

## 동적계획법을 위한 계산 그리드 작업 스케줄링 정책

조지훈<sup>○</sup> 이원주\* 전창호\*\*

코리아와이즈넷, 인하공업전문대학 컴퓨터정보과, 한양대학교 전자컴퓨터공학부

[deadwi@wisenu.co.kr](mailto:deadwi@wisenu.co.kr), [wonjoo2@inhac.ac.kr](mailto:wonjoo2@inhac.ac.kr), [chj5193@hanyang.ac.kr](mailto:chj5193@hanyang.ac.kr)

## A Computational Grid Job Scheduling Policy for Dynamic Programming

Jihun Jo<sup>○</sup> Wonjoo Lee\* Changho Jeon\*\*

Korean Wisenut Laboratory<sup>○</sup>,

Dept. of Computer Science, Inha Technical College\*,

School of Electrical and Computer Engineering, Hanyang University\*\*

그리드 환경이 가지는 대용량의 데이터 수용 능력은 동적계획법에서 요구하는 동적 테이블을 구성하는데 적합하다 또한 그리드 환경은 동적 테이블 구성에 필요한 방대한 연산 능력을 가지고 있기 때문에 동적계획법에 적합하다

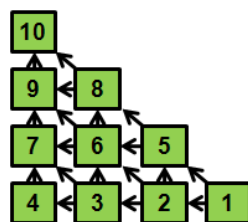
본 논문에서는 동적계획법에 적합한 계산 그리드 작업 스케줄링 정책을 제안한다. 정책은 동적 테이블 분산 할당 과정과 동적 테이블 구성 과정으로 나뉜다. 동적 테이블 분산 할당 과정은 동적 테이블을 분할하고, 특정 노드에 할당하는 과정이다. 이때 각 노드는 마스터 노드와 슬레이브 노드로 분류한다. 마스터 노드는 동적 테이블을 분산 할당하는 기능을 한다.

동적 테이블 할당 과정은 그리드 환경의 정보 획득 및 갱신 과정과 테이블 할당 과정으로 나뉘어진다. 그리드 환경의 정보 획득 및 갱신 과정은 다음과 같은 과정을 수행한다.

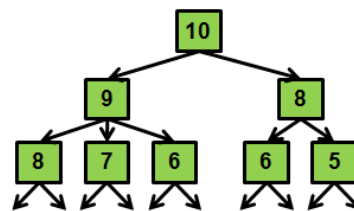
- ① 노드는 마스터 노드에 그리드 환경 참여 요청을 하고 마스터 노드로부터 노드 리스트를 반환 받는다.
- ② 모든 노드의 네트워크 지연 정보를 얻는다.
- ③ 네트워크 지연이 적은 상위 m개의 인접 노드를 선정한다.
- ④ 인접 노드의 목록을 주기적으로 갱신한다.

테이블 할당 과정은 다음과 같은 과정을 수행한다.

- ① 연관된 테이블이 할당된 노드에 인접 노드 목록을 요청한다.
- ② 네트워크 지연이 최소인 인접 노드를 선정한다.
- ③ 마스터 노드에 테이블 할당을 요청한다.
- ④ 해당 노드에 테이블을 할당한다.



(a) 상향식 접근 방법



(b) 하향식 접근 방법

그림 1. 동적 테이블 구성 방법

동적 테이블 구성 과정은 분산된 동적 테이블을 구성하는 과정이다. 동적 테이블을 할당 받은 각 노드는 자신의 테이블을 구성하는 연산을 수행한다. 동적 계획법은 일반적으로 그림 1 (a) 상향식 접근 방법으로 동적 테이블을 구성한다. 하지만 그림 1 (b) 하향식 접근 방법으로 변형한다면 독립성을 가진 하위문제로 분할 가능하기 때문에 병행성을 확보할 수 있다.

본 논문에서는 기존의 분기-한정 알고리즘에 비해 동적 계획법의 성능이 우수함을 시뮬레이션을 통하여 검증한다. LCS(Longest Common Subsequence) 문제를 대상으로 분기-한정 알고리즘과 동적 계획법의 문제 크기에 따른 총 소요시간을 측정한 결과는 그림 2와 같다.

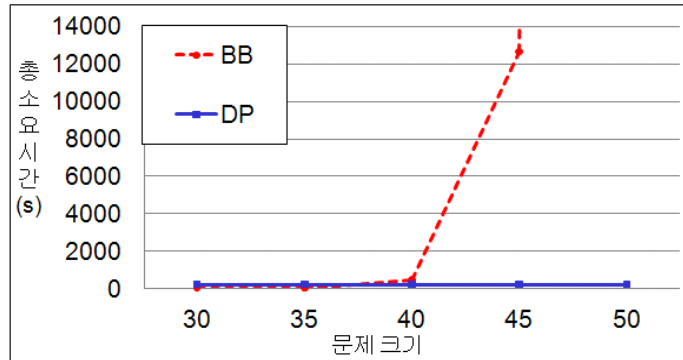


그림 2. 문제 크기에 따른 총소요시간

그림 2를 살펴보면 분기-한정(BB)의 경우 문제 크기가 40 이상에서는 총소요시간이 급격히 증가함을 알 수 있다. 하지만 동적 계획법(DP)의 총소요시간은 거의 변함이 없다. 따라서 문제 크기 40 이상에서 최적해를 구하는 것은 동적 계획법이 더 유리함을 알 수 있다.

문제 크기에 따른 계산점유율을 측정한 결과는 그림 3과 같다.

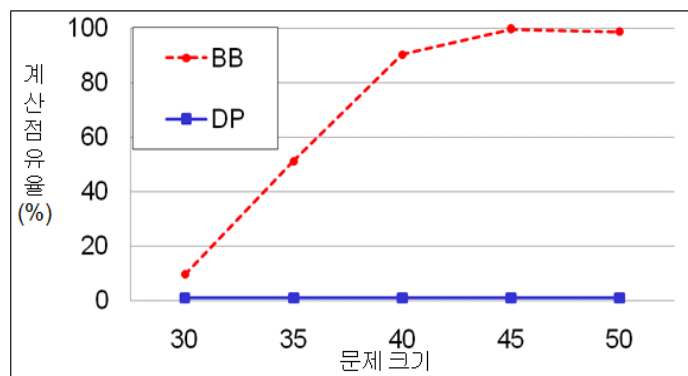


그림 3. 문제 크기에 따른 계산 점유율

그림 3을 살펴보면 분기-한정(BB) 알고리즘은 문제 크기 45 이상에서는 CPU 자원을 약 100% 쓰고 있지만 동적 계획법(DP)은 CPU 자원 점유율이 일정함을 알 수 있다.

기존의 분기-한정 알고리즘은 문제 크기가 증가하면서 총소요시간과 CPU 점유율이 급격하게 증가함을 알 수 있다. 하지만 본 논문에서 제안한 그리드 작업 스케줄링 정책은 문제 크기가 증가하여도 총소요시간과 CPU 점유율이 크게 변하지 않았다. 따라서 그리드 환경에서는 기존의 분기-한정 알고리즘에 비해 본 논문에서 제안한 동적 계획법을 위한 작업 스케줄링 정책이 우수함을 알 수 있었다.