

협력적 공진화 차등진화

신성윤^{1*} · 이현창² · 신광성² · 김형진³ · 이재완¹

¹군산대학교 · ²원광대학교 · ³전북대학교

Cooperative Coevolution Differential Evolution

Seong-Yoon Shin^{1*} · Hyun-Chang Lee² · Kwang-Seong Shin² · Hyung-Jin Kim³ · Jae-Wan Lee²

¹Kunsan Natl. University · ²Wonkwang University · ³Jeonbuk Natl. University

E-mail : s3397220@kunsan.ac.kr / hclglory@wku.ac.kr / waver0920@wku.ac.kr /

kim@jbnu.ac.kr / jwlee@kunsan.ac.kr

요약

차등 진화는 연속 최적화 문제를 해결하기 위한 효율적인 알고리즘이다. 그러나 대규모 최적화 문제를 해결하기 위해 차등 진화를 적용하면 성능이 급격히 저하되고 런타임이 기하급수적으로 증가한다. 따라서 Spark(SparkDECC로 알려짐)를 기반으로 하는 새로운 협력 공진화 차등 진화가 제안된다. 분할 정복 전략은 SparkDECC에서 사용된다.

ABSTRACT

Differential evolution is an efficient algorithm for solving continuous optimization problems. However, applying differential evolution to solve large-scale optimization problems dramatically degrades performance and exponentially increases runtime. Therefore, a novel cooperative coevolution differential evolution based on Spark (known as SparkDECC) is proposed. The divide-and-conquer strategy is used in SparkDECC.

키워드

Differential evolution, continuous optimization problems, SparkDECC, divide-and-conquer strategy

I. 서론

차등 진화(DE) 알고리즘은 실제 코딩을 기반으로 하는 전역 최적화 알고리즘이다[1]. 단순성, 효율성 및 전역 병렬화로 인해 DE 알고리즘은 최근 몇 년 동안 산업 디자인 및 엔지니어링 최적화 분야에서 성공적으로 적용되었다. 또한 연구원들은 DE 알고리즘을 개선하고 혁신하여 일부 성과를 달성했다. 예를 들어, Brest et al. [2]는 제어 매개변수에 대한 적응적 방법을 구성하고 적응적 DE 알고리즘(jDE)을 제안했다. Wang et al. 3개의 선택된 돌연변이 전략과 3개의 제어 매개변수 그룹을 무작위로 결합하는 복합 DE 알고리즘(CoDE)[3]을 제안했다. 이러한 연구는 주로 저차원 문제(30차원)에 중점을 둔다. 그러나 고차원 문제(1000차원)를 처리할 때 이러한 DE 알고리즘의 성능이 크게 저하

되고 검색 시간이 차원에 따라 기하급수적으로 증가한다. 따라서 문제는 해결하기가 매우 어렵고 차원 문제가 지속된다[4].

II. SparkDECC

CC 프레임워크는 대규모 최적화 문제를 효과적으로 해결할 수 있습니다. 그러나 인구 규모가 증가함에 따라 CC 프레임워크에 필요한 시간이 빠르게 증가합니다. CC 프레임워크의 수렴 속도를 향상시키기 위해 클라우드 컴퓨팅의 장점을 CC 프레임워크와 결합하고 SparkDECC 기반 CC 알고리즘을 제안합니다.

SparkDECC 알고리즘은 먼저 무작위 그룹화 방법을 통해 고차원 문제를 여러 개의 저차원 하위 문제로 분해합니다. 여기서 하나의 하위 문제는 하위 개체군에 해당하고 전체 문제에서 각 하위 문제의 위

* corresponding author

치 정보는 보존됩니다. key_i 값을 기반으로 저차원 하위 집단은 RDD의 해당 파티션에 분산되고 각 파티션의 하위 집단은 DE 알고리즘의 돌연변이 및 교차 선택을 병렬로 실행합니다. 부분 모집단의 개별 적합도 값을 계산할 때 최종 라운드의 최적 개체를 선택하여 완전한 모집단을 구성하고 로컬 최적화를 수행합니다. 해당 파티션에서 여러 세대의 진화 후 저차원 하위 개체군은 위치 정보를 기반으로 새로운 완전한 개체군으로 병합됩니다. 전역 검색을 통해 최적의 개인을 반환하는 SparkDECC 알고리즘의 흐름도는 그림 1와 같습니다.

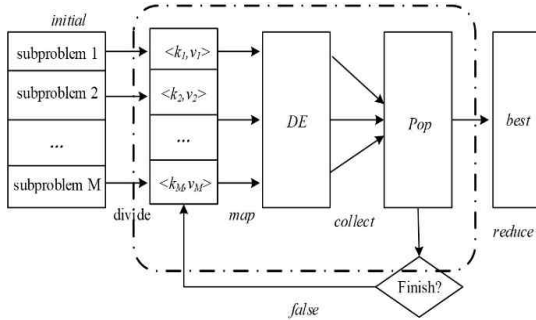


그림 1. SparkDECC 알고리즘의 흐름도

III. 실험

대규모 최적화 문제를 해결하기 위한 SparkDECC 알고리즘의 성능을 테스트하기 위해 문헌에서 선택한 13개의 테스트 함수를 사용하여 실험을 수행했습니다. 이 중 f1~f8은 단일 모드 함수, f9~f13은 다중 모드 함수, f4와 f5는 비분해 함수, 다른 모든 함수는 분해 가능합니다.

그림 2의 가속도 비율은 SparkDECC 알고리즘의 고차원 최적화 기능을 테스트할 때 파티션 수가 증가할수록 알고리즘의 실행 시간이 점차 감소하고 가속 효과가 향상됨을 보여줍니다.

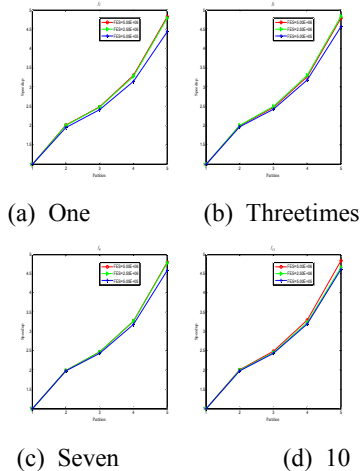


그림 2. 실험 결과(가속도 비율)

IV. 결론

본 논문에서는 새로운 반복 클라우드 컴퓨팅 모델을 사용하여 Spark 기반의 새로운 협업 클라우드 차등 진화 알고리즘(SparkDECC)을 개발했다. SparkDECC 알고리즘은 무작위 그룹화 전략을 통해 고차원 문제를 동일한 차원의 여러 저차원 하위 문제로 분해하고 각 하위 문제는 RDD 모델의 파티션과 일대일로 대응한다.

Acknowledgement

"This research is partially supported by Institute of Information and Telecommunication Technology of KNU."

References

- [1] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997. DOI: 10.1023/A:1008202821328.
- [2] J. Brest, S. Greiner, B. Boskovic B, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646-657, 2006. DOI:10.1109/TEVC.2006.872133.
- [3] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol 15, no. 1, pp. 55-66, 2011. DOI:10.1109/TEVC.2010.2087271.
- [4] F. V. d. Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225-239, 2004. DOI: 10.1109/TEVC.2004.826069.