다양한 상황에서의 Adaptive Pursuit 방법론 분석

김민혁○ 박남용
서울대학교 컴퓨터공학부 구조복잡도 연구실
rniritz@gmail.com, zest159@naver.com

Analysis of Adaptive Pursuit Methods on Various Situations

MinHyeok Kim○ Namyong Park
Structural Complexity Lab. Computer Science and Engineering, Seoul National Univ.

요 약

Adaptive Operator Selection (AOS)은 학습 진행도에 따라 변화하는 환경에 맞추어 각 연산자의 값을 유연하게 조정하는 방법으로, 현 시점까지의 결과를 바탕으로 앞으로의 상황 변화를 예측하여 최적의 연산자 조합을 제시한다. 본 논문에서는 AOS 중 하나인 Adaptive Pursuit를 소개하고, 세 종류의 Adaptive Pursuit 방법론이 간단한 시나리오 상에서 어떻게 동작하는지 살펴봄으로써 각 방법론의 특징 및 차이를 분석해 보았다.

1. 서 론

인공 신경망 (Artificial Neural Network)이나 결정 트리 (Decision Tree), 베이지안망 (Baysian Network) 등의 많은 학습 알고리즘에서는 학습에 필요한 각종 파라미터 값을 학습 전에 미리 결정하고 이 값을 이용하여 학습을 진행하는 경우가 대부분이다. 유전 알고리즘 (Genetic Algorithm)이나 유전 프로그래밍 (Genetic Programming)과 같은 진화 연산 (Evolutionary Computation) 알고리즘에서도 이러한 경향은 마찬가지로, 대체로 학습 전에 교차 (Crossover)와 변이 (Mutation) 등의 유전 연산자와 관련된 파라미터 값을 미리 설정하고 학습을 진행한다.

하지만 학습 길이가 짧다고 하더라도, 학습 초기의 모습과 학습 말기의 모습을 살펴보면 동작이나 요구사항 등이 서로 다르다. 학습 초기에는 대체로 다양한 조합을 통해 최적해를 찾기 위한 많은 가능성을 타진하는 반면, 학습 말기에서는 현재까지 찾은 해를 기반으로 점진적인 해의 향상을 꾀한다. 이렇듯 학습 진행에 따라 서로 다른 경향성을 보인다는 사실에 기인하여 학습 진행도에 따라 파라미터 등을 다르게 적용하는 연구는 오래전부터 있어왔다. 간단하게는 파라미터 값을 학습 진행도를 변수로 하는 함수로 설정하는 스케쥴링 (Scheduling) 방법이 있고, 비교적 최근 연구로는 유전 알고리즘 및 유전 프로그래밍 분야에서 적용된 바 있는 Adaptive Operator Selection (AOS) 방법을 예로 들 수 있다.

본 논문에서는 AOS 방법론 중, Adaptive Pursuits 방법에 기반을 두고, 임의로 설정된 상황에서 각 연산자 (Operator)의 변화 및 그에 따른 각 방법론의 특성을 살펴보고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 우선 관련된 AOS 방법론을 소개할 것이고, 3장에서는 본 논문에서 주로 다룰 Adaptive Pursuit 방법론에 대해서 설명한다. 이후, 4장에서 실험의 설계 및 결과를 제시한 뒤 이를 분석한 뒤, 마지막으로 5장에서 이들 결과를 정리하고자 한다.

2. 관련 연구

연산자 적용 (Operator Adaptation)에 대한 연구는 진화 연산 연구 초기부터 큰 쟁점이 된 사안으로, 해당 시점까지의 결과를 바탕으로 향후 성능을 예측하여 시스템의 성능을 극대화 하는 것을 목표로 하는 한편, 아무런 선지식이 없는 상황에서 하나의 가이드라인을 제시하는 역할을 할 수 있다. AOS 분야에서의 최근 연구로 Adaptive Pursuit 외에 Probability Matching와 Multi-Armed Bandits 방법이 있다.

2.1 Probability Matching (PM)

Probability Matching 방법론 [1]은 가장 직관적인 방법론으로, 각 연산자의 현재까지의 성능을 평가한 뒤, 전체 연산자의 평가 값 간 비율을 이용하여 각 연산자에게 새로운 적용 확률을 제공하는 방법이다.

$$Q(t+1) = Q(t) + \alpha(R(t) - Q(t)) \text{ for } all\, operators$$

$$P_a(t+1) = P_{\min} + (1 - K \bullet P_{\min}) \frac{Q_a(t+1)}{\sum_{i=1}^{K} Q_i(t+1)}$$

수식 1 Probability Matching

구체적으로는, 연산자 a에 대해서 현재 시점 t에서 얻은 적합도 기반의 결과 값 $R(t)$을 현재까지의 성능 값인 $Q(t)$에 더하여 새로운 값 $Q(t+1)$을 얻고, 이를 전체 K개의 연산자가 가진 성능 값에 대한 비율을 계산, 새롭

게 적용할 확률 값 $P_a(t+1)$을 얻는 것을 반복하면서 이루어진다. (수식 1)

2.2 Multi-Armed Bandits (MAB)

Multi-Armed Bandits 방법론 [2,3]은 Upper Confidence Bound (UCB) 알고리즘 [4]을 기반으로 Exploration과 Exploitation의 균형을 맞추면서 최적 연산자를 선택하는 방법이다.

$$UCB_a(t) = \widehat{P_{a,t}} + \sqrt{\frac{\log\sum_{}^{k} n_{i,t}}{n_{a,t}}}$$

수식 2 UCB 알고리즘

수식 2에서 볼 수 있듯이, UCB 알고리즘은 크게 두 부분으로 이루어져 있다. 앞부분은 이제까지의 성능을 구하는 부분이며, 뒷부분은 해당 연산자가 이제껏 선택된 횟수 n을 바탕으로 얼마나 선택이 되지 않았는지를 평가하는 부분이다. 즉, UCB 알고리즘은 성능이 좋은 연산자를 선택하는 한편 여태껏 선택된 기회가 적은 연산자에게도 선택될 기회를 부여한다.

3. Adpative Pursuit Methods

Adaptive Pursuit 방법론 [5]의 기본 형태는 PM과 유사한 형태를 가지고 있으나, 세부적으로 구해진 성능 값을 바탕으로 적용 확률을 제시하는 방법에서 차이가 있다. 본 논문에서는 세부 방법에 따라 세 가지 종류의 Adaptive Pursuit 방법론을 비교, 분석하고자 한다.

3.1 Adpative Pursuit (AP)

가장 기본적인 Adaptive Pursuit 방법론의 형태는 수식 3과 같다.

$$Q(t+1) = Q(t) + \alpha(R(t) - Q(t)) \text{ for } all\, operators$$
$$a^* = \arg max\{Q_i, i = 1,\ldots,K\}$$
$$P_{a^*}(t+1) = P_{a^*}(t) + \beta(P_{\max} - P_{a^*}(t))$$
$$P_a(t+1) = P_a(t) + \beta(P_{\min} - P_a(t)) \text{ for } a \neq a^*$$

수식 3 Adaptive Pursuit

AP의 성능 값 갱신 방법은 PM과 동일하다. 하지만 AP에서는 가장 성능이 좋은 연산자 a^*을 선별하여 해당 연산자의 적용 확률만 최대 확률 $P_{\max}$를 이용하여 향상시키는 한편, 나머지 연산자는 최소 확률 $P_{\min}$을 이용하여 균등하게 감소시켜서 두 집단을 차별화 시킨다.

3.2 Adpative Probability Matching (APM)

Adaptive Probability Matching (APM) 방법론 [6]은, AP와 PM 방법론이 혼합된 형태의 알고리즘으로, 가장 성능이 좋은 연산자 외의 다른 연산자의 성능 값에 차별을 두고자 만들어졌다. APM에서는 AP와 마찬가지로 가장 좋은 성능을 가지는 연산자를 선별하여 해당 적용 확률을 향상시키지만, 나머지 연산자에 대해서는 PM과 마찬가지로 성능 값의 비율에 맞추어 확률을 적용시킨다. (수식 4)

$$Q(t+1) = Q(t) + \alpha(R(t) - Q(t)) \text{ for } all\, operators$$
$$a^* = \arg max\{Q_i, i = 1,\ldots,K\}$$
$$P_{a^*}(t+1) = P_{a^*}(t) + \beta(P_{\max} - P_{a^*}(t))$$
$$P_a(t+1) = P_{\min} + (1 - P_{a^*}(t+1) - (K-1) \cdot P_{\min}) \frac{Q_a t}{\sum_{i=1, i \neq a^*}^{K} Q_i t} \text{ for } a \neq a^*$$

수식 4 Adaptive Probability Matching

3.3 Recursive Adpative Pursuit (rAP)

Recursive Adaptive Pursuit 방법론은, 본 논문에서 처음 제시하는 방법론으로, 반복적으로 AP를 적용하여 각 연산자의 확률을 차별화 하는 것이 주 특징이다. 가장 좋은 성능의 연산자의 적용 확률을 높이는 것까지는 AP와 동일하지만, 이하 연산자의 적용 확률은 남은 연산자들만을 대상으로 AP를 재귀적으로 적용하여 계산하는 방법이다. (수식 5)

$$Q(t+1) = Q(t) + \alpha(R(t) - Q(t)) \text{ for } all\, operators\, from\, the\, best\, operator$$
$$P_{a^{nth}}(t+1) = \left(1 - \sum_{i}^{a^{n-1th}} P_i(t+1)\right)\{P_{a^{nth}}(t) + \beta(P_{\max} - P_{a^{nth}}(t))\}$$

수식 5 Recursive Adaptive Pursuit

따라서 rAP에서는 기본적으로 AP와 똑같은 수식을 이용하여 적용 확률을 구하되, 전체 범위가 아닌 앞선 연산자의 적용 확률을 제외한 범위에서 식을 적용한다.

4. 실험 설계 및 결과

4.1 실험 설계

본 논문에서의 실험은 몇 가지 가정 하에 간단하게 진행된다.

우선 연산자의 수는 3개로 가정하고, 상황은 기존의 결과를 고려하여[6,7] 역시 3개를 가정한다 (표 1, 그림 1).

표 1 상황별 연산자 설정

	Op1	Op2	Op3
상황1	선형감소	선형증가	선형증가 후 감소
상황2	점진적 감소	점진적 증가	점진적 증가 후 감소
상황3	점진적 감소	점진적 증가	진동

한편 각 방법론에 필요한 파라미터 값은 이전 실험 [5,6,7]을 참고하여 표 2와 같이 설정하도록 한다.

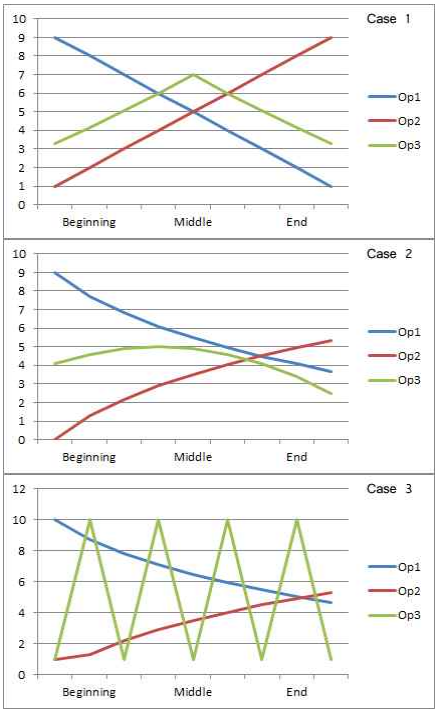

그림 1 상황별 연산자 결과 변화

표 2 AOS 파라메타 값

α	β	P_{min}	P_{max}
0.8	0.8	0.1	0.8

4.2 실험 결과

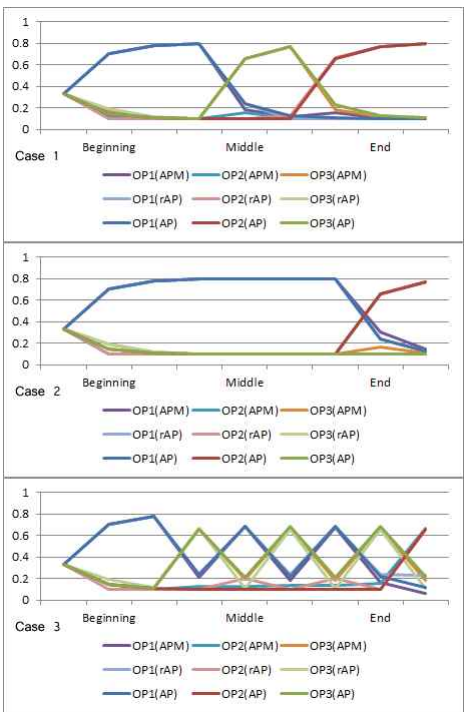

그림 2 상황별 연산자 확률 변화

모두 같은 방법 군에 속하기 때문인지, 세 가지 방법론은 서로 비슷한 경향을 보인다 (그림 2). 대체로 가장 좋은 성능의 연산자의 동작은 비슷한 가운데 이를 제외한 나머지 연산자의 동작에서 차이점을 발견할 수 있다.

AP의 경우 나머지 연산자가 어떤 성능을 가지고 있던지 간에 같은 비율로 값이 떨어지게 되며, APM은 각 성능의 비율을 그대로 반영하기에 값의 변화에 따라 비교적 민감하게 움직이지만 한편으로 값의 변화량은 적은 편이다. 마지막으로 rAP의 경우 값의 변화가 상대적으로 급격하게 변화하며, 좋은 해들의 값들이 상대적으로 높은 대신 제일 안 좋은 성능의 연산자는 최젓값 P_{min}에 가까운 값을 가지는 것을 확인할 수 있다.

한편 각 방법론의 적용 확률 값과 설정된 결과 값을 정규화 한 값의 곱의 합 (Sum of Product)를 구하여, 각 방법론의 상황별 예측도를 구해보면 표 3과 같다.

표 3 상황별 예측도

	AP	APM	rAP
상황1	0.388	0.389	**0.390**
상황2	0.370	**0.376**	0.372
상황3	**0.314**	0.312	0.301

이 수치는 성능을 제시해주는 객관적인 수치는 아니지만, 이를 바탕으로 상황에 따라 어떠한 방법론이 유리할 것인지 짐작할 수 있다.

5. 결론

이상으로 서로 다른 Adaptive Pursuit 방법론에 대해서 간단히 살펴보았다. 세 가지 방법론이 모두 같은 방법군에 속하기 때문에 큰 차이는 없었으나 각 상황에 따라서 연산자의 변화가 서로 다르게 일어나는 것을 확인할 수 있었다.

6. 참고문헌

[1] D. Goldberg, *Probability matching, the magnitude of reinforcement, and classier system bidding*, Machine Learning 5(4), 407-425, 1990.

[2] L. DaCosta, A. Fialho, M. Schoenauer, M. Sebag, *Adaptive operator selection with dynamic Multi-Armed bandits*, In: Proceedings of the 10th annual conference on Genetic and evolutionary computation, 913-920, 2008.

[3] A. Fialho, R. Ros, M. Schoenauer, M. Sebag, *Comparison-based Adaptive Strategy Selection with Bandits in Differential Evolution*, In: Proc. 11th International Conference on Parallel Problem Solving from Nature, LNCS 6238, 194-203, 2010.

[4] P. Auer, N. Cesa-Bianchi, P. Fischer, *Finite-time analysis of the multiarmed bandit problem*, Machine learning 47(2), 235-256, 2002.

[5] D. Thierens, *An adaptive pursuit strategy for allocating operator probabilities*, In: Proceedings of the 7th Genetic and Evolutionary Computation Conference, 1539-1546, 2005.

[6] MinHyeok Kim, R.I. McKay, Dong-Kyun Kim, Nguyen Xuan Hoai, *Evolutionary Operator Self-Adaptation with Diverse Operators*, In: Proceedings of 15th European Conference on Genetic Programming, LNCS 7244, 230-241, 2012.

[7] MinHyeok Kim, R.I. McKay, Nguyen Xuan Hoai, Kangil Kim, *Operator Self-Adaptation in Genetic Programming*, In: Proceedings of 14th European Conference on Genetic Programming, LNCS 6621, 215-226, 2011.