The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 23, No. 4, pp.47-52, Aug. 31, 2023. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2023.23.4.47
JIIBC 2023-4-8

사물인터넷 시스템에서 가변적인 실시간 태스크를 지원하는 자원 플래닝 정책

A Resource Planning Policy to Support Variable Real-time Tasks in IoT Systems

반효경*, 정선화**

Hyokyung Bahn*, Sunhwa Annie Nam**

요 약 기계학습의 데이터 크기 및 컴퓨팅 부하 증가로 사물인터넷 시스템에서 에너지 효율적인 자원 플래닝이 중요해지고 있다. 본 논문에서는 사물인터넷 시스템의 실시간 워크로드 변화를 지원하는 자원 플래닝 정책을 제안한다. 이를 위해 본 논문은 실시간 태스크를 고정 태스크와 가변 태스크로 나누고 다양한 워크로드 상황에 대한 자원 플래닝 최적화를 수행한다. 이를 바탕으로 사물인터넷 시스템의 자원 설정을 고정 태스크 기반으로 시작한 후, 가변 태스크가 활성화될 경우 상황에 맞는 자원 플래닝을 즉시 적용할 수 있도록 한다. 시뮬레이션을 통해 제안한 정책이 사물인터넷 시스템의 프로세서 및 메모리 소모 에너지를 크게 줄일 수 있음을 보인다.

Abstract With the growing data size and the increased computing load in machine learning, energy-efficient resource planning in IoT systems is becoming increasingly important. In this paper, we suggest a new resource planning policy for real-time workloads that can be fluctuated over time in IoT systems. To handle such situations, we categorize real-time tasks into fixed tasks and variable tasks, and optimize the resource planning for various workload conditions. Based on this, we initiate the IoT system with the configuration for the fixed tasks, and when variable tasks are activated, we update the resource planning promptly for the situation. Simulation experiments show that the proposed policy saves the processor and memory energy significantly.

Key Words : Resource planning, Machine learning, IoT system, real-time task, evolutionary computation.

Ⅰ. 서 론

최근 모바일 학습 워크로드의 데이터 크기 및 컴퓨팅 부하 증가로 사물인터넷 시스템의 AI 워크로드 수행시 에너지 효율성이 중요해지고 있다[1]. 사물인터넷 시스템의 워크로드는 상황에 따라 가변적이므로[2, 3] 이를 고려한 배터리 효율적인 자원 플래닝이 중요하다[4]. 그러나, 전통적인 사물인터넷 시스템의 실시간 태스크 스케줄링은 고정된 자원 플래닝을 사용해왔다[5]. 본 논문에서는 시간에 따라 워크로드가 변할 수 있는 실시간 태스크를 위한 새로운 자원 플래닝을 제안한다. 이를 위해 본 논문에서는 먼저 실시간 태스크를 시스템과 주변 상황에 관

*정회원, 이화여자대학교 컴퓨터공학과
**비회원, 이화여자대학교 인공지능융합전공
접수일자 2023년 7월 15일, 수정완료 2023년 7월 30일
게재확정일자 2023년 8월 4일

Received: 15 July, 2023 / Revised: 30 July, 2023 /
Accepted: 4 August, 2023
*Corresponding Author: bahn@ewha.ac.kr
Dept. of Computer Engineering, Ewha University, Korea

계없이 항상 실행되는 고정 태스크와 특정 상황에서만 활성화되는 가변 태스크로 분류한다. 그런 다음 활성화된 태스크의 데드라인을 모두 만족하면서 에너지 절감을 극대화하는 자원 플래닝을 탐색한다. 사물인터넷 시스템은 통상적으로 배터리에 의해 전력이 공급되므로 자원 플래닝 시 에너지 절감은 매우 중요한 목표이다[6, 7]. 본 논문에서는 사물인터넷 시스템의 에너지 절감을 위해 프로세서의 전압을 동적으로 조절하는 기법과 사용되지 않는 메모리 영역을 저전력 모드로 전환하는 메모리 동면 기법을 활용한다. 동적 전압 조절 기법은 워크로드의 부하에 따라 프로세서의 처리 속도를 조절하여 에너지 소모를 줄이는 기술로 실시간 시스템에 널리 사용되고 있다[6]. 메모리 동면 기법은 태스크를 전력 소모가 큰 DRAM 대신 영속 메모리에 배치하고 DRAM의 해당 영역을 저전력 상태로 변화시키는 기술로 데드라인을 어기지 않는 선에서 태스크를 느린 영속 메모리로 배치하는 저전력 기술을 뜻한다[8, 9].

상기의 저전력 기술을 적용할 경우 에너지 소모는 줄어들지만 실행 시간이 늘어나므로 실시간 태스크의 데드라인을 어기지 않는 제약 조건을 가지고 에너지 소모를 최소화하는 자원 플래닝을 탐색하는 것이 필요하다[5]. 이를 위해 본 논문에서는 각 태스크의 실행시 프로세서의 공급 전압과 저장할 메모리 매체를 결정하는 문제를 고정 태스크뿐 아니라 시간에 따라 변화하는 가변 태스크의 조합도 고려하여 최적 설정을 탐색하고 이를 워크로드 변화 시 즉시 반영할 수 있도록 한다.

다양한 실시간 워크로드에 대한 시뮬레이션 실험을 통해 본 논문의 자원 플래닝이 사물인터넷 시스템의 에너지 소모를 평균 32% 절약할 수 있음을 확인하였다. 또한, 워크로드가 돌발적으로 변화하는 상황에서도 제안한 자원 플래닝은 실시간 태스크의 데드라인을 충족함을 확인하였다.

II. 가변 태스크를 위한 자원 플래닝

전통적인 실시간시스템은 데드라인(또는 실행주기)과 최악실행시간이 미리 정해져 있는 고정 태스크들을 다루므로, 주어진 태스크 집합에 대해 오프라인으로 자원 플래닝을 결정하는 것이 일반적이다[5]. 그러나, 차세대 사물인터넷 시스템은 시간에 따른 워크로드 변화가 생길 수 있으므로 효율적인 자원 플래닝은 이러한 워크로드 변화에 즉각적으로 대처할 수 있어야 한다.

이를 위해 본 논문에서는 다양한 실시간 태스크 조합에 대한 자원 설정을 오프라인으로 미리 최적화하고 실행 태스크의 변화 시 그에 대응하는 자원 플래닝을 즉각 적용할 수 있도록 한다. 구체적으로 살펴보면 각 태스크별 프로세서의 공급 전압과 메모리 위치 결정을 고정 태스크와 가변 태스크의 가능한 조합들에 대해 미리 최적화하고 그 결과를 유지하는 방식을 사용한다. 이러한 최적화 문제는 복잡한 문제 공간 탐색을 필요로 하므로, 본 논문에서는 이를 위해 진화 연산을 사용한다[10, 11]. 즉, 가능한 각각의 태스크 조합들에 대해 데드라인 충족을 제약 조건으로 프로세서와 메모리의 에너지 소모를 최소화하는 문제 공간 탐색을 진화연산을 통해 푸는 것이다.

본 논문의 자원 플래닝 과정은 두 단계로 구성된다. 첫 번째 단계에서는 고정 및 가변 태스크로 구성된 다양한 조합에 대해 오프라인으로 자원 플래닝의 최적화를 수행한다. 각 실시간 태스크의 실행을 위한 프로세서의 공급전압과 메모리 위치가 결정되면, 고정 태스크로 구성된 워크로드로 시스템의 실행을 시작한다. 두 번째 단계에서는 시간이 흐르면서 가변 태스크가 활성화될 때 1단계에서 구한 해당 태스크 집합에 대응하는 자원 플래닝을 적용한다.

한편, 프로세서의 동적 전압 조절과 DRAM의 동면 기법을 사용할 경우 시스템의 전력 소모를 절감할 수 있지만 대신 태스크의 실행시간이 늘어나는 부작용이 발생하므로 유효한 자원 플래닝인지를 확인하는 과정이 필요하다. 이를 위해 각 태스크의 저전력 기법 적용 이후 늘어난 최악실행시간을 실행주기로 나눈 태스크별 자원 이용률을 합했을 때 그 값이 자원의 한계치를 넘어서지 않는지 확인해야 한다. 이러한 제약조건을 만족하면 EDF(earliest deadline first) 알고리즘을 통해 모든 태스크들의 데드라인을 만족하는 유효한 스케줄을 얻을 수 있다[12].

가변 태스크가 활성화될 경우 자원 플래닝은 이를 고려한 자원 이용률 테스트를 만족해야 한다. 즉, 가변 태스크를 포함한 활성화된 모든 태스크들의 최악실행시간을 실행주기로 나눈 값을 합하여 자원의 한계치를 넘지 않는지 확인하고 이를 만족하는 자원 플래닝 중 에너지 소모가 가장 적은 스케줄을 진화 연산으로 탐색하고 그 결과에 의해 프로세서의 공급 전압과 메모리 배치를 결정한다. 이러한 작업을 워크로드가 실제 실행되기 전에 태스크의 가능한 조합에 대해 수행하므로 워크로드 변동 시 적절한 자원 플래닝을 즉각 반영할 수 있다. 이러한 오프라인 플래닝이 가능한 것은 실시간 태스크가 실행 중에 변경될 수는 있지만 가능한 가변 태스크들의 조합

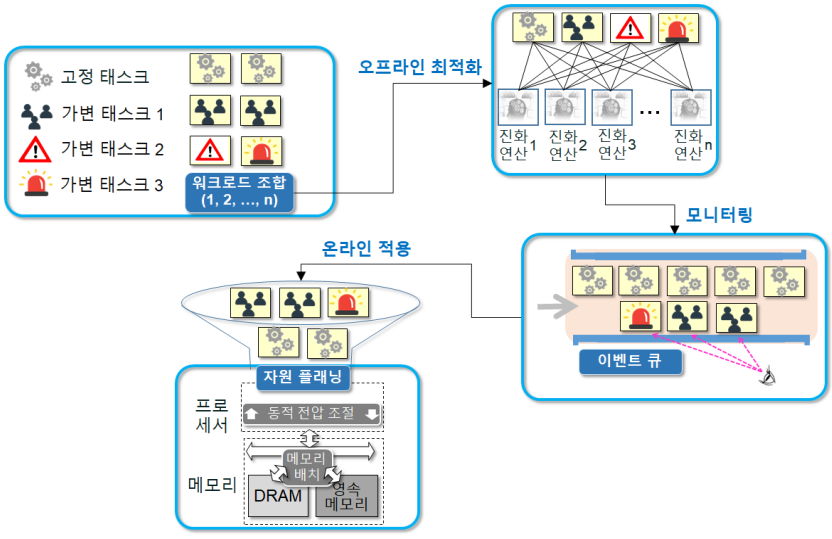

그림 1. 본 논문의 자원 플래닝 과정
Fig. 1. Resource planning process proposed in this paper.

을 미리 파악할 수 있기 때문이다. 그림 1은 본 논문의 자원 플래닝 과정을 개략적으로 보여주고 있다.

III. 진화 연산 기반의 최적화

진화 연산은 대규모 탐색 공간에 대해 가능한 모든 경우를 탐색하는 대신 집단 유전학의 개체 진화 원리를 모방하여 부분 해집합을 진화시켜 최적화를 수행하는 기법이다[10, 11]. 본 논문에서는 각 실시간 태스크의 실행 시 프로세서의 공급 전압과 메모리 위치를 결정하는 문제를 해결하고자 하며 태스크의 데드라인을 지키면서 전력 절감을 극대화하는 방식을 찾는 데에 그 목적이 있다.

이를 위해 본 논문에서는 충분히 좋은 결과값을 얻을 때까지 일부 제한된 해들로 구성된 해집합을 진화시킨다. 본 논문의 진화 연산은 먼저 랜덤하게 초기 해집합을 생성한 후 세대를 거듭하며 진화를 반복시킨다. 각 세대별 진화 과정은 먼저 해당 세대의 해집합에서 두 해를 확률적으로 선택한 후 이들을 조합하여 새로운 해를 생성하고 일정 확률로 생성 해에 대해 변이 작업을 수행한 후 그 결과를 해집합에 삽입한다. 대신 해집합 내에서 가장 가치가 낮은 해를 삭제한다. 이와 같은 진화의 반복은 해집합 내의 모든 해들이 유사한 가치를 가지는 해들로 수렴할 때까지 진행한다. 수렴된 해집합에서 가장 우수한 해를 추출하고 해당 해가 나타내는 자원 플래닝에 근거해 태스크별 프로세서 및 메모리 상태를 결정한다. 해의 가치를 평가하기 위해 본 논문에서는 각 해의 비용 함수를 정의하며, 이 값은 해당 해를 통해 결정된 프로세서 및 메모리 상태로 태스크 스케줄링을 했을 때 발생하는 에너지 소모량으로 정의된다. 한편, 에너지 소모량이 적더라도 일부 태스크에 대해 데드라인을 만족하지 못할 수 있으므로 비용 함수 내에 데드라인을 넘어선 태스크의 실행시간을 패널티 값으로 추가하여 유효하지 않은 해가 낮은 평가를 받아 멸종될 수 있도록 하였다.

IV. 성능 평가

제안한 자원 플래닝 정책의 효과를 입증하기 위해 시뮬레이션 실험을 수행하였다. 제안한 기법은 적응적 플래닝으로 명명하였으며, 이 기법과의 비교를 위해 고정 태스크 기반 플래닝, 최대 태스크 기반 플래닝, 기본 플래닝 등 3가지 자원 플래닝에 대한 시뮬레이션을 추가로 진행하였다. 기본 플래닝은 모든 실시간 태스크를 DRAM에 배치하고 프로세서를 기본 전압 모드로 실행시킨다. 고정 태스크 기반 플래닝과 최대 태스크 기반 플래닝 역시 진화 연산을 통해 프로세서와 메모리의 자원 플래닝을 최적화한다는 측면에서 적응적 플래닝과 동일하나 이들은 워크로드 변화를 고려하지 않는다. 고정 태스크 기반 플래닝의 경우 프로세서와 메모리의 상태를 고정 태스크에 기반해서 최적화하며 워크로드가 수행되는 동안 자원 플래닝을 변경하지 않는다. 최대 태스크 기반

플래닝은 고정 태스크와 가변 태스크가 최대로 실행되는 상황을 전제로 자원 플래닝을 최적화하여 이를 전체 워크로드 구간에 적용한다. DRAM과 영속 메모리의 크기는 모든 실시간 태스크를 동시에 수용할 수 있는 충분한 크기로 하되, 태스크가 영속 메모리로 배치된 경우 해당 DRAM 영역은 부분적으로 동면 상태로 전환할 수 있는 것으로 가정한다[8].

본 논문에서는 다양한 실시간 태스크 상황을 고려하기 위해 서로 다른 3가지 워크로드 시나리오에 대해 시뮬레이션을 수행하였다. 각 시나리오에서 고정 태스크는 모든 실행 구간에서 항상 활성화되며, 가변 태스크는 시간에 따라 일부 구간에서 활성화된다. 표 1은 본 논문의 3가지 시나리오에 대한 구체적인 내용을 보여주고 있다. 표에서 활성 태스크는 시간대별로 활성화된 태스크 및 그 부하를 보여준다. 예를 들어, 시나리오 2의 경우 실행 시간의 앞부분 30% 구간에서는 고정 태스크만 활성화되며 이때 고정 태스크의 부하는 0.3임을 나타낸다. 그런 다음 부하가 0.5인 가변 태스크가 이후 실행 시간의 70% 동안 활성화되며, 이 시간대에 고정 태스크도 여전히 실행 중이므로 전체 작업 부하는 0.8이 된다.

그림 2는 4가지 플래닝 정책의 에너지 소모량을 3가지 시나리오에 대해 비교해서 보여주고 있다. 그래프에서 에너지 소모량은 기본 플래닝의 에너지 소모량에 대한 정규화된 형태로 표시하였다. 그림에서 보는 것처럼 본 논문이 제안한 적응적 플래닝이 모든 워크로드 시나리오에 대해 많은 에너지 절약 효과를 나타내는 것을 확인할 수 있다. 특히, 시나리오 1과 시나리오 3에서 에너지 절약 효과가 상대적으로 크게 나타났으며, 이는 워크로드의 부하가 이 두 시나리오에서 상대적으로 낮은 편이어서 데드라인을 만족하면서 에너지 절감 기법을 적용할 수 있는 여지가 더 많기 때문으로 볼 수 있다. 적응적 플래닝 정책의 도입을 통해 기본 플래닝 대비 평균 32%의 에너지 절감 효과가 나타났으며, 비록 고정 태스크 기반 플래닝이 에너지 절감 측면에서 더 좋은 결과를 나타내었으나, 이 방법은 가변 태스크가 활성화되었을 때 실시간 태스크의 데드라인을 충족하지 못해 유효하지 않은 결과를 초래함을 확인할 수 있었다. 최대 태스크 기반 플래닝은 적응적 플래닝과 비교했을 때 경쟁력 있는 에너지 절감 효과를 나타내지 못했으며, 이는 이 정책이 가변 태스크의 활성화 여부와 무관하게 전체 실행 기간 동안 최대 태스크가 실행되는 것을 전제로 자원 플래닝을 했기 때문이다.

표 1. 워크로드 시나리오
Table 1. Workload scenarios

시나리오	활성 태스크 (작업 부하)	실행시간 비율
1	고정(0.2)-가변1(0.2)-가변2(0.6)	50%-20%-30%
2	고정(0.3)-가변(0.5)	30%-70%
3	고정(0.1)-가변1(0.4)-가변2 (0.7)	25%-50%-25%

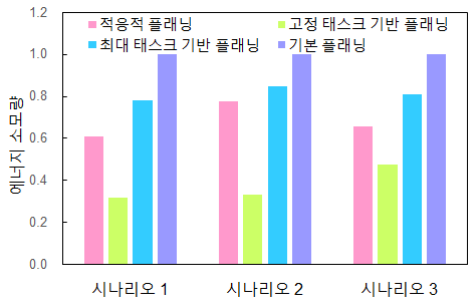

그림 2. 에너지 소모량 비교
Fig. 2. Comparison of energy consumption.

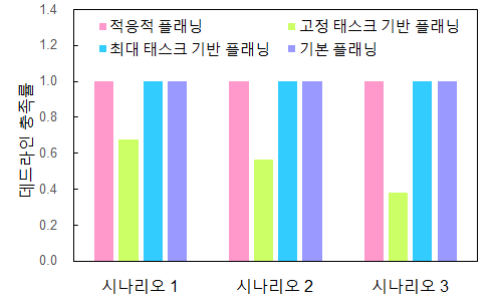

그림 3. 데드라인 충족률 비교
Fig. 3. Comparison of deadline meet ratio.

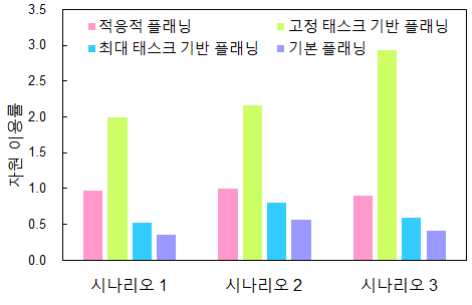

그림 4. 자원 이용률 비교
Fig. 4. Comparison of resource utilization.

그림 3은 4가지 자원 플래닝 정책에 대한 데드라인 충족률을 보여주고 있다. 그림에서 보는 것처럼 적응적 플래닝, 최대 태스크 기반 플래닝, 기본 플래닝은 워크로드 시나리오에 관계없이 모두 데드라인을 충족함을 확인할 수 있다. 참고로 적응적 플래닝은 나머지 두 정책 대비 에너지 절약 효과가 훨씬 큼에도 두 정책과 마찬 가지로 데드라인 충족률이 100%임을 확인할 수 있다. 이 세 정책과 달리 고정 태스크 기반 플래닝의 경우 데드라인 충족률이 매우 낮았으며, 이는 이 정책이 가변 태스크의 활성화를 고려하지 않기 때문이다. 시나리오 3의 경우 고정 태스크의 부하가 매우 낮은 반면 가변 태스크의 부하가 크게 증가하는 워크로드로 구성되었기 때문에 고정 태스크 기반 플래닝의 데드라인 충족률이 크게 저하되는 것을 확인할 수 있다.

그림 4는 4가지 정책의 자원 이용률을 보여주고 있다. 그림에서 보는 것처럼 적응적 플래닝이 모든 시나리오에서 0.95 이상의 높은 자원 이용률을 나타낸 것을 확인할 수 있다. 참고로 본 논문의 모든 시나리오에서는 태스크들의 최대 부하가 0.8 이하로 설정되었기 때문에 1에 가까운 이용률은 저전력 기법들을 공격적으로 적용하여 얻은 결과로 볼 수 있다. 최대 태스크 기반 플래닝과 기본 플래닝의 자원이용률은 적응적 플래닝보다 모든 경우에 있어 낮게 나타났으며 이는 실제로 부하가 높지 않은 상황에서도 이들 정책들이 높은 부하를 가정해서 자원 플래닝을 하기 때문으로 볼 수 있다. 최대 태스크 기반 플래닝의 자원 이용률이 기본 플래닝에 비해 높게 나타났으며 이는 최대 태스크 기반 플래닝이 저전력 기법들을 일정 수준까지 적용하기 때문이다. 그러나, 워크로드 변화에 대처하지 않기 때문에 그 효과는 제한적임을 알 수 있다. 고정 태스크 기반 플래닝의 경우 자원 이용률이 1을 넘는 매우 높은 값으로 나타났으며, 이는 주어진 자원의 최대치를 활용해서도 얻을 수 없는 유효하지 않은 결과로 볼 수 있다. 이는 그림 3의 데드라인 충족률이 낮은 것에서도 그 근거를 확인할 수 있다.

V. 결 론

전통적인 실시간 시스템에서는 모든 태스크의 데드라인을 보장하기 위해 오프라인 스케줄링에 기반한 자원 플래닝을 수행해왔다. 그러나, 이러한 방식은 워크로드 변화에 대처할 수 없어 비효율적인 자원 상황을 초래할 수 있다. 본 논문에서는 가변 태스크가 포함된 사물인터넷 시스템을 위한 자원 플래닝 정책을 제안하였다. 제안하는 정책은 워크로드 변화에 대처하기 위해 고정 및 가변 태스크로 구성된 워크로드 조합에 대해 자원 플래닝을 각각 실시한다. 구체적으로는 진화 연산을 통해 프로세서의 공급전압과 메모리 상태의 조합을 최적화하고 그 결과를 워크로드 조합별로 보존하여 상황 변화에 맞게 자원 플래닝을 즉시 반영할 수 있도록 한다. 시뮬레이션을 통해 제안한 정책이 사물인터넷 시스템의 전력을 평균 32%을 줄이면서 실시간 태스크의 데드라인을 충족하는 것을 확인하였다.

References

[1] J. Wan, M. Yi, D. Li, C. Zhang, S. Wang, and K. Zhou, “Mobile services for customization manufacturing systems: an example of industry 4.0,” IEEE Access, vol. 4, pp. 8977-8986, 2016.
DOI: http://doi.org/10.1109/ACCESS.2016.2631152.

[2] L. Liu, S. Lu, R. Zhong, B. Wu, Y. Yao, Q. Zhang, and W. Shi, “Computing systems for autonomous driving: state of the art and challenges,” IEEE Internet of Things Journal, vol. 8, no. 8, pp. 6469-6486, 2021.
DOI: http://doi.org/10.1109/JIOT.2020.3043716.

[3] S. Yoo, Y. Jo, and H. Bahn, “Integrated scheduling of real-time and interactive tasks for configurable industrial systems,” IEEE Trans. Industrial Informatics, vol. 18, no. 1, pp. 631-641, 2022.
DOI: http://doi.org/10.1109/TII.2021.3067714

[4] D. Kim, S. Lee, and H. Bahn, “An adaptive location detection scheme for energy-efficiency of smartphones,” Pervasive and Mobile Computing, vol. 31, pp. 67-78, 2016.
DOI: http://doi.org/10.1016/j.pmcj.2016.04.012

[5] S. Nam, K. Cho, and H. Bahn, “Tight evaluation of real-time task schedulability for processor’s DVS and nonvolatile memory allocation,” Micromachines, vol. 10, article 371, 2019.
DOI: http://doi.org/10.3390/mi10060371

[6] H.E. Ghor and E.M. Aggoune, “Energy saving EDF scheduling for wireless sensors on variable voltage processors,” Int. J. Adv. Comput. Sci. Appl., vol. 5, no. 2, pp. 158-167, 2014.
DOI: https://doi.org/10.14569/IJACSA.2014.050223

[7] M. Huh and S. Shin, “A study on the energy saving house system using IoT technology,” The Journal of The Institute of Internet, Broadcasting and Communication (JIIBC), vol. 20, no. 6, pp. 109-113, 2020.
DOI: https://doi.org/10.7236/JIIBC.2020.20.6.109

[8] S. Lee, H. Bahn, and S.H. Noh, “CLOCK-DWF: a

write-history-aware page replacement algorithm for hybrid PCM and DRAM memory architectures," IEEE Trans. Comput., vol. 63, no. 9, pp. 2187-2200, 2014. DOI: http://doi.org/10.1109/TC.2013.98.

[9] E. Lee, H. Kang, H. Bahn, and K. G. Shin, "Eliminating periodic flush overhead of file I/O with non-volatile buffer cache," IEEE Trans. Comput., vol. 65, no. 4, pp. 1145-1157, 2016. DOI: http://doi.org/10.1109/TC.2014.2349525.

[10] J. Gong and S. Lee, "A study on performance comparison of combination of genetic algorithm and ant colony optimization for multi-robot coverage path planning," Journal of KIIT, vol. 21, no. 6, pp. 55-65, 2023. DOI: http://doi.org/10.14801/jkiit.2023.21.6.55

[11] S. Yang and Y. Kim, "Optimization of process parameters of incremental sheet forming of Al3004 sheet using genetic algorithm-BP neural network," Journal of the Korea Academia-Industrial cooperation Society, vol. 21, no. 1, pp. 560-567, 2020. DOI: https://doi.org/10.5762/KAIS.2020.21.1.560

[12] H. Chetto and M. Chetto, "Some results of the earliest deadline scheduling algorithm," IEEE Trans. Software Engineering, vol. 15, no. 10, pp. 1261-1269, 1989. DOI: http://doi.org/10.1109/TSE.1989.559777.

저 자 소 개

반 효 경(정회원)

- 1997년 : 서울대학교 계산통계학과 학사
- 1999년 : 서울대학교 전산과학과 석사
- 2002년 : 서울대학교 컴퓨터공학부 박사.
- 2002년 ~ : 이화여자대학교 컴퓨터공학과 교수.
- 주관심분야 : 운영체제, 스토리지시스템, 임베디드시스템

정 선 화(비회원)

- 1986년 : 이화여자대학교 물리학과 학사
- 1996년 : Illinois Institute of Technology 컴퓨터과학과 석사
- 2018년 : 이화여자대학교 컴퓨터공학과 박사
- 2019년 ~ 2020년 : Seattle Pacific University 조교수
- 2022년 ~ : 이화여자대학교 AI융합전공 연구교수
- 주관심분야 : 운영체제, 소프트웨어 테스팅, 인공지능, 임베디드 시스템

※ This work was partly supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub) and (No.RS-2022-00155966, Artificial Intelligence Convergence Innovation Human Resources Development (Ewha University)).