



유리재단 문제에 대한 분산 합성 알고리즘

홍철의

상명대학교 컴퓨터학과

A Distributed Hybrid Algorithm for Glass Cutting

Chuleui Hong

Department of Computer Science, Sangmyung University, Seoul 03016, Korea

[요 약]

본 논문에서는 유리재단 문제에 평균장 어닐링과 시뮬레이션된 어닐링 형태의 유전자 알고리즘을 결합한 합성 알고리즘을 분산 처리하여 적용한다. 유리재단 문제는 2차원 2진 패킹 문제로 주어진 원판에 요구되는 사각형 모양의 패턴들을 버려지는 부분이 최소가 되게 배치하는 조합 최적화 문제이다. 제안된 합성 알고리즘은 유전자 알고리즘의 다양한 연산자에 시뮬레이션된 어닐링의 온도개념을 추가하여 평균장 알고리즘에 의한 빠른 평형상태 도달을 유지하게 하였다. MPI를 이용한 분산 합성 알고리즘을 유리재단 문제에 적용하여 실험한 결과 기존의 평균장 어닐링 또는 유전자 알고리즘을 단독으로 사용하였을 때보다 최적의 배치 상태를 나타내었으며 최적해 접근 특성을 유지하면서 문제의 크기에 대하여 선형적인 수행시간 단축을 보여 주었다.

[Abstract]

The proposed hybrid algorithm combines the benefits of rapid convergence property of mean filed annealing(MFA) and the effective genetic operations of simulated annealing-like genetic algorithm(SGA). This algorithm is applied to the isotropic material stock cutting problem, especially to glass cutting in distributed computing environments base on MPI called message passing interface. The glass cutting is to place the required rectangular patterns to the given large glass sheets resulting in reducing the wasted scrap area. Our experimental results show that the heuristic method improves the performance over the conventional ones by decreasing the scrap area and maximum execution time. It is also proved that the proposed distributed algorithm maintains the convergence properties of sequential one while it achieves almost linear speedup as the problem size increases.

색인어 : 분산처리, 시뮬레이션된 어닐링, 유리재단, 유전자 알고리즘, 평균장 어닐링

Key word : Distributed Processing, Genetic Algorithms, Glass Cutting, Mean Field Annealing, Simulated Annealing

<http://dx.doi.org/10.9728/dcs.2018.19.2.343>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 19 January 2018; **Revised** 04 February 2018

Accepted 26 February 2018

***Corresponding Author; Chuleui Hong**

Tel: +82-2-2287-5313

E-mail: hongch@smu.ac.kr

I. 서론

본 논문에서 사용하는 분산 평균장 유전자 알고리즘(DMGA; Distributed Mean field Genetic Algorithm)은 평균장 어닐링(MFA, Mean Field Annealing)[1]과 유전자 알고리즘(GA; Genetic Algorithm)[2]의 합성 알고리즘이다. MFA는 시스템의 상태를 평균장 근사법(mean-field approximation)으로 표현하여 시뮬레이션된 어닐링(SA; Simulated Annealing)[3]에 비하여 평형 상태에 빠르게 도달하나 최종해의 결과가 다른 경험적 알고리즘에 비하여 좋지 못하다. 따라서 본 연구에서는 다양하고 강력한 상태변환 연산자를 가지고 있는 유전자 알고리즘을 MFA와 결합한 평균장 유전자 알고리즘(MGA)을 사용한다.[4]

유전자 알고리즘은 어닐링 알고리즘에 적용하는 온도 개념을 가지고 있지 않다. 따라서 특정 온도에서의 평균장 어닐링으로부터 구해진 평형상태가 유전자 알고리즘에서는 유지되지 못한다. 그러므로 MFA에서 도달한 평형상태가 GA에서도 유지되도록 선택, 교차 및 돌연변이와 같은 상태변환 연산에서 SA에서 사용하는 Metropolis Criteria에 의해서 새로이 생성된 상태를 선택하게 하였다. SA와 GA를 결합한 알고리즘을 SGA(Simulated annealing-like Genetic Algorithm)라고 정의한다.[5]

평균장 유전자 알고리즘은 경험적 합성 알고리즘으로 수행시간이 기존의 개별 알고리즘에 비하여 오래 걸린다. 수행시간을 단축하기 위하여 MGA를 MPI(Message Passing Interface) 환경 하에서 분산처리를 수행한다. 분산처리는 우선 MFA를 수행하여 평형상태에 도달한 결과를 SGA를 통하여 최적해에 보다 빠르게 접근하게 한다.

제안된 알고리즘은 일반적으로 모든 최적화 문제에 적용할 수 있으나 본 연구에서는 최적화 문제의 일종인 유리재단(glass cutting) 문제에 적용하였다.[6]-[8] 유리재단 문제는 조합 최적화 문제 중 하나로 사각형의 여러 가지 패턴을 2차원의 원판에 버려지는 부분이 최소가 되게 배치한 후 절단하는 문제이다. 본 문제는 일종의 무 방향성 2차원 2진 패킹 문제로 NP군으로 분류된다. 따라서 본 문제를 해결할 다항식 알고리즘은 존재하지 않는다. 본 논문이 제안하는 합성 알고리즘을 사용하여 항상 최적 값은 아니더라도 사용 가능한 배치를 찾는다.

II. 유리재단 문제

균등한 재질의 2차원 재료재단 문제는 요구되는 사각형의 개수 및 이를 수용하는 사각형 모양의 원판 개수가 주어졌을 때, 원판 사각형으로부터 버려지는 부분이 최소가 되도록 요구되는 모든 사각형을 배치시키는 문제이다. 절단은 원판을 횡단하는 길로틴(guillotine) 절단만이 허용되며, 재질은 방향성이 없으므로 요구되는 사각형은 90도의 회전이 가능하다.

평판 유리는 재질이 균등하여 회전 시 재질의 방향성이 존재

하지 않으며 사각형 모양은 90도 회전 시 변화한다. 또한, 유리 특성상 절단은 원판의 한 면에서 마주보는 변으로 직선으로 이루어지는 길로틴 절단만이 가능하다. 그림1은 균등 재질의 재료 재단의 배치를 보여주며, 번호는 절단되는 순서를 나타낸다. 그림에서 보여주는 모든 절단은 길로틴 절단으로 구성되어져 있다.

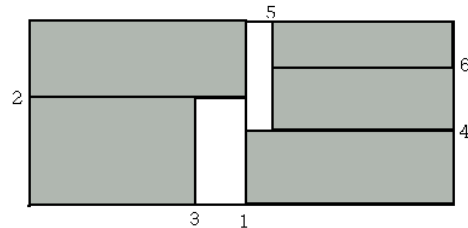


그림 1. 배치의 예 및 절단순서

Fig. 1. The Example of placements of patterns and their sequence of cutting

유리 재료의 재단 문제는 무 방향성 2차원 2진 패킹 문제로 NP-Hard 문제이다. 따라서 본 논문에서는 경험적 합성 알고리즘을 이용하여 항상 최적 배치는 아니더라도 현실적으로 받아들일 수 있는 근사 최적 배치를 찾는다.

유리재단 문제의 순서를 살펴보면, 합성 알고리즘을 적용하기 전에, 각 사각형(pattern)에 대하여 모양 함수(shape function)를 정의한 후, 두 개의 사각형을 결합하여 큰 사각형을 얻는다. 이러한 작업을 반복하여 주어진 원판에 최적의 배치를 찾아낸다. 두 개의 사각형이 결합하여 또 다른 큰 사각형을 생성한다. 이 사각형을 외부 사각형(bounding rectangle)이라 부르기로 한다. 외부 사각형은 내부의 두 사각형이 결합하는 위치에 따라서 모양이 변한다. 즉, 외부 사각형으로부터 내부의 두 사각형을 분리하기 위하여 수평 또는 수직으로 절단하는 두 가지 방법이 존재하며 각각에 대하여 4가지 결합 방법이 존재하므로 모두 8가지의 서로 다른 외부 사각형 모양이 존재한다.

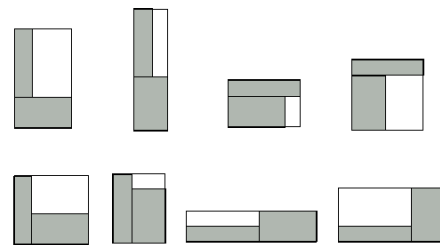


그림 2. 8가지의 외부 사각형 모양

Fig. 2. The 8 shapes of a bounding rectangles

외부 사각형은 항상 2개의 내부 사각형으로 구성되므로 길로틴 절단을 만족한다. 또한 외부 사각형을 부모 노드로 정의하였을 때, 포함되어 있는 내부 사각형은 외부 사각형의 두 개의 자식 노드로 나타내어진다. 다시 외부 사각형은 일반 사각형과 똑 같이 취급되어 두 개의 사각형을 결합하여 또 다른 외부 사

각형을 만든다. 이러한 과정을 주어진 원판(sheet)의 크기를 벗어나지 않는 한도 내에서 반복 수행하여 모든 가능한 사각형 모양을 만든다.

그러나 실제 유리 공장에서 요구되는 사각형 모양(patterns)은 다양하며 많게는 수십에서 수백 종류에 이르고 원판 모양(sheets)의 종류는 보통 많아야 3-5개이며 대부분 1개로 고정된다. 따라서 하나의 원판에 포함되는 모든 사각형의 경우의 수는 폭발적으로 많아 질수 있으므로 사각형의 개수를 효율적으로 조절하기 위해서 밀도가 너무 떨어지는 외부 사각형은 고려 대상에서 제외한다. 만들어진 모든 외부 사각형을 밀도 순으로 정렬한 후 제안하는 합성 알고리즘을 적용하여 사용되는 원판의 면적을 줄이도록 요구되는 모든 사각형 패턴을 원판에 최적으로 배치하여 생산 원가를 절감하게 한다.

경험적으로 면적이 큰 패턴을 원판에 우선 배치하는 경우 최종적으로 밀도가 증가하기 때문에 먼저 패턴을 면적이 큰 순으로 정렬한다. 다음 면적이 큰 패턴이 우선적으로 원판에 배치시키기 위하여 각 패턴의 면적의 1.5배를 하여 계산된 값을 value라 부르기로 한다. value를 면적의 1.5배로 하였을 때 실험적으로 가장 좋은 결과를 얻었다. 주어진 패턴을 임의의 원판에 최대로 배치하는 경우는 너무 많을 수 있으므로 각 원판에 배치된 패턴의 value의 합을 원판의 면적으로 나눈 값을 value-밀도라 하고 이를 큰 순(non-decreasing)으로 정렬하여 번호를 부여한다. 실험적으로 최대 10분 이내에 최적의 해를 찾기 위해서는 각 패턴당 128개의 배치가 완료된 원판 모양이 적정하다. 따라서 배치가 완료된 원판이 유전자 알고리즘의 각 유전자에 해당하며 주어진 모든 패턴이 배치된 서로 다른 원판의 조합이 개체(individual)이며 패턴의 면적의 합을 사용된 원판의 면적의 합으로 나눈 밀도가 가장 큰 개체가 찾고자 하는 최적해이다.

III. 분산 합성 알고리즘

분산 합성 알고리즘(DMGA)은 유전자 알고리즘의 강력하고 다양한 상태변환 연산자를 통하여 전역 최적해 상태를 구하는 장점과 평균장 어닐링의 최적해에 빠르게 접근하는 특성을 각 온도에 대하여 평형상태를 유지하도록 결합한 알고리즘이다.

유리재단 문제에 적용하기 위하여 먼저 요구되는 사각형 패턴을 사용 가능한 원판에 버려지는 부분이 최소가 되도록 배치한다. 배치가 완료된 원판의 경우의 수가 아주 많을 수 있으므로 각 패턴당 value-밀도가 높은 순으로 128개만 선정한다. 패턴의 순서는 패턴 면적이 큰 순서이다. 패턴 순서대로 합성 알고리즘을 적용하여 모든 패턴이 배치되는 원판의 조합인 최적해를 찾는다.

MFA에서 스핀 행렬(spin matrix)은 N 개의 유전자(gene, 배치가 완료된 원판)행과 K 개의 개체로 구성된 개체집단 (individual population)의 열로 구성되며 유전자-개체 할당 상태

를 표현한다. 다시 말하면 열에 해당하는 개체가 가능한 해를 표현한다. 즉, N×K 스핀행렬은 가능한 해의 상태를 확률적으로 표현한다고 정의할 수 있다.

스핀행렬의 행의 순서는 value-밀도가 큰 순이다. 스핀행렬의 각 열은 유전자 알고리즘에서의 개체를 나타내며 개체집단의 크기는 본 실험에서는 128로 정하였다. 스핀행렬의 원소는 각각의 스핀 값을 나타내며, 따라서 s_{ip} 는 스핀(i, p)로 유전자 i를 개체 p에 할당할 확률을 나타낸다. 여기서 s는 $0 \leq s \leq 1$ 의 연속 변수이다. MFA 알고리즘에서 최종 최적해에 도달하면 스핀 값은 1 또는 0으로 접근한다. s가 1로 수렴되면 유전자 i는 개체 p에만 할당되는 것을 의미한다. 표 1의 예는 배치가 완료된 원판 즉 유전자가 6개이며 개체집단의 크기는 4이다. 스핀 행렬로부터 구해진 개체로부터 개체집단을 구성한 후 유전자 알고리즘을 적용하여 최적해를 구한다.

표 1. 6×4스핀 행렬의 예
Table 1. A Spin Matrix of 6×4

	1	2	3	4		1	2	3	4
1	0.25	0.25	0.25	0.25	1	1	0	0	0
2	0.25	0.25	0.25	0.25	2	0	1	0	0
3	0.25	0.25	0.25	0.25	3	0	0	0	1
4	0.25	0.25	0.25	0.25	4	0	1	0	0
5	0.25	0.25	0.25	0.25	5	1	0	0	0
6	0.25	0.25	0.25	0.25	6	0	0	1	0

(a) Initial State

(b) Final State

MFA에서는 상태를 비용함수로 표현하며 비용함수의 값이 가장 큰 상태가 본 실험에서 얻고자 하는 최적해를 의미한다. 따라서 비용함수, C(s),의 최댓값은 배치가 완료된 원판의 조합 가운데서 밀도가 가장 큰 조합의 상태를 나타낸다. 즉, 찾고자 하는 최적해는 비용함수가 최대인 상태를 말한다.

유리재단 문제에서는 유전자가 같은 개체에 할당되어 요구되는 모든 패턴을 다 배치하였을 때의 모든 패턴의 value의 합은 일정하므로 사용된 원판의 면적의 합이 최소일 때가 최적해이다. 따라서 모든 패턴의 value의 합을 사용된 원판의 면적의 합으로 나눈 값 즉, value-밀도가 최대가 될 때가 최적해이다. 그러나 유전자가 개체에 할당될 때 주어진 패턴의 개수를 초과하는 경우는 제외하여야 하므로 identity함수를 적용한다.

$$C(s) = \sum_{i=1}^N \sum_{j \neq i} \sum_{p=1}^K S_{ip} S_{jp} \rho_{value} I_{ij} \tag{1}$$

N : 총 유전자 수

K : 개체집단 크기

S_{ip} : 유전자 i가 개체 p에 배정될 확률

S_{jp} : 유전자 j가 개체 p에 배정될 확률

ρ_{value} : 유전자 i와 j의 value-밀도

I_{ij} : 유전자 i와 j가 허용 패턴 수 이하이면 1 아니면 0

3-1 분산 평균장 어닐링 알고리즘(DMFA)

평균장 어닐링은 시뮬레이션된 어닐링에 물리학의 평균장 근사법 (mean-field approximation)을 적용한 방법론이다. 시뮬레이션된 어닐링이 상태 변환을 하나의 개체에 대하여 수행하는 반면에 평균장 어닐링은 평균장 근사법을 사용하여 개체집단에 대한 열평형 상태에 빠르게 도달한다.

평균장(mean field), Φ_p , 는 다음의 식으로 정의되며 유전자 i 가 개체 p에 할당될 때 얻어지는 비용함수의 증가를 나타낸다.

$$\phi_{ip} = \frac{\partial C(s)}{\partial s_{ip}} = \sum_{j \neq ip=1}^N \sum_{s_{jp}} \rho_{value} \quad (2)$$

스핀 값 s_{ip} 는 $e^{\phi_{ip}/T}$ 의 값에 비례하므로 이를 정규화 하면 다음과 같다. 여기서 T는 온도를 나타낸다.

$$s_{ip} = \frac{e^{\phi_{ip}/T}}{\sum_{q=1}^K e^{\phi_{iq}/T}} \quad (3)$$

식 (1)에 의해서 비용함수, C(s), 는 s_{ip} 에 선형적으로 비례하므로 비용 함수의 변화 ΔC 는 s_{ip} 의 변화 값 Δs_{ip} 에 선형 비례한다.

$$\Delta C = \sum_{p=1}^K \Delta C_p = \sum_{p=1}^K \phi_{ip} \Delta s_{ip} \quad (4)$$

여기서 $\Delta s_{ip} = s_{ip}^{new} - s_{ip}^{old}$ 로 정의한다.

MPI를 이용한 분산처리를 위해서 N×K 스핀행렬을 열 단위로 서로 다른 노드에 할당한 다음 모든 노드가 같은 유전자 i를 선택한 후 각 열에 해당하는 평균장, Φ_p 를 동시에 계산한 후 다른 노드로부터 평균장 값을 받아서 합한 후 s_{ip} 를 계산한다. 다음 의사 코드는 분산 평균장 어닐링 알고리즘을 나타낸다.

while (비용변화, ΔC ,가 충분히 작은 값 ϵ 보다 크다) **begin**
 각 노드에서는 같은 seed 값을 사용하여 같은 유전자 i 선택한 후 지역 평균장을 구한다.

$$\phi_{ip} = \frac{\partial C(s)}{\partial s_{ip}} = \sum_{j \neq ip=1}^N \sum_{s_{jp}} \rho_{value} \quad \text{for } 1 \leq p \leq K$$

global-sum 연산을 이용하여 새로운 스핀 값을 구한다.

$$s_{ip}^{new} = e^{\phi_{ip}/T} / \sum_{q=1}^K e^{\phi_{iq}/T}$$

global-sum 연산을 이용하여 스핀 값의 변화에 따른 새로운 비용변화를 구한다.

$$\Delta C = \sum_{p=1}^K \phi_{ip} (s_{ip}^{new} - s_{ip})$$

i-행의 새로운 스핀 값을 갱신한다.

$$s_{ip} = s_{ip}^{new} \quad \text{for } 1 \leq p \leq K$$

global-collect 연산을 이용하여 i-행의 새로운 스핀 값 저장
end

평균장 어닐링을 수행할 때 다음과 같은 온도 및 평형상태에 대한 변수가 정의된다.

- 초기 온도 : T_0
- 최종 온도 : T_f
- 일정 온도 k에서의 Markov 사슬 길이 : L_k
- 온도 감소율 : $T_{k+1} = f(T_k)$

냉각 스케줄은 문제의 특성에 따라 민감하게 작용하므로 유리제단 문제에 대한 적절한 변수 선택은 중요하다. 초기 온도는 비용의 변화가 나타나지 않도록 충분히 커야 하며, 최종 온도는 유전자의 개수(N)번의 상태변환 동안 비용의 감소가 발생하지 않는 온도로 정하였다. Markov 사슬의 길이, L_k , 는 일정 온도에서 평형상태에 도달하는데 필요한 상태 변환의 수를 나타내며 이를 유전자의 개수로 정하였다. 즉, N번의 상태변환 동안 비용 변화가 $\epsilon(=0.5)$ 보다 작으면 평형상태에 도달한 것으로 간주하였다. 온도 감소율은 0.95에서 0.98로 정의하였다.

3-2 분산 유전자 알고리즘(DSGA)

유전자 알고리즘(GA)에서는 온도에 대한 개념이 없으며 이에 따른 열 평형상태에 대한 개념 또한 없다. 따라서 평균장 어닐링(MFA)에서의 평형상태를 유전자 알고리즘에서도 유지하기 위하여 시뮬레이션된 어닐링(SA)에서와 같이 Metropolis Criterion을 이용하여 새로운 상태의 수락 여부를 결정한다. 다음의 Metropolis Criterion식에서 ΔC 는 상태변화에 따른 비용 변화를 의미하며, 비용함수가 최대일 때 최적이므로 변화된 상태의 비용으로부터 전 상태의 비용을 빼서 얻어진다. T는 온도를 나타낸다.

$$\Pr[\Delta C \text{ is accepted}] = \min\left(1, \exp\left(\frac{\Delta C}{T}\right)\right) \quad (5)$$

분산 유전자 알고리즘에서 사용되는 파라미터에 대하여 설명은 다음과 같다.

- 유전자(gene)
 배치에 필요한 사각형 패턴들의 종류(가로×세로) 및 개수와 원판의 종류 및 최대 사용 개수를 입력 받아 각 패턴의 면적이 큰 순으로 번호를 부여한 후 패턴 번호 순으로 주어진 원판에 버려지는 부분이 최소가 되도록 배치를 수행한다. 배치가 완료된 원판을 MFA에서는 행의 값으로 GA에서는 유전자로 사용하며 유전자의 수가 너무 많을 경우에는 본 실험에서는 각 패턴당 128개로 제한하였다.
- 개체(individual) 또는 염색체(chromosome)
 개체는 이미 만들어진 유전자의 조합으로 요구되는 모든 패턴 및 개수가 배치된 원판의 조합으로 원판 또한 최대

개수를 초과하지 않아야 한다. 이러한 배치를 제약 조건이 있는 절단 문제(constraint cutting)라 한다. 따라서 유리 제단 문제는 제약조건을 만족하는 해 중에서 가장 밀도가 높은 즉, 최소 원판이 소요되는 해를 찾는 최적화 문제이다.

- 개체집단(population)의 크기

본 연구에서는 개체집단의 크기를 스핀행렬의 열의 크기와 같은 128로 정하였다. 분산처리에서 각 노드는 전체 개체집단(P_{global})으로부터 노드의 개체집단(P_{sub})을 균등하게 배분 받은 후 독립적으로 유전 알고리즘을 수행한다. 예를 들어 노드의 수, P 가 4이면 P_{sub} 는 32이다.

- 비용 또는 목적 함수

평균장 어닐링 알고리즘에서 적용한 비용함수의 특성을 최대한 유지하기 위하여 개체의 유전자 즉 배치가 완료된 원판의 조합을 value-밀도가 큰 순으로 즉, 유전자 번호 순으로 제약조건이 만족하게 선택한 후 제약조건이 위배되는 유전자부터는 탐욕적인 방법으로 제약조건이 만족하도록 유전자를 번호순으로 선택하여 모든 개체가 제약조건이 만족하도록 하며 이때의 밀도를 GA에서는 비용함수로 선택하였다.

- MFA의 스핀행렬로부터 SGA의 개체집단 생성

MFA의 스핀행렬의 행이 유전자를 열이 개체를 나타내므로 각각의 열이 개체를 이루며 제약조건이 만족하도록 스핀 행렬로부터 주어진 개체의 유전자를 value-밀도가 큰 순으로 최대한 보존하면서 유전자를 조정한다.

- SGA의 개체집단으로부터 MFA의 스핀행렬 생성

개체집단으로부터 각각의 유전자가 존재할 확률을 계산하여 MFA의 스핀행렬의 열을 로 구성한다.

- 선택 연산자(selection)

개체집단의 전체 비용에 대한 각 개체의 비용 비율로 개체를 선택한다.

- 교차 연산자(crossover)

임의로 2개의 부모 개체 $P1$ 과 $P2$ 를 선택한 다음 교차지점을 랜덤하게 지정한 후 기존의 교차연산과는 다르게 많은 유전자 부분은 자식 개체 $C1$ 과 $C2$ 로 각각 그대로 복사하고 유전자 수가 적은 부분에 대하여 교차연산을 수행한다. 이 경우에도 제약조건이 만족하도록 유전자를 조정한다. 이렇게 하는 이유는 자식의 유전자가 급격히 변화하는 것을 막음으로써 선택될 확률을 높이기 위해서 이다. 교차연산 수행 확률, $P_{crossover}$ 는 0.8로 정하였다.

- 돌연변이 연산자(mutation)

돌연변이 연산 확률, $P_{mutation}$ 은 0.05로 정하였으며 개체를

선택한 후 각 1~2개의 유전자를 랜덤하게 선택하여 삭제한 후 제약조건에 맞도록 유전자를 조정한다.

- 최적 유전자 보존

개체집단에서 가장 밀도가 높은 즉, 비용함수의 값이 큰 개체는 항상 개체집단에 남아 있도록 보존한다.

- 일정 온도에서의 연산 수행 횟수

일정 온도에서의 유전연산의 수행 횟수는 유전자의 수인 128로 고정 하였다. 분산처리 적용시 노드의 수, P , 만큼 노드가 동기화를 이루어 새로운 개체집단 및 노드의 개체집단을 생성한다.

다음은 분산 유전자 알고리즘의 의사 코드이다.

MFA의 스핀행렬로부터 노드의 개체집단(P_{sub})을 구성한다.

repeat 노드 수(P) **begin**

P_{sub} 에 대한 비용함수를 계산한다.

repeat 128/ P **begin**

노드의 개체집단(P_{sub})로부터 2개의 개체를 랜덤하게 선택한 후 $P_{crossover}$ 확률로 교차연산을 수행한다. 비용변화(ΔC)를 계산한다.

if $\exp(-\Delta C/T) > \text{random}[0,1]$ **then**

새로운 개체를 수락한다.

end

임의로 선택된 개체에 대하여 $P_{mutation}$ 확률로 돌연변이 연산자를 적용한다.

비용변화(ΔC)를 계산한다.

if $\exp(-\Delta C/T) > \text{random}[0,1]$ **then**

새로운 개체를 수락한다.

end

end

P_{sub} 를 다른 모든 노드에게 브로드캐스트하여 전체 개체집단(P_{global})을 생성한다.

새로운 P_{sub} 를 구성한다.

최적의 비용을 가진 개체를 보존한다.

end

3-3 분산 합성 알고리즘(DMGA)

합성 알고리즘의 냉각 스케줄(cooling schedule)은 MFA의 스케줄을 그대로 따른다. 각 온도에서의 연산은 먼저 MFA를 수행하고 다음 열 평형상태를 유지하며 SGA를 수행을 반복한다.

다음은 분산 합성 알고리즘의 의사코드이다.

MFA의 스핀 행렬을 초기화한다.

현재온도를 초기온도로 설정한다. ($T = T_0$)

while $T \geq T_f$ (=최종 온도) **begin**

DMFA 수행한다.

DMFA의 스핀행렬로부터 DSGA 개체집단을

생성 한다.

DSGA 수행한다.
 DSGA 개체집단으로부터 DMFA 스핀 행렬을 생성한다.
 $T = aT$ 로 온도를 감소시킨다. ($a=0.95\sim 0.98$)

end

IV. 시뮬레이션 결과

유리재단 문제에 적합한 합성 알고리즘은 일반적으로 MFA나 GA를 개별적으로 사용할 때보다 최적해를 도출하는데 오랜 시간이 걸리므로 수행시간 단축을 위해서 분산처리를 수행하였다. 실험환경은 윈도우10 운영체제의 Intel Core i5, CPU 3.5GHz QuadCore 컴퓨터를 10Mbps의 인터넷으로 연결하여 MPI 환경을 구축하였다.

표2와 그림3은 유리 산업에서 사용되는 실례로서 총 4가지 모양의 사각형(patterns)을 원판에 배치하는 예를 보여 준다.

표 2. 원판 및 패턴 특성
 Table 2. The types of patterns and sheets

	No.	length(mm)	width(mm)	quantities
sheet	0	1524	3048	100
	1	2286	3353	57
	2	2438	3048	147
pattern	0	748	1524	370
	1	558	1720	24
	2	448	1124	34
	3	783	1729	82

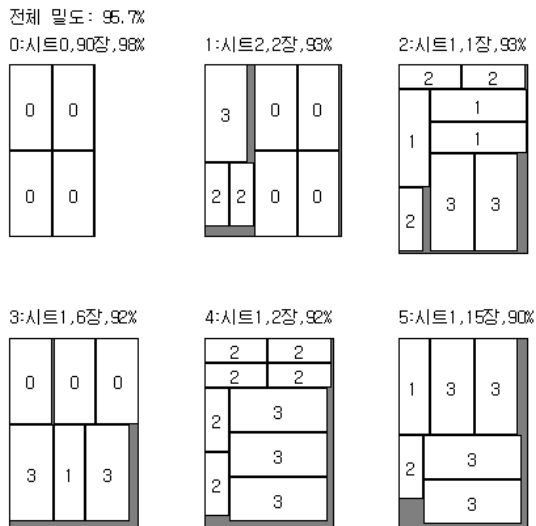


그림 3. 배치의 실제 예
 Fig. 3. The Example of placements

70개의 서로 다른 샘플에 대하여 평균장 어닐링, 유전자 알고리즘, 합성 알고리즘을 수행하여 최종결과에 대한 배치 밀도를 비교하였다. 표3에서와 같이 합성 알고리즘은 GA와 MFA 알고리즘에 비하여 수행시간은 오래 걸리나 배치 밀도는 높게 나타났다.

표 3. 각 알고리즘의 성능 비교
 Table 3. The final optimal results of each algorithms

	Waste Density(%)	Avg. Run Time(sec)
Hybrid	11.67	520
Genetic	11.92	412
Mean Field	12.53	285

3가지 알고리즘 중 MFA가 가장 안 좋았으며 유전자 알고리즘을 단독으로 사용하였을 때 MFA에 비해서 결과가 많이 향상되었다. 본 논문에서 사용한 합성 알고리즘은 유전자 알고리즘에 비하여 밀도가 크게 향상되지 못하였으나 일반적으로 전역 최적해에 가까워질수록 성능향상도가 낮게 나타나는 것을 감안하면 0.25% 정도의 향상은 의미 있는 값으로 판단된다. 실제 두 알고리즘의 샘플별 결과를 비교해 보면 합성 알고리즘의 결과와 유전자 알고리즘의 결과가 대부분의 경우 비슷하나 특정 샘플에서는 원판의 사용을 많이 줄인 것을 볼 수 있었다.

합성 알고리즘에 사용되는 평균장 어닐링과 유전자 알고리즘은 알고리즘 특성상 병렬화를 쉽게 이룰 수 있다. 평균장 어닐링에서는 스핀행렬을 열 단위로 분해하여 병렬처리를 수행하며 유전자 알고리즘에서는 개체집단을 노드 수에 따라 균등하게 배분함으로써 독립적으로 병렬화가 이루어진다. 따라서 두 알고리즘을 결합한 합성 알고리즘에서도 병렬화를 손쉽게 이룰 수 있어서 수행시간을 단축하면서도 최종 최적해의 배치 밀도를 향상하는 결과를 보여 주었다.

그림4는 서로 다른 샘플에 대해서 병렬화가 다르게 나타나는 특성을 보여준다. s1-p4는 sheet는 1종류, 패턴은 4종류로 구성된 문제를 나타낸다.

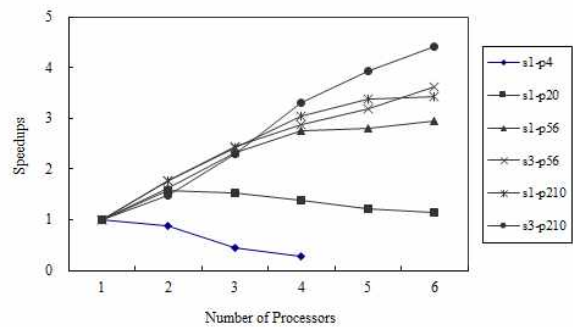


그림 4. 여러 샘플에서의 수행시간 단축
 Fig. 4. Speedups of different samples

병렬화를 통한 수행시간의 단축(parallel speedup)은 병렬화에 참여한 노드의 수뿐만 아니라 문제의 크기 즉, 원판 및 패턴 종류의 수에도 비례하였다. 이는 원판 및 패턴의 종류가 늘어남에 따라 가능해가 폭발적으로 증가하여 MFA 및 SGA에서 일정 온도에서 열 평형상태에 도달하기 어려워져 수행시간이 오래 걸리는 것으로 분석된다. 그러나 분산 합성 알고리즘은 MFA에서는 각 온도마다 동기화가 이루어지며 SGA에서는 노드 수만큼의 동기화가 추가로 이루어져서 우수한 해를 포함할 확률을 높여서 문제의 크기가 큰 유리재단 문제에서 병렬화 효과가 높게 나타난 것으로 분석된다.

V. 결 론

본 논문에서는 평균장 어닐링과 유전자 알고리즘의 장점을 결합한 합성 알고리즘을 유리재단문제에 적용하여 최적화 알고리즘을 개별적으로 사용할 때보다 좋은 결과를 내었으며 분산처리를 적용하여 최적화 문제의 단점인 수행시간을 단축하였다. 실험에서도 알 수 있듯이 분산 합성 알고리즘은 문제가 복잡해지거나 어려워질수록 더 좋은 성능을 예상할 수 있어 실제 상용화에도 기여할 것으로 기대된다. 따라서 현재 외국에서 수입하여 사용 중인 재단 도구의 국산화를 통한 기업 경쟁력 향상에도 이바지하고자 한다.

향후 연구로는 MFA에서의 임계 영역에 대한 자동 탐지 문제, 온도 감소율이 최적 값에 미치는 영향을 조사하여 어떠한 종류의 최적화 문제에 대해서도 자동적으로 어닐링 스케줄을 조정할 수 있는 냉각 스케줄에 대한 연구가 필요할 것으로 보인다. 또한 어닐링 알고리즘의 hill-climb power의 특성을 파악하여 냉각 스케줄에 자동 적용 가능하도록 하여 보다 좋은 최적해를 얻을 수 있는 연구가 필요해 보인다.

감사의 글

본 연구는 2016년도 상명대학교 교내연구비를 지원받아 수행하였음

참고문헌

[1] L. Liu, Y. Peng, W. Xu, "To converge more quickly and effectively—Mean field annealing based optimal path selection in WMN," *Information Sciences*, Vol. 294, pp.216-226, February 2015.

[2] V. Sachnev, S. Suresh, Y. S. Choi, "Cancer subtype's classifier based on Hybrid Samples Balanced Genetic Algorithm and Extreme Learning Machine," *The Journal of*

Digital Contents Society, Vol. 17, No. 6, pp. 565-579, December 2016.

[3] S. M. Sait, F. C. Oughali, F. Chikh, M. Al-Asli, "Design partitioning and layer assignment for 3D integrated circuits using tabu search and simulated annealing," *Journal of applied research and technology*, Vol. 14, No. 1, pp. 67-76, January 2016.

[4] S. Rho, "Artificial Intelligence technology R&D Trend by Patent Analysis," *The Journal of Digital Contents Society*, Vol. 18, No. 2, pp. 423-428, April 2017.

[5] Z. Li and P. Schonfeld, "Hybrid simulated annealing and genetic algorithm for optimizing arterial signal timings under oversaturated traffic conditions," *Journal of Advanced Transportation*, Vol. 49, No. 1, pp.153-170, January 2015.

[6] D. A. Wuttke, H. S. Heese, "Two-dimensional cutting stock problem with sequence dependent setup times," *European Journal of Operational Research*, Vol. 265, No. 1, pp. 303-315, February 2018.

[7] R. Andrade, E. G. Birgin, R. Morabito, "Two-stage two-dimensional guillotine cutting stock problems with usable leftover," *International Transactions in Operational Research*, Vol. 1, No. 2, pp.121-145, January 2016.

[8] M. Delorme, M. Iori, S. Martello, "Bin packing and cutting stock problems: Mathematical models and exact algorithms," *European Journal of Operational Research*, Vol.255, No. 1, pp. 1-20, November 2016.



홍철의(Chuleui Hong)

1989년 미국 뉴저지공과대학교
전산학(석사)
1992년 미국 미주리대학교
전산학(박사)

1992년~1997년 한국전자통신연구원(ETRI) 선임연구원
1997년~현재 상명대학교 컴퓨터과학과 교수
※관심분야 : 분산시스템, 인공지능, 최적화기법, 멀티미디어 응용 등