

병렬의 동일기계에서 처리되는 순서의존적인 작업들의 스케줄링을 위한 유전알고리즘†

이문규¹ · 이승주²

¹계명대학교 자동차공학부 산업공학 전공 / ²(주)서린정보기술

A Genetic Algorithm for Scheduling Sequence-Dependant Jobs on Parallel Identical Machines

Moon-Kyu Lee · Seung-Joo Lee

We consider the problem of scheduling n jobs with sequence-dependent processing times on a set of parallel-identical machines. The processing time of each job consists of a pure processing time and a sequence-dependent setup time. The objective is to maximize the total remaining machine available time which can be used for other tasks. For the problem, a hybrid genetic algorithm is proposed. The algorithm combines a genetic algorithm for global search and a heuristic for local optimization to improve the speed of evolution convergence. The genetic operators are developed such that parallel machines can be handled in an efficient and effective way. For local optimization, the adjacent pairwise interchange method is used. The proposed hybrid genetic algorithm is compared with two heuristics, the nearest setup time method and the maximum penalty method. Computational results for a series of randomly generated problems demonstrate that the proposed algorithm outperforms the two heuristics.

1. 서론

작업순서에 의존적인 작업시간(sequence-dependant processing time)을 갖는 작업들의 스케줄링 문제(Parallel-machine Scheduling Problem with Sequence-dependent Processing Times: PSPSPT)는 기계가공업체는 물론이고 섬유업체, 플라스틱 성형 업체, 석유 화학업체 등의 작업현장에서 흔히 발생하는 문제이다(Franca *et al.*, 1996; Serafini and Speranza, 1992). 예로서, 작업마다 사용하는 공구가 다르면 직전 작업에 사용되던 공구들을 제거하고 새로운 공구들을 장착해야 하는 경우를 들 수 있다.

본 연구에서는 병렬의 동일기계(parallel identical machines)로 구성된 작업장에서 발생하는 순서 의존적인 작업들의 스케줄링(scheduling) 문제를 다룬다. 이러한 문제들의 목적함수는 작업완료시간의 최대값(C_{max})을 최소화하거나 납기가 주어진 경우 지연시간(lateness time)의 최대값(L_{max})을 최소화하는 경우가 대부분이다. 여기서는 이러한 목적함수와는 달리, 작업할당 후 남은 잔여 기계가용시간(remaining machine available time:

RMAT)을 최대화하는 목적함수를 고려한다. 이는, 주어진 업무 시간 내에 작업들을 각 기계에서 수행할 때, 작업이 전혀 할당되지 않은 기계들의 가용시간과 작업이 할당된 기계들 중 최대로 남게되는 기계가용시간의 합을 가능한 최대화하여 이 시간을 다른 업무에 활용하고자 하는 현장의 일반적인 상황을 반영한 것이다. 이러한 상황을 반영하는 대표적인 작업장으로는 병렬의 동일기계들로 와이어하니스(wire harness)의 주요 부품인 압착단자(crimped terminal)를 가공하는 작업장을 들 수 있다(강상구, 1991; 임진규와 박주철, 1996). 여기서는 압착단자의 주문량이 일정치 않아서 설치된 기계들을 모두 사용하지 않는 경우가 많은데, 이 경우 작업이 할당되지 않은 기계는 다른 종류의 작업에 활용되고 있다. 따라서 주어진 작업들을 처리하고 남은 가용시간을 최대화하여 그 활용도를 높이는 것이 중요한 과제가 된다. 특히 작업 수가 많을 때는 가동해야 할 기계대수를 사전에 정확하게 예측하지 못하기 때문에 이러한 목적함수가 효과적일 것이다.

이와 같은 PSPSPT에 관한 기존의 연구는 어느 정도 진행되어 있는 상태(Franca *et al.*, 1996; Frederickson *et al.*, 1978; Guinet,

† 이 논문은 1997년 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음

1991; Ovacik and Uzsoy, 1993; Parker *et al.*, 1997; Schutten, 1996; Sumichrast and Baker, 1987)이나 이들 연구들에서는 모두 목적 함수로서 C_{max} , L_{max} 나 총 준비비용(total setup time) 등을 최소화하기 위한 발전적 해법이나 최적해의 특성을 밝힌 바 있을 뿐 본 연구대상의 목적함수는 다루어진 바 없다. 최근에 이문규와 이승주(1998)는 이 문제에 대한 최적해를 제공하는 동적계획(dynamic programming) 모형을 제안한 바 있으나 문제의 규모가 커지면 적용하기가 곤란하다. 여기서는 중대규모의 PSPSPT에도 적용할 수 있도록 조합최적화 문제(combinatorial optimization problem)에 우수한 성능을 보이는 유전알고리즘(genetic algorithm)을 응용한 효과적인 해법을 제시하고자 한다.

2. 최적화 모형

본 연구 대상의 문제는

“순서에 따라 작업시간이 달라지는 n 개의 작업들을 주어진 계획기간 내에 병렬로 구성된 m 개의 동일기계에서 처리할 때, 타 작업에 활용될 수 있는 기계가용시간인 RMAT를 최대화하도록 각 기계에서 처리될 작업들과 그 처리순서를 결정하는 것”으로 정의된다. 여기서, 계획기간은 각 기계가 동시에 작업을 처리한다는 가정하에 각 기계가 작업할 수 있는 총 가용시간을 의미한다. 여기서는 이문규와 이승주(1998)에서 사용된 용어와 수리적 모형을 소개한다.

J_i = 작업 i , $i = 1, \dots, n$;

M_k = 기계 k , $k = 1, \dots, m$;

P_j = 작업 j 의 순수 작업시간, $j=1, \dots, n$;

S_{ij} = 작업 i 가 완료된 후 작업 j 를 처리하기 위한 작업준비시간 ($i=1, \dots, n; j=1, \dots, n$);

S_{n+kj} = 기계 k 에서 작업 j 가 최초작업으로서 처리될 때 발생하는 준비시간으로서 기계에 무관하게 모두 SM_j 로 동일하다고 가정함

($k=1, \dots, m; j=1, \dots, n$);

t_{ij} = 직전 작업이 i 일 때 작업 j 의 처리시간으로서 $S_{ij} + P_j$ 임

$C_M(k)$ = 기계 k 의 완료시간;

MAT = 기계가용시간(machine available time).

RMAT는 <그림 1>에 도시된 예와 같이 작업이 할당되지 않은 기계(M_4)들의 MAT값과 가장 빨리 작업을 완료한 기계(M_3)의 완료시간과 MAT와의 절대차의 합으로 정의된다. 이러한 RMAT는 다른 유형의 작업에 활용될 수 있는 시간을 반영하는 것으로, 이를 최대화하는 것이 목적함수이다. 여기서 작업이 할당되었으나 가장 빨리 끝나지 않은 기계들(M_1, M_2)은 남은 시간이 있다 하더라도 소량일 경우가 대부분이므로 다른 작업에 활용하기가 어렵다. 따라서, 목적함수를 작업이 할당되지 않은 기계들과 가장 빨리 작업을 완료한 기계의 총 가용시간으로 설정한 것이다.

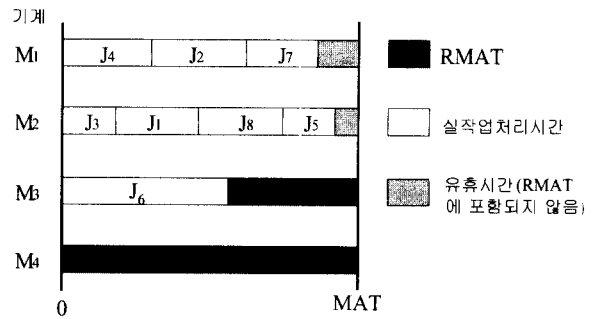


그림 1. 4대의 동일기계(M_1, \dots, M_4)에서 8개의 작업 (J_1, \dots, J_8)을 처리하기 위한 작업 스케줄링의 예.

이 문제를 문제유형 정의를 위한 용어대로 표현하면 $P/scls - g(\sigma)$ 가 되는데(Frederickson *et al.*, 1978), $g(\sigma)$ 는 주어진 스케줄 σ 에 대한 RMAT로서 작업이 할당되지 않은 기계들의 집합

$$U_M = \{M_k \mid C_M(k) = 0, k = 1, \dots, m\}$$

라고 할 때

$$g(\sigma) = MAT - X|U_M| + \max_{k \in U_M} [MAT - C_M(k)]$$

와 같이 정의된다. 여기서 $|S|$ 는 집합 S 의 원소수를 의미한다. 이 문제를 최적해가 보장되는 수리적 모형으로 정형화하면 다음과 같이 혼합정수계획(mixed linear integer programming)이 된다:

$$\max MAT - X|U_M| + \max_{k \in U_M} [MAT - C_M(k)] \quad (1)$$

s.t.

$$\sum_{i=0, i \neq j}^n \sum_{k=1}^m x_{ijk} = 1, j = 1, \dots, n \quad (2)$$

$$\sum_{i=0, i \neq h}^n x_{ikh} - \sum_{j=0, j \neq h}^n x_{bjh} = 0 \quad b = 1, \dots, n; \quad k = 1, \dots, m, \quad (3)$$

$$u_{ik} - u_{jk} + (n+1)x_{ijk} \leq n \quad 0 \leq i \neq j \leq n; \quad k = 1, \dots, m \quad (4)$$

$$C_M(k) = \sum_{j=1}^n (S_n + k, j + P_j) x_{0jk} + \sum_{i=1}^n \sum_{j=1, j \neq i}^n t_{ij} x_{ijk} \leq MAT \quad k = 1, \dots, m \quad (5)$$

$$U_M = \{M_k \mid C_M(k) = 0, k = 1, \dots, m\}$$

$$x_{ijk} \in \{0, 1\}, \quad i = 0, 1, \dots, n-1;$$

$$j = 0, 1, \dots, n; k = 1, \dots, m.$$

$$u_{ik} \geq 0 \text{ (정수)} \quad i = 0, \dots, n-1; k = 1, \dots, m.$$

여기서 결정변수 x_{ijk} 는 작업 j 가 작업 i 바로 다음에 기계 k 에

서 처리될 때는 1이고 그렇지 않으면 0으로 정의된다(단, 작업 0는 각 기계마다 가상적으로 정의되는 최초의 작업을 의미하므로 x_{0jk} 는 작업 j 가 기계 k 에서 최초 작업으로 처리될 때는 1이고 그렇지 않으면 0으로 정의되고 x_{i0k} 는 작업 i 가 기계 k 에서 마지막 작업으로 처리될 때는 1이고 그렇지 않으면 0을 의미함). n_k 는 subtour를 방지하기 위하여 주어지는 조건식 (4)에 사용되는 실수변수이다. 목적함수 (1)은 RMAT이고 제약식 (2)와 (3)은 각각 작업 j 가 오직 하나의 위치에 할당되도록 하는 제약 조건과 흐름량보존(flow conservation) 조건을 나타낸다. 제약식 (5)는 각 기계의 가용용량 초과를 방지하기 위한 조건이다.

3. 발견적 해법

앞 절과 같이 정형화된 최적화 모형은 최적해를 보장하나 문제의 규모가 커질수록 요구되는 계산시간과 저장용량이 급격히 증가하여 실제규모의 문제를 해결하기에는 많은 제약이 따르게 된다. 또한, 목적함수가 nonregular 함수이므로 대부분의 regular 목적함수를 대상으로 한 스케줄링 문제보다 다루기 어렵다. 따라서, 실제현장의 대규모 문제를 해결하기 위해서는 최적해는 보장하지 않으나 합리적인 시간 내에 해를 구할 수 있는 발견적(heuristic)해법들이 요구된다.

3.1 최단준비시간법(nearest setup time method: NSM)

본 문제에 가장 쉽게 적용할 수 있는 발견적 해법은, 현재 부분 완성된 작업처리순서를 확장시켜 나갈 때 아직까지 처리순서가 확정되지 않은 작업 중에서 준비시간이 짧은 작업을 우선적으로 스케줄링하는 최단준비시간법을 생각해 볼 수 있다. 이러한 최단준비시간법은 계산이 복잡하지 않으면서도 스케줄링 성능이 비교적 좋은 것으로 알려져 있고 기타 알고리즘의 비교대상으로 종종 이용되고 있다(안상형, 1995). 병렬기계의 스케줄링을 위한 최단준비시간법은 여러 가지로 변형이 있을 수 있는데 본 연구에서는 다음의 NSM절차에 따라 해를 구하였다.

초기의 R 을 전체 작업들의 집합으로 정의하고 그 이후의 R 은 현재까지 처리순서가 확정되지 않은 작업들의 집합으로 정의할 때, $\min_{i \in R} SM_i$ 를 만족시키는 작업 i^* 를 선택한다. $R = R - \{i^*\}$ 로 재 정의한 후 다음 작업은 순서가 결정되지 않은 작업 중 $\min_{j \in R} S_{i^*j}$ 를 만족시키는 작업 j^* 를 선택한다. 이러한 방법을 현재의 기계에 더 이상 할당할 수 없을 때까지 반복 수행한다. 한 기계의 스케줄링이 완료되면 남아 있는 기계와 작업을 대상으로 위의 과정을 반복 수행하여 모든 작업이 할당될 때까지 반복한다(기계할당 순서는 1, 2, ..., m 순으로 함).

3.2 최대 벌금법(maximum penalty method: MPM)

최단준비시간법과 함께 본 문제에 쉽게 적용할 수 있는 발견적 해법으로, 배정하지 않았을 때의 벌금값이 가장 큰 작업을 우선적으로 처리하는 방법이다.

최소준비시간법과 동일하게 R 을 현재까지 처리순서가 확정되지 않은 작업들의 집합으로 정의할 때, $R = \{1, 2, \dots, n\}$ 중에서 $\min_{i \in R} SM_i$ 를 만족시키는 작업 i^* 를 최초의 작업으로 선택한다. $R = R - \{i^*\}$ 로 재정의하고 다음 작업은 순서가 결정되지 않은 작업 중,

$$\max_{j \in R} [\min_{r \in R - \{j\}} S_{rj} - S_{i^*j}]$$

를 만족시키는 작업 j^* 를 선택한다. 이러한 과정을 현재의 기계에 더 이상 할당할 수 없을 때까지 반복 수행한다. 한 기계의 할당이 완료되면 남아 있는 기계와 작업을 대상으로 위의 과정을 반복 수행하여 모든 작업이 할당될 때까지 반복한다(기계할당 순서는 1, 2, ..., m 순으로 함).

3.3 성능평가

이러한 발견적 기법의 성능을 평가하기 위하여, 준비시간 $\sim U(0, 100)$; 가공시간 $\sim N(100, 30^2)$ 을 따르는 가상의 문제를 $n = 6, 9, 12$ 의 각 경우에 10개씩 임의로 생성하였다. 또한, MAT=480, $m=5$ 로 주어졌다. 이와 같이 생성된 문제에 대하여 각각 NSM과 MPM을 적용하여 구한 결과가 표 1에 정리되어 있다. 여기서 평가기준은 절대평균오차(mean absolute deviation: MAD)로서,

$$MAD(\%) = \frac{\text{최적해} - \text{발견적 해법의 해}}{\text{최적해}} \times 100$$

을 사용하였다. 이 결과, NSM이나 MPM의 성능은 문제에 따라서 우열이 다르게 나타나고 있으며, 이문규와 이승주(1998)에 제시된 동적계획법(Improved Dynamic Programming for Multi-machine Scheduling: IDPMS)을 이용하여 구한 최적해와 비교할 때 문제의 규모가 커짐에 따라 성능저하가 심해지는데 특히 n 의 값이 12인 경우에는 MPM의 MAD가 12.52%에 달하고 있다. 그러나, 규모가 큰 문제를 다룰 경우 성능 비교기준으로서의 가치는 있을 것이라 생각된다.

표 1. 최적해 대비 발견적 해법의 성능

No. of Jobs	MAD (%)	
	NSM	MPM
6	4.44	5.61
9	5.16	7.89
12	10.95	12.52

4. 유전알고리즘

유전알고리즘은 Holland(1997)에 의해 연구되기 시작하여 많은 조합최적화문제에 응용된 일종의 인공지능기법으로서, 자연의 진화과정을 모방하여 개발한 탐색알고리즘인데 최근에는 스케줄링 분야에 활발하게 응용되고 있다(김갑환 외, 1997; 김여근 외, 1997; Han, 1997; Cheng and Gen, 1996; Croce *et al.*, 1995). 유전알고리즘을 적용하기 위해서는 대상문제의 특징에 맞는 알고리즘, 관련된 여러 조정변수들, 유전인자의 유전염색체(chromosome)의 표현방법, 적합도(fitness) 평가함수, 모집단의 초기화 방법, 염색체 선택전략 등을 결정해야 한다. 여기서는 우선 유전알고리즘의 절차를 기술하고 위의 여러 결정항목들을 본 문제의 특성에 부합되도록 수정된 내용을 소개한다.

4.1 유전알고리즘 절차

Holland(1975)의 기본적인 유전알고리즘 절차 이후로 여러 가지 문제의 특성에 따라 변형된 절차가 다양하게 개발되어 왔다. 여기서는 병렬기계의 스케줄링을 위하여 개발된 Memetic 알고리즘(1992)을 수정하여 적용하며, 기계가용시간에 대한 제약은 새로운 개체가 제약조건을 만족하면 다음 세대 수행을 위해 새로운 집단으로 받아들여지고, 불만족할 경우 탈락시키는 변수제약법을 사용하였다. 최초의 Memetic 알고리즘은 Moscato & Norman(1992)이 제시한 것으로서 부모 염색체(parent chromosome)로부터 생성된 자손 염색체(offspring chromosome)를 곧바로 다음 세대에 유전하지 않고 그 자체를 국소최적화 탐색하여 유전하면 해공간(solution space)의 광역탐색을 효과적으로 수행할 수 있다는 것이다. 따라서, 국소최적화 탐색을 위해서 유전알고리즘 대신에 이전에 개발된 효과적인 발전적해법이나 최적화해법을 사용하는 일종의 hybrid 기법이다. $P//L_{max}$ 를 해결하기 위하여 광역최적화를 위한 탐색에서는 유전알고리즘을 사용하고 국소최적화 탐색은 잘 알려진 V-shaped 순서화기법을 이용한 것이 대표적인 예이다(Cheng and Gen, 1996).

본 연구에서 제시한 문제에서의 스케줄링은 이러한 기법이 효과적으로 적용될 수 있도록 각 기계로의 작업할당(job allocation)과 기계 내에서의 작업순서결정(job sequencing)의 두 부분으로 나누어져 있다. 작업할당은 광역탐색을 위한 유전알고리즘을 적용하고 각 기계에서의 작업순서결정을 위한 국소최적화는 앞 절에서 소개된 NSM을 단일기계를 대상으로 하여 적용한 다음 이를 이용한 이웃쌍 교환방법(adjacent pairwise interchange method: APIM)을 사용하였다. APIM은 이웃쌍의 순서를 서로 교환해 가면서 적합도를 나타내는 목적함수값을 비교·개선하는 기법으로 해를 구하는 데 소요되는 시간이 짧지만 비교적 우수한 해를 생성하기 때문에 방문판매원문제(traveling salesman problem: TSP) 등을 비롯한 많은 순서결정문

제에 가장 기본적인 기법으로 사용되어 왔다(안상형, 1995).

이상의 알고리즘을 기술하기 위하여 다음의 용어를 사용한다.

t = 세대 ($t = 0, 1, \dots, T$);

$P(t)$ = 모집단 t ;

N_p = 모집단 크기 = $|P(t)|$;

$C(t)$ = 모집단 $P(t-1)$ 로부터 생성된 자손집단 t ;

α = 모집단 대비 교차변이율(crossover rate, $0 \leq \alpha \leq 1$);

β = 교차변이율 대비 돌연변이율(mutation rate, $0 \leq \beta \leq 1$);

알고리즘의 절차는 다음과 같다.

단계 1: $t=0$;

크기 N_p 인 $P(0)$ 를 임의로 생성함;

$P(0)$ 의 각 개체의 적합도를 평가함.

단계 2: $t = t+1$; $P(t) = C(t) = \emptyset$.

단계 3: $P(t-1)$ 로부터 가장 우수한 개체를 포함하여 $(1-\alpha)N_p$ 개의 개체를 중복시키지 않게 선택하여 $P(t)$ 에 그대로 유전시킴.

단계 4: $P(t-1)$ 에서 αN_p 개의 개체를 중복에 상관없이 선택하여 교차변이와 국소최적화를 통하여 생성된 αN_p 개의 자손개체를 $C(t)$ 에 추가함.

단계 5: 단계 4에서 생성된 αN_p 개의 자손개체 중에서 중복에 상관없이 $\beta \alpha N_p$ 개의 개체를 선택함. 이로부터 돌연변이와 국소최적화를 통하여 생성된 $\beta \alpha N_p$ 개의 자손개체를 단계 4에서 교차변이된 개체 중 가장 열등한 개체와 비교하여 우수하면 교환함.

단계 6: $P(t) = P(t) \cup C(t)$;

$t \geq T$ 이면 종료하여 $P(t)$ 내의 가장 우수한 개체를 최종 해로 선택하고, 그렇지 않으면 단계 2부터 반복 수행함.

이와 같은 유전알고리즘을 수행하기 위해서는 T, N_p, α, β 의 유전 조정변수 값들이 주어져야 한다. 이 조정변수들은 알고리즘의 성능에 상당한 영향을 미치므로 충분한 분석을 거쳐 결정된 값들을 사용해야 한다. 단계 3의 선택전략은 소위 elitist 전략으로서 (Grefenstette, 1986), 기존의 가장 우수한 해를 계속 다음 세대에 유전시켜 최우수해의 탈락을 방지하고자 하는 것이다. 단계 3, 4의 선택전략은 모두 Roulette Wheel 방법(Baker, 1987)을 이용하여, 평가된 개체적합도에 확률적으로 비례하도록 우수한 개체의 유전인자를 다음 세대에 유전시키도록 한다. 그림 2에서는 이러한 선택전략에 따른 세대간 유전과정을 보여주고 있다.

위 절차에서 광역최적화 탐색은 단계 4와 5의 돌연변이와 교차변이를 통하여 각 기계에 할당된 작업들을 바꾸게 함으로써 국소최적해로부터 탈출가능하게 한다. 국소최적화 탐색은 돌연변이 또는 교차변이를 통하여 생성된 개체의 할당결과에 NSM과 APIM을 적용함으로써 적합도가 높아지도록 기계 내의 유전인자인 작업들의 순서를 재조정하게 한다.

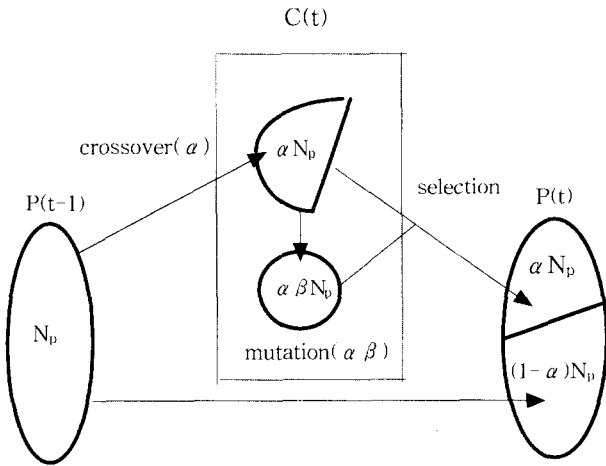


그림 2. 모집단 $P(t-1)$ 에서 자손 $P(t)$ 를 생성하는 과정.

4.2 표현방법(Representation)

유전알고리즘을 적용하기 위해서는 문제의 해가 개체표시자인 염색체로 적절하게 표현되어야 하며 이는 유전알고리즘의 탐색성과 확장성에 있어서 중대한 영향을 미친다. 본 문제에 있어서는 기계의 숫자만으로 표시하고 각 기계들을 하나의 자연수에 대응시켜 하나의 해가 1차원 배열로 표현되는 간단한 방법을 사용하고자 한다. 예로서, 3대의 기계에 9개의 작업을 스케줄링하는 경우

$$\begin{aligned} (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9) &: \text{작업번호} \\ (3\ 1\ 1\ 2\ 2\ 3\ 1\ 3\ 2) &: \text{정수문자열로 표현된 유전자} \end{aligned} \quad (6)$$

라고 하면 기계 1에 작업 {2, 3, 7}, 기계 2에 작업 {4, 5, 9}, 기계 3에 작업 {1, 6, 8}이 할당되어 있음을 의미하나, 그 작업순서는 (6)에 기재된 염색체에서는 나타나지 않는다. 즉, 염색체에는 작업할당에 대한 정보만 표시하며 작업순서는 국소최적화 단계에서 결정한다.

4.3 유전연산자(Genetic Operator)

유전연산자는 알고리즘의 성능에 결정적인 영향을 미치므로 문제의 특성을 감안하여 우수한 해를 효율적으로 탐색할 수 있도록 정의되어야 한다. 이러한 연산자는 크게 두 개의 부모개체로부터 자손개체를 생성하는 이원연산자(binary operator)인 교차변이와 하나의 부모개체로부터 생성되는 일원연산자(unary operator)인 돌연변이가 있다. 순서결정문제(sequencing problem)에 많이 사용된 교차변이 방법으로는 부분사상 교차변이(Partially Map Crossover: PMX), 순서 교차변이(Order Crossover: OX), 순환 교차변이(Cycle Crossover: CX) 등이 있으며 돌연변이로는 삽입(insertion), 교환(swap), 역위(inversion) 그리고 전위(displacement) 등이 있다(김여근과 현철주, 1994).

이러한 유전연산자들의 성능평가에 관한 연구가 지금까지 많이 이루어져 왔다. Fogel(1988)은 단위길이가 1인 정방형 내의 방문판매원문제에 대하여 가장 간단한 연산자인 삽입 돌연변이(부모의 한 유전인자를 임의의 다른 위치로 이동하여 자손을 생성)만으로 구해진 최종해가 다른 해보다 열등할 확률이 10^{-12} 에 불과하다는 결과를 제시하였다. 즉, 최종해의 수준이 최적해로부터 평균적으로 99.9999999999 percentile에 위치한다는 것이다. 이것과는 달리 김여근과 현철주(1994)와 Lim(1996)은 우수한 일원연산자와 이원연산자를 혼용하여 사용할 때 좋은 결과를 나타낸다는 발표를 한 바 있다. 본 연구에서는 이러한 상반된 결과를 종합하여 이원연산자는 순서교차변이(order crossover: OX)를, 일원연산자로서는 한 지점이 변경되는 한점 돌연변이(one point mutation)를 수정·사용하였다.

4.3.1 순서 교차변이

본 연구에서 제시한 순서 교차변이는 Davis(1985)가 제시한 OX를 본 문제에 적합하게 수정한 연산자이다. 먼저 두 개의 부모개체 P_1 과 P_2 로부터 두개의 유전인자를 임의로 선택하여 교차점으로 한다. 다음으로 P_1 의 교차점 사이의 인자들을 자손 C_1 에 그대로 유전시킨 후, 유전된 인자들을 P_2 에서 찾아 지우고 남은 인자들을 순서대로 C_1 의 비어 있는 인자위치에 유전하는 방법이다. C_2 는 P_2 에 선택된 2개의 교차점을 위와 같은 방법을 적용하여 생성한다. 예를 들어,

$$\begin{aligned} P_1 &= (1\ 2\ 1\ | 3\ 1\ 1\ 2\ 2\ 3) \\ P_2 &= (2\ 3\ 3\ | 2\ 2\ 3\ 1\ 1\ 1) \end{aligned}$$

에서 네 번째와 여섯 번째 위치가 선택되었다면 C_1 과 C_2 는 각각

$$\begin{aligned} C_1 &= (2\ 3\ 2\ 3\ 1\ 1\ 2\ 3\ 1) \\ C_2 &= (1\ 1\ 1\ 2\ 2\ 3\ 1\ 2\ 3) \end{aligned}$$

가 된다. 이 연산자를 적용하면 부모 염색체의 기계할당순서를 유전받게 되어 우수한 형질을 지켜나가게 하는 특성을 지닌다.

4.3.2 한점 돌연변이

이 연산자는 일반적으로 부모의 염색체에서 임의의 한 인자를 선택하여 그 유전인자의 특성을 임의로 바꾸어 자손을 생성하는 방법이다. 기존의 일반적인 돌연변이 방법인 교환(swap), 삽입(insertion), 역위(inversion), 전위(displacement), 접합(splice) 연산자 등(김여근과 현철주, 1994)은 본 문제에서 볼 때 기계 내에서의 작업간 교환이 이루어지는 경우가 많다. 이는 이미 국소최적화가 이루어진 상태이기 때문에 아무런 의미가 없다. 따라서, 기계간의 할당구조를 근본적으로 바꾸는 것이 가능하도록 임의의 한 유전인자의 할당기계를 바꾸는 한점 돌연변이

연산자를 사용한다. 예로서 P 와 C 를 각각 부모와 자손개체라 했을 때 네 번째 유전인자를 $3 \rightarrow 2$ 로의 한점 돌연변이하면,

$$P = (2 \ 3 \ 2 \ 3 \ 1 \ 1 \ 2 \ 3 \ 1)$$

$$C = (2 \ 3 \ 2 \ 2 \ 1 \ 1 \ 2 \ 3 \ 1)$$

와 같이 자손 C 가 생성된다. 이 경우에는 세 번째 기계에 할당되어 있던 4번 작업이 부모의 형질과 전혀 다른 두 번째 기계에 할당되는 것을 의미한다. 이와 같이 한점 돌연변이를 사용하면 기존의 국소최적점을 벗어나 전혀 다른 국소최적점으로의 탐색이 가능하여 마침내 광역최적점에 도달 가능할 것으로 판단된다.

4.4 유전조정변수(Genetic parameter)값 결정

앞서 기술한 바와 같이 제시된 hybrid 유전알고리즘을 실제 문제에 적용하기 위해서는 적합한 유전 조정변수 값을 알아야 한다. 따라서, 여기서는 조정변수인 T, N_p, α 및 β 값의 변화가 알고리즘 성능에 미치는 영향을 분석하였다.

본 예제의 경우 앞의 3.3절과 동일하게 준비시간은 0에서 100분 사이의 균등분포로부터, 가공시간은 평균 100분, 표준편차 30분의 정규분포로부터 임의로 발생시킨 값을 사용하였다. 또한, 이렇게 생성된 문제에 대하여 제안된 유전알고리즘을 적용할 때마다 난수를 이용하기 때문에 최종적으로 구해지는 해가 달라지게 된다. 그러므로 결과분석은 실험의 신뢰도를 높이기 위하여 각 경우별로 25회 반복 실험한 결과를 사용하였다.

4.4.1 종료조건의 변화

종료조건(T)의 변화에 따른 알고리즘의 성능추이를 알아보기 위하여 $N_p = 20$ 일 때 $(\alpha, \beta) = (0.9, 0.9), (0.5, 0.5), (0.5, 0.1), (0.1, 0.1)$ 의 각 경우에 대하여 실험을 하였다. 여기서, $(0.1, 0.5)$ 등의 경우를 실험에서 제외한 이유는, 예비실험을 통하여 성능 추이를 살펴본 결과 $\alpha < \beta$ 인 경우에는 대체적으로 종료조건 변화에 큰 의미가 없는 것으로 해석되었기 때문이다. 실험결과, 각 10회의 진화마다 구해진 25개 해의 평균을 도시한 <그림 3>에서 보는 것처럼 T 값이 300 이상이면 해의 개선이 완만해지고 T 값이 900 이상이면 해가 거의 개선되지 않고 수렴하게 됨을 알 수 있다. 또한, α, β 의 값이 클수록 수렴속도가 빨라지는 특성을 나타내고 있다. 특히, $(\alpha, \beta) = (0.9, 0.9)$ 의 경우에는 $T = 150$ 정도에서 수렴치에 거의 도달하는 것을 확인할 수 있다.

4.4.2 교차 및 돌연변이율의 변화

여기서는 교차변이율(α)과 돌연변이율(β)의 변화에 따른 민감도 분석을 위하여, $N_p = 20, T = 1000$ 의 상태에서 α 와 β 를 각각 0.1, 0.3, 0.5, 0.7, 0.9와 0, 0.25, 0.5, 0.75, 1.0의 5수준으로

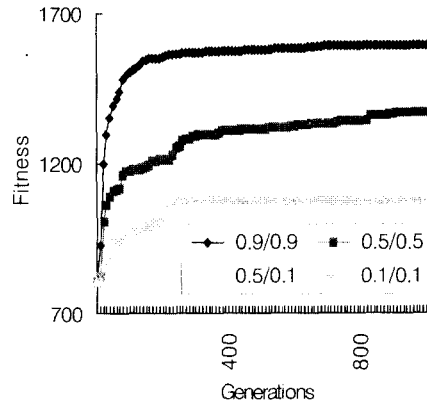


그림 3. T 값의 변화에 따른 유전알고리즘 해의 적합도 분포

조합하여 실험하였다.

실험결과, 적합도의 최대값이나 모집단 평균값 모두 유의수준 1%에서 수준간 유의차 및 교호작용(interaction)이 있었다. 또한, <그림 4>와 <그림 5>에서 보듯이 교차변이율은 $0.7 \leq \alpha \leq 0.9$ 에서 우수한 성능을 나타내었으며, 돌연변이율은 1.0에서 교차변이 수준과 관계없이 항상 좋은 결과를 갖는 것으로 나타났다. 이는 돌연변이한 개체를 교차변이한 개체와 서로 비교하여, 우수한 것만 유전시키고자 하는 전략에서 기인한 것이다. 또한 교차변이율이 0.9 이하가 되는 것은 이전 세대의 우수한 개체를 일부분 다음 세대에 유전하는 것이 바람직하기 때문이다.

α 와 β 값의 변화에 대한 전체적인 경향을 보면 β 값이 증가함에 따라 최대값과 모집단 평균값이 모두 단조증가하고 있다. 반면에 최대값은 α 값이 증가함에 따라 증가하나 0.7 이상이 되면 유의차가 없고, 모집단 평균값은 오히려 대폭 감소하고 있음을 알 수 있다. 최대값이 $\alpha \geq 0.7$ 에서 모집단 최대값이 우수한 것은 elitist 전략 때문이고, 모집단 평균값이 감소되는 것은 α 가 너무 크면(예, $\alpha = 0.9$) 부모의 우수한 유전형질을 유지할 수가 없기 때문인 것으로 풀이된다.

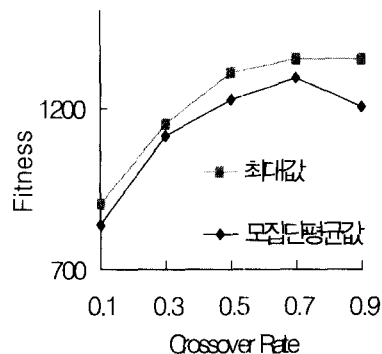


그림 4. α 의 변화에 따른 유전알고리즘 해의 적합도 분포.

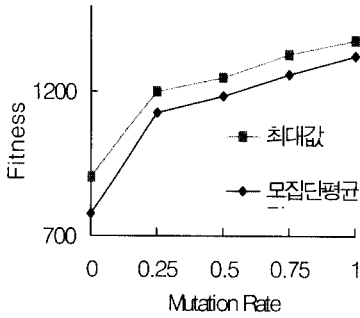


그림 5. β 의 변화에 따른 유전알고리즘 해의 적합도 분포.

4.4.3 모집단의 크기 변화

여기서는 모집단의 크기(N_p) 변화가 알고리즘의 성능에 미치는 영향을 파악하기 위하여, $T = 1000$, $0.7 \leq \alpha \leq 0.9$, $\beta = 1.0$ 으로 고정하고 N_p 값을 20, 30, 40, 50, 60의 5수준으로 변화하며 실험하였다. 실험결과, <그림 6>에서 보는 바와 같이 본 문제의 경우 모집단의 크기가 증가할수록 해의 적합도 역시 향상되었다. 이는 보다 많은 탐색공간을 가지게 되어 광역최적해에 도달할 기회가 증가되기 때문으로 해석된다. 그러나, 모집단의 크기가 증가함에 따른 전체 알고리즘 수행시간 증가를 고려할 때 유의수준 1%에서 유의차가 없으면서 상대적으로 작은 크기인 30개 정도를 유지하는 것이 바람직하다고 판단된다.

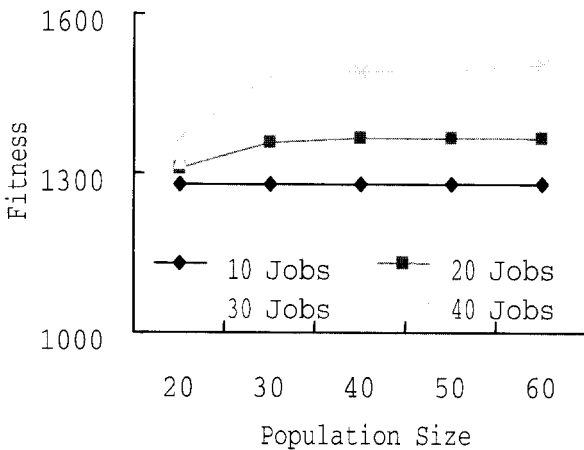


그림 6. N_p 값의 변화에 따른 유전알고리즘 해의 적합도 분포.

5. 유전알고리즘 성능평가

앞 절에서 기술한 절차를 통해 결정된 유전 조정변수값을 이용한 유전알고리즘의 PSPSPT에 대한 성능을 평가하기 위해서

는 평가대상 문제들이 확보되어야 한다. 그러나, 본 연구과제는 기존 연구실적이 전무하기 때문에 기존의 문헌으로부터 평가에 기준이 되는 문제들을 찾을 수 없다. 따라서, 앞의 3절에서와 같이 임의로 생성한 자료를 활용하였으며 통계적 신뢰도를 높이기 위하여 각 자료별로 25회 반복 실험하였다.

5.1 소규모 문제에서의 성능평가

소규모 문제에 대해서는 2장에서 소개한 최적해법을 통해 얻어진 최적해를 유전알고리즘을 적용하여 구한 해와 비교·분석하였다. 평가기준은 3.3절에 정의된 것과 마찬가지로 다음과 같이 정의되는 MAD를 사용하였다.

$$MAD(\%) = \frac{\text{최적해} - \text{유전알고리즘의 해}}{\text{최적해}} \times 100$$

작업수 6, 9, 12에 대하여 각각 25개의 문제로부터 도출된 결과가 <표 2>에 나타나 있다. 이 표에서 Best, Mean, Worst는 각각 문제당 도출된 25개의 해 중에서, 가장 우수한 적합도값, 평균 적합도 값, 가장 열등한 적합도값을 의미한다. 여기서 유전알고리즘의 해는 당연히 가장 우수한 적합도를 나타내는 것이 된다. 이로부터 MAD 값은 최대작업 수 $n=12$ 일 경우에도 0.20%에 불과하여 앞 절 3.3에 기술된 발견적 기법인 MSM과 MPM과 비교할 때 상대적으로 매우 우수함을 알 수 있다. 또한, 3가지 작업 수를 모두 고려한 유전알고리즘 해의 절대평균오차도 최대 4.03%, 최소 0%, 평균 0.12%로 좋은 결과를 나타내었다. 특히 주목할 점은, 총 75개의 문제 중 최적해를 찾지 못한 것이 7회에 불과하였으나, 1대의 기계에 할당될 수 있는 작업의 수가 증가함에 따라 최적해를 생성해 내지 못할 가능성이 증가하였다. 이는 유전알고리즘의 국소최적화 단계에서 사용된 도구가 최적해를 보장 못하는 것에 원인이 있는 것으로 생각된다.

그러나, <표 3>에 나타난 바와 같이 최적해법과 유전알고리즘간의 평균 CPU 처리시간(IBM PC 166MHz, 32M RAM)을 비교하여 보면, 유전알고리즘이 해를 구하는 데 있어 보다 효율적임을 알 수 있다. 이 표에서 평균 CPU 처리시간은 IDPMS로 구한 최적해의 경우, 한 문제의 해를 구하는 데 소요된 시간으로서, 총 25개 문제를 통해 얻어진 결과의 평균값을 나타낸다. 유전알고리즘의 해의 경우는 해 발생의 임의성을 고려하여 각각의 문제별로 역시 25회씩 반복 수행된 결과를 평균한 것이다. 여기서 First Appearance Time과 Total Execution Time은 각각 유전알고리즘의 최종해와 동일한 적합도를 갖는 해가 최초로 도출된 시간, 최종해를 구하는 데 소요된 총 CPU 시간을 의미한다.

5.2 자중규모 이상 문제에서의 성능평가

일반 산업현장에서 해결해야 할 중규모 이상의 문제에 대해서는 타당한 시간 내에 최적해를 구하기가 어렵기 때문에, 본 연구에서는 3절에 기술된 발견적해법을 이용하여 간접적인 성능비교 방법을 사용하였다. 이를 위하여, 앞의 두 발견적해법 중 보다 우수한 것을 해의 하한(lower bound)인 최저해 (= max(NSM, MPM))로 하고, 이 최저해 대비 유전알고리즘 해의 개선비율(improvement ratio: IR)을 평가지수로 하였다.

$$IR(\%) = \frac{\text{유전알고리즘해} - \text{최저해}}{\text{최저해}} \times 100$$

실험방법은 작업 수 20, 30, 40에 대해서 각각 25개씩의 문제를 생성시켜 사용하였다. 표 4에서 보는 바와 같이, 유전알고리즘의 해는 n=40인 경우 최저해 대비 최대 60.9%, 평균 13.5%, 최저 0% 내외에 분포하는 우수한 성능을 보이고 있다. 표에서의 IR값은 각 25개 문제의 최종해에 대한 IR값의 최대값, 평균값, 최소값을 의미한다. 유전알고리즘의 평균 수행시간은 약 25.53분이었으며, 이중 초기 가능해 발생에 사용되는 시간이 40% 이상 소요되었다. 그러나, 앞에서 제시한 발견적 기법을 통해 도출된 가능해를 이용하여 유전알고리즘의 초기해를 생성하였을 때에는, 도출된 해의 적합도에는 차이가 없었으나 평균 수행시간이 13.46분으로 훨씬

효율적인 것으로 나타났다.

6. 결론

본 연구에서는 순서 의존적인 작업시간을 갖는 작업들을 병렬의 동일기계에서 처리할 때, 잔여 기계가용시간을 최대화하도록 작업들을 스케줄하는 문제를 다루었다. 우선 이 문제에 대한 하나의 최저해를 생성하는 발견적 기법을 제시하여 평가 기준을 삼을 수 있도록 하였다. 다음으로, 보다 효율적이고 효과적인 해를 구하기 위하여, 본 문제의 특성에 적합하도록 해를 염색체로 표현함으로써 수정·개발된 유전연산자를 활용하고 Memetic 알고리즘에 기초한 유전알고리즘을 개발하였다. 또한, 컴퓨터 실험을 통하여 PSPSPT에 적합한 모집단의 크기, 종료조건인 최대 세대수, 교차 및 돌연변이율과 같은 유전 조정변수값을 결정하였다. 최적화된 유전알고리즘을 수행하여 작은 규모의 문제에 대해서는 최적해를 기준으로, 중규모 이상의 문제는 최저해를 기준으로 성능을 평가하였다. 실험결과, 본 연구에서 개발된 유전알고리즘은 상대적으로 효율적이고 향상된 성능을 나타내기 때문에 본 문제에서 설정된 상황에 부합하는 실제 현장에서 효과적으로 사용될 수 있을 것으로 판단된다.

표 2. 동적계획법과 유전알고리즘에 의한 평균 목적함수의 값과 MAD

No. of Jobs	IDPMS	GA			
		Best	Mean	Worst	MAD(%)
6	1157.21	1157.21	1157.21	1157.21	0.00
9	1262.97	1260.98	1247.95	1219.46	0.17
12	1471.59	1468.87	1428.05	1364.85	0.20

표 3. 평균 CPU 처리시간의 비교 (단위: sec)

No. of Jobs	IDPMS	GA	
		First Appearance Time	Total Execution Time
6	0.06	0.14	14.34
9	5.71	2.86	24.34
12	17.53	7.98	36.60

표 4. 중규모 이상 문제에서의 발견적 해법 대비 유전알고리즘 성능 비교

No. of Jobs	Heuristic Solutions			Solutions of GA						
	NSM	MPM	최저해	최종해	평균	Worst	IR(%)			
							최대	평균	최소	
20	1326.4	1358.5	1371.9	1543.9	1418.4	1288.3	29.4	13.1	2.3	
30	1164.6	1135.8	1198.2	1342.5	1190.3	1040.6	44.5	12.9	0.2	
40	1384.7	1361.3	1425.6	1597.8	1486.1	1330.7	60.9	13.5	0	

참고문헌

- 장상구 (1991), 다품종 소량 생산하의 작업순서 결정기준에 관한 연구, *산업관리공학과 석사학위 논문*, 울산대학교 산업경영.
- 김갑환, 김기영, 고창성 (1997), Loading Scheduling Using a Genetic Algorithm in Port Container Terminals, *대한산업공학회지*, 23(4), 645-660.
- 김여근, 이수연, 김용주 (1997), 혼합모델 조립라인에서 작업부하의 평활화를 위한 유전알고리즘, *대한산업공학회지*, 23(3), 515-532.
- 김여근, 현철주 (1994), 혼합모델 조립라인의 생산순서 결정을 위한 유전알고리즘, *대한산업공학회지*, 20(3), 15-34.
- 안상형 (1995), 작업준비시간을 고려한 총작업완료시간 최소화 해법에 관한 연구, *경영과학*, 12(2), 77-87.
- 임진규, 박주철 (1996), 절단압착공정의 기계별 작업순서 결정방법개선에 관한 연구, *산업공학*, 9(2), 83-94.
- 이문규, 이승주 (1998), 순서의존적인 작업시간을 갖는 작업들의 스케줄링을 위한 동적계획법, *대한산업공학회지*, 24(3), 431-446.
- Han, Y. H. (1997), Genetic algorithms with a permutation approach to the parallel machines scheduling problem, Dynamic programming approach to sequencing problems, *경영과학*, 14(2), 47-61.
- Baker, J. E. (1987), Reducing bias and inefficiency in the selection algorithm, *Proceedings of the 2nd International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, New Jersey.
- Cheng, R. and Gen, M. (1996), Parallel machine scheduling problems using memetic algorithms, *Proceedings of the 20th International Conference on Computers and Industrial Engineering*, Kyongju, 1, 17-20.
- Croce, F. D., Roberto, R. and Volta, G. (1995), A Genetic algorithm for the job shop problem, *Computers and Operational Research*, 22(1), 15-24.
- Davis, L. (1985), Applying adaptive algorithms to epistatic domains, *Proceedings of the International Joint Conference on Artificial Intelligence*, 162-164.
- Fogel, D. B. (1988), An evolutionary approach to the traveling salesman problem, *Biological Cybernetics*, 60, 139-144.
- França, M. P., Gendreau, M., Laporte, G. and Müller, F. M. (1996), A tabu search heuristic for the multiprocessor scheduling problem with sequence dependent setup times, *International Journal of Production Economics*, 43, 79-89.
- Frederickson, G., Hecht, M. S. and Kim, C. E. (1978), Approximation Algorithm for some routing problems, *SIAM*, 7, 178-193.
- Grefenstette, J. J. (1986), Optimization of control parameters for genetic algorithms, *IEEE Transactions on system, Man, and Cybernetics*, 16(1), 122-128.
- Guinet, A. (1991), Textile production systems: a succession of non-identical parallel processor shops, *J. of Opl. Res. Soc.*, 42, 655-671.
- Holland, J. H. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.
- Lim, J. M. (1996), A genetic algorithm for a single hoist scheduling in the printed-circuit-board electroplating line, *Proceedings of the 20th International Conference on Computers & Industrial Engineering*, Kyongju, 317-320.
- Moscato, P. and Norman M. (1992), A memetic approach for the traveling salesman problem: Implementation of a computational ecology for combinatorial optimization on message-passing systems, *Proceedings of the International Conference on Parallel Computing and Transportation Applications*, Amsterdam.
- Ovacik, I. M. and Uzsoy, R. (1993), Worst-case error bounds for parallel machine scheduling problems with bounded sequence-dependent setup times, *Operations Research Letters*, 14, 251-256.
- Parker, R. G., Deane, R. H. and Holmes, R. A. (1977), On the use of a vehicle routing algorithm for the parallel processor problems with sequence dependent changeover costs, *AIIE Trans*, 9(12), 155-160.
- Schutten, J. M. J. (1996), List scheduling revisited, *Operations Research Letters*, 18, 165-170.
- Serafini, P. and Speranza, M. G. (1992), Production scheduling problems in a textile industry, *European Journal of Operations Research*, 58, 173-190.
- Sumichrast, R. and Baker, J. R. (1987), An integer linear programming based heuristic for minimizing setup time, *International Journal of Production Research*, 25(5), 761-771.
- Whitley, D., Starkweather T. and Fuquay, D. (1989), Scheduling problems and traveling salesmen: The genetic edge recombination operator, *Proceedings of the Third International Conference on Genetic Algorithms*, 133-140.