HWAHAK KONGHAK Vol. 33, No. 6, December, 1995, pp. 734-746
(Journal of the Korean Institute of Chemical Engineers)

유전 알고리듬을 이용한 Earliness와 Tardiness Penalty를 가진 다품종 회분식공정의 생산계획

이창형* · 정재학† · 이인범**

*삼성엔지니어링주식회사
영남대학교 공과대학 화학공학과
**포항공과대학교 화학공학과
(1995년 3월 23일 접수, 1995년 7월 21일 채택)

Scheduling of Multi-product Batch Processes with Earliness and Tardiness Penalties Using Genetic Algorithm

Chang Hyung Lee*, Jae Hak Jung† and In-Beum Lee**

**Samsung Engineering Co., LTD*
Dept. of Chem. Eng., College of Eng., Yeungnam University
***Dept. of Chem. Eng., Pohang Univ. of Science and Technology*
(Received 23 March 1995; accepted 21 July 1995)

요 약

고객에 대한 서비스와 재고비용절감 등 제품생산에서 발생하는 간접적 생산성에 관한 요소들이 특히 다단 회분공정의 다품종 소량생산체제에서의 생산계획에서 점점 더 중요한 문제로 부각되고 있다. 이러한 관점에서 주문한 제품군의 최적생산순서나 한 제품의 주문량이 커서 여러 회분을 연속적으로 생산하는 Campaign 생산계획에 있어서 납기일 이전에 생산완료되어 발생하는 재고비용(earliness penalty)과 납기일 후에 생산완료되어 물어야 하는 납기일 위반 벌칙비용(tardiness penalty)을 최소화하는 문제가 최근 세계적으로 많은 연구진들에 의해 매우 중요하게 다루어지고 있다. 본 연구는 이러한 earliness 및 tardiness penalty의 최소화를 목적함수로 하는 회분식 다단위 다품종 최적생산계획문제를 위한 유전 알고리듬(Genetic Algorithm)을 개발하였다. 유전 알고리듬은 최근 개발된 첨단 최적화 알고리듬으로 3가지 기본 연산자인 재생(reproduction), 교배(crossover) 및 돌연변이(mutation) 연산자를 가진다. 본 연구는 유전 알고리듬을 회분식 화학공정의 최적 생산계획에 적합하게 개선하기 위해 기본연산자 외에도 교배연산자의 강화, 돌연변이 빈도수 강화 및 Crowding Factor Model과 세대차(generation gap) 등의 확장된 연산자들을 유효 적절히 개발하여 사용하였다. 본 연구를 통해 개발된 유전 알고리듬의 성능평가를 위해 UIS 및 NIS 중간저장탱크 운용방식의 회분식 화학공정의 생산계획문제를 무작위로 만들어 적용해 보았으며, 기존의 방법으로 가장 우수한 성능을 보이고 있는 Ku와 Karimi[16]의 Simulated Annealing(SA)에 비해 매우 향상된 성능을 보임을 입증하였다.

Abstract—Improving customer service and reducing inventory costs become more and more important aspects in the production scheduling of many batch processes. From this point of view, one of the important problems is to determine the optimum production sequence of a list of products or single-product campaigns so as to minimize the total penalty cost with earliness and tardiness where tardiness means any later deliveries

than the due date and earliness is any early production resulting in the inventory cost. In this paper, we present a Genetic Algorithm(GA) for multi-product batch process scheduling problems with minimum cost of earliness and tardiness penalties. For this algorithm, we have improved the three basic operators, reproduction, crossover and mutation. Additionally we have developed the extended operators, so called Crowding Factor Model, Elitist Model and Generation Gap. To evaluate the performance of this study, we have tested various scheduling problems with UIS and NIS policies and the results are compared with Simulate Annealing (SA) method by Ku and Karimi[16]. Finally the GA by this work is outperformed to SA.

1. 서 론

회분식공정은 복잡한 합성 절차를 요구하는 고부가가치 제품을 적은 양으로 비슷한 생산 공정을 통한 다품종 제품의 생산을 요구하는 공정에 적합하다. 이러한 회분식공정은 의약, 화장품, 고분자, 생화학, 전자재료, 식품 등과 같은 특수한 화학제품의 생산에 널리 이용되고 있다. 또한 최근 소비자의 기호가 다양해지고 기술개발 속도가 빨라짐에 따라 다양한 등급을 가진 다양한 아이템의 제품을 라이프사이클이내에 소량으로 생산함으로써 제품의 고부가가치화를 추구하는 다품종 소량 고부가가치 생산 체재로의 전환이 요구되고 있다. 이러한 이유로 다품종 소량생산체제의 필수적인 반연속 및 회분식공정이 다시 화학공업에서 각광을 받기 시작하였다.

이러한 회분식공정은 연속식공정보다 계절적인 수요변동과 같은 변동적인 수요변화에 대처하는 유연성이 뛰어나기 때문에 제품의 생산성과 조업효율을 향상시킬 수 있고, 이에 따라 생산비용을 줄이는 많은 연구활동이 진행 중이다. 이러한 연구분야 중에서 다품종 회분식공정의 생산계획은 가장 중요한 연구분야 중의 하나이다. 지난 10년간 다품종 회분식공정의 생산계획이 활발히 연구되어 왔고 최근 Reklaitis[1, 2], Ku 등[3]에 의해 review paper로 발표되었다.

회분식공정의 생산계획 문제는 공통적으로 목적함수를 최소가 되도록 단위제품 혹은 제품의 campaign의 생산순서와 각 제품별 각 공정에서의 생산시각을 결정하는 것이다. 이러한 생산계획의 대부분의 연구는 목적함수로써 모든 제품의 총 조업완료 시간인 makespan을 최소화하는 문제에 집중되어져 왔다[4-7]. Makespan의 최소화는 각 장치를 최대한 사용하여 전체 제품의 총 생산시간을 최소화한다.

그러나 총 생산시간을 단축시키는 것도 중요하지만 고객서비스 즉 고객주문의 납기일을 맞추는 것이 화학공업에서 보다 더 중요한 고려사항 중의 하나라는 것을 최근의 연구들[8, 9]은 말하고 있다. 또한 재고비용을 감소시키려는 경영활동으로 Just-In-Time(JIT) 생산체제로의 전환이 최근 활발히 진행 중이다[10]. JIT 관점에서 볼 때 납기일(due-date)보다 일찍 생산된 제품은 그들의 납기일까지 재고로 유지되어야 하므로 재고비용을 초래하고 제품이 납기일 이후에 생산되면 고객기업은 제품이 지연되는 시간동안 조업을 중단하는 사태를 유발할 수 있으므로 커다란 벌칙(penalty)을 물어야 할 것이다. 따라서 이상적인 생산계획은 모든 제품이 정확히 그들의 할당된 납기일에 생산을 완료하는 것이 될 것이다.

이러한 중요성의 증가에도 불구하고 납기일 위반에 따른 벌칙과 재고비용을 모두 고려한 생산계획에 관한 연구는 미비한 실정이며 상대적으로 최근의 시작된 연구분야이다. 따라서 재고비용과 납기일 위반을 고려한 earliness와 tardiness penalty로 구성된 생산계획 모형(E/T model)의 목적함수를 최소화하는 효과적인 다품종 회분식공정의 생산계획에 관한 연구의 필요성이 요구된다.

다품종 회분식공정(multi-product batch process)은 일련의 M개의 회분장치로 구성되어 있다. 이 장치들 각각은 다른 회분식장치 혹은 반연속식장치와 연결되어 있다. 이러한 시스템은 중간저장탱크의 수, 장치이용률, 원료공급, 예기치 않은 사고에 대비한 어떤 제품에 대한 최소재고레벨 등의 제약조건을 가질 수 있다. 다품종 회분식공정은 다양한 제품을 생산하지만 모든 제품은 똑같은 처리경로를 따른다. Fig. 1은 이러한 다품종 회분식공정의 예를 보여준다. 각 장치들 사이에는 펌프로써 물질 수송이 일어나고 장치효율을 증가시키기 위한

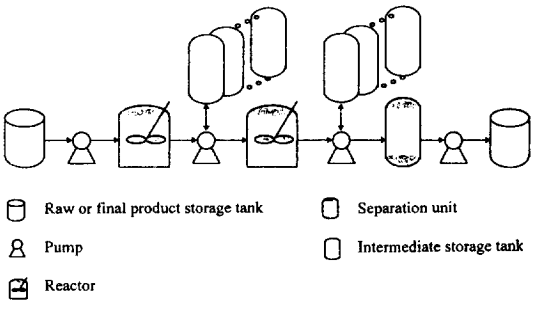

Fig. 1. Multi-product batch process.

중간저장탱크들이 필요에 따라 설치될 수 있다.

다품종 회분식공정이란 N개의 다른 제품이 M개의 장치에서 생산되는 시스템을 말하는데 여기서 각 장치 j에서의 제품 i의 처리 시간(processing time)은 t_{ij}이고 장치 j에서의 준비 시간(setup time)은 이전의 $(i-1)$번째 제품을 고려한 i번째 제품처리를 위한 장치준비 시간으로서 제품순서에 따라 그 시간이 달라지는 $s_{(i-1)ij}$가 되고 장치 j로부터의 제품 i의 수송 시간(transfer time)은 a_{ij}이다. 이러한 조업자료들은 미리 주어진다고 가정하고, 또한 제품 i에 대한 납기시간(due time)인 d_i가 주어진다. 이러한 조업정보에 의해 제품생산계획이 결정되면 조업완료결정 알고리듬(completion time algorithm)에 의해 출하시간(finishing time) F_i가 계산된다. 만일 제품 i가 d_i 이전까지 준비가 되지 않으면 이것을 "tardy" 혹은 "late"라 하며 지연벌칙비용 혹은 tardiness penalty를 유발한다. 만일 제품 i가 d_i 이전에 만들어지면 이를 "early"라고 하며 재고 비용 혹은 earliness penalty를 유발한다. 즉 모든 제품은 주문 납기일인 d_i에 꼭 맞게 조업되는 것이 가장 이상적이라는 가정으로 출발한다.

Garey 등[11]은 처음으로 E/T(earliness/tardiness) 문제가 NP-complete하다는 것을 보여주었다. NP-complete 문제는 문제의 크기가 증가하면 문제를 풀기 위한 시간은 다항식 한계(polynomial bound)를 크게 넘어서 문제의 크기(주로 제품의 수)가 증가함에 따라 search space는 기하급수적 증가 이상으로 증가한다. 비록 컴퓨터의 성능이 많은 향상을 이루었지만 아직까지 NP-complete 문제는 가장 풀기 어려운 문제들 중의 하나로 인식되고 있다.

이러한 E/T 문제를 풀기 위한 방법으로써 Fry 등[12]은 adjacent pairwise interchange의 경험적 방법을 이용하여 E/T문제를 풀었으며 이 경험적 방법은 평균적으로 최적해의 2% 이내의 준 최적해(near optimum solution)를 얻고 있다. Abdul-Razaq와 Potts[13], Fry 등[14]은 branch-and-bound 방법을 사용하여 이러한 문제를 풀었다. 그러나 그들의 결과는 20개 이상의 job에 대하여 지나친 계산 시간이 필요하기 때문에 풀기 어렵다고 제시하였다. 또한 Musier와 Evans[9]가 모든 제품의 penalty 부가 가중치가 동일한 tardiness penalty를 최소화하는 간단한 경험적 방법을 제시하였고 Ku와 Karimi[15]가 tardiness penalty를 최소화하는 heuristic lower bound를 가진 branch and bound 방법을 이용하여 15개의 제품까지 풀이하였다. 또한 Ku와 Karimi[16]는 sequence building, maximum penalty reduction, four-product enumeration의 3가지의 경험적 방법과 Simulated Annealing(SA)의 방법을 이용하여 tardiness penalty를 최소화하는 방법을 제시하였고 이 4가지 방법을 비교하였다. 이 방법 중 SA 방법이 가장 좋은 해를 구하였지만 계산 시간은 가장 많이 걸린다는 단점이 있다. 따라서 본 연구는 E/T 문제에 관해 적절한 시간 이내에 더 우수한 해와 더 많은 제품 수를 풀 수 있는 생산계획 알고리듬으로 유전 알고리듬(Genetic Algorithm)을 적용하였으며 최근까지 가장 성능이 우수하다고 인정되는 SA 알고리듬과 비교 분석하였다.

최근 combinatorial optimization 문제에서 유전 알고리듬이 매우 효과적인 최적화 알고리듬으로 각광받고 있다. Lee 등[17]은 makespan 최소화의 생산계획에서 유전 알고리듬이 SA보다 우수하며 지금까지 가장 효과적이고 강건한 알고리듬이라는 것을 보여 주었다. 유전 알고리듬은 John Holland[18]에 의해 처음으로 제안된 이후 계속해서 발전되어 왔다. 이 알고리듬은 자연도태(natural selection)와 유전학(natural genetics)의 역학에 기초한 최적화 알고리듬이다. 유전 알고리듬은 Goldberg[19, 20]에 의해 잘 정의된 framework를 가지고 있다. 이 framework는 재생(reproduction), 교배(crossover), 돌연변이(mutation)의 3가지 기본 연산자(operator)를 가지고 있다. 재생 연산자는 각 개체의 적합도(fitness)에 비례하여 다음 세대(generation)에 그들의 유전형질을 가진 자손을 재생하는 연산자이다. 교배 연산자는 자손에게 부모의 유전형질을 합성하여 보다 우수한 자손을 생산하는 연산자이다. 돌연변이 연산자는 무작위로 개체의 유전자를 변형시키는 연산자이다. 최적화 알고리듬에서 교배 연산자는 지역탐색(local search)을 수행하고 돌연변이 연산자는 uphill-climb을 가능케 하여 지역탐색으로 인한 국부 최소치를 극복하여 전체 최적해(global optimal solution)를 구할 수 있게 한다.

본 연구에서는 E/T penalty를 가진 다품종 회분식공정의 생산계획에 적합한 유전 알고리듬을 개발하였다. 본 연구에서 개발된 유전 알고리듬은 유전 알고리듬의 기본 연산자외에 회분식 화학공정의 최적 생산계획에 적합한 형태로 보다 발전했으며 이를 위해 확장된 연산자들을 개발, 적용하고 있다. 여기에는 개선된 교배율 및 방법론, 돌연변이 빈도 증가와 Crowding Model의 적용, 적절한 세대차(generation gap) 등을 심도있게 모의실험을 통해 개발하였다. 또한 유전 알고리듬이 다양한 크기의 문제로 구성된 생산계획 시스템에 관하여 범용 최적화 알고리듬으로써의 가능성을 보여주며 이를 위한 framework와 제어변수(control parameter)에 대한 실험적 연구를 통한 일반적인 가이드 값(guide value)을 제시하였다.

2. 최소 earliness 및 tardiness penalty의 생산계획문제

다품종 회분식공정(multi-product batch process)의 생산계획은 크게 3단계의 해법 절차를 가진다. 첫 단계는 생산순서 생성법칙(sequence generation rule)으로서 목적함수를 최소화하는 최적의 생산순서를 결정한다. 두번째 단계는 완료시간 결정 알고리듬(completion time algorithm)으로서 주어진 제품 생산순서에 대하여 각 장치에서의 생산시간과 각 제품의 조업 완료시간을 계산한다. 세번째 단계는 목적함수 평가(objective function evaluation) 단계로서 조업완료시간이 어느 정도로 목적하는 바에 부합하는지를 평가한다. 이 3단계 중에서 가장 핵심적인 부분은 생산순서 생성법칙으로써 대부분의 생산계획 연구가 여기에 집중되어 있다.

본 연구의 목적은 제품의 납기일을 어기지 않고 재고비용을 최소화하는 목적함수에 가장 적합한 제품 생산순서를 결정하는 생산계획 알고리듬을 개발하는 것이다. 생산계획의 이러한 두가지 상반된 목적은 상호 trade-off를 가질 수 있다. 예를 들면 모든 생산의 납기일을 맞추는 것은 큰 재고를 유지하는 것을 의미할 수도 있다. 이것은 재고비용을 상승시키고 재고비용을 최소화하는 목적에 위배된다. 따라서 어떤 생산계획결과의 최적성은 조업 전략을 반영하는 목적함수에 따른다. 즉 다른 목적함수는 다른 최적해를 가지는 것이다.

2-1. 목적함수

본 연구에서 고려되는 목적함수는 2개의 부분으로 구성되어 있다. 즉 재고 비용(inventory cost)과 납기시간 위반 비용(due time violation penalties)으로 구성된다. 이 두 가지를 고려한 일반적인 형태의 비용모형의 목적함수는 다음과 같이 계산될 수 있다.

일정 생산계획 기간 T동안의 플랜트 조업에 대한 생산계획을 고려할 경우 제품 i에 대한 재고 비용은 다음과 같이 계산된다.

$$IC_i = \sum_{t=1}^{T} (\alpha_i Q_{it})$$

여기서

t : time quantum

T : planning period or planning horizon(from start of operation to due-time)

α_i : inventory cost of product i per unit quantity per time quantum

Q_{it} : total amount of inventory of i in time quantum t

IC_i : total inventory cost of i이며,

제품의 주문에 대한 납기시간 위반(due-time violation penalty)의 비용은 다음과 같은 형태를 가진다.

$$DVP_j = FP_j + \beta_j L_j$$

여기서

FP_j : fixed penalty for missing deadline or order j

β_j : late penalty coefficient for order j

L_j : number of time quanta by which j has been delayed or late

DVP_j : deadline violation penalty of order j가 된다.

따라서 총 비용은 상기 두 비용의 총합으로 표시된다.

$$\text{Total Cost} = \sum_{i=1}^{N} IC_i + \sum_{i \in O} DVP_i$$

여기서

O : list of delayed orders이다.

위의 식들에서 비용계수들은 사용자들에 의해 주어지고 실질적인 비용함수는 상황에 따라 다소 변동될 수 있다. 이 값들은 각각의 요소(factor)들에 주어지는 상대적인 중요성을 나타낼 수 있다. 위의 목적함수 관계식은 목적함수 계산을 위하여 상당히 많은 데이터를 필요로 한다. 본 논문에서는 이러한 전체비용을 최소화하는 견지에서 해석될 수 있는 간단한 형태의 earliness와 tardiness(E/T)의 penalty를 가진 생산계획 모형(E/T model)을 목적함수로 이용한다.

만일 제품 i가 d_i까지 준비되지 않으면 이것을 tardiness라 하고 지연된 시간에 따라 기울기 β_i(\$ /h) ($\beta_i > 0$)로 선형으로 증가하는 지연비용 혹은 tardiness penalty를 유발한다. 만일 제품 i가 d_i 이전에 만들어지면 이를 earliness라 하고 저장시간에 따라 α_i(\$ /h)($\alpha_i < 0$)의 기울기로 증가하는 재고비용 혹은 earliness penalty를 유발한다. 다음 Fig. 2에 α_i와, β_i에 의해 부가되는 벌칙비용에 관한 그래프를 나타내었다. 보통 재고비용이 납기일 위반비용에 비해 작은 것이 실제산업에서의 일반적 현상이므로 α_i의 기울기의 절대치가 β_i의 기울기의 절대치보다 작은 것을 Fig. 2에서 나타내고 있다.

Earliness는 재고비용으로 해석될 수 있고 d_i 이전에 생산된 제품 i의 earliness는

$$E_i = \max[0,\ d_i - F_i]$$

이고 제품 i의 납기일 지연에 의한 tardiness는

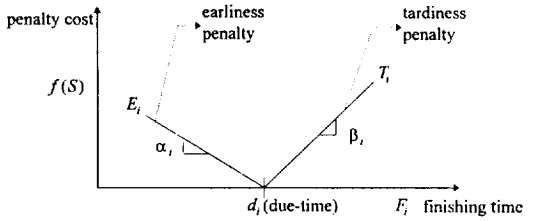

Fig. 2. Scheduling model of objective function with earliness and tardiness penalties.

$T_i = \max[0, F_i - d_i]$으로 계산된다.

따라서 목적 함수는

$f(S) = \sum_{i=1}^{N} (\alpha_i E_i + \beta_i T_i)$와 같이 요약될 수 있다.

여기서 S는 주어진 제품들의 생산순서 중 하나를 나타낸다. 위의 관계식에서 F_i는 실질적 제품 i의 출하시간이며 조업 완료시간 결정알고리듬에 의해 C_{iM}을 구한 후 다음과 같이 구해진다.

$$F_i = C_{iM} + a_{iM}$$

여기서 C_{iM}은 제품 i번째 제품이 마지막 장치 M을 떠나는 시간으로 장치 M에서의 i번째 제품의 조업 완료시간이고 a_{iM}은 제품 i가 장치 M으로부터 수송되어 다음 장치로 수송완료되는 시간이다.

2-2. M-unit 시스템

본 연구의 대상공정은 M개의 회분식 장치가 연속적으로 연결된 다품종 회분식공정이다. 각각의 제품은 원료로부터 최종제품을 만들기 위해 최소 1개의 장치로부터 최대 M개의 장치까지의 처리경로를 거친다. 즉 제품생산 처리경로에 속하지 않는 장치에서의 처리시간, 수송시간, 준비시간은 모두 0으로 둘 수 있다. 이러한 M-unit 시스템에서 조업 완료시간을 계산하기 위한 가정은 다음과 같다[21].

(1) 모든 제품은 각각의 생산 장치에서 똑같은 순서로 생산된다. 이러한 생산계획문제를 순열 스케쥴(permutation schedule)이라고 한다.

(2) 생산 조업은 비선매(non-preemptive)이다. 즉 일단 생산이 시작되면 그 조업은 끝날 때까지 정해진 순서에서 방해받지 않는다.

(3) 하나의 장치는 한번에 하나의 제품만을 생산하고 하나의 제품은 하나 이상의 장치에서 동시에 생산되지 않는다(serial processing).

(4) Set-up 시간은 sequence-dependent하다. 즉 바로 앞의 처리제품이 무엇인가, 또 현재 처리해야 할 제품이 무엇인가에 따라 set-up time은 영향을 받는다.

본 연구에서는 UIS(unlimited intermediate storage) 및 NIS(no intermediate storage) 방안의 문제에서 제품 생산순서를 p_1-p_2-p_3-$\cdots$-p_N이라고 하면 위의 가정들을 만족하는 i번째 제품의 장치 j에서의 조업 완료시간은 UIS방안의 경우

$$C_{ij} = \max[C_{i(j-1)}, C_{(i-1)j} + a_{(i-1)j} + s_{(i-1)ij}] + a_{i(j-1)} + t_{ij}$$

이고 또한 NIS방안의 경우

$$C_{ij} = \max[C_{i(j-1)}, C_{(i-1)j} + s_{(i-1)ij} + a_{i(j-1)}, C_{(i-1)(j+1)} + s_{(i-1)i(j+1)} - t_{ij}] + t_{ij} + a_{ij}$$

경계조건은 두 경우 모두 다 $C_{ij}=0$ for $i \le 0$, $j \le 0$이다. 여기서 제품 p_i는 생산순서의 i번째 순서로 생산된다. 이러한 조업 완료시간 알고리듬은 각 제품의 각 장치에서의 조업 시작시간, 완료시간 뿐만 아니라 총 조업 완료 시간인 makespan까지 계산할 수 있고 또한 이들은 쉽게 Gantt chart로 보여질 수 있다.

(예제 1) N=3, M=2인 간단한 예를 통하여 M-unit 시스템의 생산계획을 보였다. 이 시스템의 조업자료는 Table 1에 나타내었고, E/T 목적함수를 최소화시키는 최적의 생산순서는 1-2-3이며 이 때의 E/T penalty는 151이다. 이러한 생산순서가 결정될 때 각 장치에서의 조업 완료시간을 보여주는 Gantt chart를 Fig. 3에 나타내었다.

2-3. Campaign Scheduling

M-unit 시스템의 문제에서 모든 제품은 1회분으로 그 주문량이 충족된다는 가정이 고려되었지만 일반적인 경우 주문량은 1회분 이상의 경우가 대부분이다. 이러한 경우는 한 제품의 주문량 완료는 동일회분의 campaign

Table 1. Data for example 1

N	Processing times		Transfer times			Cost coeff.		Due time	Setup times					
	t_{i1}	t_{i2}	a_{i0}	a_{i1}	a_{i2}	α_i	β_i	d_i	s_{i11}	s_{i21}	s_{i31}	s_{i12}	s_{i22}	s_{i32}
1	12	29	5	2	4	1	3	54	0	4	2	0	4	3
2	8	11	1	1	3	2	4	70	5	0	3	5	0	2
3	28	16	2	5	4	2	5	69	2	4	0	3	2	0

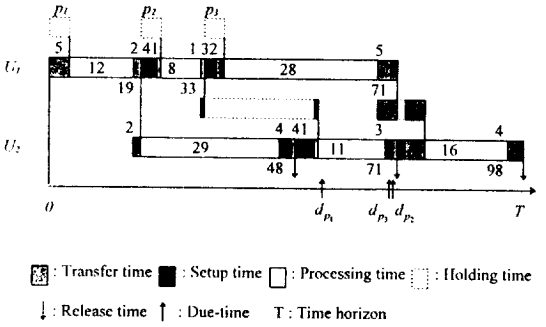

Fig. 3. Gantt chart for the optimal sequence of example 1.

의 견지에서 고려될 수 있다. 하나의 campaign은 일정 기간 일련의 동일제품을 생산하는 Single-Product-Campaign(SPC) 혹은 다른 제품들이 일정한 반복규칙의 생산순서로 생산되는 Multiple-Product-Campaign(MPC)로 구분한다. 고객의 주문량이 많고 따라서 일련의 동일회분을 생산함으로써 주문량을 맞출 수 있는 경우 campaign생산은 일반적이다. 또한 campaign생산은 하나의 플랜트에 의해 생산되는 제품들이 여러 개의 호환되는 제품들의 그룹으로 나누어지는 경우에도 가능하다. 즉 이러한 그룹들은 같은 그룹 내에서는 하나의 제품에서 다른 제품으로 전환될 때 비싼 change-over cost /time을 요구하지 않는 호환되는 혹은 동일한 제품들로 구성된다. 따라서 호환성있는 제품들을 연속적으로 생산하는 것은 set-up과 change-over 비용을 최소화하므로 많은 상황에서 바람직하다. 본 연구에서는 campaign scheduling 문제의 경우 SPC조업만을 고려하였으며 또한 UIS 방안의 경우를 가정하였다. 이러한 campaign scheduling을 위한 가정은 앞서 보았던 M-unit 시스템의 경우와 같고 다만 다음의 가정 5항이 추가된다.

(5) 한 campaign은 한 제품으로만 구성되며(SPC), 동일제품 campaign 내에서의 각 장치 들간의 set-up time은 0으로 가정한다.

위의 가정을 만족하는 campaign 생산계획의 조업 완료시간 알고리듬은 다음과 같다.

$$C_{ijk}=\max[C_{i(j-1)k},\ C_{ij(k-1)}+\delta_k a_{ij}]+a_{i(j-1)}+t_{ij}$$

여기서 $\delta_k=\begin{cases}0 & k=1\\ 1 & k>1\end{cases}$

경계조건 : $C_{ijk}=0$ for $i\le 0,\ j\le 0$

$$C_{ij0}=\max[C_{i(j-1)0},\ C_{(i-1)jn(i-1)}+a_{(i-1)j}+s_{(i-1)j}]$$
for $k=0$

여기서 i(i=1,2,⋯,N)는 campaign을 이루는 제품을 나타내고 k(k=1,2,⋯,n_i)는 campaign i의 몇 번째 회분인가를 나타내고 j(j=1,2,⋯,M)는 장치를 나타낸다. 경계조건은 campaign i가 장치 j에서 시작 가능한 최소의 시간을 결정한다.

Campaign i가 unit M에서의 실제적 조업이 완료되는 시간은 $C_{in_iM}+a_{p_iM}$이다. 그리고 모든 N개의 campaign의 생산을 완료하는 총 시간인 makespan은 $C_{Nn_NM}+a_{NM}$이다.

SPC 생산계획은 각 제품이 campaign의 생산순서를 결정하는 것으로서 M-unit 시스템의 생산계획의 경우와 같으며 단지 조업 완료시간 알고리듬의 계산시간만 더 길어질 뿐이다.

(예제 2) 순차적 다중 unit을 갖는 회분식공정의 1회 최대생산 능력이 100일 때 고객들이 2개의 제품 A, B를 각각 200, 300을 납기시간까지 주문한 경우 SPC 생산계획의 예를 보이기 위한 조업자료를 Table 2에 나타내었고 이 때의 E/T penalty를 최소화하는 campaign 순서는 먼저 A를 2번 생산하고 B를 3번 연속적으로 생산하는 것이었다. 목적함수 값은 34이며 makespan은 58이다. 이러한 SPC 생산 순서가 결정될 때 각 장치에서의 조업 완료시간을 보여주는 gantt chart를 Fig. 4에 나타내었다.

3. 생산계획을 위한 유전 알고리듬

J. Holland에 의해 개발된 유전 알고리듬은 자연도태와 유전학의 역학에 기초를 두고 자연계의 적응과 진화를 인공적으로 모델링하는 아주 효율적이고 강건한 탐색기법이다. 자연도태는 Darwin의 적자생존(The survival of the fittest)에 의해 설명될 수 있고 유전학은 유전자 재조합(gene recombination)과 돌연변이(mutation)에 의해 설명될 수 있다. 유전 알고리듬은 인공

Table 2. Data for example 2

Customer	Product	Quantity	Batches	α_i	β_i	d_i	Processing times		Setup times				Transfer times		
							t_{i1k}	t_{i2k}	s_{iA1}	s_{iB1}	s_{iA2}	s_{iB2}	a_{i0k}	a_{i1k}	a_{i2k}
1	A	200	2	2	3	40	5	7	0	3	0	2	2	1	2
2	B	300	3	1	2	54	8	4	1	0	2	0	1	2	2

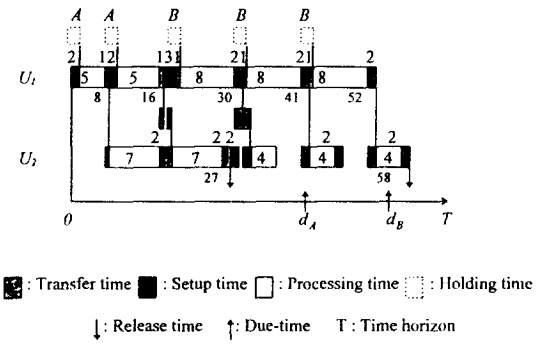

Fig. 4. Gantt chart for the optimal sequence of example 2.

개체의 집단 속에서 적자의 생존을 집행하며 매 세대마다 새로운 개체의 집단을 부모집단의 유전자 재조합과 돌연변이에 의해 재생한다.

유전 알고리듬은 확률적인 기원의 결과로부터 기인하는 전체 탐색기법이기 때문에 그들은 전통적인 최적화 알고리듬의 큰 단점 중의 하나인 국부탐색을 극복할 수 있다. 그러나 이것은 단순한 무작위 탐색(random search)에 의한 국부탐색의 극복이 아니고 높은 성능향상 확률을 가지고 탐색공간의 전체영역을 탐색하기 위해 빨리 효과적으로 과거의 정보를 활용한다. 유전 알고리듬 개발을 추진하는 철학은 다른 최적화 환경에서도 좋은 성능을 보이기 위해 필요한 효율성(efficiency)과 유효성(efficacy)사이의 균형을 나타내는 강건성(robustness)에 대한 필요성 때문이다. 인공시스템에서 강건성은 재설계의 필요성을 줄일 수 있는 장점이 있다. 자연계의 생물학적 유전과정에서의 이러한 강건한 설계절차(genetic action+survival of the fittest)는 자율치유(self-repair), 자율항해(self-guidance) 그리고 재생 그 자체와 같은 그들의 생존을 보장하는 수많은 특성을 가진 많은 생물체를 창조하여 왔다. 자연계의 그러한 특성은 인공 시스템에서는 심지어 가장 정교한 시스템일지라도 거의 존재하지 않는다. 자연계의 강건성과 적응을 모방하는 유전 알고리듬은 복잡한 공간들에서 강건한 탐색을 제공한다는 것을 J. Holland는 "Schema Theorem"에 의해 이론적으로 증명하였다[18].

현재 유전 알고리듬은 게임이론에서부터 기계설계까지 다양한 응용에서 널리 사용되고 있다. 실 세계의 과학적인 문제들과 공학적인 문제들에 대한 타당성은 많은 응용 예들에 의해 실험적으로 입증되었다. 대표적인 예는 Goldberg[19]에 의해 수행된 natural gas pipeline optimization이다. 이와 같은 특성을 가진 유전 알고

Table 3. The major differences of GA and conventional optimization algorithm

	Genetic algorithm	Conventional algorithm
Parameter	discrete	continuous
Search method	parallel	serial
Search direction	problaistic transition	deterministic gradient

리듬과 전통적 최적화 알고리듬의 차이점을 Table 3에 요약하였다. 즉 유전 알고리듬은 유전형질의 on-off특성을 다루므로 이산 시스템을 다루며 동시에 여러 탐색공간을 찾는 병렬탐색(parallel search)법이며 탐색방향설정은 높은 확률적 비확정성에 근거를 둔다.

유전 알고리듬은 이진수 혹은 십진수로 구성된 고정된 길이의 스트링을 취급하기 때문에 변수의 연속성과 목적함수에 대한 미분가능성을 필요로 하지 않는다. 많은 최적화 기법들은 negative gradient direction($-\nabla f$)과 같은 탐색방향 결정 규칙을 사용하면서 하나의 위치에서 다음 위치로 이동한다. 이러한 직렬 탐색은 여러 개의 피크를 가진 탐색공간(multimodal search space)에서 잘못된 피크를 찾을 수 있기 때문에 종종 전체 최적해를 찾지 못한다. 그러나 유전 알고리듬은 여러 개의 피크를 동시에 찾기 위하여 병렬탐색을 하기 때문에 잘못된 피크를 찾을 확률을 낮춘다. 이러한 유전 알고리듬의 병렬탐색은 최적해에 대한 발견속도를 증가시키고 복잡한 탐색공간에서 지역 최적해에서 벗어나는 것을 도와준다. 유전 알고리듬의 이러한 개념들은 재생, 교배 그리고 돌연변이의 3개의 연산자에 의해 수행된다.

3-1. 생산계획을 위한 유전 알고리듬의 스트링 설계

유전 알고리듬의 개체는 유전정보를 가지고 있는 염색체 혹은 스트링과 그 스트링의 적합도로 구성되어 있다. 보통의 유전 알고리듬의 스트링은 이진수로 구성되어 있지만 생산계획에서는 십진수의 양의정수로 구성된다. 이 십진수는 제품 명을 나타내고 스트링에서 십진수의 위치는 생산순서를 나타낸다. 그리고 스트링의 적합도는 스트링에 표현된 순서로 제품을 생산할 때의 목적함수 값을 나타낸다. 예를 들면 각 제품의 생산순서는 p_2-p_4-p_1-p_5-p_3이고 이 때의 목적 함수 값이 10이면 이를 표현하는 스트링의 구성은 [2 4 1 5 3]이고 이 스트링의 적합도는 10이다. 본 연구에서 사용한 목적함수는 즉 적합도 총 비용(total penalty cost)으로서 이 적합도를 최소화하는 문제를 설계하였다.

3-2. 재생 연산자

재생 연산자(reproduction operator)는 자연도태를 인공적으로 구현한 것으로써 새로운 개체를 재생하기 위해 부모집단으로부터 적자 생존법칙에 따라 부모를 선택한다. 지금까지 많은 재생 연산자가 개발되었지만 본 연구에서는 비대체 추계론적 잔류선택(stochastic remainder selection without replacement) 재생 연산자를 사용한다. 이 연산자는 각 개체의 기대값(expected value)의 정수부분만큼 많은 수의 개체를 교배장(mating pool)으로 복사한다. 남아있는 집단의 부분은 그들의 기대값의 소수부의 확률로써 채워진다. 그들은 성공할 확률로서 그들의 기대값의 소수부를 이용하면서 가중치를 가진 동전 던지기의 수행에 의해 선택된다. 재생 연산자에서는 선택될 부모개체의 수에 대한 조정이 특히 중요하다. 만일 전형적인 선택규칙을 따른다면 매우 우수한 적합도를 가진 개체는 한 세대의 유한집단에서 큰 비중을 차지할 것이다. 이것은 조속한 수렴(premature convergence)의 결과를 가져올 수 있고, 또한 평균 적합도가 최대 적합도에 가까울 경우 각 개체의 적합도에 대한 무차별화(indiscrimination)의 가능성을 유발할 수 있다. 이러한 상황은 대부분의 문제에서 일반적으로 발생한다. 이러한 이유 때문에 비슷한 크기의 적합도를 가진 개체들을 차별화하기 위해 적합도 스케일링(scaling)이 필요하다.

비록 선형 스케일링이 대부분의 유전 알고리듬에서 통상적으로 사용되지만 그것은 극소수의 열등 개체가 집단의 평균 적합도보다 상당히 작을 경우 음수의 적합도를 갖는 스케일된 개체가 후손 세대에 나타나는 문제점을 가지고 있다. 이러한 선형 스케일링의 단점을 극복하기 위해 본 연구에서는

$$f' = \frac{2}{1+\exp[-\lambda x]}$$

와 같은 S자형의 스케일링(sigmoid scaling)을 개발했다. Sigmoid scaling의 x값은 편차 적합도(deviation fitness)이다. 편차 적합도는 적합도에서 평균 적합도를 뺀 값으로 $x=(f-f_{avg})$이다. 그리고 sigmoid scaling의 f'은 스케일된 적합도이다. 그리고 평균 스케일된 적합도의 크기를 1로 함으로써 스케일된 적합도를 각 개체의 기대값으로 전환할 수 있다. 또한 스케일 함수의 기울기를 조절하기 위해 λ를 조절할 수 있고, 이것이 결국 최종 알고리듬성능에도 영향을 줄 수 있다. 유전 알고리듬에서 기대값의 의미는 어떤 개체가 교배를 하기 위해 선택될 수 있는 횟수이다. 즉 만일 어떤 개체의 적합도가 f=30이고 집단의 평균 적합도는 $f_{avg}=20$이라고 하면 어떤 개체의 기대값은 $E(f)=f/f_{avg}=1.5$ 이다. 따라서 그 개체는 다음 세대를 생산하기 위해 한번 선택되어 교배를 수행하고 또 다시 한번 더 선택될 확률이 0.5가 된다는 뜻이다. 이러한 표현의 장점은 양수와 음수의 적합도를 모두 사용할 수 있다는 것이다. 또한 전통적인 최적화 방법에서와 같이 목적함수의 부호를 바꿈으로써 쉽게 최소화 문제를 최대화 문제로 전환시킬 수 있다.

3-3. 교배 연산자

교배 연산자(crossover operator)는 부모보다 더 나은 후손을 생산하기 위해 부모의 유전정보를 부분적으로 교환하는 역할을 한다. 유전 알고리듬은 부모의 유전정보를 표현하기 위해 주로 이진수로 구성된 스트링을 사용한다. 그리고 유전정보를 교환하기 위해 무작위로 하나의 절단점(cutting point)을 선택하여 절단점 이전은 부(父)의 유전자와 절단점 이후는 모(母)의 유전자를 합성하여 자(子)의 유전자를 생산한다. 그러나 회분식 생산계획은 십진수의 정수로 이루어진 순열(permutation) 스트링 표현이 필요하다. 이러한 표현을 위해 Goldberg와 Lingle은 PMX(Partially Matched Crossover)를, Davis는 OX(Order Crossover)를 각각 개발했다[17].

PMX와 OX의 유전자는 십진수의 정수로 구성되어 있고 각각의 유전자는 서로 중복됨이 없어야 한다. 그리고 십진수로 구성된 유전인자의 값은 생산계획 문제에서 생산할 제품이 된다. 그리고 스트링에서 각 유전인자의 위치는 제품의 생산순서가 된다. 예를 들면 유전알고리듬에서 어떤 스트링이 s=[2 4 1 5 3]으로 구성되어 있다면, 즉 s[1]=2 s[2]=4 s[3]=1 s[4]=5 s[5]=3와 같은 스트링이 있다면 첫 번째로 제품 2를 생산하고 마지막으로 제품 3을 생산한다. PMX는 반전(inversion)과 단순교배(simple crossover)의 특징을 하나의 교배 연산자로 결합한 연산자이다. 그리고 PMX는 무작위로 2개의 교배점(crossing site)을 선택하며 2개의 교배점 내부에 있는 스트링의 교배부분(matching section)은 단순교배 연산을 수행하고 유전자가 중복되는 것을 막기 위해 position-by-position 교환 연산을 수행한다. 예를 들면 부모 스트링이 A=[9 8 4 | 5 6 7 | 1 3 2 10], B=[8 7 1 | 2 3 10 | 9 5 4 6]으로 구성되었다면 첫째로 B를 A로 사상(mapping)하면서 5와 2, 3과 6, 10과 7을 교환한다. 비슷하게 A를 B로 사상하면서 2와 5, 6과 3, 7과 10을 교환한다. PMX을 수행한 최종 자손 스트링은 A'=[9 8 4 | 2 3 10 | 1 6 5 7], B'=[8 10 1 | 5 6 7 | 9 2 4 3]으로 구성된다. OX는 PMX와 비슷하지만 유전자의 중복을 해결하기 위해 경사 이동(sliding motion)을 이용한다. 예를 들어

PMX에서와 똑 같은 부모 스트링 A, B를 이용하고 A를 B에 사상하면 중복되는 제품 5, 6, 7은 H에 의해 표시는 공백이 된다. 즉 B=[8 H 1 | 2 3 10 | 9 H 4 H]이다. 이 H들은 상위 교배 점부터 시작하는 경사 이동으로 채워진다. 즉 B=[2 3 10 | H H H | 9 4 8 1]이다. 그런 다음 A로부터 교배 부분으로 B의 공백을 채운다. 이러한 연산 후의 최종적인 자손 스트링은 A′=[5 6 7 | 2 3 10 | 1 9 8 4], B′=[2 3 10 | 5 6 7 | 9 4 8 1]이 된다. 비록 PMX와 OX는 비슷하지만 그들은 다른 종류의 유사성을 처리한다. PMX는 절대적인 제품 순서를 고려하고 OX는 상대적인 순서를 고려한다.

유전 알고리듬은 대부분의 교배 연산자에서 한번의 교배에서 부모로부터 동시에 2명의 후손을 생산한다. 그러나 이것은 부모보다 잠재적으로 우수한 자손의 발현확률을 감소시킨다. 이러한 경우를 줄이기 위해 본 연구에서는 2번의 교배를 수행하여 2명의 자손을 생산하도록 하여 우수한 자손의 발현확률을 높이는 superior reinforcement strategy(SRS)를 개발하여 적용하였다. SRS는 한번의 교배에서 단지 하나의 자손을 생산한다. 이러한 교배는 인간의 경우와 유사하다. 그리고 한번의 교배에서 발생하는 2개의 자손 중에서 하나를 선택하는 법칙은 둘중 더 낳은 자손을 선택한다. 이러한 선택은 무작위로 선택하는 경우보다 우수한 자손의 발현확률을 높일 수 있다.

3-4. 돌연변이 연산자

돌연변이 연산자(mutation operator)는 무작위로 선택된 위치의 유전정보를 변경시킨다. 이진수로 표현된 스트링의 경우 돌연변이 연산자는 0은 1로 1은 0으로 바꾼다. 그러나 십진수로 표현된 스트링은 서로의 값들이 중복됨이 없도록 하기 위해 상호교환(pairwise exchange)을 행하여 돌연변이를 수행하는 상호교환 돌연변이 연산자를 사용한다. 그러나 이러한 단순 상호교환 돌연변이는 너무 외란이 커서 유전자의 구조를 크게 변경시키기 때문에 외란을 줄이도록 하는 방안으로 인접 상호교환 돌연변이(adjacent pairwise interchange mutation) 연산자와 활주 상호교환 돌연변이(sliding interchange mutation) 연산자를 혼합하여 사용한다.

인접 상호교환 돌연변이 연산자는 돌연변이 연산을 수행하기 위해 무작위로 하나의 돌연변이점을 선택하여 인접한 유전자와 상호교환을 수행한다. 예를 들면 하나의 스트링의 유전인자 구성이 [1 2 3 4 5 6]으로 구성되었고 무작위로 선택된 돌연변이 위치가 3이라면 인접 상호교환 돌연변이 연산을 수행하면 3번째 위치의 3과 4번째 위치의 4가 서로 교환되어 [1 2 4 3 5 6]으로 바뀐다.

활주 상호 교환 돌연변이 연산자는 무작위로 2개의 돌연변이 점을 선택하고 선택된 하위 위치의 1 2 3 4 5 6 → 1 2 H 4 5 6과 같이 값을 제거한다.

그후 하위 위치로부터 상위 위치까지의 스트링 부분을 1 2 H 4 5 6 → 1 2 4 5 H 6 다같이 오른쪽에서 왼쪽으로 이동시킨다.

최종적으로 제거된 하위 위치의 값을 1 2 4 5 H 6 → 1 2 4 5 3 6과 같이 상위 위치에 삽입한다.

3-5. 확장 연산자들

유전 알고리듬의 성능을 향상시키기 위해 세가지의 기본적인 연산자인 재생, 교배, 돌연변이 외에도 Elitist Model, Expected Value Model, Elitist Expected Value Model, Crowding Factor Model 등과 같은 다양한 전략들이 De Jong에 의해 제시되었다[20]. 본 연구에서는 이들중 Elitist Model과 Crowding Factor Model을 사용하였다. 이러한 전략 중에서 Crowding Factor Model은 많은 피크가 존재하는 함수의 최적화 문제에서 집단의 다양성을 유지하기 위해 새로운 자손을 비슷한 형태의 스트링을 가진 부모와 교체하는 방법이다. 즉 부모 개체들 중 일정 개체수 집단을 선택하고 그 다수의 집단들에서 가장 열등한 개체들 중 새로이 생성된 자손개체와 가장 유사한 개체를 자손개체와 교체하여 보다 우수한 부모세대를 형성하는 방법이다. 이 전략은 또한 많은 피크를 가진 조합문제(combinatorial 문제)인 다품종 회분식공정에도 잘 적용될 수 있다[17]. Crowding Model은 두 스트링간의 유사성(similarity)을 두 스트링의 동일한 위치에서 동일한 값을 가지는 횟수로 정의한다. 만일 2개의 벡터 x=[1 2 3 4 5], y=[4 2 3 5 1]이면 x와 y의 유사성은 2/5=0.4이다.

이상의 방법을 활용한 상세한 유전 알고리듬 flow chart구성을 Fig. 5에 나타내었다. 즉 본 연구에서 구현한 GA algorithm은 초기 세대의 개체수를 무작위로 생성시킨 후 각 개체의 적합도 함수를 scaling하고 본 연구에서 개발한 새로운 기본 연산자 및 확장 연산자로써 자손세대 개체를 만들고 적합도를 판정한 후 세대간 유전을 계속 진행한다. 이 때 한 세대의 개체수는 10N+2M으로 한정하며 3N 세대까지 수행한다.

3-6. 유전 알고리듬의 제어 변수

유전 알고리듬은 주어진 문제에서 해의 성능을 향상시키기 위해 집단의 크기, 교배 확률, 돌연변이 확률, 세대 차 등의 제어 변수들을 사용한다.

3-6-1. 집단의 크기

집단의 크기는 어떤 집단에서의 개체의 수를 말한다. 일반적으로 집단의 크기가 증가함에 따라 유전 알고리

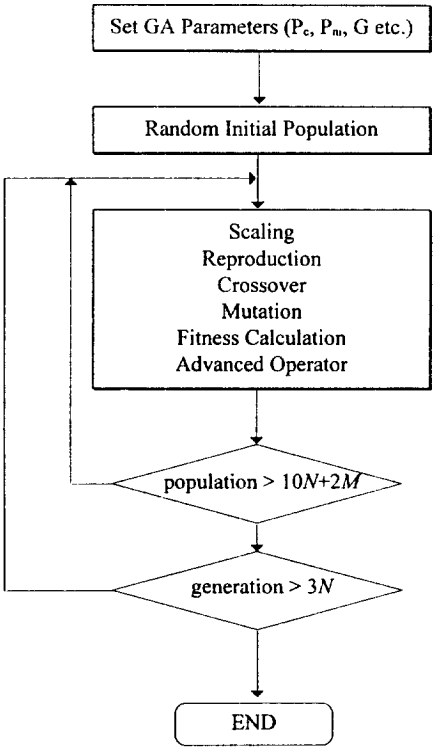

Fig. 5. Main genetic algorithm for multi-product batch scheduling.

듬의 성능은 향상된다. 그러나 또한 계산시간도 증가한다. 따라서 더 적은 계산시간으로 해의 정확성을 감소시키지 않는 적절한 집단의 크기를 정하는 것은 중요하다. 적절한 집단의 크기는 일반적으로 문제에 따라 다르고 문제가 복잡할수록 증가하는 경향이 있다. 따라서 회분식공정 생산계획에서 문제의 크기가 변함에 따라 적절한 집단의 크기도 변한다. 본 연구에서는 집단의 크기는 제품 수가 증가함에 비례하여 선형으로 증가하도록 설계하였다. 본 연구는 모의실험결과로부터 집단의 크기를 10N으로 정하였다.

3-6-2. 교배확률

교배확률(P_c)은 자손을 생산하기 위하여 선택된 배우자들이 교배를 하는 확률이다. 교배확률의 보수인 ($1-P_c$)는 자손들이 그들의 부모를 복제하는 확률이다. 만일 교배확률이 0이면 모든 자손은 그들의 부모를 복제한다. 교배확률에 대한 여러 연구들[20]은 상당히 정확한 선택 방법에 의해 표본추출의 추계론적 오차가 감소될 때 높은 교배확률($P_c=1.0$)이 더 우수하다고 제의한다. 이러한 사실로부터 본 연구에서는 $P_c=1.0$으로 선택한다.

3-6-3. 돌연변이확률

돌연변이확률(P_m)은 유전 알고리듬의 성능에 큰 영향을 끼친다. 돌연변이는 자연계에서 유기체가 진화하는 중요한 추진력이다. 돌연변이 확률은 보통 1보다 상당히 작고 집단 수에 반비례한다고 De Jong은 제시하였다[20]. 많은 문제에서 돌연변이 확률은 이진정수로 구성된 스트링의 경우 각 스트링의 비트당 0.001의 값을 가지지만 문제에 따라 가변적이다. 그리고 고정된 돌연변이 확률보다 환경에 따라 가변적인 적응 돌연변이가 훨씬 더 우수한 결과를 보여준다. 본 연구에서는 우성발현 강화 교배연산자와 연계하여 교배연산에서 만들어진 2개의 자손 스트링 중의 하나를 선택한다. 그리고 선택된 스트링의 원소 중의 하나를 임의로 선택하여 돌연변이 연산을 수행한다. 따라서 돌연변이확률은 하나의 스트링이 선택될 확률 0.5에 스트링 중에서 임의의 위치를 선택할 확률 $1/l$을 곱한 값이다. 즉 $P_c=0.5/l$이다. 여기서 l은 스트링의 길이로써 본 연구의 문제에서 제품의 수와 같다. 그러므로 제품의 수가 증가하면 스트링의 길이가 길어지며 생산계획 문제 또한 풀기 어려워진다. 따라서 이러한 생산계획 문제를 풀기 위해서는 스트링의 길이가 길어질수록 유전 알고리듬의 집단의 수가 증가해야 한다. 그러므로 유전 알고리듬의 집단의 수가 증가함에 따라 이에 반비례하여 돌연변이확률은 감소한다.

3-6-4. 세대 차

세대 차(G)는 다중 국부최소치 함수(multimodal function) 최적화 문제에서 중첩 집단(overlapping population)을 허용하기 위해 De Jong에 의해 도입되었다[20]. 세대 차는 현 세대에서 다음 세대를 생산하기 위해 현 세대에서 만든 자손들로 현세대를 대체하는 비율이다. G는 0과 1사이의 값으로써 G=1일 경우 비중첩 집단이며 G가 1보다 작을 경우는 중첩집단이다. 중첩집단에서 (n×G)개의 개체를 선택하여 다음 세대를 위한 자손을 생산한다. 그리고 현 집단에서 무작위로 (n×G)개의 개체를 선택하여 이들을 새로운 자손들로 대체한다. 대부분의 최적화 문제에서는 비중첩 집단 모델이 더 좋은 결과를 보여 준다고 알려진다. 그리고 본 연구의 생산계획 문제들을 테스트한 결과 비중첩집단이 중첩집단보다 더 빨리 우수한 해를 찾았다.

4. 실험적 결과

납기일은 $d_i=d(i=1,2,\cdots,N)$인 공통의 납기시간(common due-time)과 다른 납기시간(different due-time)으로 구분될 수 있다. 또한 납기일의 분포에 따라 3가지의 문제 유형으로 나누어 질 수 있다.

Type I은 highly constrained 문제들로 d_i의 분포가 $\min\{F_i\}$ 쪽으로 치우친 경우로써

$$\forall d_i < \min\{F_i\}$$

인 경우이다. 이 유형은 tardiness penalty만이 적용되고 납기일 위반에 대한 벌칙비용이 각 제품에 대하여 똑같다면 makespan 최소화 문제와 같은 유형이 된다.

Type II는 highly relaxed 문제로 d_i의 분포가 $\max\{F_i\}$ 쪽으로 치우친 경우로써

$$\forall d_i > \max\{F_i\}$$

인 경우이다. 이 유형은 단지 재고 비용만이 적용된 경우로써 재고 비용을 최소화하는 형태의 문제가 된다. 이러한 경우는 time horizon을 납기일 쪽으로 이동시킴으로써 재고 비용을 줄일 수 있다.

Type III는 neither too constrained nor too relaxed 문제로써 Type I과 II의 중간이다. 대부분의 문제는 Type III의 경우라고 가정하고 본 논문에서는 Type III의 문제를 중심으로 테스트하였다.

본 연구에서 평가되는 모든 문제들의 조업 정보는

$$t_{ij} \in [0, 30]$$
$$s_{(i-1)ij},\ a_{ij} \in [1, 5]$$

의 정수로 무작위로 만들어지고 생산계획의 시간 단위는 시(hour)단위로 가정한다. 또한 E/T model에서 이용되는 비용 계수들은

$$\alpha_i \in [0, 3]$$
$$\beta_i \in [0, 5]$$

의 무작위로 만들어지는 정수로 이루어진다. Tardiness의 비용이 earliness보다 더 크다는 가정 아래 β_i의 범위가 α_i의 범위보다 더 크도록 하였다.

본 연구의 성능평가를 N이 8보다 큰 대규모크기의 제품 수를 가진 문제에 관하여 평가하였다. 따라서 N=8, 10, 20, 30, 40과 M=2, 4, 6, 8의 각각의 조합에 대하여 50개의 테스트 문제를 풀어서 유전 알고리듬의 성능과 SA의 성능을 비교 조사하였다. 이러한 조건으로 다품종 회분식공정의 UIS 및 NIS방안의 여러 가지 문제를 풀이해 보았다.

본 연구에서는 생산계획 알고리듬의 해에 성능평가

Table 4. Numerical test results of GA and SA

N×M	UIS Proportion of optimal solution(%)		NIS Proportion of optimal solutions(%)		Mean CPU time of UIS and NIS(sec)	
	GA	SA by Ku and Karimi/SA by this work	GA	SA by Ku and Karimi/SA by this work	GA	SA
8×2	100	100/100	100	−/100	0.22	0.50
8×4	100	100/100	100	−/100	0.63	0.78
8×6	100	100/100	100	−/98	0.66	1.11
8×8	100	−/100	100	−/100	0.72	1.35
10×2	100	100/100	100	−/100	0.72	0.75
10×4	100	98/100	100	−/96	0.98	1.19
10×6	100	96/98	100	−/98	1.14	1.59
10×8	100	−/96	100	−/98	1.85	2.03
20×2	100	74/70	100	−/62	4.62	4.73
20×4	100	62/66	100	−/52	7.57	7.61
20×6	100	64/58	100	−/56	10.30	10.69
20×8	100	−/52	100	−/38	11.77	13.56
30×2	78	74/68	78	−/54	16.38	20.66
30×4	70	38/42	64	−/22	26.77	33.10
30×6	70	20/26	70	−/18	35.73	46.09
30×8	72	−/16	62	−/6	45.47	59.30
40×2	48	−/0	30	−/0	42.26	62.42
40×4	32	−/0	24	−/0	62.89	101.09
40×6	30	−/0	24	−/0	85.15	140.74
40×8	30	−/0	26	−/0	106.77	181.98

측정으로 가장 많이 이용되는 최적해의 비율(proportion of optimal solution)과 계산 시간인 CPU 시간을 사용하여 본 연구의 GA와 KU와 Karimi[16]의 SA 알고리듬에 대해 같은 문제들을 풀이한 결과를 비교, 분석하였다. 최적해의 비율은 테스트 문제들에서 최적해를 발견하는 백분율 혹은 확률이다.

N이 8보다 작은 소규모크기의 문제에서는 complete enumeration을 통한 전체구역 탐색으로 최적해를 구할 수 있지만, N이 8보다 큰 대규모 크기의 문제에서는 적절한 시간 이내에 최적해를 발견하는 것은 어렵다. 그래서 대규모 크기의 문제에 대한 본 연구방법의 최적성평가를 하는 방법으로 다음과 같은 평가법을 사용하였다. 어떤 임의의 생산순서를 정한 다음 이 생산계획에서 각 제품의 출하일인 $F_i = d_i$가 되도록 납기일을 정한다. 따라서 이러한 문제들의 최소의 목적함수값 즉 penalty의 값은 정확히 0이 된다. 생산계획 알고리듬의 계산시간에 대한 성능평가 측정은 DEC 3000 workstation에서 이루어졌다.

또한 사용된 유전 알고리듬의 제어 변수들은 $P_c = 1.0$, $P_m = 0.5/N$, $G = 1.0$, $n = 10N + 2M$이고 최대의 세대수는 3N으로 제한하였다. 본 연구의 결과와 SA법의 결과를 Table 4에 비교하여 나타내었으며 본 연구의 GA법의 경우 문제의 크기가 $8 \times M$ 및 $10 \times M$일 때 모든 문제에서 최적해를 얻고 있음을 보여준다. 문제의 크기가 커질수록 최적해를 구하는 경우가 차츰 줄어듬을 볼 수 있으나 GA의 경우는 문제가 커지더라도 상당히 높은 확률로 최적해를 찾고 있고, SA의 경우는 문제가 커질수록 그 성능이 GA에 비해 크게 뒤짐을 알 수 있었다.

5. 결 론

본 논문에서는 다품종 회분식공정의 생산계획 문제로써 납기일과 재고 비용을 고려하는 E/T 모형에 대한 유전 알고리듬을 제시하고 이를 UIS 및 NIS 방안의 생산계획 문제에 적용하였다.

유전 알고리듬은 다양한 생산계획 문제에 대해 놀라울 만큼 향상된 성능을 보여주는 빠르고 강건한 방법임을 보였다. 재생, 교배, 돌연변이의 기본 연산자와 보다 개선된 확장 연산자들을 가진 유전 알고리듬은 우수한 준최적해를 찾는 생산계획 알고리듬이며 쉽게 프로그램될 수 있고 사용할 수 있는 특징이 있다. 또한 유전 알고리듬은 더 복잡한 형태의 생산계획 알고리듬에 적용될 수 있으며 그 발전가능성이 매우 큰 방법임을 보여준다. 끝으로 무작위로 만들어진 많은 테스트 문제에서 GA는 그 성능이 SA보다 월등하여 보다 짧은 컴퓨터 계산시간 내에 보다 더 우수한 결과를 보여주고 있다.

감 사

본 연구는 한국학술진흥재단의 신진연구인력 연구비 지원과 1994학년도 영남대학교 학술연구조성비에 의해 이루어진 것으로 연구비를 지원해 주신 한국학술진흥재단과 영남대학교에 감사의 뜻을 표하는 바입니다.

사용기호

a_{ij} : transfer time of batch i out of unit j
C_{ij} : completion time at which i th product in the sequence finishes processing on batch unit j
E(f) : expected value
f(S) : objective function
G : generation gap
GA : genetic algorithm
M : number of batch units
n : number of generation
N : number of batches
P : probability
P_c : probability of crossover
P_m : probability of mutation
SA : simulated annealing
$s_{(i-1)ij}$: setup time of batch i after (i−1) on unit j
t_{ij} : processing time of product i on batch unit j
exp[ ] : exponential function
max[ ] : maximum of the quantities in the brackets
min[ ] : minimum of the quantities in the brackets

참고문헌

1. Reklaitis, G. V.: *AIChE Symp. Ser.*, **78**, 119(1982).
2. Reklaitis, G. V.: "Perspectives on Scheduling and Planning of Process Operations", Proceedings of 4th International Symp. on PSE, Montebello, Canada(1991).
3. Ku, H. M., Rajagopalan, D. and Karimi, I.: *Chem. Engng Prog.*, **August**, 35(1991).
4. Johnson, S. M.: *Naval Research Logistics Quarterly*, **1**, 1(1954).
5. Dannenbring, D. G.: *Management Science*, **23**, 1174 (1977).
6. Rajagopalan, D. and Karimi, I.: "Scheduling in Serial Mixed-storage Multiproduct Processes with Transfer and Set-up Times", Computer-Aided Proc. Oper., CACHE and Elsevier, New York, 679

(1987).
7. Ku, H. M. and Karimi, I.: *Ind. Eng. Chem. Res.*, **27**, 1840(1988).
8. Wellons, M. C. and Reklaitis, G. V.: "Problems in the Scheduling of Batch Processes", Presented at the TIMS/ORSA Joint National Meeting, Washington, D.C., 1988.
9. Musier, R. F. H and Evans, L. B.: *Comput. & Chem. Eng.*, **13**, 229(1989).
10. Baker, K. and Scudder, G.: *Operations Research*, **38**, 22(1990).
11. Garey, M., Tarjan, R. and Wilfong, G.: *Math. Opns. Res.*, **13**, 330(1988).
12. Fry, T., Armstrong, R. and Blackstone, J.: *IEEE Trans.*, **19**, 445(1987).
13. Abdul-Razaq, T. and Potts, C.: *J. Opnl. Res. Soc.*, **39**, 141(1988).
14. Jung, J. H., Lee, H., Yang, D. R. and Lee, I.: *Comput. & Chem. Eng.*, **18**, 537(1994).
15. Ku, H. M. and Karimi, I.: *Ind. Eng. Chem. Res.*, **29**, 580(1990).
16. Ku, H. M. and Karimi, I. A.: *Comput. & Chem. Engng.*, **15**, 283(1991).
17. Lee, C. H., Jung, J. H. and Lee, I.: *Comput. & Chem. Eng.*, resubmitted, (1994).
18. Holland, J.: "Adaptation in Natural and Artificial Systems", University of Michigan Press(1975).
19. Goldberg, D. E.: *Engineering with Computers*, **3**, 35(1987).
20. Goldberg, D. E.: "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, (1989).
21. Rajagopalan, D. and Karimi, I.: *Comput. & Chem. Eng.*, **13**, 175(1989).