DNA 서열 디자인을 위한 다중 목적 함수 진화 알고리즘

김동민[0], 신수용, 장병탁
서울대학교 컴퓨터 공학부
{dmkim[0], syshin, btzhang}@bi.snu.ac.kr

Multiobjective Evolutionary Algorithm for DNA Sequence Design

Dong-Min Kim[0], Soo-Yong Shin, Byoung-Tak Zhang
School of Computer Science and Engineering,
Seoul National University

요 약

DNA 컴퓨팅은 차세대 컴퓨팅 방법으로서 주목받고 있으나, 실제 생화학 분자인 DNA의 특성에 의한 오류 가능성을 내포하고 있다. 근래에 들어 이러한 문제점을 극복하고 DNA 컴퓨팅의 신뢰도를 향상시킬 방법으로서 실험에 사용될 DNA 서열의 생성 단계에서 그 오류의 가능성을 예측하고 이를 최소화하고자 하는 방법이 많이 연구되고 있는데, 본 논문에서는 DNA 서열의 적합도를 측정할 함수를 적절하게 정의할 경우 서열 생성 문제가 수치 최적화 문제로 쉽게 환원될 수 있음에 주목하고 이러한 관점에서 실제 실험에서 발현되는 DNA의 다양한 특징을 반영하고 그 최적화를 위하여 다중 목적 함수 진화 알고리즘을 적용하고자 시도하였다. 구현된 알고리즘은 진화의 각 단계마다 우열을 판별할 수 없는 여러개의 서열 묶음을 효과적으로 찾아내었다.

1 서 론

실제 생화학 분자인 DNA를 계산 매체로 이용함으로써, DNA 컴퓨팅은 초병렬성을 비롯하여 상보성 (Complementarity)과 같은 유용한 특성을 확보하게 됨과 동시에 이들의 화학적 특성에 의한 문제점을 안게 되었다. 이러한 어려움을 극복하기 위하여 많은 연구들이 진행되고 있고 그중에서도 오류 가능성을 최소화하는 방향으로 DNA 서열을 디자인하고자 하는 연구가 활발히 이루어지고 있다. 또, 그 연구 결과로서 DNA 서열의 적합도를 판별하고자 하는 목적 함수들이 다양하게 밝혀지고 정의되고 있다. 이러한 적합도 함수들을 통해 DNA 서열의 오류 가능성을 최소화하는 알고리즘들이 또한 다양하게 제시되고 있는데, 이제까지 개발된 알고리즘들은 크게 두 부류로 구분해 볼 수 있다.

첫번째 부류는 확정적 (deterministic)인 알고리즘들이다. Marathe [1]가 제시한 Hamming distance와 Free energy를 이용한 동적 프로그래밍 (dynamic programming) 방법과 Frutos [2]가 사용한 template-map strategy등이 대표적이라 할 수 있다. 이밖에도 Hartemink [3]가 프로그램 "SCAN"에서 구현한 exhaustive 탐색과 Feldkamp [4]가 "DNASequenceGenerator"에서 개발한 그래프 탐색등 역시 이 부류에 속하는 것으로 분류할 수 있다.

두번째는 진화 (evolutionary) 알고리즘들이다. Arita [5]의 유전자 알고리즘과 random generate-and-test 알고리즘, Ruben [4]의 "PUNCH"에 구현된 진화 알고리즘을 예로 들 수 있으며, 부가적으로 Tanaka [4]등이 사용한 Simulated Annealing은 진화 연산의 일종은 아니나 이 부류의 알고리즘으로 볼 수 있다.

본 논문에서는 적합도 함수들을 적절하게 정의하면 DNA 서열 생성 문제를 쉽게 수치 최적화 문제로 변환할 수 있다는 점과 문제의 복잡도, 그리고 이전 연구 [6]에 기반하여 두번째 접근 방법을 택하였다. 특별히 문제에 적용될 적합도 함수가 단일하지 않다는 점에 주목하여 다중 목적 함수 진화 (multiobjective evolutionary) 알고리즘의 기술들을 적용하고자 시도하였는데 다음 장에서 DNA 서열 생성을 위해 설계한 알고리즘을 기술한다. 3장에서 단순 유전자 알고리즘을 이용한 DNA 서열 생성과의 비교 실험 및 결과를 살펴보며 마지막으로 4장에서 결론을 제시하고자 한다.

2 DNA 서열 생성 알고리즘

앞서 언급한 바와 같이, DNA 서열 생성은 수치 최적화 문제로 변환하여 생각할 수 있다. 더우기 실제 실험에서의 오류가 DNA 서열의 다양한 특성에서 비롯된다는 점을 감안한다면, 각 서열의 오류 가능성 최소화 문제는 특정한 것을 수치화하는 함수들의 묶음에 대한, 동시적인 최적화를 의미하는 것으로 볼 수 있다. 이런 관점에서 현재까지 개발된 많은 알고리즘들이 여러 목적 함수를 동시에 충족시키는 최적의 서열 묶음을 찾고자 시도하였다. 하지만, 많은 경우에 있어서 이와 같은 최적의 해가 존재하지 않거나 존재한다고 하더라도 효과적으로 찾기 어렵다는 문제가 발생하였다. 본 논문에서는 이와 같은 문제를 피하기 위하여, 단 하나의 최적해를 찾는 알고리즘을 지양하고 대신 여러 목적 함수를 고려할 때에 서로 간의 우열을 판별할 수 없는 파레토 (Pareto) 해를 전부 탐색한 후 이들을 추천하였다.

이와 같은 방식의 알고리즘 설계를 위하여 현재까지 개발된 알고리즘 중 Goldberg [7]가 파레토 해를 찾기 위한 방법으로서 제시한 Nondominated Sorting 개념을 구현한 NSGA (Nondominated Sorting Genetic Algorithm) [8]를 확장하였다. 원래의 NSGA는 단순 유전자 알고리즘에서 선택 연산자 (selection

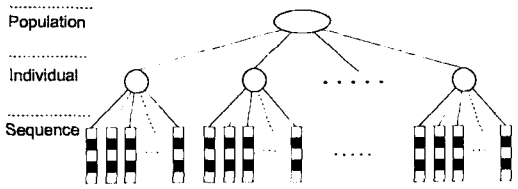

그림 1: DNA 서열 생성을 위한 개체군 구조

operator)만을 변형하였고, 하나의 비트 스트링이 하나의 개체를 표현한 것과는 달리 DNA 서열 생성을 위한 유전자 알고리즘에서 여러개의 4진 스트링이 모여 하나의 개체를 이루게 되는 점 (그림 1 참조)을 반영하기 위하여 교차 (crossover) 연산자와 돌연변이 (mutation) 연산자를 추가적으로 확장하였다. 다음 프로시저는 선택 연산자에서 사용하는 Nondominated Sorting을 나타낸다. 다음 의사코드에서 개체간의 우열을 평가하기위한 기호 $\prec$은 다음과 같이 정의된다.

$p = (p_1, p_2, \ldots, p_n)$, $q = (q_1, q_2, \ldots, q_n)$
$q \prec p$ *iff* $\exists i\ (i = 1, 2, \ldots, n)$ *s.t.* $q_i < p_i$

nondominated-sort(P)
  $i = 0$
  while $P \neq \varnothing$
    for each $p \in P$
      for each $q \in P$
        if $(q \prec p)$ then $n_p = 1$
    for each $p \in P$
      if $(n_p = 0)$ then $\mathcal{F}_i = \mathcal{F}_i \cup \{p\}$
      else $n_p = 0$
    $\mathcal{G} = \mathcal{G} \cup \mathcal{F}_i, i = i + 1$

nondominated-sort(P)를 통해 현재 세대의 개체들은 그 적합도에 따라 적절한 파레토 프런트 (pareto front) $\mathcal{F}_i$에 속하게 되며, 이 프런트들이 모여 프런트 리스트 $\mathcal{G}$를 형성하게 된다. 선택 연산자는 각 프런트들의 순위 (ranking)에 기반하여 작동하며, 같은 프런트에 속한 개체들의 선택 확률은 모두 같다. 즉 하나의 개체를 선택할 때에 먼저 좋은 순위의 파레토 프런트를 높은 확률로 선택한 후, 해당 프런트에 속한 모든 개체들 중 하나를 무작위로 선택하게 되는 것이다.

교차 연산자와 돌연변이 연산자는 그림 1의 Individual 수준 (Step1)과 Sequence 수준 (Step2)의 두단계로 각각 나누어 구현하였다. 먼저 Step1 교차 연산자는 각각의 개체들이 자신에게 속한 DNA 서열들 중 일부를 교환하는 것으로 이해할 수 있다. 이렇게 교환되는 과정 중에 Step2 교차 연산자가 적용되며 이는 단순 유전자 알고리즘의 다중점 교차 연산 (multi-point crossover)이 각각의 서열에 적용되는 것과 동일하다. (그림 2) 돌연변이 연산자 역시 교차 연산자와 비슷한 방식으로 작동하는데 다만 Step1 돌연변이 연산자는 Step2 돌연변이 연산자가 적용될 DNA 서열들이 속한 개체를 선택하는 과정일 뿐, 각 개체의 유전자형 (genotype)에 변화를 일으키지 않는다는 점에서 Step1 교차 연산자와 차이가 있다. 다음 의사 코드는 한 세대의 진화 과정에서 일어나는 모든 연산자의 작동을 보여주는데, 마지막의 *set-pareto*$(\mathcal{F}_0)$는 이제까지 구해진 모든 파레토 해 $(\mathcal{H})$와 현재 세대에서 구해진 파레토 해 $(\mathcal{F}_0)$를 합한 후, 다시 그들중의 파레토 해를 구하는 프로시저이다. 따라서, 이 프로시저는 진화 과정 중에 나타나는 모든 파레토 해들을 진화에 필요한 개체군과 별개의 개체군을 유지하는 역할을 하며 이들이 진화의 첫단계에서 기존의 세대에 병합됨으로써 이를 통해 elitism을 구현하였다.

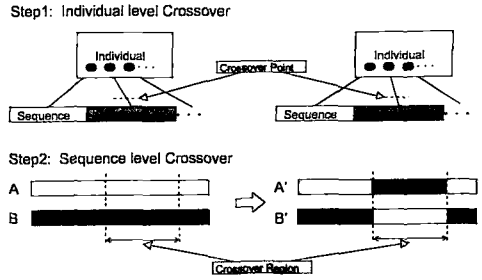

그림 2: 교차 연산자

evolve(P)
  $P = P \cup \mathcal{H}$
  evaluate(P)
  nondominated-sort(P)
  init-front-selection(P)
  while $|Q| < N$
    $\mathcal{F}_1$ = select-front$(\mathcal{H})$, i_1 = select-individual$(\mathcal{F}_1)$
    if (step1 crossover) then
      $\mathcal{F}_2$ = select-front$(\mathcal{H})$, i_2 = select-individual$(\mathcal{F}_2)$
      if (step2 crossover) then
        $Q = Q \cup$ {crossover2(i_1, i_2)}
      else
        $Q = Q \cup$ {crossover1(i_1, i_2)}
      if (mutation) then mutate(Q_{i_1}, Q_{i_2})
    else
      $Q = Q \cup \{i_1\}$
      if (mutation) then mutate(Q_{i_1})
  set-pareto$(\mathcal{F}_0)$

마지막으로 각 개체의 적합도는 [9]에서 정의한 7개의 적합도 함수를 통해 측정한다.

3 실험 및 결과

먼저 알고리즘의 바른 작동을 검증하기 위하여, 진화 정도에 따른 각 목적 함수의 적합도 정도를 살펴보았다. 그림 3에 세대별로 각 목적 함수에 대해 가장 잘 적합된 개체의 적합도를 표시하였다. 진화가 진행됨에 따라, 특정한 목적 함수에 더욱 적합한 개체가 탐

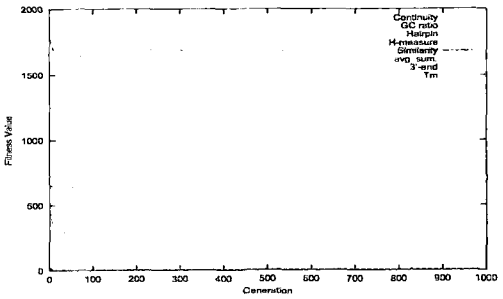

그림 3: 적합도 함수값 변화 추이

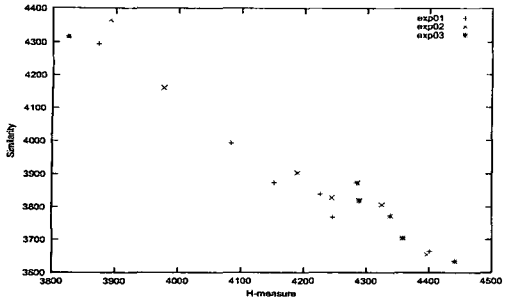

그림 4: H-measure와 Similarity에 대한 알고리즘의 해

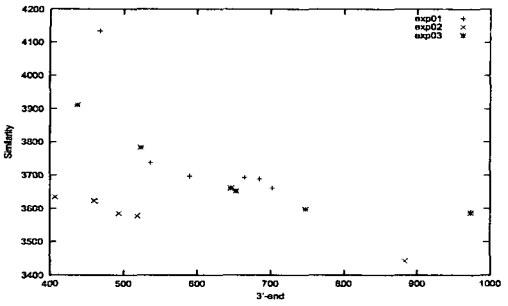

그림 5: 3'-end와 Similarity에 대한 알고리즘의 해

색됨을 알 수 있으며, 특히 Hairpin, Melting Temperature, GC ratio등은 초기 세대에서 쉽게 최적화된 개체가 생성됨을 보여준다. 반면 모든 적합도 함수값의 평균을 기준으로 보면 진화의 정도가 적은 결과를 나타내는데, 이는 전체 적합도 함수값의 합을 기준으로 몇몇 목적 함수들의 상호 보상 (trade-off) 관계를 나타내기 때문인 것으로 생각된다. 바꿔 말하면, 하나의 목적 함수를 최적화하기 위해서 다른 목적 함수에 대한 최적화 정도를 포기하는 현상이 발생하는 것이다. 설계한 알고리즘이 이와 같은 관계의 목적 함수들 사이에서 효과적으로 파레토 해를 찾아내는가를 확인하기 위하여 그림 3에서 진화에 가장 어려움을 보인 H-measure와 Similarity만을 선택하여 다시 실험하였고, H-measure와 매우 유사한 적합도 함수인 3'-end에 대해서도 Similarity와 비교를 반복하였다. 그림 4와 5는 각각 두 목적 함수에 대하여 최종적으로 6개의 파레토 해를 보여주는 실험을 3회 반복한 결과이다. 각 실험으로부터 Similarity는 H-measure와 3'-end에 대하여 분명한 상호 보상 관계임을 알 수 있으며, 또한 작성한 알고리즘이 의도한 바대로 동작하는 것을 볼 수 있다. 실험 결과로부터, DNA 서열의 적합도 측정에서는 상호 보상 관계의 적합도 함수들이 존재하며 이들을 동시에 고려할 경우에는 단순합이나 가중합에 대한 최적화로는 모든 목적 함수를 균형있게 적합화하기 어려울 것으로 생각할 수 있다.

4 결 론

본 논문에서는 DNA 컴퓨팅을 위한 서열 생성에 여러가지의 DNA 서열 적합도 함수를 이용하는 다중 목적 함수 진화 알고리즘을 적용하고자 시도하였다. 구현된 알고리즘은 이제까지의 알고리즘들이 단일한 최적해를 찾고자 시도하였던 것과 달리 하나 이상의 목적 함수에 대하여 다른 해들보다 나은 적합도를 보이는 파레토 해들을 탐색하여 제시함으로써 실제 실험에 보다 적합한 서열 묶음을 사용자가 선택할 수 있도록 하였다.

앞으로는 보다 많은 DNA 서열의 특성을 고려할 수 있도록 적합도 함수들을 추가 구현하고, 각종 연산자의 계산 효율을 향상시켜 보다 높은 성능을 확보하고자 한다. 또한, 진화중에 유지되는 파레토 해들의 개체군에 대하여 클러스터링 기법등을 도입하여 더욱 유효한 서열 묶음의 추천 능력을 확보하고자 한다.

감사의 글

본 연구는 산업자원부 차세대신기술사업과 BK21 프로그램에 의하여 일부 지원되었음. 이 연구를 위해 연구 장비를 지원하고 공간을 제공한 서울대학교 컴퓨터신기술공동연구소에 감사드립니다.

참고문헌

[1] A. Marathe, A. E. Condon, and R. M. Corn, "On combinatorial DNA word design," in *Proceedings of 5th DIMACS Workshop on DNA Based Computers*, pp. 75-89, 1999.

[2] A. G. Frutos, A. J. Thiel, A. E. Condon, L. M. Smith, and R. M. Corn, "DNA computing at surfaces: 4 base mismatch word design," in *Proceedings of 3rd DIMACS Workshop on DNA Based Computers*, pp. 238, 1997.

[3] A. J. Hartemink, D. K. Gifford, and J. Khodor, "Automated constraint-based nucleotide sequence selection for DNA computation," in *Proceedings of 4th DTMACS Workshop on DNA Based Computers*, pp. 227-235, 1998.

[4] N. Jonoska and N. C. Seedman (Eds.), Preliminary Proceedings of 7th international Workshop on DNA-Based Computers, University of South Florida, Tampa, FL, June10-13, 2001.

[5] M. Arita, A. Nishikawa, M. Hagiya, K. Komiya, H. Gouzu, and K. sakamoto, "Improving sequence design for DNA computing," in *Proceedings of Genetic and Evolutionary Computation Conference 2000*, pp. 875-882, 2000.

[6] B.-T. Zhang and S.-Y. Shin, "Molecular algorithms for efficient and reliable DNA computing," in *Genetic Programming 1998*, Morgan Kaufmann., pp. 735-742, 1998.

[7] D. E. Goldberg, "Genetic algorithms in search, optimization and machine learning," Addison-Wesley, 1989.

[8] N. Srinivas, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 3, no. 2, pp. 221-248, 1995.

[9] S.-Y. Shin, D.-M. Kim, and B.-T. Zhang, "Evolutionary Sequence Generation for Reliable DNA Computing," *Proc. of Congress on Evolutionary Computatioin 2002.*