시뮬레이티드 어닐링을 이용한 이종병렬기계에서의 일정계획 수립
(Job Scheduling for Nonidentical Parallel Machines Using Simulated Annealing)

김경희*, 나동길*, 박문원*, 김동원*
(전북대학교 산업공학과)

Abstract

This paper presents job scheduling for non-identical parallel machines using Simulated Annealing (SA). The scheduling problem accounts for allotting work parts of L lots into M parallel machines, where each lot is composed of N homogeneous jobs. Some lots may have different jobs while every job within each lot has common due date. Each machine has its own performance and set up time according to the features of the machine, and also by job types. A meta-heuristic, SA, is applied in this study to determine the job sequences of the scheduling problem so as to minimize total tardiness of due. The SA method is compared with a conventional steepest descent(SD) algorithm that is a typical tool for finding local optimum. The comparison shows the SA is much better than the SD in terms of tardiness while SA takes longer , but acceptable time.

1. 서론

본 연구에서는 한 화합물 반도체 제조업체의 웨이퍼 절단 공정의 일정계획 문제에 대해서 다룬다. 이 공정은 병목공정으로, 이 공정에서의 생산효율이 전체시스템의 생산효율에 큰 영향을 미친다. 기계가 고가이기 때문에 단순히 기계의 수를 늘이는 것보다는 현재의 생산능력을 효율적으로 사용하여 주문생산 환경에서 납기를 줄이는 것을 가치 있는 일이라 하겠다.

이 공정을 수행하는 기계들은 노후화와 종류의 차이로 인하여 수행시간의 차이를 보이는 이종병렬기계이다. 또 작업물에 따라, 그리고 그것을 수행하는 기계에 따라 서로 다른 셋업시간이 발생한다. 작업물의 흐름은 일정수가 모여 배치를 이루어 이동하고 있는데, 각 기계는 한번에 한 작업물만 수행 가능하다.

이종병렬기계에 대하여 지연시간을 최소화하는 문제에 대한 연구는 다음과 같다. Suresh and Chaudhuri(1994)은 작업시간과 납기의 다양한 조합에 대하여 휴리스틱한 방법을 제안하였고, Adamopoulos and Pappis(1998)는 각 작업의 납기가 동일한 경우에 대하여 연구하였다.

최근에는 복잡한 조합최적화 문제를 해결하기 위해 개발된 메타 휴리스틱(Meta Heuristic) 방법을 일정계획 문제에 적용하여 좋은 해를 얻고 있다. Koulamas(1997)가 SA를 이용한 병렬기계의 일정계획을 제시하였는데, 동일한 기계를 대상으로 하며, 제시한 방법에서 SA가 이용된 부분은 분할(Decomposition)에 의해 기계에 할당된 작업을 교환하는 부분이다. 그리고 Park and Kim(1997)은 작업능력이 동일한 병렬기계의 일정계획에 대해서 SA와 Tabu Search를 이용하여 비교하였는데, 작업물이 뱃취를 이루고 이 작업물에 대하여 준비시간(ready time)과 납기(due date)를 만족시키면서 완성품과 재공품의 유지비용 최소화를 목적으로 하였다.

2. 문제정의 및 특성

본 연구의 대상은 J개의 작업이 하나의 로트를 이루고 있고, 이러한 로트가 L개 있는 작업장에 대해서, M대의 서로 다른 능력을 가진 병렬기계가 작업을 수행할 때, 총 납기지연의 최소화를 목표로 하는 일정계획이다. 각각의 작업은 할당되는 기계에 따라 서로 다른 작업시간을 갖는데, 한 로트의 각 작업은 특정기계에서 동일한 작업시간을 갖는다. 그리고 한 대의 기계는 한번에 하나의 작업만 수행 가능하고, 진행중인 작업이 완료되기 전에 새로운 작업이 투입될 수 없다. 이때 한 로트의 작업은 여러 대의 기계에서 분산되어 수행될 수 있고, 이 공정에서 한 로트의 완료시점은 이 로트를 구성하는 작업들 중에서 가장 늦게 완료되는 작업의 완료시점이 된다. 그리고 셋업시간은 기계의 종류, 작업순서에 따라 다른 값을 가질 수 있다. 하나의 로트는 동일한 작업들로 이루어져 있으므로 이 작업들 간의 셋업시간은 0이다. 납기는 로트에 따라 정해져 있다. 위의 문제에 관한 기호 정의는 다음과 같다.

P_{ijk} : 로트 i의 j번째 작업을 기계 k에서 수행할 때 수행시간 ($P_{i1k} = P_{i2k} = P_{i3k} \cdots$)

$S_{ii'k}$: 로트 i에서 i'로 전환할 때 기계 k에서의 셋업시간

3. Simulated Annealing

SA는 조합 최적화 문제의 한 해법으로 Kirkpatrick et al.(1983) 등에 의해 제안되었다. 실제문제에 SA를 적용하기 위해서는 미리 결정해야 할 것들이 있다. 우선 초기해를 기초로 하여 이웃해 생성과정을 정의하는 것이 필요하고, 이 이웃해의 생성을 효과적으로 하기 위해서 결정해야 하는 모수로는 초기온도, 온도를 내리는 비율, 한 온도에 대해서 몇 회의 이웃해 생성을 시도할 것인지를 결정하는 내부루프의 반복수, 종료기준이다. 좋은 해를 계산하기 위한 이 모수들의 조합은 문제에

따라 다르다. 본 연구에서 이들을 결정한 방법은 다음과 같다.

3.1 초기해

본 연구에서는 납기가 로트에 따라 정해져 있으므로 그것에 의해 로트의 순서를 정한 후에 기계에 할당한다. 즉, 납기가 가장 빠른 로트를 선택한 후에 그 로트의 작업들이 하나씩 기계 인덱스 순서로 모두 할당한 후에 다음으로 빠른 납기를 가진 로트의 작업들을 할당한다.

3.2 이웃해 생성

문제가 로트를 다루고 있고 셋업시간이 있기 때문에, 로트내의 한 작업 단위로 작업순서를 변화시켜서 이웃해를 생성하는 것보다는 로트와 셋업시간을 고려한 이웃해 생성이 타당하다 하겠다. 실제 이들을 고려하지 않고 한 작업단위로 작업순서를 바꾸는 방법을 SA에 적용한 결과 다음에 소개될 이웃해 생성 방법보다 더 많은 탐색시간을 갖게 하더라도 좋지 않은 해를 나타내었다. 이웃해 생성은 다음에 소개될 6가지 생성방법을 통하여 생성된다

.

• 로트 교환(Lot Interchange)

이 방법에서는 두 로트를 선택하여 서로 교환한다. 여기서의 로트란 한 로트를 이루는 작업전체를 말하는 것이 아니고, 한 로트를 이루는 작업들 중에서 한 기계에서 수행되도록 계획된 부분을 말한다. 먼저 교환할 첫 번째 로트를 선택할 때, 납기지연이 많이 발생할수록 선택될 확률을 높여준다. 즉 로트 A가 선택될 확률은 로트 A의 납기지연/총 납기지연이 된다. 그리고 로트의 지연은 그 로트의 작업들 중 완료시간이 가장 큰 작업에 의해 결정되므로, 선택된 로트를 수행하도록 계획되어진 기계들 중에서 그 로트에 속한 작업들의 완료시간이 가장 큰 값을 가지는 작업의 집합을 선택한다. 나머지 한 로트는 임의의 로트를 선택하고, 그 로트를 수행하도록 계획되어진 기계들 중에서 임의의 기계에서 수행되는 것으로 한다. 만일 그림 2와 같이 첫 번째 로트로 기계 1의 로트 5가 선택되고, 두 번째 로트로 기계 2의 로트 2가 선택되었을 때, 선택된 대로 두 로트를 교환하면 기계 2에서 같은 로트 5에 속한 작업들이 분할되어 수행되게 된다. 이는 불필요한 셋업시간의 증가를 초래 할 수 있으므로 로트 5의 작업들이 연속하여 처리되도록 한다.

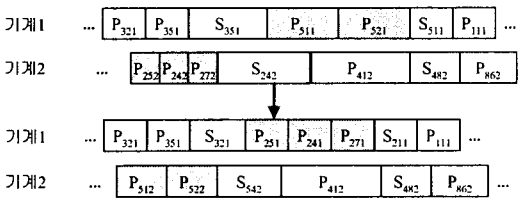

그림 1. 로트 교환(Lot Interchange)

• 로트 삽입(Lot Insert)

두 로트를 선택하여 한 로트를 다른 로트의 뒷부분으로 삽입한다. 두 로트의 선택기준은 로트 교환 방법과 같다. 로트 교환과 마찬가지로 선택된 로트가 현재해에서 상대 기계에서 수행되고 있는 경우에는 한 기계에서 한 로트가 나뉘어져 수행되지 않도록 같은 로트에 연결시킨다.

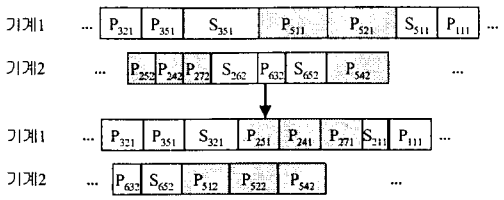

그림 2. 로트 교환 : 상대 기계에 같은 로트가 있는 경우

• 로트 병합(Lot Merge)

다른 기계에서 수행되는 같은 로트에 속해 있는 작업의 두 집합을 선택하여, 두 집합이 한 기계에서 수행되도록 한다. 이는 같은 로트에 속한 작업들이 나뉘어져 수행되었을 때 초래될 수 있는 셋업시간을 줄여 목적함수를 개선하기 위해서다. 로트의 선택기준은 로트 교환의 첫 번째 로트를 선택하는 기준과 같이 납기지연의 크기에 비례하여 선택하도록 한다. 그리고 이 로트를 이루고 있는 작업의 집합 중에서 완료시간이 가장 큰 작업의 집합을 선택하고, 나머지 한 집합은 임의의 기계에서 수행되는 것으로 한다.

• 로트 분할(Lot Split)

한 기계에서 수행되도록 계획되어진 한 로트의 작업들을 두 기계에서 수행되어지도록 한다. 이는 완료시간을 크게 하는 로트의 부분집합을 분할하여 이 로트의 지연시간을 줄여보기 위함이다. 로트의 선택기준은 로트 병합 방법과 같다. 이 로트의 작업 중에서 완료시간이 가장 큰 작업의 집합을 선택하여 이 집합의 임의의 위치에서 분할한 후, 그 로트가 수행되지 않도록 계획되어진 기계 중에서 임의의 기계, 임의의 위치에서 수행되도록 한다. 분할된 작업이 수행될 임의의 위치를 선택할 때 한 로트의 작업이 연속되고 있는 사이는 다른 작업이 삽입될 때 셋업시간의 증가를 초래할 수 있으므로 임의의 위치선택에서 제외시킨다.

• 작업 교환(Job Interchange)

로트 단위로 이동하는 것보다 더 미세한 개선을 위하여 작업 단위로 이동시키는 것을 생각해 볼 수 있다. 작업 교환은 임의의 두 기계를 선택하고, 이들로부터 각각 하나의 작업을 선택하여 교환한다. 이때 한 로트의 작업이 연속되고 있는 작업들 중에 양끝이 아닌 작업이 선택되어 교환될 경우에는 연속되던 작업들 사이로 다른 종류의 작업이 삽입됨으로써 셋업시간의 증가를 초래할 수 있으므로 이 경우는 피한다.

• 작업 삽입(Job Insert)

임의의 두 기계로부터 각각 하나의 작업을 선택하여 한 작업을 다른 작업의 뒤로 삽입시킨다. 두 번째 작업을 선택할 때 바로 뒤에 같은 작업 종류가 있는 경우에는 첫 번째 작업이 삽입됨으로써 셋업시간의 증가를 초래할 수 있으므로 이 경우는 피한다.

이웃해 후보를 정하는 기준에 있어서 6가지 생성방법에 임의의 경험적 확률을 주어 정하는 방법과 모든 방법에 대해 총지연(total tardiness)을 계

산한 후에 가장 좋은 값을 갖게 하는 것으로 정하는 방법을 비교하였다. 그 결과 후자가 시간은 더 걸리는 반면, 해의 질은 더 좋았다. 그러나 이 시간은 다음에 소개될 수행도 평가에서 볼 수 있는 바와 같이 타당한 시간이므로 이 방법을 택하기로 한다.

3.3 모수결정

SA는 이론적인 조건을 맞추면 1의 확률로 전체 최적해에 수렴하게 되는 이론(Kirkpatrick et. el., 1983)이지만 수렴속도를 고려하여 이 조건을 맞추지 않게 되므로 결국 근사 최적해를 얻게 된다. 따라서 문제 특성에 맞게 초기온도, 냉각스케줄, 내부루프의 반복수, 종료기준을 선택해야 한다. 본 연구의 실험대상은 한 화합물 반도체 회사의 일주일 생산량을 기초자료로 한다. 20개의 작업이 한 로트를 이루고 있고, 이러한 로트가 50개 있다. 기계의 대수는 10으로 한다. 각 기계가 로트의 한 작업을 처리하는 수행시간은 30분과 60분 사이에서 일양분포(Uniform distribution)를 이루고 있고, 한 작업을 처리하고 다음 작업으로 넘어가는데 소요되는 셋업 시간은 전후 작업의 종류가 다른 경우에는 10분과 90분 사이에서 일양분포를 이루며, 전후 작업의 종류가 같으면 0분이다.

• 표준모수

문제 특성에 맞게 일반적으로 적용되고 있는 표준모수를 결정한 후에, 각각의 모수가 다른 값을 가질 때의 해의 질, 수행시간을 고려하여 최종적으로 문제에 적용할 모수를 결정한다.

초기온도는 흔히 SA의 실행 전에 예비적인 실험을 통해 설정된 수락률(=수락된 이동/총이동수) 이상이 되는 최소의 온도를 초기온도로 정한 후 본격적인 SA의 과정으로 들어가는 방법을 취하고 있다(Johnson, D. S., 1989). 본 논문에서는 반복루프 10000회의 실험결과를 기준으로 해의 수락률 60%가 되게 정한다.

현재까지 대표적인 냉각 스케줄은

$T_k = \alpha\, T_{k-1}$, k=1,2, …, $0 < \alpha < 1$

와 같이 기하스케줄에 의해 흔히 정한다. α는 대개 0.5와 0.99사이에서 정해지는데 본 연구에서는 0.96을 사용하기로 한다.

내부루프의 반복수는 보통 이웃해가 생성할 수 있는 경우의 수에 비례하도록 선택되는데 냉각스케줄과 더불어 알고리즘의 성능에 큰 영향을 준다. 그런데 본 연구에서는 이웃해의 경우의 수가 너무 크므로 다음과 같이 작업의 수에 비례하게 정하고자 한다.

$$Epoch\ Length = NmberOfLot \times SizeOfLot \times \beta$$

β는 경험적으로 1.1을 사용하기로 한다.

종료기준은 알고리즘이 끝나는 조건으로서 이론적으로는 온도가 0에 수렴하는 곳에서 끝나야 하지만 알고리즘이 낮은 온도에서 보내는 시간이 너무 길기 때문에 보통 외부루프가 일정횟수 반복될 때까지 해의 향상이 없으면 종료하는 방식을 취하고 있다. 본 연구에서는 작업의 수에 비례하게 정하고자 한다.

$$End\ Count = NmberOfLot \times SizeOfLot \times \gamma$$

γ는 경험적으로 10을 사용하기로 한다. 그리고 충분히 낮은 온도로 생각되는 0.1이 되면 강제로 종료시킨다.

• 모수수정

다음은 앞에서 정한 표준모수를 기준으로 풀고자 하는 문제에 적당하게 모수를 수정하고자 한다. 결정기준은 해의 질과 계산시간을 함께 고려하고자 한다. 앞에서 제시한 실험대상 환경에 대하여 결정

표 1. 최종해와 계산시간

로트 크기	기계 대수	납기 (단위:분)	SA		SA_J		SD	
			최종해(단위:분)	계산시간(단위:초)	최종해(단위:분)	계산시간(단위:초)	최종해(단위:분)	계산시간(단위:초)
10	5	T	4604	241	10121	268	12865	89
		M	8021	236	15518	323	20703	87
		L	14979	255	20057	303	25398	90
	10	T	4405	320	10861	133	11617	174
		M	6511	452	11665	245	14887	180
		L	10728	347	13240	391	20022	178
	15	T	2692	504	5545	350	9547	252
		M	5100	802	5881	515	12152	255
		L	7933	610	7649	681	14570	257
20	5	T	3185	856	26365	939	15221	444
		M	11235	767	29975	2126	29369	460
		L	22415	672	46692	2123	34988	438
	10	T	7283	963	13870	2282	26633	923
		M	13943	1325	32021	635	31682	913
		L	21186	1177	31133	2316	38987	939
	15	T	5330	1494	7787	2386	18448	1487
		M	8877	1592	11788	2361	21787	1508
		L	13262	1485	18233	2349	27229	1546
30	5	T	12946	1422	47680	2890	29626	1041
		M	30991	1886	78762	2908	57600	1085
		L	51516	1338	95278	2912	78343	966
	10	T	7177	1506	26819	3311	25177	2267
		M	14795	2135	33255	3287	40208	2414
		L	23217	1493	49947	3284	51748	2492
	15	T	9404	2278	16894	3374	23691	3737
		M	16382	2397	27843	3362	31706	3778
		L	22678	2357	34802	3372	39302	4196

하고자 하는 모수를 제외한 모든 모수를 표준모수로 고정시킨 후, 결정하고자 하는 모수의 여러 경우 각각에 대하여 10회의 임의변수를 발생하여 비교하였다. 비교한 결과, 초기온도는 해의 수락률 57%에 해당하는 온도로, 냉각스케쥴은 α 값을 0.95, 내부루프의 반복수는 β를 1.1, 종료조건의 γ는 14로 정하였다.

4. 수행도평가

제안한 알고리즘의 효율성을 증명하기 위해 다른 알고리즘과 비교하고자 한다. 하나는 로트, 셋업시간등의 문제특성을 고려하지 않고 하나의 작업단위로 교환하고 삽입하는 기본적인 SA(SA_J)이고, 다른 하나는 국부최소해를 찾는 대표적인 방법인 Steepest Descent(SD) 방법이다. SD 방법은 탐색할 수 있는 해 공간에서 가장 좋은 해로 이동하는 방법으로 SA와 가장 큰 차이점은 나쁜해로의 이동을 허용하지 않는다는 것이다. 타당성 있는 비교를 위하여 SD의 이웃해 탐색방법은 모든 가능한 공간을 비교하여 가장 좋은 해로 이동한다는 것을 제외하고는 적용한 SA 알고리즘에서 제안했던 불필요한 해 공간을 탐색하는 것은 피하는 등 다른 조건은 같게 한다.

제안한 알고리즘인 SA, 작업단위로 해를 찾아나가는 SA_J, 국부최적해를 찾는 SD를 비교하기 위해서 기계의 대수, 로트의 크기, 납기의 다른 조합을 고려하였다. SA_J는 이웃해를 생성하는 방법이 작업단위로 교환하고 삽입하는 2가지이므로 제시한 알고리즘의 1/3수준이다. 해의 질의 타당성 있는 비교를 위하여 SA_J의 내부루프 반복수를 SA의 3배로 늘려 비교하였다. 기준으로 했던 기계의 대수 10, 로트의 수 50, 로트의 크기 20인 환경에 더하여서 기계의 대수 5, 10, 로트의 크기 10, 30을 추가로 고려하고 로트의 납기는 Potts and Van(1982)이 제시한 방법으로 다음의 범위에서 일양분포를 따르게 한다.

$$\left[P\left(1-\tau-\frac{\rho}{2}\right), P\left(1-\tau+\frac{\rho}{2}\right)\right]$$

P는 makespan, τ는 긴급도를 조절하는 모수이고, ρ는 범위를 조절하는 모수이다. P의 정확한 계산이 불가능하므로 작업에 대한 기계의 수행시간, 셋업시간에서 가장 빨리 처리되는 시간으로 계산하였다. ρ는 0.8로 고정시키고, τ는 0.4(tight), 0.45(moderate), 0.5(loose)인 경우에 대하여 비교한다.

각 알고리즘은 C++로 코딩되었고, Pentium 500에서 계산되었다. 표 1은 로트의 수가 50인 경우에 대하여 다른 비교기준들의 조합을 나타내었다. 세가지 비교조합에서 계산시간에 영향을 주는 요인은 SA, SA_J, SD 모두 로트의 크기, 기계의 대수, 납기 순으로 나타났다. 그리고 문제크기가 작을 때는 SD가 SA, SA_J에 비해 계산속도가 빠르나 문제가 커질수록 SA와 SA_J의 계산속도가 상대적으로 빠름을 볼 수 있다. 그리고 SA_J는 이웃해 생성 방법의 수가 적어 내부루프의 반복수를 조절하여 반복수는 같게 했으나 이웃해끼리 비교하는 횟수가 증가하여 SA보다는 긴 계산시간을 보이고 있다. 그러나 몇몇 경우에서는 SA_J가 높은 온도에서 장시간 해의 개선이 이루어지지 않아 종료조건이 만족되어서 알고리즘이 일찍 종료되는 경우를 볼 수 있다. 해의 질에 있어서는 SA가 모든 경우에 있어서 다른 두 방법보다 좋았으며, SA_J와 SD는 비슷한 결과를 보였다.

5. 결론

본 연구에서는 수행능력이 서로 다른 병렬기계에 대해서, 납기지연의 최소화를 목표로 셋업시간, 배치공정을 고려하여 Simulated Annealing을 이용한 일정계획을 수립하였다.

기존연구에서 제시하고 있는 표준모수 설정 방법을 기준으로 문제 특성에 맞는 모수를 실험을 통하여 제시하였고, 이웃해 탐색방법은 로트 교환(Lot Interchange), 로트 삽입(Lot Insert), 로트 병합(Lot Merge), 로트 분할(Lot Split), 작업 교환(Job Interchange), 작업 삽입(Job Insert)을 통하여 문제 특성을 반영하였다.

Simulated Annealing을 이용해 제시한 알고리즘은 국부최적해를 찾는 방법인 Steepest Descent와 셋업시간 배치공정 등을 고려하지 않고 작업 단위로 교환, 삽입하는 방법으로 이웃해를 찾는 Simulated Annealing(SA_J)과 비교되었다. 비교기준은 기계의 대수, 로트의 크기, 로트의 수, 납기의 긴급도에 따라 비교되었다. 해의 질은 비교한 모든 경우에 대하여 SA가 다른 두 방법보다 좋았다. 계산시간은 본 논문에서 대상으로 했던 표준 환경에서는 SD가 SA와 SA_J보다 짧은 결과를 보였으나, 문제의 크기가 커지면 오히려 SD가 다른 두 방법보다 길었다. 이는 고려해야 하는 경우의 수가 많아지면 SD의 특성상 모든 가능한 경우를 탐색하여 가장 좋은 결과를 보이는 선택을 하기 때문이라고 생각된다.

참고문헌

Adamopoulos, George I.; Pappis, Costas P., Scheduling under a common due-date on parallel unrelated machines, *European Journal of Operational Research*, 105(3), 494-501, 1998.

Johnson, David S.; Aragon, Cecilia R.; Mageoch, Lyle A.; Schevon, Catherine, Optimization by simulated annealing: An experimental evaluation; Part 1, Graph partitioning, *Operations Research*, 37(6), 865-892, 1989.

Kirkpatric, S.; Gelatt Jr. C. D.; Vecci, M. P., Optimization by simulated annealing, *Science*, 220, 671-680, 1983.

Koulamas, Christos, Decomposition and hybrid simulated annealing heuristics for the parallel-machine total tardiness problem, *Naval Research Logistics*, 44, 105-125, 1997.

Park, Moon-Won; Kim, Yeong-Dae, Search heuristics for a parallel machine scheduling problem with ready times and due dates, *Computers & Industrial Engineering*, 33(3-4), 793-796, 1997.

Potts, C. N.; Van Wassenhove, L., Decomposition algorithm for the single machine total tardiness problem, *Operations Research Letters*, 1(5), 177-181, 1982.

Suresh, V.; Chaudhuri, Dipak, Minimizing maximum tardiness for unrelated parallel machines, *International Journal of Production Economics*, 34(2), 223-229, 1994.