

유연 생산 시스템에서의 최적 버퍼 할당과 다른 요소와의 상호관계에 관한 연구

이용균, 김경섭

연세대학교 산업시스템 공학과 물류 시뮬레이션 연구실

본 연구는 유연 생산 시스템에서 버퍼 할당 최적화 알고리즘을 제시한다. 기존의 연구들과는 달리 제안된 알고리즘은 시스템의 복잡성과 Combinatorial 특성을 모두 다룰 수 있다. 알고리즘은 시뮬레이션을 사용하여 시스템의 복잡성을 모델링하고 수정된 유전 알고리즘을 사용하여 Combinatorial 특성을 다루며 버퍼를 최적으로 시스템에 할당하게 한다. 제안된 알고리즘은 첫 번째 제한된 버퍼가 있는 상황에서 시스템 Output을 최대화하는 목적함수를 사용하여 최적 버퍼 할당을 찾아내는 것과, 두 번째 원하는 Output을 달성할 수 있는 최소의 총 버퍼 수와 할당을 찾아내는 두 가지 목적함수에 적용된다. 마지막으로 유연 생산 시스템의 성능을 결정짓는 다른 요소들과의 관계를 살펴보기 위해 무인 운반 시스템의 발주방식과 무인 운반차의 수 등을 변화시켜 실험을 수행하고 그 결과를 분석한다.

1. Introduction & Literature Review

유연 생산 시스템은 소비자들의 기호가 다양해짐에 따라 생산의 형태를 쉽고 빠르게 변화시킬 수 있는 시스템이 필요해지면서 최근에 대두되었다. 유연 생산 시스템에서 버퍼는 시스템의 최종 Output의 증가와 생산라인의 안정성 보완에 그 목적이 있다. 시스템에 따라서 시스템 내에 버퍼가 하나도 없어도 문제가 발생하지 않는 특별한 경우도 있지만 대부분의 경우 버퍼의 부족은 시스템의 성능을 크게 저하시킨다. 이러한 버퍼는 시스템의 공간적인 제약과 버퍼의 추가에 의해 발생하는 비용 등을 고려하면 시스템 내에서 제한되어 있는 자원으로 생각할 수 있다. 유연 생산 시스템에서의 버퍼의 중요성을 인지하고 있음에도 불구하고 유연 생산 시스템의 복잡성 때문에 수학적 분석이 어려워 많은 연구가 이루어지지 못하고 있다.

기존의 버퍼 할당에 관한 연구를 살펴보면 크게

수학적 접근방법을 사용한 연구, 휴리스틱을 사용한 연구 그리고 버퍼의 최적 할당 특성을 분석한 연구 이렇게 세 가지로 나누어 볼 수 있다. 우선 수학적 접근방법을 사용한 연구들은 시스템이 복잡해지거나 커지면 상태 공간 폭발 등의 문제가 발생하기 때문에 대부분 간단한 생산라인에 대해서만 연구가 수행되었다. 시스템의 복잡성을 모델링 할 수 없는 수학적 접근 방법의 한계를 극복하려는 노력으로 휴리스틱 접근방법이 최근에 많이 제시되고 있다. Lutz, Vorous, Park 등이 Tabu Search, Knowledge Base, Dimension Reduction & Beam Search 등을 이용하여 문제에 접근하였다. 마지막으로 버퍼의 최적 할당 특성을 분석한 연구들은 Inverted Bowl 현상을 분석하고 작업 시간 분포의 변화나 기계의 고장 그리고 이어 지는 작업간의 Coefficient Of Variation의 영향에 Inverted Bowl 현상이 어떤 식으로 변화하는가에 대한 관찰과 분석을 수행한 연구가 대부분이다. 그러나 위의

기존 연구들은 버퍼 할당 문제가 갖고 있는 Combinatorial 특성을 반영하여 복잡한 모델에 적용하기 힘들거나, 모델의 복잡성은 반영을 했으나 특정한 몇 가지 해 집합만을 고려하여 Combinatorial 특성을 반영하지 못하는 문제점이 있다. 본 연구에서는 이러한 한계를 극복하고자 복잡성을 처리하기 위해서 시물레이션을 사용하고 최적 Buffer Profile을 찾아내기 위해 수정된 유전 알고리즘을 사용하는 알고리즘을 제시한다. 결론적으로 버퍼라는 제약된 자원을 최적으로 할당하여 시스템의 최종 Output을 증가시키는 것이 본 연구의 하나의 목적이다. 이와는 반대로, 한 공장의 Output은 시스템 설계 시에 버퍼 할당 부분이 아닌 다른 부분에서 결정이 되므로 이미 존재하는 시스템에 대해서는 주어진 Output에 대하여 총 버퍼 수를 최소화하는 것이 타당하며 본 연구에서는 위의 문제도 제안된 알고리즘으로 해결한다.

2. Algorithm

2.1 System Configuration & Problem Definition

그림<2-1>에 제시된 것처럼 시스템은 7개의 기계와 3대의 무인 운반차, 그리고 Load/Unload Station으로 이루어져 있는 유연 생산 시스템이다. 제품 타입에 따른 생산순서는 4가지 타입 모두 틀리며 한 제품의 같은 기계에서의 작업도 생산 순서 상 다른 작업이라면 작업시간이 틀리다. 기존의 연구들이 하나의 제품 타입만을 고려하는 반면 이 시스템에서 생산되는 제품의 타입은 4가지이며 각 타입은 다른 생산 순서를 가진다. 여러 가지 제품 타입과 생산순서를 고려함으로써 최적 버퍼 할당 문제는 더 복잡해지고 기존의 생산라인 시스템에서 제시된 알고리즘들과 다른 알고리즘을 필요로 하게 된다.

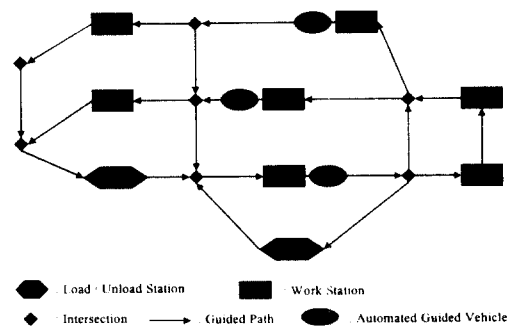


그림 2-1. 유연 생산 시스템 Configuration

본 연구에서 고려하는 첫 번째 문제는 위의 시스템과 같은 유연 생산 시스템에서의 Output을 최대화하는 최적의 버퍼 할당이다. 고려하는 문제에서의 제약으로는 시스템에는 총 할당 가능한 버퍼의 제한과 각 Workstation의 입/출력 버퍼에 할당 가능한 버퍼의 제한이 있다. 이 문제를 수식적으로 표현하면 다음과 같다.

Max Throughput

$$s.t \sum_{i=1}^N b_i \leq \text{Total Buffer Capacity}$$

$$b_i \leq \text{Each Buffer Capacity } i=1...N$$

N : Total Buffer Location (InSystem=14)

두 번째로, 이미 존재하는 시스템에 대해서는 생산계획 등에 의하여 의도된 Output을 만족시키는 최소의 총 버퍼 수와 Buffer Profile을 찾는 것이 더 타당하다. 마지막으로 Value가 다른 타입을 고려하여 전체 Value를 최대화하는 Buffer Profile을 찾는 문제도 고려하며 이 문제들을 수식적으로 표현하면 다음과 같다.

Min Total Buffer Capacity

$$s.t \text{ DesiredOutput} \leq \text{Output}$$

$$b_i \leq \text{Each Buffer Capacity } i=1...N$$

$$\text{Max} \sum_{\text{Part Type } = 1}^4 \text{Output}_{\text{part type}}$$

$$s.t \sum_{i=1}^N b_i \leq \text{Total Buffer Capacity}$$

$$b_i \leq \text{Each Buffer Capacity } i=1...N$$

이 문제들은 Combinatorial 특성을 지니며 시스템이 유연 생산 시스템과 같이 복잡한 시스템이면 기존의 알고리즘으로는 해결할 수 없다. 따라서 본 연구에서는 시뮬레이션을 이용하여 시스템의 Throughput을 측정하고 수정된 유전 알고리즘을 사용하여 각 문제에서 최적 버퍼 할당을 찾는 알고리즘을 제안한다.

2.2 Algorithm Step

본 연구에서는 해의 형태를 14개의 각 버퍼용량을 의미하는 유전자를 갖는 하나의 개체로 구성한다. 예를 들어 하나의 해가 (0, 1, 2, 1, 1, 0, 0, 2, 2, 1, 0, 0, 2, 1)의 형태를 갖는다면 1번 기계의 입/출력 버퍼의 수가 0과 1이 된다는 것을 의미한다.

Step 1. Buffer Profile의 초기 모집단을 생성한다.

유전 알고리즘은 해의 집합을 운영하여 해를 개선시키므로 항상 모집단을 구성해야 하고 모집단의 크기와 초기 모집단 생성 알고리즘은 알고리즘의 성능에 많은 영향을 미친다. 본 연구에서는 모집단의 크기를 10으로 설정하고 초기 모집단은 임의의 수를 생성하는 방법으로 구성하였다. 초기해에서는 각 기계 입/출력 버퍼 용량의 총 합이 시스템 전체의 버퍼 용량 제한을 넘는 비가능해가 나오지 않도록 초기해 임의 생성 방법을 조정하였다.

Step 2. 시뮬레이션을 수행한다.

본 연구에서는 시뮬레이션의 Random 특성을 고려하여 하나의 Buffer Profile당 10번의 반복시행 결과값의 평균을 시스템의 Output으로 사용한다.

Step 2.1 VBA를 이용하여 Arena Model의 각 버퍼 용량을 설정한다.

VBA란 Arena와 Visual Basic을 연결해 주는

Add-In으로 Visual Basic에서 생성된 모집단의 Buffer Profile에 맞게 Arena Model의 버퍼 용량을 설정한다. 첫 번째 세대에서는 Step1.에서 생성된 초기해를 사용하고 다음 세대부터는 Step3.을 거처 나오는 해 집합을 사용하여 Arena Model의 버퍼 용량을 설정한다.

Step 2.2 하나의 버퍼 프로파일, 즉 모집단에서 하나의 해의 시뮬레이션을 주어진 반복회수만큼 반복하여 시스템의 Output을 결정하고 Step 2.3으로 이동한다.

여기서 만약 Buffer Profile의 총 합이 시스템 전체의 제한된 Buffer 크기보다 크다면 비가능해이기 때문에 시스템의 Output을 반으로 줄여준다. 이것은 Step 3에서 비가능해가 다음 세대로 유전되는 확률을 줄여주기 위함이다.

Step 2.3 한 해집합 내의 모든 Buffer Profile에 대한 시뮬레이션이 끝나면 Step3으로 이동하고 아니라면 다음 Buffer Profile로 Arena Model을 설정하고 Step 2.2로 이동한다.

Step 3. 시뮬레이션의 결과값들을 사용하여 새로운 해집합을 생성한다.

Step 3.1 다음세대에 유전될 개체를 선별한다.

선별의 방법에는 여러 가지 방법이 있지만 본 연구에서는 토너먼트 선별방식을 사용한다. 우선 10개의 Buffer Profile중 5개를 먼저 선정하고, 그 다섯 Buffer Profile중 최대의 Output을 보이는 것을 다음 세대로 유전되는 집합에 포함시킨다. 위의 과정을 모집단의 총 수인 10개가 선정될 때까지 반복하여 우선 10개의 유전되는 개체의 집합을 만들어 낸다.

Step 3.2 교차연산을 수행한다.

유전 알고리즘에서 교차연산은 이전 세대의 좋은 해의 형태를 이어가기 위한 연산이다. 본 연구에서는 여러 가지 교차연산 방법 중 일점교차 방

법을 사용한다.

Step 3.3 돌연변이연산을 수행한다.

생태계에서 돌연변이는 대개 나쁜 영향을 미치나 때로는 유익한 돌연변이가 일어나기도 한다. 유전 알고리즘에서 돌연변이는 해의 탐색중 국부해에 빠지는 것을 방지하는 역할을 한다. 본 연구에서는 최적해의 개선이 있을 때에는 돌연변이율을 줄여주고 개선이 없을 시에는 일정한 확률까지 단계적으로 높여주는 방법을 사용하고 있다. 또한, 모델에서 각 버퍼의 효율을 측정해 효율이 크면 버퍼를 추가하는 방향으로 돌연변이 연산을 하고 작으면 제거하는 방향으로 돌연변이 연산을 수행한다. 이렇게 시스템 모델에서 의미 있는 정보들을 돌연변이 연산에 추가함으로써 유전 알고리즘의 성능 개선도 이룰 수 있다.

Step 4. 종료조건을 검사한다.

유전 알고리즘에서 쓰이는 종료조건의 종류에는 최대의 세대수를 사용하거나 원하는 목적함수 값을 달성하는 등 여러 가지가 있으나 본 연구에서는 최적해의 개선이 40세대 동안 일어나지 않으면 종료하도록 종료조건을 설정하였다.

주어진 Output을 만족하는 최소의 Buffer Profile을 찾는 문제에서는 위 알고리즘 단계 중 Step2.2의 비가능해를 구분하는 부분에서 원하는 Output보다 작은 Output을 보이는 Buffer Profile을 비 가능해로 구분해야 하며 Step3.1에서 선별시에 목적함수의 값이 총 버퍼 양이 되어 최소의 버퍼 양을 가지는 Buffer Profile을 선별하여야 한다. 그리고, 각 타입의 Value가 다른 전체 Value최대화 문제에서는 Step3.1에서 목적함수의 값이 각 타입의 Value와 각 타입의 Output을 곱한 것의 합이 되어야 한다.

본 연구에서는 위의 단계들을 Arena와 VBA를

이용하여 지정된 반복시행이 끝날 때마다 자동적으로 수정된 유전 알고리즘이 수행되게 설계하였다.

3. 실험 결과 및 분석

본 연구에서는 Output 최대화 문제에 대한 실험을 위해서 시스템의 총 버퍼 용량을 12개에서부터 30개로 늘어가면서 실험을 수행하였고 각 버퍼의 용량은 모든 실험에서 5개로 제한하였다. 기존에 유연 생산 시스템을 위한 버퍼 할당 알고리즘이 제시되지 않았기 때문에, 제안한 알고리즘의 성능을 비교하기 위하여 단순한 라인 생산 시스템에 적용 가능한 Hillier등이 제시한 알고리즘의 버퍼 할당 시 시뮬레이션의 결과값과 제안된 알고리즘에서의 버퍼 할당 시 결과값을 비교하였다. <표 3-1>에 그 결과값이 표시되어 있으며 제안된 알고리즘의 성능이 총 버퍼 용량이 12일 때를 제외하고 전체적으로 약 6.9% 향상이 되었음을 알 수 있다. 특이한 점은 총 버퍼 용량이 12에서 성능의 차이가 다른 곳에서의 차이보다 훨씬 큰 것을 볼 수 있는데 이는 전체 시스템 성능이 총 버퍼 용량이 작을 때에 버퍼의 추가나 제거에 더 민감하게 반응한다는 기존의 연구 결과와 비슷한 의미를 갖는다. 그러나 본 연구의 결과를 보면 이 때에도 버퍼의 위치를 적절하게 설정한다면 크게 시스템의 성능이 향상함을 알 수 있다. 기존의 연구와 같은 결과를 보이는 것은 Hillier의 알고리즘에서 총 버퍼 양이 12에서 14로 늘어나면서 Output이 크게 늘어나는 것이다. 또한, Hiller의 알고리즘과 제안된 알고리즘 모두에서 총 버퍼 용량이 커져 갈수록 전체적인 성능의 향상이 둔화됨을 보인다.

Park이 주장한 Stagnant Area의 존재는 Hiller 알고리즘 결과에서 버퍼의 총 용량이 늘어나도 시스템 Output이 증가하지 않는 중간 단계가 있는

것으로 알 수 있다. 그러나 돌연변이 연산이 국부 해에 빠지지 않도록 도와주기 때문에 Stagnant Area가 나타나지 않음을 알 수 있다. 두 번째로, 총 버퍼수를 최소화하는 문제는 제안된 Output을 380으로 설정하였고 <표 3-2>의 실험 결과에서 보이듯이 총 8개의 버퍼만 있으면 제안된 Output을 넘도록 생산 할 수 있음을 알 수 있다. 이는 총 버퍼양이 12인 Hillier의 알고리즘에서 Output이 300을 넘지 않는 것에 비하면 상당히 많은 성능의 향상이 있음을 보인다. 마지막으로, 각 타입이 Value가 있는 경우에 대해서는 <표 3-3>에 그 결과가 나타나 있으며 타입의 Value를 고려하지 않은 최대화에서의 Output과는 약 7.2%의 차이를 보여주었다. 여기서 각 타입의 Value는 네 타입에 대하여 15, 3, 5, 7로 설정하였다.

4. 결론 및 향후 연구 방향

본 연구에서 유연 생산 시스템 같은 복잡한 시스템에서 Combinatorial 특성을 지니는 버퍼 할당 문제에 대한 근사 최적해를 시뮬레이션과 수정된 유전 알고리즘을 사용하여 개발된 알고리즘으로 제시했다. 기존의 연구들과 달리 제시된 알고리즘은 시스템의 복잡성과 Combinatorial 특성을 모두 다룰 수 있고 기존의 알고리즘들이 Stagnant Area에 빠질 수 있는 반면 제시된 알고리즘은 Stagnant Area에 빠지지 않음을 보였다. 제시된 알고리즘의 성능 검증을 위하여 Hillier의 알고리즘과 비교를 했고 전체적으로 6.9%의 성능향상이 있음을 보였다. 그리고 제안된 Output이 존재하는 경우의 총 버퍼 최소화에 대한 문제와 각 타입이 Value를 가지는 문제에 대하여 실험을 수행하였고 그 결과를 제시하였다. 향후의 연구로는 시스템의 특성을 약간 변경하여 AGV 발주 방식 등에도 영향을 받도록 한 후 여러 발주 방식과 버퍼 할당과의 상관관

계를 알아보도록 하겠다.

총 버퍼 양	Buffer Profile	Output	차이
12	1,1,2,1,0,1,1,1,0,1,0,1,1,1	404	87.04
14	3,1,1,1,0,1,2,1,0,1,0,1,1,1	424	3.41
16	2,1,2,1,1,1,2,1,0,1,0,1,2,1	437	6.33
20	2,1,2,1,1,1,3,1,1,2,1,1,2,1	454	6.82
25	3,1,3,1,1,1,5,2,1,1,1,1,3,1	470	9.56
30	5,1,3,1,1,1,5,2,1,1,1,3,4,1	475	4.63

표 3-1. 제안된 알고리즘의 Buffer Profile, Output과 Hillier 알고리즘과의 성능차이

Buffer Profile	총 버퍼 양
0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1	8

표 3-2. Min Total Buffer Capacity 실험 결과

총 버퍼 양	Buffer Profile	Output	차이
12	1,1,0,1,1,1,1,1,0,1,1,1,1,1	2810	6.27
14	1,1,3,1,0,2,1,1,0,1,0,1,1,1	3030	11.6
16	1,1,4,1,1,1,2,1,0,1,0,1,1,1	3170	7.02
20	2,1,3,1,0,1,2,1,1,1,1,1,3,2	3220	5.74
25	4,1,5,1,1,2,2,2,1,1,1,1,2,1	3440	8.48

표 3-3. 각 Type의 Value가 다른 실험의 Output과 기존 알고리즘 Output의 차이

Reference

1. H. Pierreval and L. Tautou (1997), Using evolutionary algorithms and simulation for the optimization of manufacturing systems. *IIE Transactions*, Vol 29, 181-189.
2. Hideaki Yamashita and Tayfur Altioek (1998), Buffer capacity allocation for a desired throughput in production lines. *IIE Transactions*,