 때, 소위 토너먼트 방식으로 뽑을 수 있다. 먼저 연속해서 서로 다른 두 개체를 무작위로 뽑는다. 이들을 i, j 라고 부른다. 물론 $j \neq i$ 이다. 이때, i 의 목적함수가 j 의 것보다 더 최적화되었다면, 우리는 토너먼트 승자인 i 를 부모 중 하나로 선택한다. 마찬가지로 부모가 될 또 다른 개체 하나를 선택할 수 있다. 이를 토너먼트(tournament) 모델이라고 한다. 토너먼트를 여러 라운드(round)로 전개할 수도 있을 것이다. 목적함수가 더 최적화될수록 부모가 될 확률이 더 증가한다. 한 세대 내부에서 심하게 불평등한 계층구조가 생기게 된다. 모든 개체가 동일한 확률로 부모가 될 수 있으면 균등(uniform) 모델이라고 부른다. 목적함수 최적화 정도를 전혀 고려하지 않고 선택하는 방법이다. 절단(truncation) 모델은 특정 최상위 후보해들만이 선택될 수 있게 하는 방식이다. 후보해들 사이의 다양성 확보를 위해서는 균등모델을 고려할 수 있다.

소위, 선택압력(selection pressure)이 높다는 것은 결국 경쟁을 통과한 개체만이 선택될 수 있게 한다는 뜻이다. 적자생존을 대변하지만 다양성을 파괴하는 측면이 있다. 매우 풀기 어려운 광역 최적화 문제풀이의 경우 다양성을 더 존중하는 선택 방식이 유리하다. 유사한 개체들이 득세하면 전체적인 퇴보가 올 수 있다. 따라서 최대한 다양성을 존중하는 절차가 반드시 내포되어 있어야 한다. 다시 말해서, 목적함수는 다소 좋지 않더라도 반드시 특이한 ‘형식’의 후보해들은 받아 들여야 한다.

변이와 교차

변이·교차 작업을 통해서 여러 개의 시도 해들을 변화시키는 것이 유전 알고리즘의 핵심이다. 따라서 유전 알고리즘은 진화 시뮬레이션(simulated evolution)이라고 볼 수 있다. 단일 후보해의 변이($\mathbf{x} \rightarrow \mathbf{x}'$) 대신에 집단적인 후보해들의 변환($\{\mathbf{x}\} \rightarrow \{\mathbf{x}'\}$)을 실행하는 것이다.

변이·교차 연산 둘 중 하나만 사용하면 안 된다. 반드시 두 가지 모두를 사용해야 유전 알고리즘이라고 부른다. 변이만 사용하면 무작위 탐색 방법이 된다. 엄마, 아빠의 특징을 자손이 대략 반반씩 직접 물려받는 것이다. 꼭, 반반씩 물려받을 필요는 없다. 예를 들어, 다차원 벡터 $\mathbf{x}$ 가 있을 때, 대략 절반의 성분들을 첫 번째 후보해에서 복사해오고, 나머지 성분들을 두 번째 후보해에서 복사하여 새로운 하나의 벡터를 만들 수 있다. 이 벡터가 새로운 후보해(chromosome)가 되는 것이다.

교차 작업에 의한 후보해 생성은 매우 독특한 방식이다. 소위 마르코프 사슬(Markov chain)에서는 쉽게 찾아볼 수 없는

변환 방식($x \rightarrow x'$)이다. 분자 동역학, Monte Carlo 계산의 경우, 새로운 상태는 직전의 상태에만 심하게 의존한다. 효율적인 다차원 공간의 탐색을 위해서는 이러한 의존성에서 벗어나는 작업이 반드시 필요하다. 마르코프 사슬에서 찾을 수 있는 유사한 변환($x \rightarrow x'$)은 넓은 공간을 탐색하는데 효율적이지 못하다. 좀 더 공격적이고 다양한 변환이 필요하다.

돌연변이에 의한 개체 변이는 유전이 된다. 따라서 유전 알고리즘에서는 교차 이외에 변이를 취급한다. 유전 알고리즘에서 변이는 교차와 달리 새로운 후보해를 찾아 갈 수 있게 해주는 기능을 수행한다. 교차에는 기존의 후보해들을 더 잘 이용하는 아이디어가 숨어 있다. 무리 자체가 특정 정보를 가지고 있음을 인정하는 것이다. 따라서, 유전 알고리즘은 더 새로운 양식의 후보해를 찾아내고(exploration) 동시에 기존의 후보해들을 효율적으로 잘 이용하는(exploitation) 방식들을 포함하고 있다.

대치

중복세대(overlapping-generation) 모델 또는 비중복세대(non-overlapping generation) 모델, 둘 중 하나를 골라서 후보해를 대치한다. 전자는 자손과 부모세대가 경쟁하는 모델이고 후자는 부모세대와 자손세대가 경쟁하지 않는 모델이다. 하나의 후보해는 기존세대의 모든 후보해들과 경쟁을 통하여 대치되거나 폐기된다. 기존세대 중에서 목적함수가 최적화된 우수한 개체들은 다음 세대에서도 그대로 생존한다. 이것을 엘리트주의(elitism)라고 한다. 통상 이 모델을 취하여 유전 알고리즘을 구현한다. 따라서 엘리트주의는 부모세대와 자손이 경쟁하는 중복세대 모델이다. 중복세대 모델에서는 목적함수 최적화가 결코 퇴보하지 않게 된다.

일반적으로, 유사한 후보해들이 지속적으로 대치되지 않는다면 중복세대 모델을 활용하고 충분한 개체수를 확보한 경우 한 세대에서 후보해들의 다양성을 확보할 수 있다. 유전 알고리즘에서 얻어진 새로운 후보해를 대치하는 방법에서 다양성을 확보하는 방식을 취하고 있어야 한다. 전통적인 유전 알고리즘에는 대치 작업이 너무 단순하다. 보완이 필요한 부분이다. 유사한 후보해들을 계속해서 대치할 필요가 없다. 기존의 후보해와 유사한 자손 후보해가 나올 경우, 하나로 단일화시키는 과정이 필요하다. 유사한 후보해들이 너무 득세하지 못하도록 다양한 후보해가 만들어질 수 있는 환경을 만들어 주어야 한다.

구배 벡터와 국소 최적화

목적함수를 광역 최적화할 때, 목적함수 구배 벡터(gradient

vector)를 얻을 수 있다면, 국소 최적화 알고리즘을 꼭 활용하여야 한다. BFGS(Broyden-Fletcher-Goldfarb-Shanno) 방법과 같은 국소 최적화 알고리즘을 활용하는 것이 매우 효율적이다.^[2] 많은 경우에는 목적함수 구배 벡터가 알려져 있지 않다. 목적함수가 미분 불가능한 경우도 있다. 이러한 경우 Nelder-Mead 방법을 활용할 수 있다.^[3] 전통적인 유전 알고리즘은 국소 최적화 알고리즘을 사용하지 않는다. 하지만, 일반적으로 국소 최적화 알고리즘을 활용하는 것이 더 효율적이다. 이를 잡종(hybrid) 유전 알고리즘이라고 한다. 효율의 기준은 광역 최적화를 이루는데 필요한 목적함수 계산 횟수로 정의할 수 있다. 일반적으로 목적함수 계산은 많은 컴퓨터 자원을 필요로 하는 경우가 대부분이다.

아울러 전통적인 유전 알고리즘에서는 후보해를 비트 문자열(bit string)로 표시하였다. 즉, '0', '1'로 표현된 긴 문자열로 후보해를 표시하였다. 예를 들면, '010011010101' 같은 방식의 비트 문자열이 가능하다. 연속 변수로 후보해를 표현해야만 앞서 언급한 국소 최적화 방법을 활용할 수 있다.

유전 알고리즘의 병렬화

현대의 유전 알고리즘은 대부분 연속 변수들을 이용하여 국소 최적화 알고리즘을 활용하고 있다. 국소 최적화 알고리즘은 매우 많은 컴퓨터 자원을 필요로 하는 것이다. 따라서 유전 알고리즘에서는 다수의 국소 최적화 작업들을 동시에 수행하는 방식으로 쉽게 유전 알고리즘의 병렬 효율성을 높일 수 있다.

국소 최적화 작업에 필요한 컴퓨터 자원에 비해서 선택, 변이, 교차, 그리고 대치 작업은 많은 컴퓨터 자원을 필요로 하지는 않는다. 따라서 소위 '왕-군인'(king-soldier, master-slave) 방식의 병렬 프로그래밍이 적합하다. '왕'은 선택, 변이, 교차, 그리고 대치 작업을 수행하고 나머지 국소 최적화 작업들은 '군인들'이 수행하는 방식이다. 국소 최적화 작업들은 동시에 시작하여 동시에 종료되지 않는 작업 특성이 있다. 따라서, 하나의 국소 최적화 작업이 종료되면 새로운 국소 최적화 작업이 즉시 시작될 수 있도록 해야 한다. '왕'은 대치 작업을 수행하고 교차 또는 변이된 새로운 후보해를 만들어 새로운 국소 최적화 작업이, 조금 전 계산이 끝난 CPU에서, 곧바로 진행되게 해야 한다. 일하지 않는 CPU가 있으면 병렬 효율성이 떨어진다.

REFERENCES

- [2] A. Mordecai, *Nonlinear Programming: Analysis and Methods* (Dover Publishing, 2003).
- [3] J. A. Nelder and R. Mead, *Comput. J.* **7**, 308 (1965).

응용 문제에 따라서 병렬화 방식은 많이 달라질 수 있다. 컴퓨터 메모리 상에서 모든 계산들이 완료될 수 있는 것이 가장 이상적이다. 이러한 경우, MPI(Message Passing Interface) 라이브러리를 활용한 병렬화가 가장 적합하다. 매우 많은 컴퓨터 자원을 사용하는 응용 문제풀이도 있을 수 있다. 이러한 경우, 시스템 함수를 불러서 프로그램 외부에서 별도의 독립된 계산이 진행되게 하고 그 결과를 파일로부터 읽어 들이는 방식이 편리하다. 다수의 폴더를 만들고 각각의 폴더에서 독립적인 국소 최적화 작업을 진행할 수 있다. 물론, 각각의 국소 최적화 작업은 병렬계산 방식으로 진행될 수도 있다. 이렇게 될 경우, 여러 개의 폴더를 이용한 초고도 병렬계산이 이루어지게 된다. 이때, PBS(Portable Batch System)와 같은 병렬계산 작업관리 프로그램을 활용하여 단위 계산들을 수행하면 매우 편리하다. 또한, 목적함수 계산이 단일 프로그램으로 수행될 수 없을 경우도 있다. 여러 프로그램들을 순차적으로 불러서 원하는 계산을 마무리할 수 있다.

다양성 확보를 위한 조치

광역 최적화를 성공적으로 수행하기 위해서는 후보해들 사이의 다양성의 생성과 유지를 확보해야만 한다. 꼭 필요한 유전자들이 완전히 소멸되지 않게 하는 것이다. 소수의 개체로 이루어진 집단에서 자연선택이 아니라 무작위 확률에 의한 유전자 부동(genetic drift) 효과가 발생하지 않게 해야 한다. 최적화에 필요한 유전자들이 무작위로 사라지는 경우가 발생할 수 있다. 유전자 부동은 자연선택설과 대비되는 개념이다.

한 세대에서 다양한 개체들이 포함될 수 있게 하는 니칭(niching) 방법이 필요하다. 이것이 목적함수 값, 그 값만으로 단순히 부모를 선택하거나 자손을 대체해서는 안 되는 가장 큰 이유이다. 전통적인 유전 알고리즘에서는 후보해들의 다양성을 확보할 수 있는 일반적인 방법이 없다.

임의의 구조를 무작위로 새로 도입하거나 자손들 중 하나를 무조건 대체하는 방법들이 개체들의 다양성 유지를 위해서 취할 수 있는 방법이 될 수 있다. 특정 비율의 부모들을 강제로 자손들로 대체하는 방법이 있을 수 있다. 선택에서도 목적함수가 최적화되지 못한 개체도 확률적으로 부모가 될 수 있게 하는 것이 좋다. 하나의 변수 x 가 $[a, b]$ 구간에서 정의된 것이라면, $a + b - x \rightarrow x'$ 처럼 변환하는 것도 좋은 아이디어이다. 아울러, 대체에서 유사한 후보해들을 단순히 목적함수 최적화 조건이 만족된다고 계속해서 대체를 허용하면 다양성을 훼손할

수 있다. 유사한 후보해들은 하나의 후보해로 단일화시켜서 유사한 해들이 득세할 수 없게 해야 한다. 다시 말하면, 가능하면 공간적으로 분리하여 후보해들을 경쟁시켜야만 한다.

위에서 기술한 전통적인 유전 알고리즘을 보완하고 응용 문제풀이에 사용될 수 있게 한 하나의 광역 최적화 알고리즘이 CSA(Conformational Space Annealing) 알고리즘이다.^[4,5] CSA 알고리즘은 Lennard-Jones 클러스터 구조 탐색, 외판원 문제, 단백질 구조 결정 및 예측, NMR 또는 X-ray 데이터를 활용한 분자구조 결정 및 예측, 네트워크 커뮤니티 구조, 제일원리 전자구조 최적화, 화학반응 경로 계산 등에 다양하게 응용되었다.^[6-9] 최근 CSA 알고리즘을 파이썬 컴퓨터 언어로 구현한 pyCSA라는 컴퓨터 프로그램이 일반에 공개되어 있다.^[10]

나오는 글

유전 알고리즘은 최고의 병렬 효율성을 가지고 있는 일반적인 광역 최적화 알고리즘으로서 이공계에서 다양한 응용 영역을 확보하고 있다. 최근 컴퓨터 자원의 발달로 과거에는 불가능해 보였던 최적화 문제들이 현대의 유전 알고리즘을 통하여 최적화되는 사례들이 늘어나고 있다. 목적함수가 너무나도 복잡하고 계산량이 많을 경우, 보다 더 간단한 목적함수를 설계함으로써 유전 알고리즘을 곧바로 적용해 볼 수도 있다. 이렇게 후보해를 얻고 난 후 보다 더 정밀한 계산을 수행할 수 있을 것이다. 물론, 얻어진 후보해를 더 잘 고찰함으로써 풀고 있는 문제의 일반적인 해를 얻을 수도 있다. 현대의 유전 알고리즘을 활용한 물리학 문제풀이에서 새로운 성공적인 모범사례들이 더욱더 많이 나오기를 희망한다.

REFERENCES

- [4] J. Lee, H. A. Scheraga and S. Rackovsky, *J. Comput. Chem.* **18**, 1222 (1997).
- [5] J. Lee, I.-H. Lee and J. Lee, *Phys. Rev. Lett.* **91**, 080201 (2003).
- [6] J. Lee, S. P. Gross and J. Lee, *Sci. Rep.* **3**, 2197 (2013).
- [7] K. Joo, I. Joung, J. Lee, J. Lee, W. Lee, B. Brooks, S. J. Lee and J. Lee, *Proteins: Structure, Function, and Bioinformatics* **83**, 2251 (2015).
- [8] I.-H. Lee, Y. J. Oh, S. Kim, J. Lee and K. J. Chang, *Comput. Phys. Commun.* **203**, 110 (2016).
- [9] J. Lee, I.-H. Lee, I. Joung, J. Lee and B. R. Brooks, *Nat. Commun.* **8**, 15443 (2017).
- [10] I. Joung, J. Y. Kim, S. P. Gross, K. Joo and J. Lee, *Comput. Phys. Commun.* **223**, 28 (2018).