

# MIN-based 다중 처리 시스템을 위한 효율적인 병렬 Branch-and-Bound 알고리즘 설계 및 성능 분석

## Design and Performance Analysis of a Parallel Optimal Branch-and-Bound Algorithm for MIN-based Multiprocessors

梁 明 國\*

(Myung-Kook Yang)

### 요 약

본 논문에서는 다중 연결 구조(Multistage Interconnection Network, MIN)를 기반으로 하는 병렬 컴퓨터 환경에서 효과적으로 운용할 수 있는 병렬 Optimal Best-First search Branch-and-Bound 알고리즘(pobs)을 제안하고, 성능을 분석하였다. 제안된 알고리즘은 먼저 해를 얻고자 하는 문제를 임의의  $G$ 개 부분 문제로 분할하고, 소수 프로세서로 구성된 프로세서 그룹들에 할당하여 각각의 지역 해를 산출하도록 하였다. 따라서  $N$ 개의 프로세서를 갖는 시스템은  $G$ 개 프로세서 그룹으로 구분되고, 각 프로세서 그룹은  $P = N/G$ 개 프로세서를 보유하게 된다. 각 프로세서 그룹은 할당된 부분 문제의 지역 해를 얻는 과정에 병렬 sub-Global Best-First B&B 알고리즘을 수행한다. 프로세서 그룹들이 산출한 지역 해들 가운데 최선의 값을 갖는 지역 해가 문제의 전역 해로 결정되는데, 이를 위하여 각 프로세서 그룹의 대표 프로세서는 할당된 부분 문제의 지역 해를 다른 그룹들에게 전파하도록 하였다. 지역 해 전파는 프로세서 그룹들의 지역 해 비교를 통한 전역해 선정 기능과 함께 프로세서 그룹간 작업 불균형 문제를 상당 부분 해소하는 효과를 제공한다. 알고리즘 설계에 이어 성능 평가를 위한 분석 모형을 제시하였다. 제안한 모형은 B&B 알고리즘 수행에 따른 연산 소요시간과 통신 소요시간을 분리하여 처리함으로써 병렬 처리 환경에서 보다 실질적인 알고리즘 성능 평가가 가능하게 함과 동시에, 다양한 컴퓨터 연결 구조에서의 알고리즘 성능 예측을 용이하게 하였다. B&B 알고리즘의 확률 특성을 토대로 작성된 성능 분석 연구의 실효성 검토를 위하여 MIN 기반 시스템을 대상으로 병행된 시뮬레이션 결과는 상호 미세한 오차 범위 내에서 일치하는 결과를 보여, 제시한 성능 분석 기법의 타당성을 입증하였다. 또한, 본 논문에서 제안한 병렬 알고리즘을 MIN 기반 시스템에 적용하여 기존 알고리즘의 성능과 비교 평가 결과, 제안한 pobs가 문제 해결 과정에서 전개되는 부분 문제 수를 줄이고, 프로세서간의 효율적인 작업 분배 효과를 제공하는 한편, 프로세서간의 주된 통신 활동 범위를 국부적으로 제한하여 성능면에서 우수함을 입증하였다.

### Abstract

In this paper, a parallel Optimal Best-First search Branch-and-Bound(B&B) algorithm(pobs) is designed and evaluated for MIN-based multiprocessor systems. The proposed algorithm decomposes a problem into  $G$  subproblems, where each subproblem is processed on a group of  $P$  processors. Each processor group uses the sub-Global Best-First search technique to find a local solution. The local solutions are broadcasted through the network to compute the global solution. This broadcast provides not only the comparison of  $G$  local solutions but also the load balancing among the processor groups. A performance analysis is then conducted to estimate the speed-up of the proposed parallel B&B algorithm. The analytical model is developed based on the probabilistic properties of the B&B algorithm. It considers both the computation time and communication overheads to evaluate the realistic performance of the algorithm under the parallel processing environment. In order to validate the proposed evaluation model, the simulation of the parallel B&B algorithm on a MIN-based system is carried out at the same time. The results from both analysis and simulation match closely. It is also shown that the proposed Optimal Best-First search B&B algorithm performs better than other reported schemes with its various advantageous features such as: less subproblem evaluations, proper load balancing, and limited scope of remote communication. Keywords: Parallel Best-First search Branch-and-Bound algorithms, Processor group, Performance (speed-up) analysis, Computation time, Communication overhead

\* 蔚山大學校 電氣電子 및 自動化 工學部

(School of Electrical Engineering, Univ. of Ulsan)

接受日: 1997年7月7日, 修正完了日: 1997年10月16日

## I. 서 론

Branch-and-Bound(B&B) 알고리즘은 Traveling Salesman Problem, Knapsack Problem, Vertex Covering Problem, 및 Integer Programming 등 NP-hard Combinatorial Search Problem을 효과적으로 해결하는 기법으로 널리 인식되고있다<sup>[1-6]</sup>. B&B 알고리즘은 주어진 문제를 임의의 부 문제들로 분할 전개하고, 이들 부 문제들을 다시 분할 전개하는 작업을 반복하여 얻어진 말단 부 문제들의 해를 비교 평가하여 전체 해를 찾아내는 휴리스틱 기법이다. 알고리즘은 전개(Branching) 단계, 선택(Selection) 단계, 제거(Elimination) 단계, 및 종료(Termination) 단계 등 크게 4단계로 구성되는데, 이들 중 전개 방식, 제거 방식, 및 종료 방식은 알고리즘을 적용하는 문제의 성질에 따라 결정되는 반면, 선택 방식은 알고리즘 자체의 특성에 따라 정해진다. 선택 단계는 현재 분할 전개된 부 문제들 중 하나의 부 문제를 선정하여 다음 전개의 모체가 되도록 하는 단계로, B&B 알고리즘의 효율을 결정하는 중요한 요소이다. B&B 알고리즘은 선택 방식의 채택에 따라 Breadth-First search B&B 알고리즘<sup>[1]</sup>, Depth-First search B&B 알고리즘<sup>[1]</sup>, Best-First search B&B 알고리즘<sup>[6-8]</sup>, 그리고 Random search B&B 알고리즘<sup>[9]</sup> 등으로 구분된다. 이들 가운데 Best-First search 기법이 성능면에서 다른 방식과 비교하여 월등한 것으로 밝혀졌다.

Wah<sup>[6-8]</sup> 등은 Parallel Global Best-First Search B&B 알고리즘(*pgbs*)을 제안하고 이의 성능을 시뮬레이션을 통하여 제시하는 한편, 알고리즘을 효율적으로 지원할 수 있도록 설계된 MANIP 시스템을 발표하였다. *pgbs*의 성능은 시스템이 가지고있는 프로세서 개수와 선형 비례하는 것으로 보고되었으나, 시뮬레이션 과정에서 프로세서간의 통신 소요시간을 고려하지않아 실질적인 알고리즘 성능으로 인정되지 못하였다. 병렬 처리의 경우 프로세서간의 통신 소요시간은 연산 소요시간과 함께 알고리즘 성능을 결정하는 중요한 요소로 평가되고있다. 특히 통합처리(Globalization) 방식은 알고리즘 수행 과정에 프로세서간 과도한 데이터 이동이 요구되어 경우에 따라 통신 시간이 연산 시간을 능가하여 전체 알고리즘 효율을 저하시키는 결과를 초래하기도 한다.

통합처리 방식과 달리, 분할처리(Decomposition) 방식

은 프로세서간의 원거리 통신을 최소화하는 기법으로 다양한 병렬 B&B 알고리즘의 기반으로 활용되고있다. 분할처리 기법은 먼저 주어진 문제를 프로세서 개수 만큼의 부 문제로 전개하여 각 프로세서에 할당하고, 각 프로세서는 할당된 부 문제를 다른 프로세서의 도움 없이 단독으로 처리하여 지역 해를 구하게 된다. 따라서 알고리즘 수행 과정에 요구되는 대부분의 통신 활동이 해당 프로세서와 지역 메모리 사이에서 발생하여 통신 소요시간을 최소화한다. 그러나 분할처리 방식은 각 프로세서에 할당하는 작업 부하가 부 문제의 특성에 따라 일정하지않아 프로세서간 작업 불균형 현상을 초래하고, 이로 인하여 알고리즘 효율 저하 현상을 유발한다. Karp와 Zhang<sup>[10]</sup>은 Parallel Decomposite Best-First Search B&B 알고리즘(*pdfs*)을 제안하며, 프로세서간의 작업 균형 유지를 위하여 무작위 작업 분배(randomized load balancing) 기법을 발표하였다. 알고리즘 성능 평가에서 Karp와 Zhang은 프로세서간의 통신 시간을 단위 시간으로 일정하게 설정하고, 네트워크 병목 현상이 발생하지 않는다는 가정을 적용하여 *pdfs*의 우수함을 주장하였다. Kumar와 Rao<sup>[11-13]</sup>는 Parallel Decomposite Depth-First search B&B 알고리즘(*pdds*)을 그리고 Janakiram과 Agrawal<sup>[9]</sup> 등은 Parallel Decomposite Random search B&B 알고리즘(*pdrs*)을 발표하였다. Depth-First search 방식이나 Random search 방식은 Best-First search 방식에 비해 성능이 떨어질 뿐 아니라 Kumar와 Rao가 제안한 프로세서간의 작업 균형 유지 기법은 프로세서간 데이터 이동을 요구하여, 원격 통신 시간이 과중한 시스템의 경우 알고리즘 성능을 급격히 저하시키는 요인이 된다. 또한 Janakiram과 Agrawal이 제시한 알고리즘 성능 분석 모형은 문제의 복잡도가 증가할수록 시뮬레이션 결과와 수학적 계산 결과 간의 오차가 과도하게 증가하는 양상을 보이고있다.

Yang과 Das<sup>[14-15]</sup>는 다층 연결 구조(Multistage Interconnection Network, MIN)를 기반으로 하는 시스템에 병렬 Decomposite Best-First search B&B 알고리즘(*pdfs*)을 적용하고, 시험 분석하였다. 분할처리 기법을 근거로 제안된 알고리즘은 프로세서간 작업 불균형 문제를 보완하기 위하여 각 프로세서가 할당받은 부 문제의 지역 해를 방송하도록 함으로, 제한된 원격 통신 활동으로 만족할만한 작업 균형 효과를 거두었다. Yang 과 Das는 시뮬레이션과 수학적 분석 결과를 통하여 MIN을 기반으로 하는 대규모 시스템에서 *pdfs* 방식이 *pgbs* 방

식 보다 우월한 것으로 평가하였다. 이는 분할처리 기법을 기반으로 제안된 *pdbs*가 프로세서간 원거리 통신 활동을 최소화하도록 설계됨으로 얻어진 결과이다.

MIN환경에서는 시스템 규모가 커지면 프로세서간 원격 통신에 소요되는 시간이 증가함과 동시에 네트워크 상에서 데이터 충돌 확률도 높아져 프로세서간 지속적인 정보교환을 요구하는 *pgbs* 방식의 성능이 *pdbs* 방식에 비하여 상대적으로 열세에 놓이게 된다. 그러나 시스템 규모가 작은 경우 ( $N \leq 4$ ) 프로세서간의 정보 교환에 소요되는 시간이 감소하여, Yang과 Das의 분석에서 보인 바와 같이, *pgbs* 방식이 *pdbs* 방식보다 성능면에서 앞서는 결과를 보인다. 본 논문에서는 다중 연결 구조(MIN)를 기반으로 하는 병렬 컴퓨터를 대상으로 시스템 규모에 관계없이 모든 프로세서를 효율적으로 운용하여 성능을 향상시킬 수 있는 Parallel Optimized Best-First search Branch-and-Bound 알고리즘 (*pobs*)을 제안하고 성능을 분석하였다. 제안된 알고리즘은 프로세서의 개수가 적을 때 우수한 성능을 보이는 *pgbs* 방식의 장점과 시스템 규모가 커질수록 더 좋은 성능을 제공하는 *pdbs* 방식의 장점을 공히 활용하도록 설계하였다. 먼저 주어진 문제를 임의의  $G$ 개 부 문제로 분할하고, 각 부 문제의 지역 해를 얻는 과정에 소수의 프로세서로 구성된 프로세서 그룹이 Parallel sub-Global Best-First-search B&B Algorithm(*psgbs*)을 수행하도록 하였다. 따라서  $N$ 개의 프로세서를 갖는 시스템은  $G$ 개 프로세서 그룹으로 나뉘고, 각 프로세서 그룹은  $P = N/G$ 개 프로세서를 보유한다. 알고리즘 설계에 이어, 알고리즘의 확률 특성을 근거로 성능 분석 모형을 제시하였다. 본 논문에 기술된 성능 분석 모형은 B&B 알고리즘 수행에 따른 연산 소요시간과 통신 소요시간을 분리하여 처리함으로써 병렬 처리 환경에서 보다 실질적인 알고리즘 성능 평가가 가능하게 함과 동시에, 다양한 컴퓨터 연결 구조에서의 알고리즘 성능 예측이 용이하게 하였다. 수학적 성능 분석 연구의 실효성 검토를 위하여 MIN 기반 시스템을 대상으로 3000개의 상이한 문제 공간에 수행된 시뮬레이션 결과는 상호 미세한 오차 범위 내에서 성능 분석 모형의 예측 데이터와 일치하는 결과를 보여, 제시한 분석 모형의 타당성을 입증하였다. 또한 제안한 병렬 Optimal Best-First search Branch-and-Bound 알고리즘 (*pobs*)을 BBN Butterfly 시스템<sup>[16]</sup>에 적용하여 기존 알고리즘들의 성능과 비교 평가 결과, *pobs*가 다양한 규모의 다중 연결 구조 시스템 환경에서 우수한 성능을 제공하는 것으로 입증되었다.

본 논문의 구성은 다음과 같다. I절 서문에 이어 II절에서

는 본 논문에서 제안한 *pobs* 알고리즘을 소개하였다. III 절에서는 *pobs* 알고리즘 수행에 따른 연산 및 프로세서간 통신에 소요되는 시간 분석을 위한 수학적 모형을 제시하고, 이를 이용하여 얻은 성능 분석 데이터를 시뮬레이션 결과와 비교하였다. 이어서 제안한 *pobs* 알고리즘과 이미 발표된 병렬 B&B 알고리즘들의 성능을 비교하여 IV절에 기술하였다. 마지막 절에는 본 연구의 효과 및 결론을 요약하였다.

## II. 병렬 Optimal Best-First search B & B 알고리즘

본 논문에서 제안한 병렬 Optimal Best-First search B&B 알고리즘(*pobs*)은 다음과 같은 3단계 과정으로 나누어 수행된다.

- 단계 1: 부 문제 전개 및 그룹 별 작업 분배
- 단계 2: 프로세서 그룹 별 Parallel sub-Global Best-First search B&B Algorithm (*psgbs*) 수행
- 단계 3: 그룹 별 지역 해 방송 및 전역 해 선정

단계 1은 주어진 문제를 특성에 따라 적절한 방법으로 분할-전개하여  $G$ 개의 부 문제로 나누고 이를 해당 프로세서 그룹에 분배하는 과정으로 모든 병렬 B&B 알고리즘에서 공통적으로 수행되는 과정이다. 각 프로세서 그룹은  $P$ 개의 프로세서로 구성되어, 전체 시스템이 보유한 프로세서가  $N$ 개일 경우,  $N = G \times P$ 가 된다.

부 문제 생성에 이어 작업 분배가 완결되면 각 프로세서 그룹은 해당 부 문제의 지역 해를 얻기 위하여 단계 2 과정을 수행한다. 이때 각 프로세서 그룹은 보유하고 있는 소수( $P \leq 4$ )의 프로세서를 이용하여 *psgbs*를 수행한다. 프로세서 그룹별 *psgbs* 실행은 프로세서간의 정보 교환 범위를 그룹내부로 제한하여, 기존 *pgbs* 방식의 문제점으로 지적된 네트워크 소통량 과다 현상을 극복함과 동시에, 그룹내부 프로세서간 효율적인 작업 부하 균형 유지효과를 통하여 *pdbs* 방식보다 성능면에서 앞서는 결과를 제공한다.

마지막으로 단계 3에서는 단계 2에서 얻은  $G$ 개 지역 해들을 비교하여 가장 이상적인 지역 해를 선정, 전역 해로 지정하게 된다. 단계 3에서 각 프로세서 그룹은 할당된 부 문제의 지역 해를 얻는 즉시 다른 그룹들에게

전파(Broadcast)하도록 하였다. 이와 같은 지역 해 전파 기법은 각 프로세서 그룹이  $psgbs$ 를 수행하여 얻은 연산 결과를 비교하는 기능과 함께, 프로세서 그룹간의 작업 불균형 문제를 상당 부분 해소하는 효과를 제공한다. 또한 최대  $G$ 번의 프로세서 그룹간 데이터 전송을 필요로 함으로, 이로 인한 성능 저하 정도가 대단히 미세한 것으로 밝혀졌다. 지역 해 전파 기법을 간단히 정리하면 다음과 같다. 먼저 각 프로세서 그룹은 지역 해 변수(Local Solution: LS) 및 중간 해 변수(Global Feasible Solution, GFS) 등 두개의 데이터 변수를 유지하며, 2 단계 시작 시점에 각 프로세서 그룹은 그룹 별로 이들 두 변수를 초기화시킨 후  $psgbs$ 를 수행한다. 제일 먼저 할당된 부 문제를 해결한 그룹은 해당 그룹 내 LS 및 GFS 데이터를 연산 결과 얻은 지역 해로 갱신하고, 이를 다른 프로세서 그룹에 전파한 후 작업 종료로 선언한다. 처음 전송된 지역 해는 일단 모든 그룹의 GFS에 저장 되고, 작업 중인 프로세서 그룹은 이를 이용하여 그룹 내 전개 대기 부 문제(Candidate Subproblem(CS)) 리스트를 재정비 하여 필요 없는 CS를 리스트에서 제거한 후 작업을 계속한다. 이후 할당된 부 문제를 해결한 그룹은 순차적으로 작업 종료로 선언하게 되는데, 이에 앞서 새로 얻은 지역 해를 기존의 GFS와 비교하여 필요에 따라 GFS 갱신하고 이를 다른 프로세서 그룹에 전파한다. 이와 같은 과정을 반복하여 시스템 전체적으로는 총  $G$ 개의 지역 해를 연산하고, 필요에 따라 GFS를 수정하게 된다. 마지막으로 각 프로세서 그룹은 각자 유지하고있는 LS와 GFS 변수를 검색, 서로 같은 값을 가질 경우 해당 지역 해 관련 데이터를 시스템 전체 대표 프로세서로 전송하여 주어진 문제의 전역 해를 구하게 된다.

그림 1은 2진 트리 형태로 전개되는 문제에 제한한 알고리즘을 적용하여 처리할 경우 각 단계별 동작 내용과 통신 활동 범위를 보여주고 있다. 알고리즘 운용을 위하여 그림 1에서는 전체 시스템을 네개 프로세서 그룹으로 분리하고, 각 프로세서 그룹은 소수 프로세서들로 구성된 것으로 가정하였다. 그림 1의 단계 1: 부 문제 전개 및 그룹별 작업 분배 과정에는 먼저 각 프로세서 그룹의 대표 프로세서들이 네개 부 문제를 생성한다. 이 같이 구성된 부 문제는 그룹 대표 프로세서간의 데이터 전송을 통하여 해당 그룹에 할당된다. 단계 2: 프로세서 그룹 별  $psgbs$  수행 과정에는 그룹 내부의 프로세서들이 지역 CS 리스트를 유지하며 그룹 내부 프로세서들간의

작업 부하 균형을 이루고, 나아가서 할당된 부 문제의 해를 산출한다. 2단계 과정은 제안한 알고리즘 작동 시 대부분의 시간을 소모하는 단계로, 지역 CS 리스트를 중심으로 그룹 내부 프로세서간의 데이터 이동이 활발하게 진행된다. 단계 3: 그룹 별 지역 해 전파 및 전역 해 선정 과정은 각 그룹에서 산출한 지역 해들 가운데 가장 이상적인 해를 찾아내는 단계로, 각 그룹 대표 프로세서가 해당 지역 해를 다른 그룹에 전파하는 그룹 대표 프로세서간 데이터 이동이 요구된다.

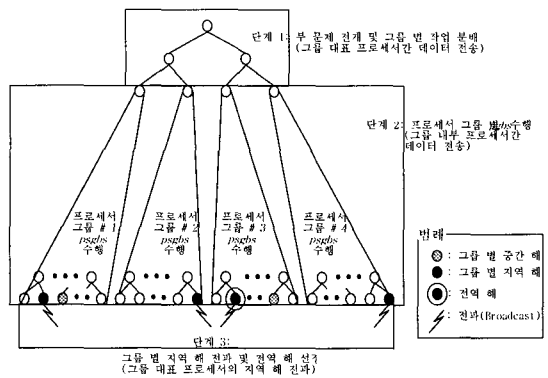


그림 1. 병렬 Optimal Best-First search B&B 알고리즘 ( $G = 4$ )

Fig. 1. Parallel Optimal Best-First search B&B Algorithm( $G = 4$ ).

### III. 성능 분석

제한한 병렬 Optimal Best-First search B&B 알고리즘의 성능을 예측할 수 있는 수학적 분석 모형을 얻기 위하여 먼저 B&B 알고리즘의 동작 특성을 검토하고, 알고리즘 실행에 따른 순수 연산 소요시간과 데이터 이동 소요시간을 분리하여, 각각의 수학적 확률 모형을 제시하였다. 본 절에 기술한 수학적 성능 분석 모형은 단일 프로세서 시스템 환경에서 B&B 알고리즘을 이용하여 Traveling Salesman Problem(TSP), Integer Programming(IP) 등 잘 알려진 최소치 추구 NP-hard combinatorial search 문제의 해를 구하는 과정에 얻어지는 결과 트리의 특성을 확률적으로 해석하여 제시되었다. 다양한 문제들의 해결과정에서 얻어진 결과 트리를

검토하여 요약한 B&B 알고리즘 결과 트리 특성을 요약 하던 다음과 같다.

- 1) 알고리즘 수행에 따라 트리 내 임의의 부 문제는  $a$  개의 부 문제로 전개되며, 이들 부 문제의 전개를 반복적으로 수행하여 트리 만단에 이르러 중간 해 혹은 최종 해를 얻게 된다. 따라서 전체 문제 공간은  $a$ -ary 트리 형태로 전개되고, 결과 트리의 크기는 주어진 문제의 복잡도에 따라 임의의 깊이 (depth) $D$ 로 주어진다.
- 2) 임의의 부 문제의 연산 결과치가 이미 얻어진 중간 해 보다 작은 값을 가질 경우 해당 부 문제는 물론이고, 이로부터 전개되는  $a$ 개의 하위 부 문제의 연산 결과치도 검색한다.
- 3) 임의의 어미 부 문제의 연산 결과치와 이로부터 전개되어 생성된 부 문제의 연산 결과치의 차이  $x_i$ 은 최소값  $x_b$ 와 최대값  $x_c$ 사이에서 균등 분포된 정수 값으로  $x_b$ 와  $x_c$ 사이에서 무작위로 결정된다. 이때  $x_b$ 와  $x_c$ 는 문제의 특성에 따라 주어진다.

B&B 알고리즘으로 처리된 결과 트리 특성 가운데 제 3)항은 B&B 알고리즘 연구에 일반적으로 주어지는 가정으로 기 발표된 연구 논문에서 가정의 타당성을 입증하였다. 특히 3)항에서 정수 값으로 주어진  $x_i$ 은 실수 영역으로 확대 해석할 수 있다. 예를 들어 최소값  $x_b(= 1)$ 와 최대값  $x_c(= 100)$ 사이에서 균등 분포된 정수 값으로 주어지는  $x_i$ 은 최소값  $x_b(= 0.1)$ 와 최대값  $x_c(= 10)$ 사이에서 0.1 간격으로 균등 분포된 실수 값으로 주어지는  $x_i$ 의 경우와 동일한 결과를 제공하게 된다. 또한 3)항에 의거 트리 형태로 전개된 전체 문제 공간 모형의 임의의 부 문제의 연산 결과치( $v_c$ )는 해당 어미 부 문제의 연산 결과치( $v_p$ )와 해당  $x_i$ 의 합으로 생각할 수 있다. 즉  $v_c = v_p + x_i$ . 따라서 제안한 B&B 알고리즘 결과 트리 모형에서 임의의 깊이  $d$ 에 위치한 부 문제의 연산 결과치는

$$v_d = v_{root} + \sum_{i=1}^d x_i \quad (1)$$

로 계산된다. 여기서  $v_{root}$ 는 트리 정점에 위치한 개괄적 문제의 연산 결과치를, 그리고  $x_i$ 는 해당 계보의 깊이 ( $i - 1$ )에 위치한 어미 부 문제의 연산 결과치와 깊이  $i$ 에 위치한 하위 부 문제의 연산 결과치 간의 차이를 나타낸

다. 식 (1)에서  $v_d=1$ 은 트리 깊이 1에 위치한 부 문제의 연산 결과치로, 3)항의 가정에 의해 ( $v_{root} + x_b$ )와 ( $v_{root} + x_c$ ) 사이에 균등 분포된 정수 값 중 하나로 주어진다. 또한  $v_d$  값은 트리의 깊이가 증가하며 ( $v_{root} + x_b \times d$ )와 ( $v_{root} + x_c \times d$ ) 사이에 분포 확률이 표준 분포 형태로 존재하는 정수 값 중 하나로 계산된다.

서론에 기술한 바와 같이 Best-First search 기법은 다른 B&B 알고리즘에 비하여 성능이 우수한 것으로 밝혀졌다. 이는 Best-First 선택 방식이 차기 전개 대상 부 문제 선정 시 문제가 추구하는 최종 해에 근사한 연산 값을 갖는 부 문제를 먼저 선정하는데 기인한다. 이로 인하여 최종 해 도달 시간이 다른 알고리즘에 비하여 현저하게 빠르고, 문제 해결 과정에 연산되는 부 문제 수도 줄어들게 된다. 최종 해 도달 시간이 빠르다는 의미는 최종 해에 이르기 전 거치는 중간 해의 수가 적다는 것으로, 다양한 TSP 시뮬레이션 결과, 제일 처음 얻은 중간 해가 이후 부 문제 전개 과정을 지나 그대로 최종 해로 결정되는 경우가 자주 관찰되었다. 따라서 본 논문에서 제안한 성능 분석 모형은 결과 트리의 특성 가운데 제 2)항의 내용을 Best-First search 기법의 특수성을 고려할 때 최소 결과치 추구 문제의 경우 "임의의 부 문제의 연산 결과치가 최종 해보다 작은 값을 가질 경우 해당 부 문제는 물론이고, 이로부터 전개되는  $a$ 의 하위 부 문제들의 연산 결과치도 검색하여야 한다."로 수정하여 개발되었다.

제한한 병렬 알고리즘 분석 모형의 이해를 돕고, 문제 해석을 용이하게 하기 위하여 알고리즘 수행에 요구되는 프로세서 그룹의 수  $G$  및 각 그룹이 보유하는 프로세서 수  $P$ 와 임의의 부 문제로부터 전개되는 하위 부 문제 수  $a$  간에  $G = a^\eta$  및  $P = a^\omega$  ( $\eta, \omega$  : 정수)의 관계가 성립하는 것으로 가정하였다. 이와 같은 조건하에서 병렬 B&B 알고리즘의 제 1단계 및 2단계 초반의 부 문제 생성 및 작업 분배 과정에서 수행되는 단계별 분배 기법(step-by-step spread technique)<sup>[15]</sup>은 각각  $\eta(= \log_a G)$ 번 및  $\omega(= \log_a P)$ 번의 부 문제 전개를 시행하고, 각 부 문제 전개 때마다  $\theta(= \lceil \log_2 a \rceil)$ 번의 그룹간 정보 이동이 수행된다. 주어진 가정은 병렬 B&B 알고리즘을 효과적으로 수행하기 위하여 일반적으로 요구되는 처리 환경이며, 이와 같은 가정하에 기술된 본 절의 병렬 B&B 알고리즘 성능 평가 기법은 MIN을 기반으로 하는 모든 시스템에 해당 파라미터 수정으로, 그리고 일반적인  $P, G$  및  $a$  값이 주어지는 환경에 다소의 분석 모형 수정으로 용이하게 적용할 수 있다. 제안한 수학적 성능 분석 기법의 실효성은 시뮬레이션

결과와 비교를 통하여 입증하였다.

### III-1. 연산 소요시간 분석

제안한 *pobs* 알고리즘을 이용하여 주어진 문제를 깊이  $D$ 의  $a$ -ary 트리 형태로 전개하여  $G$ 개 부 문제들을 생성하고, 시스템이 보유한  $N$ 개 프로세서를  $G$ 개 그룹으로 분할하여 동작시켜 문제의 해를 구하는데 소요되는 연산 시간,  $TE(D, a, N)$ ,은 각 프로세서가 알고리즘을 수행하면서 처리하는 부 문제의 수,  $E(D, a, N)$ ,를 분석함으로 구할 수 있다. 먼저 *pobs* 알고리즘의 제 1단계 부 문제 생성 및 작업 분배 과정은 각 프로세서 그룹의 대표 프로세서가 부 문제를 할당받아 처리하는 과정으로 해당 대표 프로세서를 제외한 다른 프로세서는 대기상태에 있게 된다. 단계별 분배 기법<sup>[15]</sup>을 적용하여 처리되는 제 1단계 과정에서 모두  $\sum_{i=0}^{l-1} a^i$  개의 부 문제가 처리되는데, 이를 위하여  $\eta (= \log_a G)$  번의 부 문제 전개가 시행되고 각 부 문제 전개 시 해당 대표 프로세서가 하나의 부 문제를 할당받아 연산 처리하게 된다. 따라서 제 1단계 과정에서 시스템이 부 문제 생성 및 작업 분배를 위하여 소모하는 시간,  $TE(D, a, N)_{phase1}$ , 은

$$TE(D, a, N)_{phase1} = \eta \times t_e \quad (2)$$

로 계산된다. 여기서  $t_e$ 는 하나의 부 문제를 연산하는데 소요되는 시간이다.

프로세서 그룹 별 *psgbs*를 수행하는 제 2단계에서 각 프로세서 그룹은  $P$ 개 프로세서를 동작시켜 할당받은 부 문제를 정점으로 전개를 수행하여 깊이  $D^*(= D - \eta)$ 의  $a$ -ary트리 형태로 부 문제들을 생성하고, 문제의 지역 해를 구하게 된다. 이때 그룹 내 각 프로세서에서 처리되는 부 문제의 수,  $E(D^*, a, P)_{phase2}$ , 는

$$E(D^*, a, P)_{phase2} = \max \{E(D^*, a, 1)/P, \omega + 1 + a \times (D^* - \omega)\} \quad (3)$$

와 같이 계산된다<sup>[15]</sup>. 여기서  $\omega = \log_a P$ 로 그룹 내 모든 프로세서가 대표 프로세서로부터 전개된 부 문제를

할당 받을 때까지 해당 프로세서가 처리하는 부 문제의 수를 나타낸다. 또한  $E(D^*, a, 1)$ 는 단일 프로세서 시스템이 Best First search B&B 알고리즘을 이용하여 깊이  $D^*$ 의  $a$ -ary트리 형태로 전개되는 문제를 해결할 때 처리하여야 할 부 문제 수를 파라미터화 한 것으로, Yang과 Das<sup>[15]</sup>에 의하여 정의되었다. 식 (3)에서  $\max$  표현의 첫 항으로 주어진  $\{E(D^*, a, 1) / P\}$ 는  $P$ 개 프로세서가 깊이  $D^*$ 의  $a$ -ary트리 형태로 전개되는 검색 공간을 *psgbs* 기법으로 처리할 때 각 프로세서가 해결하게 되는 부 문제의 수를 수식화 한 것이다. 이때 그룹 내  $P$ 개 프로세서는 그룹 내에서 전개되는 부 문제들을 공평 분배하여 처리하여, Wah<sup>[8]</sup> 등에 의해 연구된 바와 같이, 프로세서 개수에 따라 선형 성능 향상을 피할 수 있게 된다. 또한 식 (3)의  $\max$  표현 두번째 항으로 기술된  $\{\omega + 1 + a \times (D^* - \omega)\}$ 은  $P$ 개의 프로세서로 깊이  $D^*$ 의  $a$ -ary트리 형태로 전개되는 검색 공간을 병렬 처리할 때 임의의 프로세서가 해결하게 되는 최소 부 문제 수를 나타낸다. 여기서  $\omega$ 는 그룹에 할당된 부 문제를 전개하여 그룹 내 모든 프로세서에게 문제를 할당하기까지 단계별 전개 기법을 수행하는 과정에서 해당 프로세서가 처리하는 부 문제의 개수이다. 그리고  $(1 + a \times (D^* - \omega))$ 는 그룹 내 지역 해를 최단 시간에 얻었을 경우 각 프로세서가 처리하게 되는 부 문제의 수를 나타낸다. 이때 임의의 프로세서는 트리의 각 레벨에서 하나의 부 문제만 할당받아 전개하고, 이로부터  $a$ 개의 하위 부 문제를 생성하여 연산하게 된다. 따라서 *pobs* 제 2단계 처리 과정에서 최소 부 문제를 전개하고 종료될 경우 각 프로세서는  $(D^* - \omega)$  레벨을 한번씩만 거치며 각 레벨에서  $a$ 개 부 문제를 전개하고, 이를 통하여 해당 그룹 내 지역 해를 얻게 된다. 식 (3)의  $\max$ 식은  $\{E(D^*, a, 1) / P\}$ 가  $\{\omega + 1 + a \times (D^* - \omega)\}$ 보다 산술적으로 작은 값으로 주어질 경우, 이를 바로잡도록 유도하고있다. 식 (3)의 결과에 따라 *pobs* 제 2단계 과정에서 각 프로세서 그룹이 지역 해를 구하는데 소요되는 시간,  $TE(D^*, a, P)_{phase2}$ ,는

$$\begin{aligned} TE(D^*, a, P)_{phase2} &= E(D^*, a, P)_{phase2} \times t_e \\ &= \max \{E(D^*, a, 1)/P, 1 + \omega + a \times (D^* - \omega)\} \times t_e \end{aligned} \quad (4)$$

이고, 따라서 총 연산 소요시간,  $TE(D, a, N)_{pobs}$ , 은

식 (2)와 식 (4)로부터

$$\begin{aligned} TE(D, a, N)pobs \\ = TE(D, a, N)_{phase1} + TE(D^*, a, P)_{phase2} \\ = [\eta + \max\{E(D^*, a, 1)/P, 1 + \omega + a \times (D^* - \omega)\}] \times te \end{aligned} \quad (5)$$

와 같이 계산된다.

## II-2. 데이터 이동 소요시간 분석

제안한 병렬 Optimal Best-First search B&B 알고리즘(*pobs*) 실행에 따른 데이터 이동 형태는 앞서 기술한 알고리즘의 단계별 수행에 따라 다음과 같이 구분된다.

- 단계 1: 그룹 별 작업 분배를 위한 그룹 간 데이터 이동
- 단계 2: *psgbs* 수행에 따른 그룹 내부 프로세서 간 데이터 이동
- 단계 3: 전역 해 선정을 위한 그룹 별 지역 해 전파

그룹 별 작업 분배가 수행되는 제 1단계 과정에서  $G$  개의 부 문제를 생성하기 위하여  $a$ -ary 트리 전개에 경우,  $\eta (= \log_a G)$  번의 부 문제 전개가 요구된다. 또한 임의의 부 문제로부터 전개된  $a$ 개의 하위 부 문제는 단계 별 전개 기법에 의해 해당 프로세서로 분배되는데, 이를 위하여 각 부 문제 전개 시마다  $\vartheta (= \lceil \log_2 a \rceil)$  번의 그룹간 데이터 이동이 시행된다. 다층 연결 구조를 기반으로 하는 시스템에서 다량의 데이터를 분할하여 다수 해당 프로세서들에게 네트워크 충돌 없이 효율적으로 분배할 수 있도록 제안된 단계별 분배 기법은  $a$ 개 부 문제 분배를 위하여  $\vartheta$ 번 데이터 이동을 관리하게 되는데, 처음  $\gamma[1] (= \lfloor a / 2 \rfloor)$ 개 부 문제 데이터, 두 번째  $\gamma[2] (= \lfloor \gamma[1] / 2 \rfloor)$ 개 부 문제 데이터,  $j$ 번째 이동되는 데이터  $\gamma[j] (= \lfloor \gamma[j-1] / 2 \rfloor)$ 개 부 문제 데이터, 그리고  $\vartheta$ 번째  $\gamma[\vartheta] (= 1)$ 개 부 문제 데이터가 되도록 한다. 따라서 단계 1에서 소요되는 데이터 이동 시간,  $TC(D, a, N)_{phase1}$ , 은

$$TC(D, a, N)_{phase1} = \eta \times \sum_{j=1}^{\vartheta} \gamma[j] \times \sigma \times t_{tb} \quad (6)$$

과 같이 계산된다. 여기서  $t_{tb}$ 는 네트워크를 통한 단위 데이터 불럭 전송 시간을 나타내고,  $\sigma$ 는 전송되는 하나의 부 문제가 차지하는 데이터의 크기를 불럭 단위로 전환한 값을 표시한다.

제 2단계에서 각 프로세서 그룹은 할당받은 부 문제를 깊이  $D^* (= D - \eta)$ 의  $a$ -ary 트리 형태로 전개하며 *psgbs*를 수행, 해당 부 문제의 지역 해를 구하게 된다. 제 2단계 초반에는 1단계와 유사한 부 문제 전개 및 작업 분배를 위한 데이터 이동이 수행되어, 각 그룹 내  $P$ 개 프로세서들이 순차적으로 부 문제를 할당 받는다. 따라서 각 프로세서 그룹의 대표 프로세서는 알고리즘 1단계 종료 시 분배된 부 문제를 전개하여  $P$ 개 부 문제로 분할하고 이를 그룹 내 각 프로세서에게 분배하게 되는데, 이를 위하여  $\omega (= \log_a P)$  번의 부 문제 전개가 요구된다. 또한 각 부 문제 전개 시마다  $\vartheta (= \lceil \log_2 a \rceil)$  번의 그룹간 정보 이동이 시행된다. 제안한 *pobs* 제 2단계 초반에 소요되는 통신 시간,  $TC(D^*, a, P)_{phase2.1}$ , 은

$$TC(D^*, a, P)_{phase2.1} = \omega \times \sum_{j=1}^{\vartheta} \gamma[j] \times \sigma \times t_{tb} \quad (7)$$

로 얻어진다. 여기서  $\gamma[j]$ ,  $\sigma$ , 그리고  $t_{tb}$ 는 식(6)에서 정의된 내용과 동일하다.

일단 모든 프로세서가 제 2단계 초반을 지나며 부 문제를 할당 받게 되면, 이때부터 그룹 대표 프로세서는 그룹 내 각 프로세서가 전개한 부 문제를 모으고, 검색하여 다음에 처리할 부 문제를 선정한 후, 이를 다시 각 프로세서에 할당하는 한편, 선정에서 제외된 부 문제는 CS 리스트에 저장하여 추후 처리에 활용하게 된다. 알고리즘 2단계 후반에 요구되는 세부 통신 활동 내역을 정리하면 다음과 같이 6개 과정으로 구분된다.

- 과정 1(전개 종료 시험): 임의 프로세서가 할당 받은 부 문제를 전개하여 생성한 하위 부 문제의 전개 종료 여부 시험을 위한 현 지역 해와의 비교
- 과정 2(부 문제 소집): 과정 1을 통과한 하위 부 문제를 그룹 대표 프로세서에게 전송.
- 과정 3(CS 리스트 정리): 전개한 부 문제의 연산 결과가 새로운 지역 해로 판명될 경우 이를 이용하여 CS 리스트상의 불필요한 부 문제 제거

과정 4(부 문제 선정): 다음 단계에서 그룹 내부의 각 프로세서가 전개할 P개 부 문제 선정.

과정 5(CS 리스트 정리III): 과정 2에서 소집된 부 문제 중 과정 4에서 선정되지 못한 부 문제를 CS 리스트에 삽입.

과정 6(부 문제 분배): 과정 4에서 선정된 부 문제를 그룹 내 프로세서들에게 분배.

과정 1, 3, 4, 5는 해당 프로세싱 소자 내부에서의 데이터 이동으로 처리되는 반면, 과정 2와 6의 부 문제 소집 및 분배는 다층 연결 구조를 통한 원거리 통신이 요구되어, 제 2단계 통신 소요시간의 대부분을 차지한다. 본 절에서는 먼저 *pobs*의 제 2단계: 그룹 별 *psgbs* 수행에 따른 부 문제 소집 및 분배에 소요되는 통신 시간을 분석하였다.

### III-2-1. 부 문제 소집

임의 프로세서 그룹이 *psgbs* 기법으로 할당 받은 부 문제를 처리하는 과정에서 그룹 대표 프로세서는 각 프로세서가 전개한 부 문제 가운데 지역 해 보다 작은 값을 갖는 하위 부 문제들, 즉 전개 종료 시험을 통과한 부 문제들을 소집하여 모으고, 이들로부터 다음에 전개할 P개 부 문제를 선정하게 된다. 트리 각 레벨마다 전개된 부 문제 가운데 그룹 대표 프로세서로 전송되는 부 문제는 Best-First search B&B 알고리즘의 수정된 특성 제 2항에 의거 해당 부 문제의 연산 결과치가 그룹 지역 해보다 작은 값을 가지는 부 문제들로 근사화하여 정의할 수 있다. P개 프로세서로 *psgbs* 알고리즘을 수행할 경우 하나의 지역 중간 해 선정을 위하여 최대  $(P \times a)$ 개 중간 해를 검색하게 됨으로 단일 프로세서 환경에서보다 최적 중간 해에 도달하는 시간이 단축되고, 나아가서 처음 얻은 중간 해가 그대로 지역 해로 결정될 확률이 높아져 성능 분석 과정에 Best-First search B&B 알고리즘의 수정된 특성 제 2항 적용으로 인한 오차 범위가 더욱 미세하여진다. 따라서 P개 프로세서로 구성된 임의 프로세서 그룹이 할당 받은 부 문제를 *psgbs* 기법으로 처리하여 깊이  $D^*$  a-ary 트리 형태로 전개할 경우, 그룹 대표 프로세서로 소집되는 총 부 문제 수,  $AS[D^*, a, P]$ , 는

$$\begin{aligned}
 AS[D^*, a, P] &= a^{\theta+1} \\
 &+ v \sum_{d=\theta+2}^{D^*-1} \theta_{d-1} \times \sum_{y=v_{d-1}(min)}^{v_{parent}(lim)} (P\{v_{d-1} \\
 &= y\} \times \sum_{j=1}^a \left[ \binom{a}{j} m^j (1-m)^{a-j} \times j \right] ) + \\
 &\theta_{D^*-1} \times \sum_{y=v_{D^*-1}(min)}^{v_{parent}(lim)} (P\{v_{D^*-1} \\
 &= y\} \times \sum_{j=1}^a \left[ \binom{a}{j} m^j (1-m)^{a-j} \right] ) \\
 &+ AS[D^*-1, a, P], \tag{8}
 \end{aligned}$$

와 같이 계산된다. 여기서  $D^* > \theta (= \log_a P)$ ,  $\theta_{d-1} = (a^{\theta+1} - P) \times a^{d-2-\theta}$ ,  $v_{parent}(lim) = \lfloor \Phi D^* - x_b \rfloor$ ,  $m = \min \{ \Phi D^* - y - x_b + 1, x_c - x_b + 1 \} / (x_c - x_b + 1)$ , 그리고  $AS[\theta + 1, a, P] = a^\theta = P$  이다. 식 (8)에서  $v_{d-1}$ 은 트리 내부 임의 깊이  $(d - 1)$ 에 위치한 임의 부 문제의 연산 결과치를 표시하며,  $v_{d-1}(min)$ 은 트리 내부 임의 깊이  $(d - 1)$ 에 위치한 임의 부 문제가 제공할 수 있는 최소 연산 결과치를, 그리고  $v_{parent}(lim)$ 은 임의 부 문제  $\zeta$ 로부터 하위 부 문제가 전개되기 위한  $\zeta$ 의 최대 연산 결과치를 나타낸다. 임의 부 문제  $\zeta$ 의 연산 결과치가  $v_{parent}(lim)$ 일 경우, 이로부터 전개되는 하위 부 문제의 연산 결과치가 최소 증가분  $(= x_b)$  만큼 증가할 때 해당 하위 부 문제는 그룹 대표 프로세서로 전송되어 향후 처리 과정에 참여하게 된다. 즉  $[v_{parent}(lim) + x_b] = \Phi D^*$  ( $=$  지역 해). 다시 정리하면, 임의 부 문제  $\zeta$ 의 연산 결과치가  $v_{parent}(lim)$ 를 초과할 경우 부 문제  $\zeta$ 는 더 이상의 전개 가치를 상실하여 대표 프로세서로 소집되지 않는다.

$\Phi D^*$ 는 임의 그룹에 할당된 부 문제를 깊이  $D^*$ 의 a-ary 트리 형태로 전개하여 얻은 지역 해를 나타낸다. 지역 해  $D^*$ 는 앞서 기술된 B&B 알고리즘 특성 제 3항에 의거  $v_{D^*}(min) (= v_{subroot} + x_b \times D^*)$ 와  $v_{D^*}(max) (= v_{subroot} + x_c \times D^*)$  사이의 값으로 결정되며,  $v_{D^*}(min)$ 과  $v_{D^*}(max)$  사이에 존재하는 각 수치가 지역 해로 결정될 확률은 표준 분포 형태를 보이게 된다. 따라서 지역 해  $\Phi D^*$ 는



$$\Phi D^* = \sum_{k=v_D^*(min)}^{v_D^*(max)} k \times \phi_{D^*}(k) \quad (9)$$

와 같이 계산된다. 여기서  $k$ 는 지역 해로 가능한  $v_{D^*}(min)$ 과  $v_{D^*}(max)$  사이에 존재하는 각 지역 해 후보의 연산 결과치를, 그리고  $\phi_{D^*}(k)$ 는 지역 해가 임의 값  $k$ 로 결정될 확률을 나타낸다. 깊이  $D^*$ 의  $a$ -ary 트리 형태로 전개된 부 문제는 총  $aD^*$ 개의 지역 해 후보를 생성한다. 이들 가운데 연산 결과치  $k$ 를 갖는 임의의 지역 해 후보가 최종 지역 해로 결정될 확률,  $\phi_{D^*}(k)$ , 는

$$\begin{aligned} \phi_{D^*}(k) &= \sum_{j=1}^{D^*} P\{v_{D^*} \geq k+1\}^{j-1} \times P\{v_{D^*} \\ &= k\} \times P\{v_{D^*} \geq k\}^{(a^{D^*}-j)} \end{aligned} \quad (10)$$

이다. 식 (10)은 총  $aD^*$ 개의 지역 해 후보 중  $j$ 번째 지역 해는  $k$ 값을 가지며, 처음  $(j-1)$ 개 지역 해와  $j$ 번째 이후 나머지  $(aD^* - j)$ 개 지역 해는  $k$ 값 보다 크거나 같은 값을 가질 확률을 수식으로 나타내고 있다.

임의 그룹에 할당된 부 문제가 깊이  $(\omega + 1)$ 인 트리로 전개가 종료될 경우 그룹 내 각 프로세서는  $a$ 개의 중간 해를 생성하고, 이들을 비교하게 된다. 각 프로세서는  $a$ 개 중간 해 비교 결과, 최소값을 갖는 중간 해 하나를 선정하여 그룹 대표 프로세서에 전송한다. 따라서 회기적 기법으로 정리된 식 (8)의 근원  $AS[\omega + 1, a, P]$ 는  $P = a^\omega$ 가 된다. 그룹 대표 프로세서는 이들  $P$ 개 중간해 가운데 최종 해를 선정한다.

식 (8)의 이해를 돕기 위하여 그림 2는 임의의 프로세서 그룹이 B&B 알고리즘을 이용하여 검색하게 될 부 문제 트리 공간을 4부분으로 구분하여 보여주고 있다. 먼저 그림 2의 “가” 부분은 그룹 내  $P$ 개 프로세서가 전개한  $a^{(\omega+1)} (= P \times a)$ 개 부 문제들로 구성되며, 이들은 모두 그룹 대표 프로세서로 전송된다. 식 (8)의 첫 번째 덧셈 항은 그림 2의 “가” 부분에서 전송되는 부 문제의 수를 계산하고있다. 그룹 대표 프로세서는 전송 받은  $a^{(\omega+1)}$ 개 부 문제를 비교 분석하여 각 부 문제의 연산 결과치에 따라 두개 집합  $\alpha$ 와  $\beta$ 로 분류한다. 집합  $\beta$ 는  $a^{(\omega+1)}$ 개 부 문제 중 연산 결과치가 작은 순서대로  $a^\omega (= P)$ 개 부 문제를, 그리고 집합  $\alpha$ 는 나머지  $(a^{(\omega+1)} - P)$ 개 부 문제를 포함하게 된다. 그림

2의 “나” 부분과 “다” 부분은 집합  $\alpha$ 에 속한 부 문제들로부터 전개되는 하위 부 문제와 중간 해들을 구분하고있다. 식 (8)의 두 번째 덧셈 항은 그림 2의 “나” 부분에서 그룹 대표 프로세서로 전송되는 부 문제의 수를 계산한다. “나” 부분 임의의 깊이  $d$ 에 위치한 부 문제가 대표 프로세서로 소집되기 위한 조건은 해당 부 문제 연산 결과치가 부 문제 전개 임계치  $v_{parent}(lim)$  보다 작아야 하며, 이는 다음 두 가지 경우를 동시에 만족시킴으로 충족된다.

- i) 깊이  $(d - 1)$ 에 위치한 임의의 상위 부 문제의 연산 결과치( $v_{d-1} = y$ )가  $v_{parent}(lim)$  보다 작거나 같아야 한다. 즉,  $v_{d-1}(min) \leq y \leq v_{parent}(lim)$ .
- ii) i)의 조건을 만족하는 임의의 상위 부 문제로부터 전개된 하위 부 문제의 연산 결과치,  $v_d$ ,가  $v_{parent}(lim)$  보다 작거나 같아야 한다. 즉, 임의의 상위 부 문제와 이로부터 전개된 하위 부 문제간의 차이( $= v_d - y$ )의 범위가  $x_b \leq (v_d - v_{d-1}) \leq \min\{\Phi D^* - y, x_b\}$ 로 주어져야 한다.

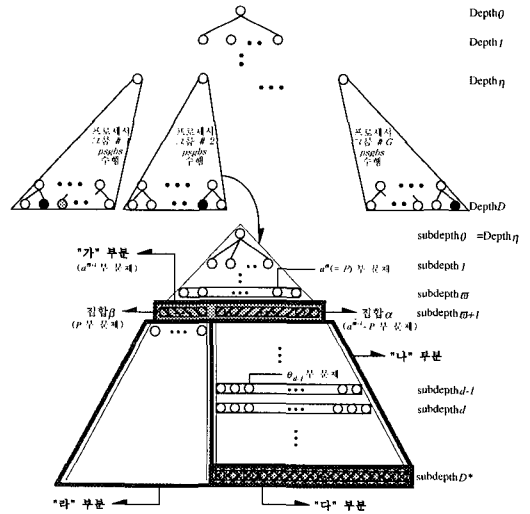


그림 2. 통신 소요시간 분석을 위한 부 문제 검색 공간 구분

Fig. 2. Decomposition of a subproblem search space to analyze the communication overhead.

따라서 식 (8)의 두 번째 덧셈 항은 그림 2의 “나” 부분 트리 깊이  $(\omega + 2)$ 와  $(D^* - 1)$ 사이에 위치한 부

문제를 대상으로 앞서 기술한 부 문제 소집 조건에 의거 각 부 문제별 소집 확률을 계산하고 있다. 먼저  $\sum_{d=\theta+2}^{R-1} \theta_{d-1}$ 은 각 깊이 별 전체 상위 부 문제 수를 나타내고,

$$\theta_{d-1} \times \sum_{y=v_{d-1}(\min)}^{v_{parent}(\lim)} (P^{v_{d-1}} = y) \times \sum_{j=1}^a \left[ \binom{a}{j} m^j (1-m)^{a-j} \times j \right]$$

는 깊이  $(d-1)$

에 위치한 임의 상위 부 문제의 연산 결과치  $(v_{d-1})$ 가  $v_{d-1}(\min) \leq y \leq v_{parent}(\lim)$ 을 만족시키는  $y$  값을 가질 확률을, 그리고 각각  $y$  값에 따라 이로부터 전개되는  $a$ 개 하위 부 문제의 연산 결과치가  $v_{parent}(\lim)$ 보다 작거나 같을 확률을 수식으로 표현한 것이다. 깊이  $(d-1)$ 에 위치하고 연산 결과치가  $v_{d-1}(\min) \leq y \leq v_{parent}(\lim)$ 을 만족시키는  $y$  값을 가지는 부 문제는  $a$ 개의 하위 부 문제를 생성하게 된다.  $a$ 개 하위 부 문제 가운데 연산 결과치가  $v_{parent}(\lim)$ 보다 작거나 같은  $j$  ( $0 \leq j \leq a$ )개 하위 부 문제는 향후 처리를 위하여 대표 프로세서로 소집된다.  $m$ 은 깊이  $d$ 에 위치한  $a$ 개 하위 부 문제 가운데 임의 하위 부 문제가 대표 프로세서로 소집될 확률, 즉 상 하위 부 문제간의 차이가  $x_b$ 에서  $\min\{\Phi D^* - y, x_c\}$  사이에 있을 확률을 의미한다. 여기서  $x_b, x_c$ 는 상 하위 부 문제간의 차이의 최소, 최대 허용치, 그리고  $(\Phi D^* - y)$ 는 상위 부 문제의 연산 결과치가  $y$ 값을 가질 때, 이로부터 전개된 하위 부 문제의 연산 결과치가  $v_{parent}(\lim)$ 보다 작거나 같은 값을 갖게 되기 위한 상 하위 부 문제간의 차이이다. 그림 2의 “다” 부분은 “나” 부분의 트리 깊이  $(D^* - 1)$ 에 위치한 부 문제로부터 전개되는 중간 해들로 구성된다. “다”부분에 속한 중간 해 부 문제들의 전개 여부는 “나” 부분에서의 경우와 동일하다. 그러나 “나” 부분에서와 달리 임의 프로세서에서 전개된  $a$ 개 중간 해 가운데 대표 프로세서로 전송되는 중간 해는 최대 한개가 된다. 식 (8)의 세 번째 덧셈 항은 그림 2의 “다” 부분에서 대표 프로세서로 전송되는 중간 해의 수를 나타내고있다. 그림 2의 “라” 부분은 “가”, “나”, “다”, 그리고 “라” 부분으로 형성된 초기 검색 공간보다 트리 깊이가 한 단계 감소한 검색 공간으로 볼 수 있다. 따라서

“라” 부분에서 대표 프로세서로 전송되는 부 문제의 수는 회기적 기법을 이용하여 식 (8)의 네 번째 덧셈 항과 같이  $AS[D^* - 1, a, P]$ 로 계산된다.

임의 프로세서 그룹 내부에서 대표 프로세서로 이동하는 전체 부 문제가 그룹 내 각 프로세서에서 동등하게 생성된다고 가정하면, 각 프로세서에서 전송되는 부 문제의 수는  $AS[D^*, a, P]/P$ 로 계산된다. 부 문제 전송 과정에서는 데이터 충돌 현상을 방지하기 위하여 단계별 소집 기법<sup>[15]</sup>을 사용한다. 단계별 소집 기법은 단계별 분배 기법의 역순으로 진행되어,  $P$ 개 프로세서에서 그룹 대표 프로세서로 각각 한 개씩의 부 문제 데이터 블록을 전송하고자 할 때  $\xi (= \lceil \log_2 P \rceil)$  단계의 데이터 전송을 수행한다. 이때  $\xi$ 번째 소집 단계에서 이동되는 데이터는 최대  $2^{\xi-1}$ 개 데이터 블록이 되고, 마지막  $\xi$ 번째 소집 단계에서는  $(P - 2^{\xi-1})$ 개 데이터 블록이 이동된다. 따라서 프로세서 그룹 별  $psgbs$  수행에 따른 부 문제 소집 과정에서 발생하는 프로세서 간 데이터 전송에 요구되는 시간,  $T_{aggr}$ , 은

$$T_{aggr} = \frac{AS[D^*, a, P]}{P} \times \left( \sum_{j=1}^{\xi-1} 2^{j-1} + (P - 2^{\xi-1}) \right) \times \sigma \times t_{rb}$$

(11)

와 같이 계산된다. 여기서  $\xi = \lceil \log_2 P \rceil$  이고,  $\sigma$ 와  $t_{rb}$ 는 식(6)에서 정의된 내용과 동일하다.

### III-2-2. 부 문제 분배

부 문제 소집 과정을 거쳐 그룹 내 각 프로세서로부터 새로 전개된 부 문제들이 그룹 대표 프로세서로 집결되면, 소집된 부 문제들을 검색하여 연산 결과치가 작은 순서대로  $P$ 개 부 문제를 선정하고, 선정된 부 문제들을 다시 그룹 내 프로세서에게 분배하게 된다. 이때 만일 전 단계에서 소집된 부 문제의 수가  $P$ 보다 적으면 지역 대표 프로세서는 CS 리스트에 저장된 부 문제 중에서 추가 선발하여 최대  $P$ 개의 부 문제를 분배하도록 한다. 앞서 부 문제 소집 과정의 분석에서와 같이 임의 그룹 소속  $P$ 개 프로세서가  $psgbs$ 를 수행하여 할당 받은 부 문제를 깊이  $D^*$ 의  $a$ -ary 트리 형태로 전개할 경우 그룹

대표 프로세서가 분배하여야 할 총 부 문제 수,  $DS[D^*, a, P]$ 는

$$DS[D^*, a, P] = a^\omega + \sum_{d=\omega+1}^{D^*-1} \theta_{d-1} \times \sum_{y=\nu}^{\nu_{parent}(lim)} \sum_{(min)}^{(lim)} (P^{v_{d-1}=y} \times \sum_{j=1}^a \left[ \binom{a}{j} m^j (1-m)^{a-j} \times j \right]) + DS[D^* - 1, a, P], \quad (12)$$

와 같이 얻어진다. 여기서  $D^* > \omega$ , 그리고  $DS[\omega + 1, a, P] = 0$  이다.  $\theta_{d-1}$ ,  $\nu_{parent}(lim)$ ,  $m$ , 그리고  $\nu_{d-1}(min)$  등은 식 (8)에서 정의된 것과 같다. 임의 그룹에 할당된 부 문제가 깊이 ( $\omega + 1$ )의  $a$ -ary 트리로 전개가 종료 될 경우 각 프로세서는 최종 해에 가장 근사한 중간 해 하나를 선정하여 그룹 대표 프로세서에 전송하게 되고, 그룹 대표 프로세서는 이들 중 하나를 최종 해로 결정한다. 따라서 더 이상의 부 문제 전개가 요구되지 않음으로, 이 경우 그룹 대표 프로세서가 분배하여야 할 총 부 문제 수  $DS[\omega + 1, a, P]$ 는 0이 된다.

$DS[D^*, a, P]$ 를 산출한 식 (12)는  $psgbs$  수행 과정에 그룹 대표 프로세서로 소집되는 총 부 문제 수를 계산한 식 (8)과 유사한 개념으로 그림 2를 이용하여 유도하였다. 먼저 그림 2의 “가” 부분에서는 앞서 기술한 대로 ( $a \times P$ )개의 부 문제가 그룹 대표 프로세서로 소집된다. 이들 중 집합  $\beta$ 에 속한  $P$ 개 부 문제는 그룹 내 각 프로세서로 분배되고, 나머지 집합  $\alpha$ 에 속한 ( $a^{\omega+1} - P$ )개 부 문제는 향후 처리를 위하여  $CS$  리스트에 저장된다. 그림 2의 “가” 부분의 집합  $\alpha$ 와 “나” 부분으로부터 분배되어야 할 총 부 문제 수는 식 (12)의 두 번째 덧셈 항과 같이 계산된다. 그림 2의 “다” 부분에서는 더 이상의 부 문제 전개가 요구되지 않아, 분배되는 부 문제가 없다. 또한 식 (12)의 마지막 덧셈 항은 그림 2의 “라” 부분에서 분배되는 부 문제 수를 계산한 것으로 회기적 기법으로 표현하였다.

단계별 분배 기법을 이용하여 그룹 대표 프로세서가  $DS[D^*, a, P]$ 개의 분배 대상 총 부 문제를  $P$ 개씩 구분하여 그룹 내 각 프로세서에 분배하는데 소요되는 시간,  $T_{dist}$ ,는

$$T_{dist} = \frac{DS[D^*, a, P]}{P} \times \sum_{j=1}^{\xi} 2^{\xi-j} \times \sigma \times t_{rb} \quad (13)$$

과 같이 계산된다. 여기서  $\xi = \lceil \log_2 P \rceil$  이고,  $\sigma$ 와  $t_{rb}$ 는 식(6)에서 정의된 내용과 동일하다. 그룹 대표 프로세서가  $P$ 개 부 문제를 단계별 분배 기법으로 분산시키기 위하여  $\xi$  단계의 데이터 전송이 요구되며,  $j$ 번째 단계에서 전송되는 부 문제 데이터 크기는  $2^{\xi-j}$  블록이 된다.

### III-2-3. 기타 통신 소요시간

제안한  $pobs$ 의 제 2단계(그룹 별  $psgbs$ ) 수행에 요구되는 데이터 이동 내용 가운데 앞서 기술한 과정 2(부 문제 소집)와 과정 6(부 문제 분배)을 제외한 과정 1(전개 종료 시험), 3( $CS$  리스트 정리), 4(부 문제 선정), 및 5( $CS$  리스트 정리II)는 각 해당 프로세싱 노드 내부에서 처리된다. 이와 같은 데이터 이동 현상은 병렬 시스템 환경이 아닌 단일 프로세서 환경에서 B&B 알고리즘을 수행할 경우에도 공히 요구되는 과정이다. 다만 병렬 처리 환경에서 과정 1은  $P$ 개 프로세서가 각자 전개한 부 문제를 대상으로 검색을 시행함으로 단일 프로세서 환경에 비교하면 하나의 프로세서가 검색하게 될 부 문제 수가 선형 감소하여 소요시간이 줄어드나, 나머지 과정 3, 4, 5는 그룹 대표 프로세서가 단독으로 처리하게 됨으로 이에 소요되는 시간이 단일 프로세서 환경에서와 같다. 따라서 본 연구에서는 이들 네 과정에서 소요되는 데이터 이동 시간을 단일 프로세서 환경에서  $a$ -ary 트리 깊이  $D^*$ 로 전개된 문제 공간을 Best-First search B&B 알고리즘으로 처리할 때 소요되는 통신 시간  $TC(D^*, a, 1)_{sb\&b}$ <sup>[15]</sup>로 근사치 계산하였다. 이들 네가지 데이터 이동 과정에서 소모되는 통신 시간은 과정 1과 6에서 요구되는 원거리 통신 시간과 비교할 때 전체 데이터 이동 시간에 미치는 영향이 미세하여 앞서 기술한 근사치 계산법이 실제 통신 시간의 오차 범위 내에서 무시해도 무방할 만큼 매우 제한된 영향을 제공하였다. 따라서  $pobs$ 의 제 2단계(그룹 별  $psgbs$ ) 수행에 소요되는 통신 시간,  $TC(D^*, a, P)_{phase2}$ ,는

$$TC(D^*, a, P)_{phase2} = \omega \times \sum_{j=1}^{\xi} r[j] \times \sigma \times t_{rb} + T_{aggr} + T_{dist} + TC(D^*, a, 1)_{sb\&b} \quad (14)$$

와 같이 계산된다.

제안된 알고리즘 제 3단계의 통신 활동은 세가지 다른 형태의 프로세서간 데이터 전송: 지역 해 전파, 그룹별 작업 종료 전파, 그리고 최종 해 데이터 전송으로 구분된다. 이들 가운데 지역 해 전파 및 그룹별 작업 종료 전파는 각각 한번의 원거리 쓰기 시도가 요구된다. 그룹 대표 프로세서는  $psgs$  수행 결과 얻은 지역 해를 타 그룹이 산출한 지역 해와 비교 평가하고, 나아가서 그룹 별 작업 불균형 현상을 완화하기 위한 수단으로 지역 해 전파를 수행한다. 이때 만일 새로 얻은 지역 해가 타 그룹으로부터 전파된 기존의 지역 해보다 큰 연산 결과치를 가질 경우 해당 그룹 대표 프로세서는 지역 해 전파 대신 그룹 작업 종료 방송을 수행한다. 따라서 모든 그룹 대표 프로세서는 할당된 부 문제의 지역 해를 얻는 즉시 지역 해 전파 혹은 작업 종료 전파를 선택 시행하게 되는데, 이에 소요되는 시간은  $G \times t_{rw}$ 이 된다. 여기서  $t_{rw}$ 는 원거리 쓰기에 소요되는 시간이다. 시스템의 모든 그룹들이 해당 그룹 지역 해 전파를 통하여 최종 해가 선정되면, 이를 시스템 대표 프로세서로 전송하게 된다. 최종 해 데이터 전송 과정에는 최종 해와 함께, 최종 해 관련 데이터 블록이 함께 전송되며, 소요시간은  $\rho \times t_{rb}$ 로 계산된다. 따라서 제 3단계 통신 활동에 소요되는 시간,  $TC(D, a, N)_{phase3}$ , 은

$$TC(D, a, N)_{phase3} = G \times t_{rw} + \rho \times t_{rb} \quad (15)$$

과 같이 계산된다.

제안한 알고리즘의 통신에 소요되는 시간,  $TC(D, a, N)_{pobs}$ ,는 식 (6), (14), 그리고 (15)의 결과를 합하여

$$\begin{aligned} TC(D, a, N)_{pobs} &= \eta \times \sum_{j=1}^{\theta} \gamma[j] \times \sigma \times t_{rb} \\ &+ \omega \times \sum_{j=1}^{\theta} \gamma[j] \times \sigma \times t_{rb} + T_{agg} \\ &+ T_{dist} + TC(D^*, a, 1)_{sb\&b} \\ &+ G \times t_{rw} + \rho \times t_{rb} \end{aligned} \quad (16)$$

과 같이 계산된다. 여기서  $\gamma[0] = a$ , 그리고  $\gamma[\theta] = 1$  이다.

$N$ 개 프로세서를 가진 다층 연결 구조 시스템이 제안한  $pobs$  알고리즘을 이용하여 주어진 문제를 깊이  $D$ 의  $a$ -ary 트리 형태로 전개하여 처리하였을 경우 소요되는 총 시간,  $T(D, a, N)_{pobs}$ ,은

$$T(D, a, N)_{pobs} = TE(D, a, N)_{pobs} + TC(D, a, N)_{pobs} \quad (17)$$

과 같이 식 (5)의 부 문제 연산 시간,  $TE(D, a, N)_{pobs}$ , 과 식 (16)의 통신 소요시간,  $TC(D, a, N)_{pobs}$ ,을 합하여 구할 수 있다. 따라서 병렬 처리에 따른 처리 속도 이득 율,  $S(N)_{pobs}$ , 는

$$S(N)_{pobs} = \frac{T(D, a, 1)_{sb\&b}}{T(D, a, N)_{pobs}} \quad (18)$$

이다. 여기서  $T(D, a, 1)_{sb\&b}$ <sup>[15]</sup>는 단일 프로세서 시스템이 B&B 알고리즘을 이용하여 주어진 문제를 처리하며 깊이  $D$ 의  $a$ -ary 트리 형태로 문제를 전개하였을 경우 소요되는 시간을 나타낸다.

표 1. 시뮬레이션 결과와 수학적 분석 결과의 비교

Table 1. Comparison of data obtained via simulation and analysis( $x_b=1, x_c=10, D=15, P=4, \sigma=1.0, t_e=500 \mu sec, t_r=0.53 \mu sec, t_w=0.39sec, t_b=72 \mu sec, t_{rw}=4 \mu sec$ ).

프로세서 개수	전개된 부 문제 수		통신 소요 시간 $\mu$ ( $\mu sec$ )		총 소요 시간(msec)	
	simulation	analysis	simulation	analysis	simulation	analysis
1	1375.14	1359.29	2192.5	2063.8	689.762	681.709
2	666.77	679.64	46341.2	42752.1	379.726	382.572
4	329.40	339.82	69992.4	64598.0	234.602	234.508
8	229.73	237.38	48322.9	45020.7	163.188	163.711
16	161.85	169.82	34361.5	32208.8	115.287	117.119
32	114.47	119.16	24237.2	22566.6	81.471	82.147
64	81.70	86.31	16965.9	16270.1	57.816	59.425
128	59.23	61.86	11791.0	11544.3	41.406	42.474
256	43.98	46.19	8363.3	8465.7	30.378	31.561

표 1은 제안한 알고리즘을 다양한 규모의 MIN-based 시스템( $N = 1, 2, 4, 8, 16, 32, 64, 128, 256$ )에 적용하여 깊이 15의 2진 트리 형태로 전개되는 문제를 처리할 때 소요되는 부 문제 연산 시간, 통신 시간, 및 총 소요

시간에 대한 시뮬레이션 결과 치의 평균과 본 절에 기술된 수학적 분석 결과를 비교하여 보여주고 있다. 각 프로세서 그룹은 네개의 프로세서( $P = 4$ )로 구성되도록 하였고, 분석 모델에 사용된  $t_{ir}$ ,  $t_{iw}$ ,  $t_{rw}$ , 및  $t_{rb}$  등 각종 파라미터들은 BBN Butterfly GP1000<sup>[16]</sup> 자료를 근거로 설정하였다. 본 연구에서 제시한 수학적 분석 모형의 실효성을 입증하기 위하여 수행된 시뮬레이션은 3000개의 상이한 문제를 대상으로 알고리즘을 적용하고 결과를 얻는 과정에서 전개되는 부 문제 수, 부 문제 연산 시간, 및 통신 소요 시간 등 각 파라미터 별 평균 값을 사용하였다. 시뮬레이션 결과와 수학적 분석 결과는 전반적으로 약 7% 이내의 오차를 보여, 본 연구에서 제시한 수학적 분석 모형의 우수성을 입증하였다.

#### IV. 병렬 Best-First Search B&B 알고리즘 비교

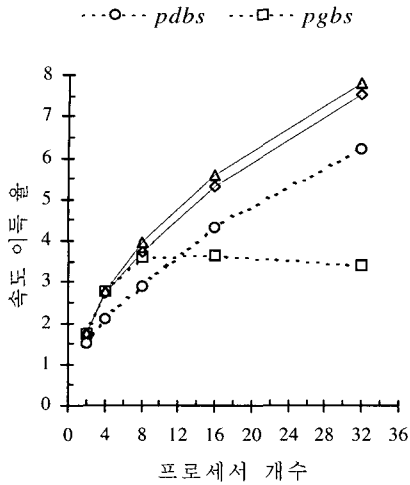
B&B 알고리즘은 전개 대상 부 문제 선택 기법에 따라 Breadth-First search, Depth-First search, Best-First search, 그리고 Random search 등으로 구분되어 발표되었다. 이들 다양한 선택 기법에 따라 제안된 알고리즘 가운데 Best-First search 기법이 최적 검색 경로를 우선적으로 추적할 확률이 높아, 성능면에서 다른 방식과 비교하여 우수한 것으로 밝혀졌다. Best-First search 기법을 기조로 설계된 병렬 Best-First Search B&B 알고리즘은 기존 발표된 병렬 Global Best-First search B&B 알고리즘( $pgbs$ )<sup>[7]</sup>과 병렬 Decomposite Best-First search B&B 알고리즘( $pdb$ s)<sup>[15]</sup>, 그리고 본 연구에서 제안한 병렬 Optimal Best-First search B&B 알고리즘( $pobs$ ) 등으로 구분된다. 이들 알고리즘들은 다중 프로세서들을 대상으로 각각 독특한 작업 분배 방식을 가지고있어, 시스템 환경에 따라 성능 면에서 큰 차이를 보이고있다. 본 절에서는 MIN을 기반으로 하는 시스템에, 각 알고리즘을 적용하여 주어진 문제를 처리할 때 얻어지는 속도 이득 율을 시뮬레이션 결과를 통하여 비교하였다. 그림 3은 BBN Butterfly 컴퓨터 환경에서 앞서 나열한 세가지 병렬 알고리즘들의 처리속도 이득 율을 보여주고 있다. 그림 3에 제시된 각 알고리즘 별 시뮬레이션 데이터는 III절 및 참고문헌 [15]에 기술된 수학적 분석 모델로부터 산출한 데이터와 미세한 오차 범위 내에서 일치하

는 결과를 보였다.

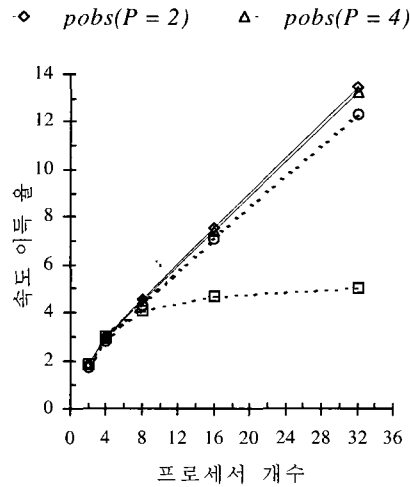
$pgbs$  기법은 시스템 전체에 오직 하나의 CS 리스트만을 유지하고, 모든 부 문제 분배 및 소집이 CS 리스트를 보유한 전체 대표 프로세서에 의하여 일괄 조정되므로 프로세서간 작업 분배가 고르게 이루어지는 장점이 있다. 따라서 부 문제 연산 소요시간 만을 고려하면  $pgbs$ 의 처리 속도 이득 율은 프로세서 개수의 증가에 따라 선형 증가하고, 경우에 따라 처리 속도 과도 증가 현상<sup>[17]</sup>을 보이기도 한다. 그러나 전체 대표 프로세서로 집중된 부 문제 분배 및 소집 조정 기능은 지속적인 프로세서간 원 거리 통신 활동을 불가피하게 하고, 이로 인하여 통신 소요시간이 실제 문제 연산에 소요되는 시간을 능가하는 결과를 초래하기도 한다. 특히 MIN을 기반으로 한 Butterfly 컴퓨터의 경우 시스템 규모가 커질수록 부 문제 분배 및 소집에 과중한 통신 시간이 소요되어, 결국 그림 3에 보인 바와 같이 전체 알고리즘 성능을 저하시킨다.

주어진 문제를 프로세서 개수만큼의 부 문제들로 분할하여 분배하고, 각 프로세서는 각자의 CS 리스트를 가지고 할당 받은 부 문제의 해를 구하는  $pdb$ s 기법은 시스템 전체에 총  $N$ 개의 CS 리스트를 분산 유지한다. 이와 같은 분산 처리 기법은 각 프로세서 별 문제 해결에 필요한 대부분의 통신 활동을 프로세서 내부로 제한하여,  $pgbs$  기법에서 문제로 지적된 프로세서간 원거리 통신 활동으로 인한 성능 저하를 최소화한다.  $pdb$ s 기법은 Best-First search B&B 알고리즘의 장점과 함께 통신 소요시간을 최소화하여 시스템 크기의 변화와 무관하게 처리 속도 이득 율 증가를 꾀할 수 있다. 그러나 프로세서간 통신 활동의 과도한 제한으로 인하여 작업 분배의 불균형 현상과 함께, 프로세서 별 문제 해결 과정에서 전개되는 부 문제 수가  $pgbs$  기법과 비교하여 증가하는 것으로 관찰되었다. 따라서 그림 3에 보인 바와 같이 원 거리 통신 소요 시간이 증가하는 대규모 시스템의 경우  $pdb$ s 기법은 비교적 안정된 처리 속도 이득 율 증가와 함께 성능 향상을 도모할 수 있는 반면, 소규모 시스템의 경우 각 프로세서가 처리하는 부 문제 연산 시간이 전체처리 시간의 주된 요소로 작용하여  $pgbs$  보다 성능 면에서 열세를 보이게 된다.

본 연구에서 제안된  $pobs$ 는 위의  $pgbs$ 와  $pdb$ s의 단점을 보완하고 장점을 조화시킨 병렬 알고리즘으로, 먼저 프로세서들을 소 그룹으로 나누어 프로세서간 원거리 통신 범위를 각 그룹 내부로 제한하고, 각 그룹마다 대표



(a).  $x_b = 1, x_c = 10, D = 12, a = 2, t_e = 200 \mu sec$



(b)  $x_b = 5, x_c = 14, D = 15, a = 2, t_e = 500 \mu sec$

그림 3. BBN Butterfly 컴퓨터에 적용된 병렬 Best-First search B&B 알고리즘들의 속도 이득을 비교

Fig. 3. Speed-up comparisons of the parallel Best-First search B&B algorithms under BBN Butterfly computer.

프로세서가 CS 리스트를 유지하며 부 문제 소집 및 분배를 제어하여 그룹 내부 프로세서 간의 작업 불균형 문제를 해결한다. 또한 각 프로세서 그룹이 Best-First search 기법으로 할당된 부 문제를 처리하여 그룹 지역 해 산출 과정에서 전개되는 부 문제의 수를 줄이는 한편, 알고리즘을 운용하는 전체 시스템의 CS 리스트를 그룹 별로 분산시켜 pgb에서 야기되었던 단일 CS 리스트 병목현상을 해소하였다. MIN을 기반으로 하는 시스템 환경에서 pobs 알고리즘은 전개하는 부 문제 수를 줄이고, 알고리즘 수행 과정에서 요구되는 프로세서간 통신 범위를 대부분 그룹 내부로 제한하여 앞서 발표한 알고리즘들과 비교하여 보다 나은 성능을 제공한다. 그림 3은 제안한 알고리즘이 기존의 두 가지 알고리즘과 비교하여 다양한 규모의 시스템 환경에 보다 유연하게 적용하며 우수한 성능을 제공함을 보여주고 있다.

### V. 결 론

본 연구에서는 새로운 병렬 Best-First Search B&B 알고리즘을 설계 제안하고, 다층 연결 구조를 기반으로 하는 병렬 컴퓨터를 대상으로 성능을 분석하였다. 제안한 알고리즘(pobs)은 이미 발표된 pgb와 pobs의 장단점을 상호 보완하고 조화시켜 설계한 알고리즘으로, 먼저 주어진 문제를 소수 부 문제로 분할하고 이를 각 프로세서 그룹에 할당하여 그룹 별 지역 해를 연산하도록 하였다. 각 프로세서 그룹에서 얻은 지역 해는 다층 연결 구조 네트워크를 통하여 타 그룹에 전파하고, 이를 전송받은 각 그룹 대표 프로세서는 기존의 중간 해와 비교하여 최종 전역 해 선정에 참조하는 한편, 그룹내 전개 대기 부 문제(CS) 리스트 정리에 활용하여 프로세서 그룹간의 작업 불균형 문제를 상당 부분 해소하도록 하였다. 알고리즘의 성능 평가를 위하여 B&B 알고리즘의 확률적 특성을 근거로 제시된 분석 모형은 알고리즘 수행에 따른 연산 소요시간과 통신 소요시간을 분리하여 처리하므로 다양한 병렬 시스템 환경에서 보다 실질적인 성능 평가를 가능하게 하였다. 제안한 분석 모형에 의한 성능 예측 기법은 과거 B&B 알고리즘 성능 측정 시 실제 알고리즘 작동 결과 혹은 시뮬레이션 데이터에 의존하던 방식을 탈피하여, 모형이 요구하는 각종 파라미터에 의한 수학적 연산으로 단 시간에 성능을 예측할 수

있도록 하였다. 또한, 다층 연결 구조를 기반으로 하는 시스템에 제안한 알고리즘을 적용, 3000개의 상이한 문제를 처리하여 얻은 진행된 시뮬레이션 결과는 성능 분석 모형의 예측 데이터와 상호 7% 이내의 오차 범위 내에서 일치하는 결과를 보여, 제시한 분석 기법의 타당성을 입증하였다. 본 논문에서 제안한 병렬 최적 Best-First search Branch-and-Bound 알고리즘(pobs)은 기존의 알고리즘의 성능과 비교 평가 결과, 프로세서 간 효율적인 작업 분배와 함께 문제 해결 과정에서 전개되는 부 문제 수를 줄이고, 또한 통신 활동의 범위를 국지적으로 제한하여 전반적으로 우수한 성능을 제공하는 것으로 입증되었다.

#### 참 고 문 헌

- [1] E. L. Lawler and D. W. Wood, "Branch-and-Bound methods: A survey", *Oper. Res.*, Vol. 14, pp. 699-719, 1966.
- [2] E. Horowitz and S. Sahni, "Fundamentals of Computer Algorithms", *Computer Science Press*, 1978.
- [3] A. V. Aho, J. E. Hopcraft, and J. D. Ullman, "Data Structures and Algorithms", *Addison-Wesley*, Apr. 1987.
- [4] M. J. Quinn, "Parallel Computing", Theory and Practice, McGraw Hill, 1994.
- [5] D. R. Smith, "Random Trees and the Analysis of Branch-and-Bound procedures", *Journal of the ACM*, 31, No 1, pp. 163-188, Jan. 1984.
- [6] B. W. Wah and C. F. Yu, "Stochastic Modeling of Branch-and-Bound Algorithms with Best-First Search", *IEEE Trans. on Software Eng.*, Vol. SE-11, No. 9, pp. 922-934, Sep. 1985.
- [7] B. W. Wah and Y. W. Eva Ma, "MANIP - A Multicomputer Architecture for solving Combinatorial Extremum-Search Problems", *IEEE Trans. on Computers*, Vol. C-33, No. 5, pp. 377-390, May 1984.
- [8] G. -J. Li and B. W. Wah, "Computational Efficiency of Parallel Approximate Branch-and-Bound Algorithm", *Proc. Intl. Conf. on Parallel Processing*, pp. 473-480, Aug. 1984.
- [9] V. K. Janakiram, D. P. Agrawal, and R. Mehrotra, "A Randomized Parallel Branch-and-Bound Algorithm", *Proc. Intl. Conf. on Parallel Processing*, pp. 69-75, Aug. 1988.
- [10] R. M. Karp and Y. Zhang, "A Randomized Parallel Branch-and-Bound Procedure", *Proc. of the ACM Symposium on Theory of Computing*, pp. 290-300, May, 1988.
- [11] V. N. Rao and V. Kumar, "Parallel Depth-First Search on Multiprocessors Part I: Implementation, and Part II: Analysis", *Intl. Journal of Parallel Programming*, Vol. 16, No. 6, 1987.
- [12] V. Kumar and V. N. Rao, "Parallel Depth-First Search Ring Architecture", *Proc. Intl. Conf. on Parallel Processing*, pp. 128-132, Aug. 1988.
- [13] S. Arvindam, V. Kumar and V. N. Rao, "Floorplan Optimization on Multiprocessors", *IEEE Intl. Conf. Symp. on Computer Design*, pp. 109-114, Oct. 1989.
- [14] M. K. Yang and C. R. Das, "Analytical Modeling of a Parallel Branch-and-Bound Algorithm on MIN-Based Multiprocessors", *Proc. Intl. Parallel Processing Symposium*, pp. 254-257, Mar. 1992.
- [15] M. K. Yang and C. R. Das, "Evaluation of a Parallel Branch-and-Bound Algorithm on a Class of Multiprocessors", *IEEE Trans. on Parallel and Distributed Systems*, Vol. 5, No. 1, pp. 74- 86, Jan. 1994.
- [16] "Butterfly GP1000 - Overview", *BBN Advanced Computer Inc.*, Nov. 1988.
- [17] T. -H. Lai and S. Sahni, "Anomalies of Parallel Branch-and-Bound Algorithms", *Comm. of the ACM*, pp. 594-602, June 1984.

---

저 자 소 개

---



梁明國 (正會員)

1957년 1월 4일생. 1983년 한양대학교 전자공학과 졸업. 1978년-1982년 육군 기술장교. 1983년-1984년 (주)금성사 전자 개발실 연구원. 1987년 Fairleigh Dickinson Univ. 전기공학과 졸업(석사). 1992년 The Pennsylvania State Univ. 전기 및 컴퓨터공학과 졸업(공학박). 1992년-1993년 현대전자산업(주) Software 연구소 책임연구원. 1993년-현재 울산대학교 공과대학 전기 전자 및 자동화공학부 조교수.