

전처리 방식의 복수지역 제약공정 스케줄링*

홍민선** · 임석철***† · 노승종****

Preprocessing based Scheduling for Multi-Site Constraint Resources*

Min-Sun Hong** · Suk-Chul Rim*** · Seung J. Noh****

■ Abstract ■

Make-to-order manufacturers with multiple plants at multiple sites need to have the ability to quickly determine which plant will produce which customer order to meet the due date and minimize the transportation cost from the plants to the customer. Balancing the work loads and minimizing setups and make-span are also of great concern. Solving such scheduling problems usually takes a long time. We propose a new approach, which we call 'preprocessing', for resolving such complex problems. In preprocessing scheme, a 'good' a priori schedule is prepared and maintained using unconfirmed order information. Upon the confirmation of orders, the preprocessed schedule is quickly modified to obtain the final schedule. We present a preprocessing solution algorithm for multi-site constraint scheduling problem (MSCSP) using genetic algorithm; and conduct computational experiments to evaluate the performance of the algorithm.

Keyword : Scheduling, Preprocessing, Genetic Algorithm, Multi-site, Constraint Resource, TOC

논문접수일 : 2007년 09월 26일 논문게재확정일 : 2007년 12월 20일

* 본 논문의 심사과정에서 많은 심도 깊은 지적과 조언을 해주신 익명의 심사위원들께 감사드립니다.

** i2 테크놀러지

*** 아주대학교 산업정보시스템공학부

**** 명지대학교 경영학과

† 교신저자

1. 서론

오늘날의 글로벌 경쟁 환경에서 제조기업은 구매, 생산, 판매 등 공급사슬 상의 제반 활동을 고객의 수요와 동기화하는 일이 운영효율 향상에 절대적으로 중요하다는 사실을 인식하게 되었다. 특히 주문생산(Make-to-order) 위주의 제조기업의 경우에는 수요 변화에 대응하여 실시간으로 생산 스케줄링을 변경함으로써 운영효율성을 제고할 수 있는 능력이 절대적으로 요구된다. 예를 들어 제조공장이 여러 지역에 위치한 경우에는 어느 공장에 어떤 주문을 할당하느냐에 따라 납기와 생산비용, 운송비 등이 달라지는데, 이러한 할당 의사결정에는 납기준수 가능성, 운송비용, 품목교체시간(셋업시간), 생산가능성 등 다양한 변수와 제약조건을 복합적으로 신속하게 고려해야 한다. 그러나 셋업시간과 로드 밸런싱 등의 상세한 제약사항을 모두 고려한 스케줄링을 수립하는 데에는 시간이 너무 많이 소요되기 때문에 대략적인 계획만을 수립함이 일반적이고, 실제로 생산을 개시하면 대략적으로 설정한 스케줄이 실행 가능하지 않거나 비효율적임을 발견한다.

스케줄링 문제의 또 다른 어려움은 스케줄링이 한 시점의 불변하는 데이터를 사용하는 정적(static), 확정적(deterministic) 문제가 아니고 시간의 흐름에 따라 데이터가 변하는 동적(dynamic), 확률적(stochastic) 문제라는 점에 있다. 예를 들어 반도체 제조공정은 품질결과에 따라 재작업(re-entrance) 여부가 결정되는 등 공정순서와 공정 소요시간이 변하는 경우가 많다. 이렇듯 데이터가 수시로 변하는 상황에서는 이를 반영한 실시간 스케줄링이 바람직하다.

본 연구에서는 상세한 제약사항까지 고려하는 스케줄링을 신속하게 수행하는 새로운 방법으로서 '전처리'(preprocessing)라는 개념을 제안한다. 전처리란 아직 확정되지 않은 rolling plan상의 주문 예상치를 활용하여 미리 좋은 스케줄을 만들고, 데이터에 변화가 생길 때마다 스케줄을 수정 보완하다가 주문이 확정되면 이를 간단히 갱신하여 신속

하게 좋은 스케줄을 얻는 동적 스케줄링 방법이다. 다른 한편으로 본 연구에서는 여러 지역에 공장이 위치한 주문생산 스케줄링 문제를 복수지역 제약공정 스케줄링 문제(MSCSP : multi-site constraint scheduling problem)라 정의하고, 전처리 방법론과 유전자 알고리즘(genetic algorithm)을 사용하여 공장별 제약공정만을 대상으로 신속하게 스케줄링할 수 있는 알고리즘을 제시한다. 또한 기존의 스케줄링 방식과 대비한 수행도를 시뮬레이션을 통해 평가한다.

2. 관련 연구

본 연구와 관련하여 Kutanoglu and Wu(2004)는 스케줄링의 견고성(robustness)을 향상시키기 위한 방법으로 1단계 전처리를 거친 후 2단계 동적 적응(dynamic adaptation)을 통해 job shop의 생산 스케줄을 결정하는 절차를 제시하였다. 그들은 각 공정소요시간(processing time)의 불확실성만을 논문에서 다루고 있으나 본 연구에서는 주문의 추가, 삭제, 및 변경을 모두 고려하는 더 넓은 범위를 다루고자 한다.

이 외에도 본 연구와 관련된 스케줄링 분야의 연구결과는 매우 많다. 특히 Jop shop 스케줄링 분야에서는 유전자 알고리즘의 성능 향상을 위하여 simulated annealing과 같은 메타 휴리스틱 알고리즘을 혼합하여 성능을 높이는 연구가 다수 진행되었다. Koonce and Tsai(2000)은 데이터 마이닝이 유전자 알고리즘에 의해 생산된 job shop 스케줄링 결과의 스키마를 활용하여 학습될 수 있음을 실험적으로 보였다. Ghedjati(1999)는 job shop 스케줄링 문제를 풀기 위하여 우선순위를 활용하여 설비를 선택하는 휴리스틱 방법론을 혼합한 유전자 알고리즘을 제시하였다. Yu and Liang(2001)은 기존의 job shop 스케줄링 문제에서 납기와 유한 복수 설비 등을 포함하는 EJSSP(expanded job shop scheduling problem)을 모형화하고 이를 풀기 위하여 neural network과 genetic algorithm을 혼합

한 hybrid approach를 제안하였다. Liu and Tang (1999)은 준비시간을 갖는 단일기계 스케줄링 문제의 해법으로 필터링 방식을 도입한 수정된 유전자 알고리즘을 제시하였다. Feldmann and Biskup (2003)은 납기 미준수에 대한 손실(penalty)이 존재하는 단일기계 스케줄링 문제에 대하여 진화전략(evolutionary strategy), simulated annealing, threshold accepting 등 메타 휴리스틱 알고리즘의 성능을 평가하였다. 김석준과 이채영(1992)은 병렬 유전자 알고리즘이 수렴속도 면에서 단순 유전자 알고리즘보다 평균적으로 많은 시간을 요구하나, 해의 질적인 측면에서는 항상 우월함을 보였다.

한편, 제약공정만을 대상으로 스케줄링을 수립하는 방법으로 복잡한 스케줄링 문제를 단순화한 연구도 있다. Simons 등(1996)은 제약이론의 DBR(Drum-Buffer-Rope, Srikanth and Umble, 1997) 방식을 적용한 DBRCSP(drum-buffer-rope constraint scheduling problem) 모델을 제시하였으나 단일공장만을 고려하고 있다. 본 연구는 DBRCSP의 확장으로서 복수지역에 공장이 위치한 주문생산 스케줄링 문제를 복수지역 제약공정 스케줄링 문제(MSCSP : multi-site constraint scheduling problem)라 정의하고, 그 제약공정만을 빠르고 정확하게 스케줄링 할 수 있는 전처리 방식의 알고리즘을 제안하고자 한다. 스케줄링에 사용되는 유전자 알고리즘은 입력값이 고정된 값이라고 전제하고 있으나, 본 연구에서는 시간에 따라 변하는 입력값을 활용하는 동적 유전자 알고리즘을 제안한다.

본 연구의 출발이 되는 Simons 등(1996)의 논문을 참조한 후속연구 중 본 연구와 관련 있는 연구를 정리하면 다음과 같다. Wu and Yeh(2006)는 반도체공정과 같이 제품이 제약공정을 여러 번 거치는 re-entrant flow의 경우에 대하여 drum, 즉 제약공정의 스케줄을 개발하는 DBR 스케줄링 방법을 제안하였다. Steele 등(2005)은 DBR과 MRP를 심층 비교하면서 시뮬레이션을 통하여 일반적으로 DBR이 MRP보다 생산계획 및 통제성과 측면에서 우수하다는 사실을 보였다. 또한 MRP 운영

정책을 DBR 원칙에 입각하여 보완한다면 두 방법의 성과차이를 크게 줄일 수 있어 현재 MRP를 사용중인 기업이 굳이 DBR방식으로 전환하지 않아도 된다고 주장하고 그 근거와 방법을 제시하였다. Atwater and Chakravorty(2002)는 시스템 내의 가장 가동률이 높은 두 자원의 가동률의 변화가 DBR 스케줄링의 성능에 미치는 효과를 시뮬레이션을 통해 분석하였다. Uzsoy and Wang(2000)은 제약자원을 갖는 job shop 스케줄링 시 shifting bottleneck의 성능을 분석하였다. Simons 등(1999)은 시스템 내에 복수의 제약자원이 존재하는 경우 이를 동시에 스케줄링 하는 것이 순차적으로 하는 것보다 최대지연(max. tardiness)값은 다소 감소하나 평균 제조시간(flow time) 등은 오히려 증가함을 보였다.

3. 복수지역 제약공정 스케줄링 문제

본 연구에서 다루는 복수지역 제약공정 스케줄링 문제(MSCSP)는 Simons 등(1996)이 다룬 DBRCSP를 확장한 문제이다. DBRCSP는 단일 공장을 대상으로 하는데 비하여 MSCSP는 여러 지역에 위치한 다수 공장을 대상으로 주문할당과 스케줄링을 동시에 결정한다. 각 주문의 생산완료가 납기일보다 늦어지는 경우 늦어진 시간에 비례하는 손실비용이 초래되며, 한 주문이 어떤 공장에 할당되면 생산종료 후 수요지까지의 운송비가 발생한다. 이 밖에도 DBRCSP에서는 고려하지 않은 작업순서에 의해 달라지는 셋업시간 문제와 생산라인이나 공장 간의 로드 밸런싱 문제도 동시에 고려한다. MSCSP의 수리모형은 다음과 같다.

모수

a_j : 주문 j 의 단위시간당 지연에 따른 지연비용

β_{jk} : 주문 j 를 공장 k 에서 고객에게 인도하는 운송비

h_j : 주문 j 의 납기일

θ : a large number

s_{ijk} : 공장 k 에서 주문 i 직후 주문 j 를 생산할

때 필요한 준비시간(셋업시간)

n_j : 주문 j 의 공정 배치(batch) 크기(수량)

p_{jk} : 주문 j 에 대한 단위제품별 공장 k 의 제약공정 소요시간

J : 총 주문수

결정변수

f_{ik} : 주문 i 가 공장 k 에서 완성되는 시각(finish time)

y_{ijk} : 공장 k 에서 주문 i 가 주문 j 의 직전 작업이면 1, 아니면 0

z_{ik} : 주문 i 가 공장 k 에 할당되면 1, 아니면 0

MSCSP

$$\text{Minimize } \sum_i \sum_k \alpha_i \text{Max}[f_{ik} - h_i, 0] \quad (1)$$

$$+ \sum_i \sum_k \beta_{ik} z_{ik}$$

subject to :

$$f_{jk} + \theta(1 - y_{ijk}) \geq f_{ik} + s_{ijk} + n_j p_{jk} \quad (2)$$

for all $1 \leq i \leq J, 1 \leq j \leq J,$

$$1 \leq k \leq K$$

$$\sum_j \sum_k y_{ijk} = 1 \quad \text{for all } 1 \leq i \leq J \quad (3)$$

$$\sum_k z_{ik} = 1 \quad \text{for all } 1 \leq i \leq J \quad (4)$$

식(1)은 MSCSP의 목적함수로서 첫째 항은 납기지연(tardiness)이 초래하는 지연비용의 합이고, 둘째 항은 생산완료 후 소요되는 운송비용이다. 식(2)는 제약공정에 할당된 주문 간의 선후 관계를 표시한 것으로, 주문 j 의 종료시각은 직전주문 i 의 종료시각에 준비시간과 가공시간을 더한 값보다 빠를 수 없음을 나타낸다. 식(3)은 한 주문의 직후 주문은 오직 하나뿐임을 보장하며, 식(4)는 각 주문이 단 하나의 공장에 할당되도록 한다. 위 수리모형은 J 개의 주문을 일시에 K 개의 공장에 배치(할당)하는 정적(static)인 문제임에도 최적해를 구하기가 어렵다. 그러나 현실의 MSCSP는 시간이 지남에 따라 주문정보가 지속적으로 변화하는 동적(dynamic)

문제이므로 이러한 수리모형으로는 해결할 수 없다. 이하에서는 전처리 방식의 제약공정 유전자 알고리즘을 제안한다.

4. 전처리 방식의 제약공정 유전자 알고리즘

본 장에서는 앞 장에서 제시한 MSCSP의 해를 구하기 위한 전처리 방식의 제약공정 유전자 알고리즘을 제안한다.

4.1 전처리의 기본개념

최근 들어 공급망관리(SCM)는 공급망을 형성하는 여러 기업을 동기화(synchronization)하는 방향으로 발전하고 있고, 이를 구현하고자 하는 RTE(Real-Time Enterprise) 개념이 본격적으로 도입되고 있다. 2002년 가트너 그룹이 처음 제시한 RTE는 이벤트가 발생하는 즉시 그 사실과 근본 원인을 인지하고, 이를 실시간으로 분석, 대응, 실행함으로써 핵심 비즈니스 프로세스의 지연을 최소화하는 개념이다. 개념상으로는 바람직한 방향이지만 이를 실제로 구현하기는 쉽지 않다.

대규모 제조기업은 RTE가 아니더라도 공급사와의 협업을 위해 공급사와 생산계획을 공유한다. 한 예로 국내의 대표적인 전자업체인 모 기업은 향후 50주의 생산계획을 주별로 갱신하여(rolling plan) 공급사에 제공하고, 공급사는 이를 자사의 수요예측치로 삼아 RTF(Return to Forecast)라는 납품가능수량을 산출하여 역시 주별로 제조기업에 제공한다.

하지만 rolling plan은 다양한 원인으로 변동되게 마련이며, 이에 따라 협업 프로세스의 효과가 저하되고 있다. Rolling plan이 변할 뿐만 아니라 제조기업은 복잡한 생산 환경-다수 공장, 다양한 납기, 할당정책, 라인밸런스-하에서 여러 생산자원의 다양한 제약사항을 제대로 고려하지 못한 상태에서 스케줄링 함이 일반적이다. 이렇듯 현실을 정확하

게 반영하지 못한 생산스케줄은 재고적체나 납기 지연, 과도한 운송비용 등을 초래하여 비효율의 원인이 된다.

본 연구에서 제시하는 전처리란 확정된 주문만으로 생산 스케줄을 일괄 작성하는 기존의 방식과는 달리 RTF와 같은 잠정적 데이터를 활용하여 좋은 스케줄을 생성해 두고, 주문이 확정되는 시점에서는 변동만을 반영하여 스케줄을 짧은 시간 내에 생성하는 방법이다. 기존의 방법은 확정된 데이터만을 사용하여 스케줄링하기 때문에 컴퓨터의 연산시간은 주문이 확정되는 시점부터 시작하여 상대적으로 오래 소요되는 데 반해서 본 연구가 제시하는 전처리 방식은 주문 데이터가 변동할 때 이미 만들어둔 스케줄을 간단히 갱신하는 방식이므로 처리속도가 빠른 장점이 있고, 보다 많은 상세한 제약조건을 고려하여 스케줄링 할 수 있어서 보다 좋은 해를 얻을 수 있다.

4.2 전처리의 예

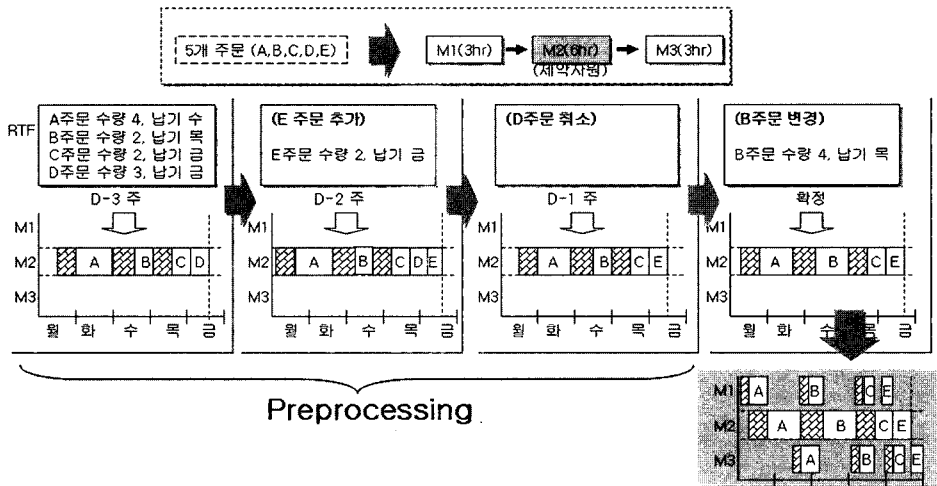
예를 들어 [그림 1]과 같이 한 공장에 3개의 순차적 공정(M1, M2, M3)이 있고 이 중에서 M2가 제약공정인 상황에서 5개의 주문이 접수된 상황을 생각해보자. 전처리 방식은 D-3주에 도착한 주문

으로 제약공정의 스케줄을 만들고, D-2주에 주문 E가 추가되면 이를 반영하여 다시 결과를 만드는 과정을 반복한다. 주문이 확정된 다음 제약공정 스케줄이 완성되면 비제약공정은 제약공정에 종속(subordinate)시킨다.

본 연구에서 제시하는 전처리 모델은 두 가지의 장점을 갖는다. 첫째, 셋업시간이나 load balance 등 다양한 제약조건을 동시에 고려하여 스케줄을 작성하기 때문에 스케줄의 feasibility를 높일 수 있다. 둘째, 스케줄링 연산을 미리 충분히 오래 수행함으로써 해의 품질을 개선할 수 있다. 대부분의 공급망 스케줄링 모델의 문제점은 변수가 많고 프로세스가 복잡하여 가능해를 도출하는 데에도 상당한 시간이 걸리고, 다양한 제약사항을 복합적으로 고려하지 못한다. 반면 전처리 방식은 rolling plan 으로부터 얻어지는 불완전하지만 가능성이 높은 데이터를 활용하여 주문 확정 이전에 좋은 스키마를 저장함으로써 스케줄링 시간을 크게 단축시킬 수 있다.

4.3 전처리 기반의 유전자 알고리즘

본 연구가 제안하는 전처리 기반의 유전자 알고리즘은 제약이론의 DBR 스케줄링과 유전자 알고리즘



[그림 1] 전처리 방식의 제약공정 스케줄링의 예

을 결합한 것으로서, 이를 요약하면 [그림 2]와 같다.

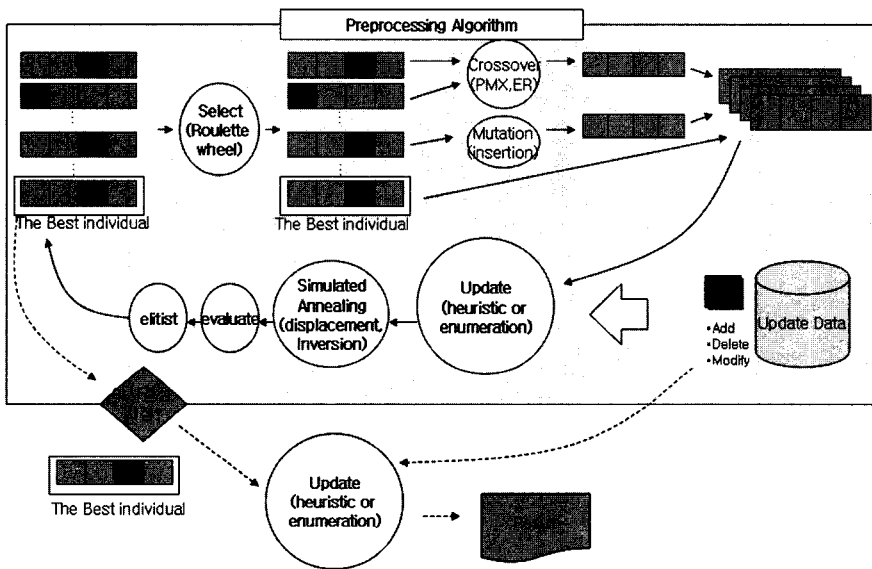
Step 0. 개체(염색체)와 모수의 정의

일반적으로 유전자 알고리즘에는 모집단의 크기 (n), 반복수(N), 교배율(Pc), 돌연변이율(Pm) 등과 같은 모수가 있고, 이 값에 따라 알고리즘의 성능이 크게 좌우된다. 모수의 값을 정하는 일정한 규칙은 없으며, 시행착오를 통해서 적절한 값을 구하는 것이 일반적이다. 개체는 [그림 3]과 같이 1차원 배열구조로 되어 있으며, 제약공정 상의 작업순서가

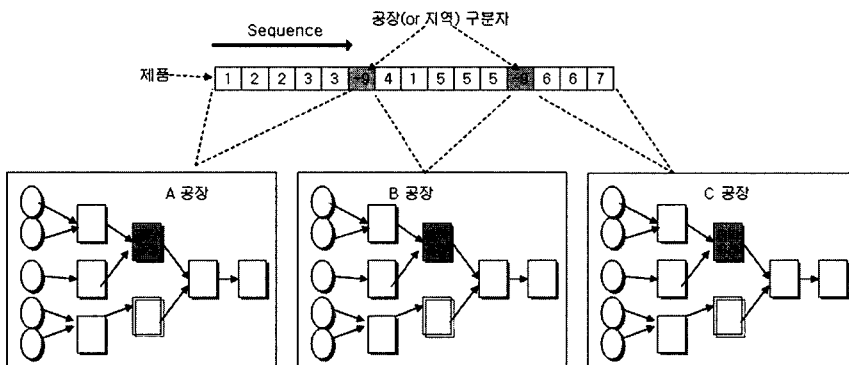
site간의 구분자(-9)로 분리되어 있다. 이러한 개체 구조는 주문의 순서와 위치를 변화시킴으로써 산업 규칙과 같은 다양한 제약식을 수용할 수 있다는 장점이 있다. 또한 1차원 배열구조이기 때문에 유전 연산자를 만들기 쉽고 가능해를 유지하기도 쉬운 장점이 있다.

Step 1. 선별(selection) 연산자

초기 해는 현재의 공정 스케줄을 기초로 임의로 생성시킨 개체 중 제약조건에 위배되는 해를 버리는



[그림 2] 전처리 기반의 유전자 알고리즘의 구조



[그림 3] 전처리 기반의 제약공정 유전자 알고리즘의 염색체 구조 예

방식으로 구한다. 선별 연산자는 한 세대(population) 개체들 중 좋은 스키마를 갖는 개체를 다음 세대에 넘기는 역할을 한다. 이렇게 선별된 개체들은 교배(crossover) 연산의 대상이 된다. 선별 방식으로는 토너먼트 방식이나 엘리티즘(elitism)과 룰렛휠(roulette wheel) 방식 등을 사용할 수 있다. 본 연구에서는 sort 기능이 없는 룰렛 휠 방식을 사용하였으며, 복수개의 목적함수를 갖는 특징을 살려 전체 목적함수, 순서기반 목적함수(셋업타임), 위치기반 목적함수(납기, 할당)의 3가지로 구분하여 선별하였다.

Step 2. 유전 연산자(genetic operators)

유전 연산자에는 교배(crossover) 연산자와 돌연변이(mutation) 연산자가 있다. MSCSP는 복합 목적함수를 갖기 때문에 개체의 위치와 순서가 모두 중요하다. 이 때문에 스케줄링 문제에서 가장 효율적이라고 알려진 PMX(partial matched crossover) 연산자와 순서기반 ER(edge recombination crossover) 연산자를 교배연산자로 사용하였다(Koonce and Tsai, 2000). 두 부모(parents) 모두 위치 적합도가 평균보다 좋은 경우에는 다음 세대의 스키마도 이 시간대를 고수해야 한다. 또한 할당의 경우에도 가장 좋은 공장에 할당되어야 한다. 이를 위해서 PMX 연산자로 교배하여 더 좋은 상태를 유지하도록 하였다. 또한 두 부모 모두 순서 적합도가 평균보다 좋은 경우에도 다음 세대의 스키마가 좋은 순서를 유지해야 한다. 이 값이 높은 적합도를 갖는 개체는 ER 연산자를 활용하여 교배하였다. 이 외의 경우에는 두 교배 연산자를 임의로 선택하여 다음 세대에 더 좋은 값이 넘어 갈 수 있도록 하였다.

Step 3. 개체 갱신 알고리즘

주문이 추가되거나 취소되는 경우, 주문 데이터가 변경되는 경우에는 개체나 유전자의 구조를 바꾸어야 한다. 이를 위해 [그림 4]에서 보는 바와 같은 휴리스틱 알고리즘을 적용하였다. 주문이 추가되는 경우에는 그 숫자만큼 유전인자가 늘어나 염색체의 구조가 바뀐다. 주문이 추가되면 최적의 공

```

Procedure 새로운 주문 추가
begin
  best_eval ← 0 ;
  best_gene ← 0 ;
  for i ← 사이트가 다른 공장 사이까지 do
    for k ← 납기일부터 동일한 셋업타임 작업까지 do
      if 동일한 셋업타입이 있는 경우 then
        동일한 셋업타임 작업 뒤에 새로운 작업 배치 ;
      end
    end
  end
  current_eval ← 현사이트 평가 ;
  if current_eval > best_eval then
    best_gene ← current_gene ;
  end
end
end
end
    
```

(a) 새로운 주문 추가

```

Procedure 주문 취소
begin
  for i ← 사이트가 다른 공장 사이까지 do
    for k ← 납기일부터 동일한 셋업타임 작업까지 do
      if 취소작업과 동일한 작업이 있는 경우 then
        작업 제거 ;
      end
    end
  end
  current_eval ← 현사이트 평가 ;
end
end
end
    
```

(b) 주문 취소

```

Procedure 주문데이터 변경
begin
  for i ← 사이트가 다른 공장 사이까지 do
    for k ← 납기일부터 동일한 셋업타임 작업까지 do
      if 주문 변경 작업이 발견된 경우 then
        작업 내용 변경 ;
      end
    end
  end
  current_eval ← 현사이트 평가 ;
end
end
end
    
```

(c) 주문데이터 변경

[그림 4] 주문 변동 heuristic algorithm

장에 할당된 다음 납기를 기점으로 시간을 거슬러 올라가며 가장 나중 주문부터 하나하나 순서를 바꾸어가면서 같은 setup type을 갖는 주문의 바로 뒤까지 계속 위치를 바꾸어 가장 높은 적합도를 갖는 위치에 배치한다. 이렇게 함으로써 기존의 스키마가 파괴되는 일을 최소로 줄일 수 있다. 주문 데이터에 변경이 발생하는 경우 염색체의 구조에는 변화가 없다. 다만, 순서의 변화가 생기고, 이를 유전자 알고리즘 상의 돌연변이(mutation)로 처리한다. 주문이 추가되는 경우 유전인자를 삽입하는 정책으로는 tabu search, simulated annealing, neural networks와 같은 메타 휴리스틱을 사용할 수 있다. 주문이 삭제되는 경우에는 염색체에서 삭제되는 주문수만큼 유전인자가 줄어들게 되지만 주문추가 기능이나 변경과 같은 변화는 생기지 않는다.

Step 4. 국지적 탐색(genetic local search)

유전자 알고리즘이 국지적 최적해 탐색에 취약한 약점을 극복하기 위해 simulated annealing 알고리즘을 활용한다. Simulated annealing은 온도를 조절함으로써 전역탐색(global search)과 국지탐색(local search)의 정도를 조절할 수 있는 장점이 있다. 본 논문에서 사용된 SA는 온도 값을 낮추어 국지적 탐색에 가깝도록 조절하여 사용하였다. Simulated annealing 연산자 역시 위치 적합도 개선을 위하여 임의로 구성된 구간을 역순으로 바꾸는 reverse 연산자를 활용하였고, 순서 적합도 개선을 위하여 임의의 구간을 옮기는 transport 연산자를 사용하였다.

Step 5. 적합도 함수(fitness function)

적합도 함수는 개체(염색체)의 성능을 나타내는 평가 함수로서 앞에서 제시한 MSCSP의 목적함수와 동일하게 설정하였다. 또한 교배 연산자를 위하여 부분적으로 위치 적합도와 순서 적합도를 함께 저장하여 선별연산자와 교배 연산자에서 활용하도록 하였다.

Step 6. Elitist

선택 연산자로 룰렛휠을 사용하였기 때문에 가장 좋은 값이 다음 세대로 넘어가지 않을 확률이 있다. 이를 방지하기 위하여 Elitist 연산자를 사용하여 현 세대에 가장 좋은 값이 다음 세대에서도 살아남도록 하였다.

Step 7. 의사결정 단계

Step 1부터 step 6은 연속적으로 순환되는 구조이다. 의사결정이 필요한 때에는 현 세대에서 적합도가 가장 우수한 개체를 획득한다. 이러한 구조의 장점은 실시간으로 의사결정에 필요한 해를 얻을 수 있다는 점이다. 따라서 일반적인 정적(static) 스케줄링뿐만 아니라 동적(dynamic) 스케줄링에도 적용할 수 있다는 장점이 있다.

5. 시뮬레이션

본 연구가 제시한 알고리즘의 우수성을 보이기 위하여 전통적 방식의 알고리즘과 본 연구의 알고리즘의 수행도를 시뮬레이션을 통해 평가하였다. VisualC++.net을 사용하여 주문 도착시간, 주문의 종류, 크기, 납기 등을 확률적으로 생성하였고, 데이터 입출력은 데이터베이스 시스템인 MySQL을 활용하였다.

5.1 시뮬레이션 실험 조건

성능평가 시뮬레이션을 위한 가정은 다음과 같다.

- (1) 2개의 공장
- (2) 24시간 연속 생산
- (3) 공장별 단일 생산라인
- (4) 제품의 종류별 routing은 <표 1>과 같으며, 제약공정은 M3 공정으로 설정
- (5) 제약공정의 생산시간은 4시간, 6시간, 8시간, 10시간, 12시간이 각각 10%, 20%, 40%, 20%, 10% 비율로 분포
- (6) 주문별 납기는 접수 후 1, 2, 3, 4, 5일이 각각 10%, 20%, 40%, 20%, 10% 비율로 분포

<표 1> 제품 종류별 공정순서(routing)

제품	Routing (괄호안은 가공시간, hr)
A	M1(1)→M2(2)→M3(4)→M4(2)→M5(1)
B	M1(2)→M2(3)→M3(6)→M4(2)→M5(1)
C	M2(5)→M1(3)→M3(8)→M4(3)
D	M1(2)→M3(10)→M5(1)→M2(2)→M4(2)
E	M2(2)→M3(12)→M4(2)

시뮬레이션은 <표 2>와 같이 평균 주문수를 30 개에서 200개까지 변동시키면서 수행하였고, 세대별 주문 변동이 3, 4, 5회 발생할 확률을 각각 25%, 50%, 25%로 정하였다.

<표 2> 실험 조건 별 주문수

실험 형태	총 주문수	기간
실험조건 1	29	5일
실험조건 2	43	7일
실험조건 3	53	9일
실험조건 4	64	11일
실험조건 5	77	13일
실험조건 6	91	15일
실험조건 7	101	17일
실험조건 8	114	19일
실험조건 9	128	21일
실험조건 10	142	23일
실험조건 11	151	25일
실험조건 12	164	27일
실험조건 13	174	29일
실험조건 14	187	31일
실험조건 15	201	33일

주문 변동은 추가, 삭제, 변경이 각각 1/3의 빈도로 발생하도록 설정하였다. 새로운 주문이 추가되는 경우에는 기존 확률분포를 사용하여 주문을 생성하고 변동 리스트에 추가하였다. 주문이 삭제되는 경우는 기존의 주문을 랜덤하게 선택하여 삭제하였다. 주문 변경의 경우 주문을 임의로 선정하여 납기가 1일 또는 2일 당겨지거나 연기되는 네 경우

를 각각 25%의 확률로 생성하였다.

실험 방법은 앞서 언급한 확률분포를 따라 초기 데이터베이스를 확률적으로 생성한 후 전처리 기반의 제약공정 스케줄링 알고리즘으로 시뮬레이션을 수행하였다. 세대수는 문제크기(주문수)의 3배로 설정하였다. 성능의 비교를 위하여 세 가지 조건을 고려하였다. 첫째 조건으로 전처리로 만든 제일 좋은 해를 가져다가 4.3절에 기술한 7개의 과정을 거친 후 결과를 저장하였다. 그리고 둘째의 경우 마지막 주문 데이터를 사용하여 유전자 알고리즘을 3,000회 연산 후의 결과를 저장하였고 셋째의 경우에는 유전자 알고리즘을 15,000회 거친 후 결과를 저장 하였다. 한편, MSCSP에서 스키마는 다음과 같은 사항을 가정하였다.

- ① rolling plan을 바탕으로 향후 50주간 스케줄을 미리 정하고, 주문이 발생 혹은 변경될 때마다 스케줄을 갱신한다.
- ② 모든 주문은 납기를 준수한다.
- ③ 완제품 재고를 줄이고 설계변경에 따른 손실을 가급적 줄이기 위해 모든 주문은 가급적 최대한 늦추어 처리한다.
- ④ 주문 간 셋업시간은 순서의존적(sequence-dependent)이다.

5.2 시뮬레이션 결과

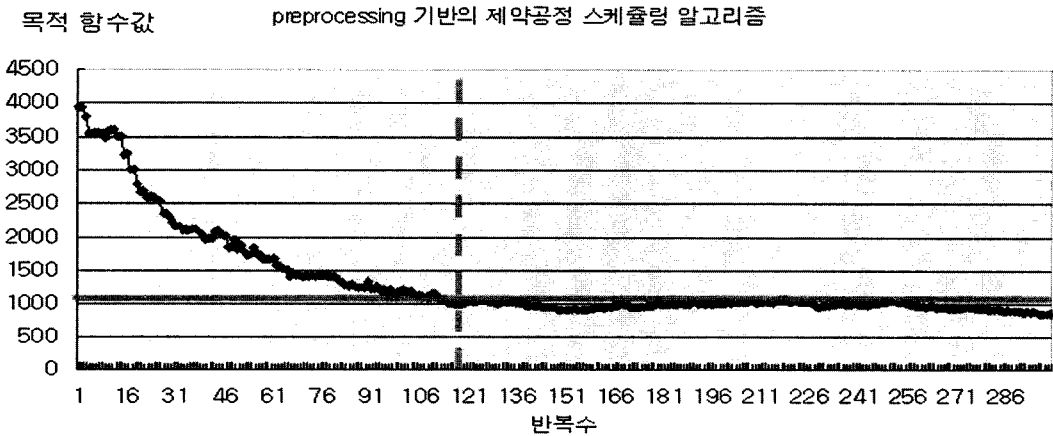
본 절에서는 전처리 기반의 제약공정 스케줄링 알고리즘 성능을 평가하기 위해 일반 유전자 알고리즘 방식과 전처리 기반의 제약공정 스케줄링 알고리즘의 성능을 비교하였다. 먼저 위에서 언급된 실험조건을 기준으로 전처리 기반의 제약공정 스케줄링 알고리즘의 경우 약 2일 동안 전처리중인 해를 매 10분마다 목적함수 값을 총 약 300회 읽어서 도식화한 것이 [그림 5]이다. 이때 전처리 이후 step 7은 약 8분이 소요되었으며, 동일한 8분 동안에 배치 방식의 일반 유전자 알고리즘 기반의 알고리즘은 반복수가 약 3,000회에 달하며 그 목적함수 값의 변화는 [그림 6]과 같다.

적합도 측면에서도 일반 유전자 알고리즘을 사용한 경우에는 [그림 6]에서 보듯이 8분 동안에 목적함수 값이 1,955에 도달하였으나, 전처리 기반의 유전자 알고리즘을 사용한 경우에는 [그림 5]에서 보듯이 동일한 8분 동안에 최종 목적함수 값이 844로서, 일반 유전자 알고리즘에 비해 56.8%가 감소한 훨씬 좋은 결과 값을 얻었다.

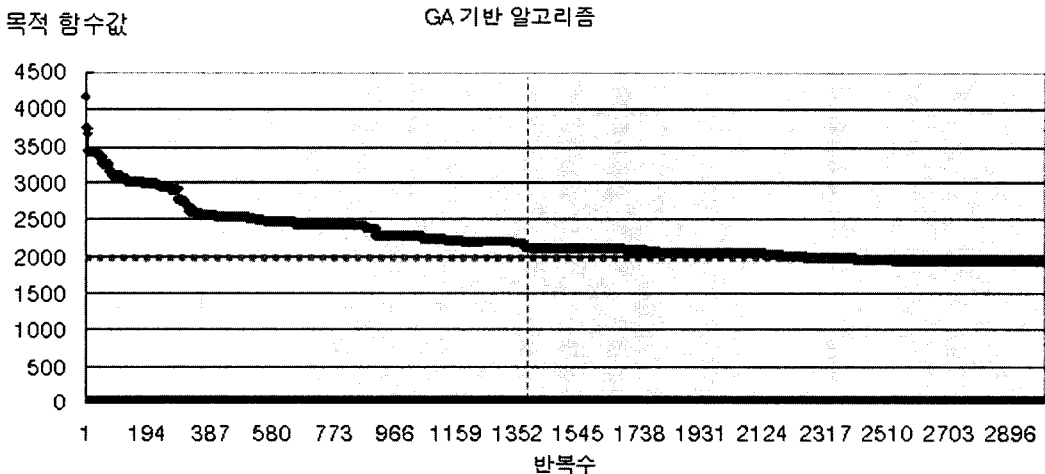
분석하기 위해 29개의 다양한 문제를 풀어 <표 3>의 결과를 얻었으며, [그림 7]은 이를 그래프로 표현한 것이다. <표 3>에서 실험조건이란 <표 2>에 보인 실험조건이며, 총 주문수란 일정기간 동안 추가, 취소, 및 변경된 주문의 합계이고, 시작 작업수란 전처리를 시작하기 전의 주문수를, 그리고 최종 작업수란 전처리가 끝난 후 스케줄을 작성하기 직전의 주문수를 각각 말한다. 또한 목적함수 값에서 PA는 전처리 기반의 제약공정 스케줄링 알고리즘의 목적함수 값을 의미하고, GA(3K)와 GA(15K)는 기존 방식의 유전자 알고리즘을 각각 3,000회

5.3 광범한 시뮬레이션 분석

전처리 기반 유전자 알고리즘의 보편적인 수행도를



[그림 5] 전처리 기반의 유전자 알고리즘 적용 시 목적함수 값



[그림 6] 동일시간 동안 일반 유전자 알고리즘 적용 시 목적함수 값

〈표 3〉 Preprocessing 방식과 일반 방식의 비교

번호	실험 조건	총 주문수	시작 작업수	최종 작업수	목적함수 값		
					PA	GA(3K)	GA(15K)
1	1	412	28	22	132	136	132
2	1	410	28	23	146	148	146
3	3	424	52	57	288	332	288
4	5	311	76	31	182	198	182
5	7	347	100	78	422	604	470
6	4	326	63	56	320	434	336
7	6	371	90	103	494	818	608
8	6	293	90	100	454	942	580
9	5	275	52	80	404	624	490
10	5	255	52	71	352	498	414
11	5	260	52	72	350	538	412
12	8	293	113	101	476	808	576
13	6	281	90	70	372	530	432
14	6	288	90	113	540	928	748
15	9	325	127	136	622	1592	910
16	11	355	150	155	710	1978	1330
17	10	341	141	137	650	1720	1178
18	8	305	113	106	500	992	690
19	13	364	173	162	780	2370	1534
20	12	362	163	162	788	2466	1620
21	9	333	127	143	658	1764	1030
22	15	402	200	197	1012	4574	2862
23	11	338	150	146	760	2470	1500
24	12	322	163	123	594	1754	1112
25	12	346	163	122	556	1270	774
26	14	380	186	177	894	3736	2400
27	13	360	173	168	796	2472	1780
28	15	391	200	202	1132	5130	4016
29	15	397	200	193	958	3970	2476

및 15,000회 반복한 경우의 목적함수 값을 말한다.

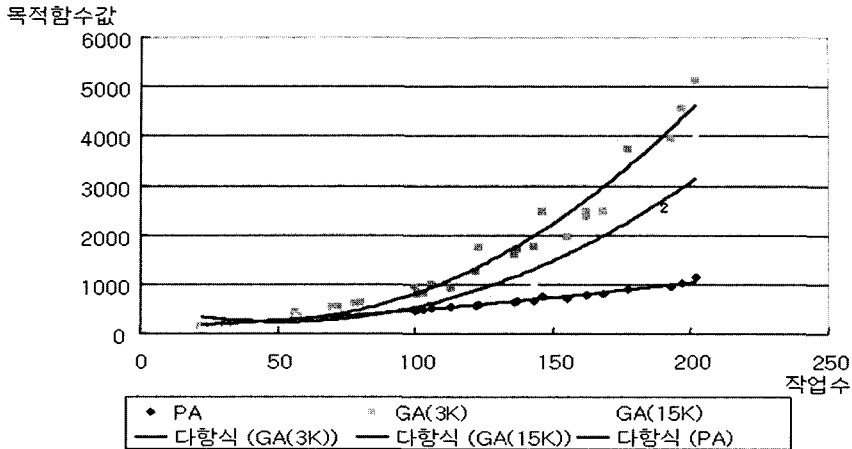
[그림 7]에서 보듯 주문수가 40개 이하에서는 별반 차이가 없으나, 주문의 수가 늘어날수록 PA 방식이 GA(3K)나 GA(15K)보다 훨씬 낮은 목적함수 값을 생성하였다. 전처리 방식이 일반 배치 방식에 비해 더 좋은 성능을 보이는 이유는 일반 배치 방식의 유전자 알고리즘의 경우 주문이 확정된

후 갱신 과정만으로 아주 짧은 시간에 생성한 값이지만 전처리 방식은 이미 성숙된 데이터를 활용하여 몇 시간에서 며칠의 긴 시간동안 연산을 하였기 때문이다. 이러한 이유로 전처리 방식의 목적함수가 훨씬 좋은 값을 가지게 되며, 그 차이는 [그림 7]에서 보듯 주문수가 많을수록 더 큰 차이를 보이는 것으로 나타났다.

6. 결론

본 연구에서는 주문생산 방식의 제조업체가 주문이 확정되면 생산 스케줄링을 작성하는 방식에서 한 걸음 더 나아가 보다 좋은 스케줄을 작성할 수 있는 새로운 접근방법인 전처리 방식을 제안하였다. 전처리란 아직 확정되지는 않았으나 가능성이 높은 주문 정보들을 사용하여 미리 좋은 스케줄을 작성해 두고, 주문이 확정되는 시점에서는 변동분만을 반영하여 짧은 시간 내에 고품질의 생산 스케줄을 작성하는 방법이다. 본 논문에서는 이러한 전처리 기반의 유전자 알고리즘을 제시하였다. 또한 본 연구에서는 여러 지역에 위치한 다수의 공장에 주문을 최적 할당하고 스케줄링하여 납기 지연비용과 운송비의 합을 최소화하는 '복수지역 제약공정 스케줄링 문제' (MSCSP : multi-site constraint scheduling problem)를 정의하고, 개발한 알고리즘을 이용하여 기존 방식보다 빠르고 좋은 해를 구할 수 있음을 보였으며, 문제의 크기가 커질수록 개선 폭은 더 증가하는 것으로 나타났다. 이는 시간경쟁(time-based competition)이 경쟁우위의 한 요소가 되고 있는 오늘날의 치열한 제조업 환경에서 매우 유의한 결과라고 사료된다.

본 논문에서 제시한 전처리 기반의 알고리즘을 사용하면 다수의 공장을 갖는 주문생산방식 제조업체가 수시로 주문 변동이 발생하는 환경에서 주문이 최종 확정된 후 동일한 시간 내에 좋은 스케줄을 작성할 수 있게 된다. 복수지역의 공장할당뿐만 아니라 납기, 셋업시간 등 다양한 제약조건을 동시에 고려한 전체 최적화된 planning이나 scheduling이 가능하기 때문에 자원을 효율적으로 활용하여 비용



[그림 7] 전처리 방식과 일반 유전자 알고리즘 방식의 성능비교

절감을 꾀할 수 있다. 또한 real time 데이터를 활용하기 때문에 공정이 동적으로 바뀌는 생산계획에 대응할 수 있게 되며, 납기확약 시스템의 정밀화와 rolling plan의 안정화, RTE (Real Time Enterprise) 환경의 supply planning의 안정화 등에 기여할 것으로 기대된다.

향후 제안된 알고리즘에서 사용한 유전인자들 사이의 순서 특성과 위치 특성을 더욱 효과적으로 보전시키기 위한 유전 연산자의 개발이 필요하고, 제안된 알고리즘을 더욱 진보시켜 전처리로 인하여 확보된 시간을 더 잘 활용할 수 있는 연구도 필요하다. 본 연구에서 제시된 전처리 개념은 스케줄링 분야뿐만 아니라 배차계획 등 다양한 순서기반 문제에 응용할 수 있을 것으로 사료된다.

참고 문헌

- [1] 김석준, 이채영, "작업일정계획문제 해결을 위한 유전알고리즘의 응용," 한국경영학회지, 제17권, 제3호(1992), pp.1-12.
- [2] Atwater, J.B. and Chakravorty, S.S., "A study of the utilization of capacity constrained resources in drum-buffer-rope systems," *Production and Operations Management*, Vol.11, No.2(2002), pp.259-269.
- [3] Feldmann, M. and Biskup, D., "Single-machine scheduling for minimizing earliness and tardiness penalties by meta-heuristic approaches," *Computers and Industrial Engineering*, Vol.44 (2003), pp.307-323.
- [4] Ghedjati, F., "Genetic algorithms for the job-shop scheduling problem with unrelated parallel constraints : Heuristic mixing method machines and precedence," *Computers and Industrial Engineering*, Vol.37(1999), pp.39-42.
- [5] Koonce, D.A. and Tsai, S.C., "Using data mining to find patterns in genetic algorithm solutions to a job shop schedule," *Computers and Industrial Engineering*, Vol. 38(2000), pp.361-374.
- [6] Kutanoglu, E. and Wu, S., "Improving scheduling robustness via preprocessing and dynamic adaptation," *IIE transactions*, Vol.36, No.11(2004), pp.1107-1124.
- [7] Liu, J. and Tang, L., "A modified genetic algorithm for single machine scheduling," *Computers and Industrial Engineering*, Vol. 37(1999), pp.43-46.

- [8] Simons, J.V., Simpson, W.P., and Carlson, J.J., "Formulation and solution of the drum-buffer-rope constraint scheduling problem (DBRCSP)," *Int. J. of Production Research*, Vol.34, No.(1996), pp.2405-2420.
- [9] Simons, J.V., Stephens, M.D., and Simpson, W.P., "Simultaneous versus sequential scheduling of multiple resources which constrain system throughput," *Int. J. of Production Research*, Vol.37, No.1(1999), pp.21-33.
- [10] Srikanth M.L. and Umble, M.M., *Synchronous management : profit-based manufacturing for the 21st century*, The Spectrum Publishing Co. 1997.
- [11] Steele, D.C., Philipoom, P.R., Malhotra, M.K., and Fry, T.D., "Comparison between drum-buffer-rope and material requirements planning : a case study," *Int. J. of Production Research*, Vol.43, No.15(2005), pp.3181-3208.
- [12] Uzsoy, R. and Wang, C. S., "Performance of decomposition procedures for job shop scheduling problems with bottleneck machines," *Int. J. of Production Research*, Vol.38, No.6(2000), pp.1271-1286.
- [13] Yu, H. and Liang, W., "Neural network and genetic algorithm-based hybrid approach to expanded job-shop scheduling," *Computers and Industrial Engineering*, Vol.39(2001), pp.337-356.
- [14] Wu, H.H. and Yeh, M.L., "A DBR scheduling method for manufacturing environments with bottleneck re-entrant flows," *International Journal of Production Research*, Vol.44, No.5(2006), pp.883-902.