Journal of the Korean Institute of Industrial Engineers
Vol. 30, No. 4, pp. 285-293, December 2004.

셀룰러 생산시스템에서 생산 리드타임의 최소화를 고려한 셀 구성 방법

임동순[1] · 우훈식[2†]

[1]한남대학교 산업시스템공학과 / [2]대전대학교 인터넷정보공학과

Cell Formation Considering the Minimization of Manufacturing Leadtime in Cellular Manufacturing Systems

Dong-Soon Yim[1] · Hoon-Shik Woo[2]

[1]Department of Industrial and Systems Engineering, Hannam University, Daejeon 306-791

[2]Department of Internet and Information Engineering, Daejon University, Daejeon 300-716

In this study, a machine grouping problem for the formation of manufacturing cells is considered. We constructed the problem as minimizing manufacturing leadtime consisting of parts' processing, moving, and waiting time. Specifically, the main objective of the defined problem is established as minimizing inter-cell traffic in order to minimize the part's moving time. In addition, to reduce the waiting time of parts, the load balance among cells is implicitly included as constraints. Since this problem is well known as NP-complete and cannot be solved in polynomial time, a genetic algorithm is implemented to obtain solutions. Also, a local optimization algorithm is applied in order to improve the solution by the genetic algorithm. Several experiments show that the suggested algorithms guarantee near optimal solutions in a few seconds.

Keywords: cellular manufacturing systems, group technology, genetic algorithm

1. 서 론

셀룰러 생산(cellular manufacturing)은 그룹 테크놀로지(group technology: GT)를 실현하기 위한 생산 철학으로, 유사한 작업물과 기계들을 각각 작업물군과 기계군으로 묶어 생산 유사성의 장점을 이용하는 것이다. 이는 궁극적으로 자재 취급 비용의 감소, 준비시간의 감소, 소수의 재공품 등의 결과를 가져오도록 한다. 그 외에 작업자 숙련도 증가, 공구 조정 능력의 향상, 단순한 스케줄링, 흐름시간 감소 등의 부수적인 결과를 가져온다(Groover and Zimmers, 1984).

기계들을 묶어 셀룰러 생산에서 추구하는 셀로 구성하기 위하여 우선적으로 기계 간의 유사성을 정의하여야 한다. 정의된 기계 간 유사성을 이용하여 큰 유사성을 갖는 기계들은 같은 셀에 있도록 하고, 유사성이 작은 기계들은 서로 다른 셀에 있도록 할 수 있다. 유사성을 위한 자료로 많이 이용되는 것은 0-1 part-machine incidence matrix, timed machine-part matrix, flow machine-machine matrix 등이다. 0-1 part-machine incidence matrix에 기초한 기계 그루핑은 가급적 동일한 작업을 요구하는 작업물과 기계들을 그룹으로 묶는 단순한 방법이다. 만약 두 작업물이 동일한 기계들을 필요로 한다면, 이들 작업물들은 하나의 작업물 그룹으로 묶을 수 있고, 요구되는 기계들 또한 하나의 셀로 묶을 수 있다. 이는 단지 비슷한 작업을 요구하는 작업물을 위하여 기계를 그루핑하는 것으로 작업물의 라우팅, 작업시간, 로트 크기 등을 고려치 않아 실제적인 상황에서 바람직하지 않은 결과를 가져올 수 있다. 이러한 단점을 보완하기 위하여 timed machine-part matrix가 이용될수 있다(Zolfaghari and Liang, 2002). 이는 셀과 기계 간 유사성을 최대화하고, 기계 용량에 대한 작업 부하 불균등을

† 연락저자 : 우훈식 교수, 300-716 대전시 동구 용운동 96-3 대전대학교 정보통신인터넷공학부, Fax : 042-284-0109, E-mail : hswoo@dju.ac.kr
2004년 2월 접수; 2004년 8월 수정본 접수; 2004년 9월 게재 확정.

최소화하는 목적에서 이용된다. 그러나 기계의 작업시간, 로트 크기를 고려하는 장점은 가지지만 0-1 part-machine matrix와 마찬가지로 기계 간 유사성을 직접적으로 표현하지는 못한다. 작업물들의 라우팅에 기초하여 기계 간 작업물 흐름량을 결정한 flow machine-machine matrix는 셀 간의 작업물 이동량을 최소화하는 목적에서 이용된다(Plaquin and Pierreval, 2000). 따라서, 기계 간 작업물 이동량을 유사성으로 간주하나 기계들의 부하를 고려치 않는 이유로 작업량이 균형된 셀 구성을 할 수 없다는 단점을 갖는다.

고려되는 생산시스템에서 각 작업물에 대한 생산 리드타임은 가공시간, 이동시간, 그리고, 대기시간으로 구성되어 있다. 생산 리드타임의 감소 측면에서 주어진 기계들을 셀로 그루핑할 때, 바람직한 일차적인 목적은 이동시간과 대기시간을 최소화하는 것이다. 만약, 이동시간만을 최소화한다면 셀 간의 작업물 이동량을 최소화하여야 한다(Gupta *et al.*, 1995). 셀 내의 작업물 이동 시간에 비해 셀 간의 이동시간이 일반적으로 크기 때문이다. 따라서, 두 기계 간에 많은 작업물의 이동이 발생한다면 이 두 기계는 가급적 같은 셀에 있도록 하는 것이 유리하다. 이 경우 flow machine-machine matrix에서의 두 기계 간 작업물 이동 정도를 유사성으로 설정할 수 있다.

작업물의 대기는 시스템 내에 혼잡이 있을 때 가중된다. 혼잡은 주로 작업물의 공정 순서와 기계 용량에 기인한다. 어느 특정 기계에 용량을 초과하는 작업이 주어지고, 더욱이 그 기계를 요구하는 작업물의 수가 많은 경우 대기시간은 길어진다. 작업량이 상대적으로 많은 이러한 기계들을 같은 셀로 구성한다면 혼잡은 가중되어 대기시간이 더 커질 가능성이 높아진다. 따라서, 혼잡을 줄이기 위하여 각 셀에 비슷한 정도의 작업을 가지도록 셀들의 작업 부하를 균등(load balancing)하게 하는 것이 유리하다. 그러나 이러한 목적에서 두 기계 간의 유사성을 설정하는 것은 쉬운 일이 아니다.

*J*개의 작업물 형태가 있고, 이들을 생산하기 위한 *m*개의 기계가 있다고 하자. 또한, 각 작업물의 형태는 고정된 공정 순서와 공정시간을 갖고 있다고 하자. 이들 *m*개 기계들을 그룹 테크놀로지에 의해 *n*개의 셀로 구성한다고 하자. 본 연구에서는 생산 리드타임을 최소화하는 목적에서 이러한 기계 그루핑 문제를 다룬다. 가공시간은 주어진 상수에 속한다는 가정 하에 여기서 다루는 문제는 작업물 이동시간과 대기시간을 최소화하는 것이다. 이미 언급된 바와 같이 작업물 이동시간의 최소화를 위한 기계 간 유사성 정도의 수치적 표현은 가능하나, 대기 시간의 최소화를 위한 기계 간 유사성 정도의 수치적 표현은 쉽지 않다. 이러한 상황을 고려하여 각 셀의 작업량을 균등하게 하는 함축적 제한 조건하에서 각 셀 간 작업물 이동량을 최소화하는 목적을 고려한다.

본 연구에서 다루는 기계 그루핑 문제는 일반적인 군집화(clustering) 문제와 유사하다. 이러한 군집화 문제는 NP-complete 문제에 속하여 제한된 시간 내에 최적해를 구하기가 어렵다. 따라서, 경험적 방법과 같이 최적해는 아니지만 우수한 해를 빠른 시간 내에 구하는 방법들이 주로 이용된다. 기계 그루핑 문제에 대한 최근의 많은 연구들은 보다 우수한 해를 구하기 위한 방법으로 메타 휴리스틱(meta heuristic) 방법에 속하는 유전자 알고리듬을 이용한다(Gutpta *et al.*, 1995; Plaquin and Pierreval, 2000; Zolfaghari and Liang, 2002). 본 연구에서도 유전자 알고리듬을 이용하며 보다 효과적인 염색체 표현 방법을 모색하였다. 또한, 유전자 알고리듬의 해를 보다 좋은 해로 변환시키기 위한 부분 최적화 방법을 적용하였다. 마지막으로 실험을 통해 제안된 방법의 유효성을 검증하였다.

2. 기호 및 문제 정의

본 연구에서 사용하는 기호는 다음과 같다.

m : 기계 수

p_i : 기계 i에서의 총 가공시간

n : 셀의 수

f_{ij} : 기계 i 에서 기계 j 로의 작업물 이동량

F_{ik}: 기계 i와 셀 k간의 작업물 이동량

T_k : 셀 k에 허용되는 총 가공시간

C : 각 셀에 포함되는 최소 기계 수

G_k: 셀 k에 포함된 기계

x_{ik}: 기계 i가 셀 k에 포함되면 1, 그렇지 않으면 0

기계 i에서의 총 가공시간(p_i)은 정해진 시간 동안 시스템에서 생산될 작업물의 형태, 로트 크기, 공정 순서, 가공시간의 정보로부터 구할 수 있다. 두 기계 간의 이동량인 f_{ij} 역시 이러한 정보로부터 구해진다. 기계와 셀 간의 총 이동량인 F_{ik}의 계산은 다음 식에 의한다.

$$F_{ik} = \sum_{j \in G_k} (f_{ij} + f_{ji})$$

m개의 기계를 n개의 셀로 그루핑하기 위한 문제는 다음과 같다.

$$Min \sum_{k=1}^{n} \sum_{\substack{l=1 \\ l \neq k}}^{n} \sum_{i=1}^{m} \sum_{j=1}^{m} f_{ij} x_{ik} x_{jl} \tag{1}$$

$$s.t) \sum_{i=1}^{m} p_i x_{ik} \le T_k, k = 1, \cdots n \tag{2}$$

$$\sum_{i=1}^{m} x_{ik} \ge C, k = 1, \cdots, n \tag{3}$$

$$\sum_{k=1}^{n} x_{ik} = 1, i = 1, \ldots m \tag{4}$$

$$x_{ik} = 1 \text{ or } 0, i = 1, \dots m, k = 1, \dots n \qquad (5)$$

위 문제의 목적함수인 식 (1)은 셀 간의 작업물 이동량을 최소화하는 것이다. 제한 조건의 첫 번째 식 (2)는 각 셀의 작업 부하를 균등하게 하기 위하여 셀의 작업량(총 작업시간)을 어느 한계 이하로 설정토록 한다. 만약, 제한 조건 없이 위의 목적함수만을 최소화한다면 셀 간의 작업물 이동량을 적게 하기 위하여 다수의 기계가 한 셀로 그루핑될 것이다. 극단적인 경우, 하나의 셀에 모든 기계가 그루핑되어 셀 간의 작업물 이동량은 없게 되기 때문이다. 따라서, 제한 조건에 있는 각 셀의 최대 작업량을 적절히 낮은 값으로 설정하여 작업 균등을 가져오도록 하여야 한다. 그러나, 셀의 최대 작업량을 너무 낮게 설정하면 제한 조건을 만족하는 유효해를 구하기가 불가능할 수 있다. 결국, 바람직한 작업 균등을 이루는 동시에 셀 간 작업물 이동량을 최소화하는 해를 구하기 위한 최대 작업량의 설정이 중요하다.

두 번째 제한 조건인 식 (3)은 각 셀에 포함되는 기계 수의 하한치를 설정한다. 첫 번째 제한 조건에 부가하여 한 셀에 너무 적은 수의 기계가 그루핑되는 것을 방지하는 목적을 갖는다. 식 (4)는 한 기계는 한 셀에만 포함될 수 있다는 것을 의미하며, 식 (5)는 기호에서 설명된 결정변수에 대한 정의를 의미한다.

위에서 정의된 문제는 일반적인 군집화 문제의 특별한 경우로 생각될 수 있다. 일반적인 군집화 문제는 주어진 수의 객체를 어느 수 만큼의 군집으로 묶는 문제로 동일한 군집의 객체 간 유사성은 크게 하고, 서로 다른 군집에 포함된 객체 간 유사성은 적도록 한다. 일반적인 군집화 문제에서 해의 대안 수는 다음과 같다(Bhuyan *et al*., 1991).

$$\frac{1}{m!}\sum_{j=1}^{m}(-1)^{m-j}\,{}_{m}C_{j}\times j^{n}$$

만약, 100개의 기계를 5개의 셀로 그루핑한다면 10^{68} 개 정도의 대안이 있게 된다. 즉, 기계의 수가 증가할수록 해의 대안 수는 기하급수적으로 증가하여 문제의 정확한 해보다는 제한된 시간 내에 근사해를 제공하는 것이 더욱 의미가 있다. 따라서, 정확한 해를 구하기 위한 최적화 방법 보다는 근사해를 구하기 위한 경험적 방법이 많이 사용되고 있다.

본 연구에서 제안하는 기계 그루핑 방법은 유전자 알고리듬과 부분 최적화 방법을 혼용한다. 유전자 알고리듬을 통하여 일차적인 해를 생성하고, 이 해를 부분 최적화 방법에 의해 보다 좋은 해로 변환한다.

3. 유전자 알고리듬

3.1 군집화 문제에서의 유전자 알고리듬

유전자 알고리듬은 여러 개의 개체가 동시에 병렬적으로 주어진 환경에 따라 적자생존의 방법으로 진화하여, 궁극적으로 최적의 상태에 도달하는 생태계의 진화이론에서 도입되었다. 이 알고리듬은 여러 개의 개체로 구성된 군집이 진화할 때 구 세대가 얻은 환경에 대한 정보는 염색체에 저장되어 다음 세대로 전달된다. 이 때 조상의 염색체가 그대로 복제되어 자손에게 전달되는 것이 아니라 조상의 염색체에 교배(crossover), 돌연변이(mutation), 전위(inversion) 등의 연산을 가하여 얻은 염색체로 전달된다. 구 세대 중에서 한 개체가 선택되어 자손에게 유전정보를 남길 확률은 일반적으로 그 개체가 주위 환경, 그리고 나머지 개체와 어떻게 상호 작용하는가에 의존하는 적응값(fitness)에 따라 변한다. 일반적으로 개체 값이 좋을수록 자손을 남길 확률이 높아지는 적자생존의 법칙이 적용된다. 이 알고리듬은 여러 가지 종류의 최적화 문제에 응용되어 좋은 결과를 낳고 있다.

유전자 알고리듬을 기계 그루핑 문제에 적용하기 위하여는 이 문제의 해결에 적합한 염색체의 표현 방법(chromosome representation)과 그 표현에 알맞는 교배, 돌연변이, 전위연산자 등을 정의하여야 한다. 다음에 설명될 군집화에서의 염색체 표현 방법들이 기계 그루핑 문제에 적용될 수 있다.

3.1.1 그룹 번호 표현 방법

기계 그루핑의 경우 각 기계들이 속한 셀 번호를 문자열로 표현하는 매우 단순한 방법이다. 즉, *m*개의 기계가 있을 경우 이를 *n*개의 셀로 분할하는 표현 방법은 $(i_1, i_2, \cdots i_n)$이다. 여기서 *j* 번째 정수 i_j는 *j* 번째 기계가 속한 셀 번호를 의미한다. 제한 조건이 없는 일반적인 군집화 문제에서 그룹 번호 표현 방법은 표준적인 돌연변이, 교배 등의 연산자 사용을 가능케 한다. 그러나, 표준적인 교배 연산자에 의한 자손은 *n*개 이하의 그룹을 가져올 수 있다(Jones and Bertramo, 1991). 또한, 동일한 그루핑 결과를 가져오는 두 부모로부터의 자손은 두 부모의 서로 다른 셀 번호 체계로 인하여 완전히 다른 결과가 될 수 있어, 이를 수정할 수 있는 복구 알고리듬이 요구된다.

3.1.2 분리자를 포함한 순열 표현 방법

이 표현 방법에서 각 그루핑은 *m*+*n*-1 개의 서로 다른 정수로 구성된 리스트로 표현된다. 즉, *m*개의 기계가 있을 경우 이를 *n*개의 셀로 분할하는 표현 방법은 $(i_1, i_2, \dots, i_{m+n-1})$ 이다. 여기서 1부터 *m*까지의 정수는 각 기계를 나타내고, *m*+1부터 *m*+*n*-1까지의 정수는 분리자를 나타낸다. 예를 들어, 그룹 번호 표현 방법에 의한 표현인 (1,1,2,2,2,3,3)은 9개의 정수인 (1,2,8,3,4,5,9,6,7)로 표현되어, 정수 8과 9는 분리자이다. 일반적인 군집화 문제에서의 연산자로는 외판원 문제(Traveling Salesman Problem: TSP)를 위한 각 도시의 순서를 순열로 표현한 방법에서의 연산자를 사용할 수 있다. 그러나, 교배의 경우에 자손이 유효한 결과가 될 때까지 연산자 적용을 반복하여야 한다.

3.1.3 경험적 방법의 해석이 필요한 순열 표현 방법

이 표현 방법에서 각 분할은 m개의 서로 다른 정수로 구성된 리스트로 표현된다(Jones and Bertramo, 1991). 즉, 기계 그루핑의 경우 m개의 기계가 있을 때, 이를 n개의 셀로 분할하는 표현 방법은 $(i_1, i_2, \cdots\ i_n)$이다. 그러나, 이를 하나의 해로 복호화(decoding)하기 위하여 경험적 방법에 의한 부가적인 해석 절차가 필요하다. 그 중 한 방법은 다음과 같다.

우선, 앞에서부터 n개의 수는 n개의 셀을 초기화하기 위한 것으로 각 셀에 속하는 첫 번째 기계 번호를 나타낸다. 나머지 수(기계 번호)들은 하나씩 각 셀에 포함되도록 하는데 적응 값이 가장 크게 되는 셀에 합치도록 한다. 이 방법에서의 연산자는 단순하다. 어느 염색체도 유효한 분할을 가져올 수 있으므로, TSP 문제에서의 일반적인 연산자를 사용 가능케 한다. 물론, 이 표현 방법의 성능은 적응 값을 고려한 염색체 해석 방법으로 인해, 위에서 언급한 두 가지 표현 방법보다 우수한 결과를 가져올 것이다. 그러나 염색체를 해로 복호화하는 데 필요한 부가적인 계산이 요구된다.

위에서 설명된 염색체 표현 방법들을 제한 조건이 있는 기계 그루핑 문제에 적용하기 위해서는 보다 복잡한 절차를 필요로 한다. 그룹 번호 표현 방법과 분리자를 포함한 순열 표현 방법의 경우, 하나의 염색체가 제한 조건을 만족하는 유효한 해를 나타내도록 별도의 방법이 필요하다.

이는 크게 두 가지 절차로 우선 유효한 초기해를 생성하여야 하고, 그 다음에 유효한 자손을 생성하는 유전 연산자를 고안하여야 한다. 예를 들어, 기계 그루핑 문제에 많이 이용되는표현 방법인 그룹 번호 표현에서 Plaquin and Pierreval(2000)는 유효한 초기해를 생성하기 위하여 트리 구조에 기반한 탐색 방법을 사용하였다. 또한, 유효한 해를 생성하는 교배 연산자를 위해 경험적 방법에 기반한 알고리듬을 사용하였고, 돌연변이 연산자로는 재배치와 교환에 기반한 방법을 사용하였다. 단, 유효한 해가 생성되지 않을 경우에는 돌연변이 연산을 취소하도록 하였다.

그러나 경험적 해석 방법이 필요한 순열 표현 방법의 경우에는 어떠한 염색체도 유효한 해를 나타내도록 할 수 있다. 단지, 염색체를 해로 복호화하기 위한 경험적 방법을 고안할 때 제한 조건을 고려하면 된다. 따라서 이 방법에서는 표준적인 연산자의 사용이 가능할 뿐 아니라 보다 우수한 해를 가져올 수 있는 가능성이 존재한다.

3.2 유전자 알고리듬을 이용한 기계 그루핑

본 연구에서 사용하는 유전자 알고리듬의 염색체 표현 방법은 기본적으로 순열 표현 방법이다. 그러나 순열로 표현된 하나의 염색체가 유효한 해를 나타낼 수 있도록 경험적 해석에서와 같은 특별한 복호화 작업이 필요하거나 또는 분리자가 요구된다. 경험적 해석에 의한 방법은 각 객체가 어떤 그룹에 포함되어야 하는지를 결정하기 위하여 평가함수에 의한 적응값을 구하나, 이 작업은 일반적으로 많은 계산시간이 요구되어 효율성이 떨어진다. 분리자를 염색체 표현에 추가하는 방법은 단순함을 유지할 수 있으나, 복구 알고리듬이 요구된다. 본 연구에서는 경험적 해석이 필요한 순열 표현 방법을 이용하고, 보다 단순한 복호화 절차를 고안하였다.

3.2.1 순열 표현 방법

이 표현 방법에서 각 분할은 m개의 서로 다른 정수로 구성된 리스트로 표현된다. 즉, m개의 기계가 있을 경우 이를 n개의 그룹으로 분할하는 표현 방법은 $(i_1, i_2, \ldots, i_n)$이다. 이를 유효한 해로 변환하기 위하여 본 연구에서는 효율성을 증가시키는 단순한 복호화 절차를 사용한다. 즉, 앞에서부터 n개의 수로 n개의 셀을 초기화하고, 나머지 수들을 하나씩 각 셀에 포함되도록 하는데 제한 조건을 만족하는 범위 안에서 이동량이 가장 많은 기계를 포함하는 셀에 우선적으로 할당한다.

예를 들어, 4개의 기계를 순열로 표현하여 (1, 3, 4, 2)라고 하자. 각 기계의 작업 시간이 10이고, 한 셀에 있는 기계들의 총 작업시간이 20으로 제한되어 있고, 각 셀에 하나 이상의 기계가 있어야 한다고 하자. 각 기계 간 이동량이 <Table 1>과 같다면, 2개의 셀로 구성된 유효한 해를 생성하는 복호화 방법은 다음과 같다.

우선, 순열의 앞에 위치한 두 기계인 기계 1과 기계 3을 각각 셀 1과 셀 2에 있도록 한다. 순열의 그 다음 기계인 기계 4는 셀 1과 셀 2의 기계 중 가장 이동량이 많은 기계의 셀과 합친다. 그러나, 제한 조건이 있는 경우 조건을 만족하는 셀이어야 한다. 따라서, <Table 1>에 따라 기계 4는 기계 1이 있는 셀 1에 할당된다. 그 다음 기계인 기계 2는 <Table 1>에 따라 기계 1이 포함된 셀 1에 할당되어야 한다. 그러나 셀 1의 총 작업시간은 30이 되어 제한 조건을 만족하지 못한다. 따라서 기계 2는 셀 2에 할당된다.

Table 1. Traffic between machines

	Machine 1	Machine 2	Machine 3	Machine 4
Machine 1	0	18	15	20
Machine 2		0	12	15
Machine 3			0	10

3.2.2 유전 연산자

본 연구에서의 유전자 알고리듬은 한 세대에서의 초기 염색체들에 대하여 적응 값에 기초한 선택 확률을 구하고 룰렛 휠(roulette wheel) 방식에 의해 염색체들을 선택한다. 선택된 염색체들은 교배와 전위 연산을 통하여 다음 세대의 염색체들을 생성한다. 각 염색체의 적응 값은 본 연구에서 정의된 문제의 목적함수에 기초한다.

순열 표현 방법에 적용될 수 있는 교배 연산자는 외판원 문제에 적용되는 패스(path) 표현 방법에서의 교배 연산자인 PMX(Davis, 1985), OX(Goldberg and Lingle, 1985), CX(Oliver *et al*., 1987) 등을 사용할 수 있다. 본 연구에서는 교배 연산자로 PMX를 사용한다.

순열 표현 방법에서는 전형적인 돌연변이 연산자를 적용하기가 불가능하다. 대신에 전형적인 전위 연산자의 사용은 매우 간단히 적용될 수 있다. 이 연산자는 임의의 두 위치를 선택하여 이 위치의 수를 서로 교환시키나, 제한 조건을 만족하는 경우에만 적용된다.

4. 부분 최적화 알고리듬

본 연구에서는 유전자 알고리듬의 해를 보다 좋은 해로 변환하기 위하여 부분 최적화 방법을 적용한다. 고려되는 부분 최적화 방법은 기존의 K-군집화(K-clustering) 방법에 속한다. K-군집화 방법은 초기에 객체들을 K개의 그룹으로 분리한 후 주어진 평가함수를 최적화시키기 위하여 각 객체들을 바람직한 그룹으로 이동시킨다. 가장 잘 알려진 최적화 기준은 각 객체의 특성치가 벡터로 표현됐을 때 이용되는 오차 제곱합 기준으로 각 객체와 객체가 속한 그룹의 중심 간의 거리에 대한 제곱합이다. 이러한 기준을 사용하는 K-Means 알고리듬(MacQueen, 1967)은 클러스터링 방법으로 많이 사용되고 있다. 특성 벡터 대신에 객체 간의 유사성 계수를 입력으로 수행될 수 있는 K-군집화 방법으로 K-medoid(Kaufman and Rousseeuw, 1990)가 있다. K-medoid 방법은 우선 K개의 대표 객체(representative 또는 medoid)를 결정하고, 대표 객체가 아닌 객체들은 가장 유사성이 높은 대표 객체와 합해 그룹을 구성하도록 하여 해를 생성한다. 그 다음에는 반복적으로 대표 객체와 대표 객체가 아닌 객체들을 교환하여 새로운 해를 생성하고 비용함수 값의 향상을 조사한다. 비용함수로는 각 그룹에 대하여 대표 객체와 대표 객체가 아닌 객체 간의 유사성 합으로 한다.

본 연구에서의 기계 그루핑 문제에 개선된 K-군집화 방법을 적용하기 위하여 두 가지 객체 이동 방법인 재위치(relocation)와 교환(exchange)을 고려한다. 이들 이동 방법은 Harhalakis *et al*.(1990)이 시뮬레이티드 애닐링 방법에서 주변해를 생성하기 위한 방법으로 사용한 것에 기초한다. 재위치는 기존의 K-Means 방법에서와 같이 한 기계를 선택하여 현재의 셀에서 다른 셀로의 이동을 모색한다. 그러나 고려되는 기계 그루핑 문제에서 각 셀의 부하를 균등하도록 하기 위하여 셀의 총 가공시간은 제한되어 있다. 따라서 현재의 해가 부하 균등을 이루고 있다면 재위치시킬 수 있는 기계는 매우 제한적일 수밖에 없어, 보다 좋은 해의 생성은 어려워진다. 이러한 단점을 보완하기 위하여 교환을 부가적으로 실시한다. 교환은 서로 다른 셀에 포함된 두 기계를 선택하여 서로 상대 셀로의 이동을 모색한다.

결국 한 기계가 현재 속한 셀에서 다른 셀로 이동을 했을 경우, 또한, 서로 다른 두 셀의 기계들을 서로 교환하는 경우, 평가함수 값(z)의 변화량을 계산하고, 변화량에 따라 기계의 이동 여부를 결정한다. 기계 i가 셀 k 에 속해 있다고 하자. 기계 i의 셀 내부 이동량(intra-cell traffic)은 F_{ik}이고, 기계 i와 다른 셀 l 간의 이동량(inter-cell traffic)은 F_{il}이다. 이 기계 i를 셀 l 로 재위치시킨다고 하자. 재위치에 따르는 목적함수 값의 변화량(Δz)과 기계와 셀 사이의 변경된 이동량(F_{ij}')은 다음과 같다.

$$\Delta z = F_{ik} - F_{il}$$

$$F_{ik}' = F_{ik}$$

$$F_{il}' = F_{il}$$

$$F_{vk}' = F_{vk} - f_{vi} - f_{iv}, v \neq i$$

$$F_{vl}' = F_{vl} + f_{vi} + f_{iv}, v \neq i$$

또한, 셀 k에 속해 있는 기계 i와 셀 l에 속해 있는 기계 j를 교환할 때 목적함수 값의 변화량과 변경된 이동량은 다음과 같다.

$$\Delta z = F_{ik} - F_{il} + F_{jl} - F_{jk} + 2\,(f_{ij} + f_{ji})$$

$$F_{ik}' = F_{ik} + f_{ij} + f_{ji}$$

$$F_{il}' = F_{il} - f_{ij} - f_{ji}$$

$$F_{jk}' = F_{jk} - f_{ji} - f_{ij}$$

$$F_{jl}' = F_{jl} + f_{ji} + f_{ij}$$

$$F_{vk}' = F_{vk} - f_{vi} - f_{iv} + f_{vj} + f_{jv}, v \neq i, j$$

$$F_{vl}' = F_{vl} - f_{vj} - f_{jv} + f_{vi} + f_{iv}, v \neq i, j$$

위 식들을 이용한 부분 최적화 방법은 다음과 같다.

4.1 부분 최적화 알고리듬

입력: (초기해)

$c_i, i = 1, ..., m$ // 기계 i 가 속한 셀 번호

$f_{ij}, i = 1, ..., m, j = 1, ..., m$

// 기계 i에서 기계 j로의 이동량

$p_i, i = 1, ..., m$ // 기계 i 에서의 총 가공시간

$T_k, k = 1, ..., n$ // 각 셀에 허용되는 최대 총 가공시간

출력: (새로운 해)

$c_i, i = 1, ..., m$ // 기계 i 의 새로운 셀 번호

절차:

1. 초기화

1.1 각 기계(i)에 대해 셀(k) 간의 총 이동량(F_{ik})을 구한다.

1.2 각 셀(k)에 대하여 총 가공시간을 구한다.

2. 다음 절차를 재위치와 교환이 불가능할 때까지 반복한다.

2.1 (재위치) 다음 절차를 기계의 재위치가 불가능할 때까지 반복한다.

2.2.1 모든 각 기계(i)에 대하여 다른 각 셀로 이동을 했다고 가정했을 때 제한 조건을 만족한다면 평가함수의 순 변화량 Δz 을 구한다.

2.1.2 Δz 중 가장 작은 것을 선택하여 $\Delta z < 0$이면 해당 기계를 이동시키고, 각 기계에 대해 새로운 F_{ik}를 구한다.

2.2 (교환) 다음 절차를 두 기계의 교환이 불가능할 때까지 반복한다.

2.2.1 모든 각 기계(i)이 대하여 다른 각 셀의 각 기계와 교환했다고 가정했을 때 제한 조건을 만족하면 평가함수의 순 변화량 Δz 을 구한다.

2.2.2 Δz 중 가장 작은 것을 선택하여 $\Delta z < 0$이면 해당되는 두 기계를 교환시키고, 각 기계에 대해 새로운 F_{ik}를 구한다.

5. 실험 및 분석

5.1 최적해를 아는 경우

최적해를 아는 경우의 문제를 위해 Plaquin and Pierreval (2000)의 연구에서와 유사한 방법으로 데이터를 생성하였다. 총 61개의 기계를 7개의 셀로 구성하도록 기계 간 작업물 이동량을 설정하였다. 즉, 61×61의 기계 간 작업물 이동량 행렬을 7개의 서로 다른 블록으로 나누어, 대각선을 제외한 블록 안의 값은 0부터 4사이의 난수로 하고, 나머지 부분은 0으로 하였다. 이에 부가하여 각 기계의 작업시간은 10으로 하고, 모든 셀에 대하여 총 작업시간은 200, 최소 기계 수는 2로 하였다. 따라서, 최적의 적응값은 0이 된다.

개체 수를 100으로 하고, 교배확률을 0.6, 전위확률을 0.1로 설정하여 유전자 알고리듬을 적용한 결과, 첫 번째 세대에서 최적해를 생성하였다. 이는 유전자 알고리듬의 염색체 표현 방법이 최적의 해를 매우 빠르게 생성하는 효과적인 방법임을 의미한다. 제한 조건이 본 연구의 문제와 다르나, 목적함수가 동일한 경우의 유전자 알고리듬을 제안한 Plaquin and Pirreval (2000)의 연구 결과가 100세대 이상에서 최적해를 생성하였다는 것을 고려할 때 본 연구에서 제안된 유전자 알고리듬의 우수성을 입증한다.

5.2 최적해를 모르는 경우

최적해를 모르는 경우의 문제를 위해 7개의 셀 구성을 위한 61개의 기계를 대상으로 모든 기계 간 작업물 이동량을 0부터 5 사이의 난수로 생성하였고, 각 기계의 작업시간은 5부터 15 사이의 난수로 생성하였다. 또한, 각 셀당 최소 기계 수는 2로 하였고, 각 셀에 제한된 총 작업시간은 모든 셀에 동일하게 하였다.

이렇게 생성된 데이터를 대상으로 셀에 제한된 총 작업시간을 80부터 150까지 10씩 증가하도록 하여 총 8번의 실험을 수행하였다. 기타 유전자 알고리듬에 필요한 파라미터는 최적해를 아는 경우의 실험과 동일하게 설정하였다.

<Figure 1>은 각 셀의 총 작업시간을 100으로 제한했을 때 유전자 알고리듬만을 적용한 경우와 이에 부가하여 국부 최적화

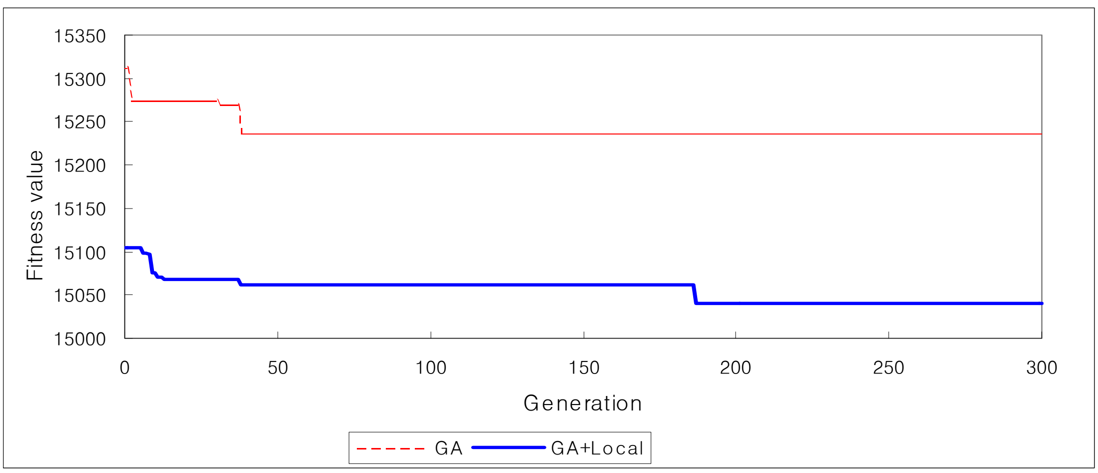

Figure 1. Trends of solutions.

알고리듬을 적용한 결과를 나타낸다. 그림은 세대수에 따른 해의 수렴을 나타내어 유전자 알고리듬만을 적용한 것에 비하여 각 세대의 해에 국부 최적화 알고리듬을 부가한 경우가 보다 좋은 해를 가져옴을 보여준다.

<Table 2>는 최대 작업시간 설정에 따른 유전자 알고리듬과 이에 부가한 국부 최적화 알고리듬의 결과, 그리고 각 셀의 총 작업시간을 나타낸다. 유전자 알고리듬의 해에 비해 국부 최적화를 부가한 알고리듬의 해가 셀 간의 작업물 이동량에서 평균 1.2% 향상된 결과를 보인다. 또한, 최대 작업시간의 증가에 따라 셀 간의 작업물 이동량은 감소하나, 셀들의 총 작업시간에 대한 표준편차는 증가한다. <Figure 2>는 표준 편차의 증가에 따른 적응 값의 추이로서, 이러한 결과를 잘 나타낸다. 표준편차로 표현되는 셀 간의 작업량 균등과 적응 값으로 표현되는 셀 간의 흐름량 감소는 서로 상반된 성격을 갖는다. 즉, 각 셀의 작업량을 균등화시킬수록 셀 간의 흐름량은 증가하는 특성을 갖는다.

각 알고리듬은 펜티엄 4의 컴퓨터에서 JAVA 언어로 프로그래밍되었다. 300세대 동안 유전자 알고리듬만을 적용했을 때 평균 4초 정도의 계산시간이 소요되었고, 이에 부가하여 국부 최적화 알고리듬을 적용했을 때 평균 5초 정도가 소요되었다. 따라서, 국부최적화 알고리듬은 평균 1초 정도의 계산시간이 소요돼 짧은 시간 안에 근사 최적해를 도출하였다.

5.3 논의

지금까지의 결과에서 보듯이 셀 간 작업물 이동량의 최소화와 셀 간의 작업 균등이라는 두 가지 목적은 서로 상반된 특성을 갖고 있다. 더욱이 두 목적은 동일한 척도로 표현될 수 없는 이유로 이 둘을 조화시키는 최적의 해를 도출하기가 어렵다. 최적해를 모르는 경우의 실험 결과에서와 같이 작업물 이동량만을 최소화한다면 가급적 하나의 셀에 많은 기계들을 그루핑하는 것이 유리하게 되어 부하가 불균형된 셀들을 형성하게 된다. 반대로 부하 균등만을 최대화한다면 작업물 이동량을 고려치 않아 셀 간의 작업물 이동이 많아지게 되는 결과를 초래한다. 따라서, 본 연구에서 정의된 문제의 경우에 각 셀들에 허용되는 최대 작업량을 적절히 설정하여 바람직한 결과를 가져오도록 하여야 한다. 이를 위한 방법 중의 하나는 본 연구에서 수행한 것과 같이 각 셀에 제한한 최대 작업량을 변경하여 이에 따른 결과를 이용하는 것이다.

실제적인 문제의 경우, 기계 간 작업물 이동량은 최적해를 아

Table 2. Experiment results

Max. processing time	traffic between cells		cell processing times							
	GA	GA + Local	cell 1	cell 2	cell 3	cell 4	cell 5	cell 6	cell 7	standard deviation
80	15437	15271	79	78	79	80	76	59	80	8
90	15153	14971	88	89	90	90	89	61	24	25
100	14841	14683	99	86	27	98	100	97	24	35
110	14619	14414	110	109	21	27	104	110	50	41
120	14250	14092	119	21	104	119	22	27	119	49
130	14142	13895	127	130	27	17	127	79	24	53
140	13744	13605	140	137	137	24	22	47	24	59
150	13364	13248	146	144	26	149	20	21	25	66

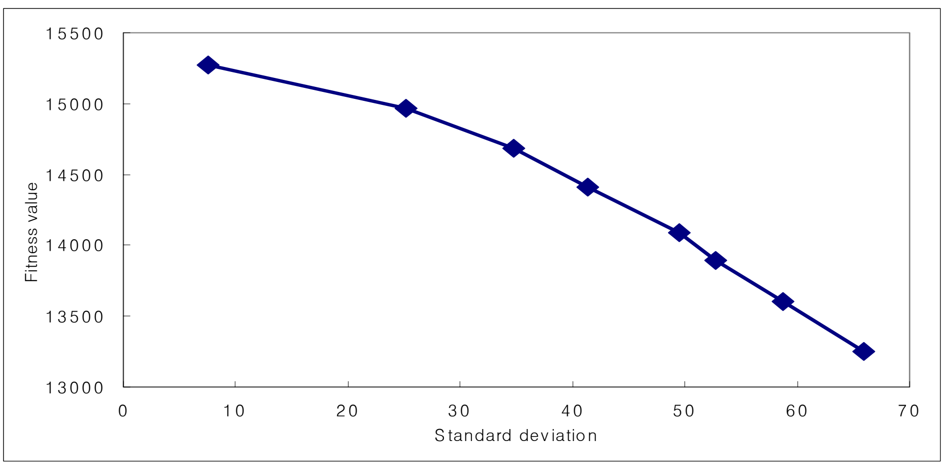

Figure 2. Fitness values with standard deviations.

는 경우와 최적해를 모르는 경우(모든 기계 간 이동량을 난수에 의해 할당) 사이의 특성을 가지게 될 것이다. <Figure 3>은 이전에 설명된 최적해를 아는 경우의 데이터에 노이즈를 추가하여 생성한 61개 기계 간 작업물 이동량 행렬로서 실제적인 상황에 근사하다고 할 수 있다. 두 기계 간 이동량이 클수록 그림의 행렬 성분에 보다 어두운 색의 점으로 표시하였다. 기타 다른 데이터는 최적해를 아는 경우와 동일하게 하여 유전자 알고리듬과 국부 최적화 알고리듬을 적용한 결과는 671의 적응 값을 가지고, 그림의 사각형으로 그루핑되어 바람직한 해를 가져옴을 알 수 있다. 그러나, 유전자 알고리듬만을 사용한 경우는 해가 약간 상이하여 적응값이 680이었다.

한 셀에 최대 20개의 기계가 그루핑될 수 있지만(각 셀의 총 작업량은 200으로 설정), 실험 결과는 이동량 데이터의 특성으로 인하여 원하는 해를 생성할 수 있음을 보여준다. 각 셀에 허용되는 총 작업시간을 150으로 줄여도 위와 동일한 결과를 생성하였다. 이는 최적해를 모르는 경우의 실험 결과에 비해 최대 작업량 설정이 해에 큰 영향을 가져다 주지 못함을 의미한다.

6. 결 론

셀룰러 생산을 위한 기계 그루핑 문제는 NP-complete 문제에 속하여 최적해를 구하기 위하여 많은 계산시간을 필요로 한다. 본 연구에서는 유전자 알고리듬과 국부 최적화 알고리듬을 혼용하여 우수한 근사 최적해를 짧은 시간 내에 구할 수 있는 방법을 도출하였다. 특히, 셀 간 작업물 이동량을 최소화하고, 셀 들 간의 작업 부하를 평준화하기 위한 함축적 제약식을 써서 정형적인 문제를 정의하였다. 정의된 문제의 해를 구하기 위한 유전자 알고리듬은 염색체를 순열로 표현하여 이를 유효한 해로 복호화하는 방법을 사용하였다. 유전자 알고리듬에 의한 해를 보다 좋은 해로 변형하기 위한 국부 최적화 알고리듬은 기계의 재위치와 두 기계의 교환 방법에 기초하였다.

제안된 알고리듬의 유효성과 성능을 분석하기 위한 실험을 통하여 셀 간 작업물 이동량 최소화와 셀 들의 작업 부하 균등화 목적은 서로 상반된 결과를 가져옴을 알 수 있다. 따라서, 보다 바람직한 해를 도출하기 위하여 작업 부하 균등을 위한 제한 조건의 설정이 매우 중요하다. 그러나 기계 간 작업물 이

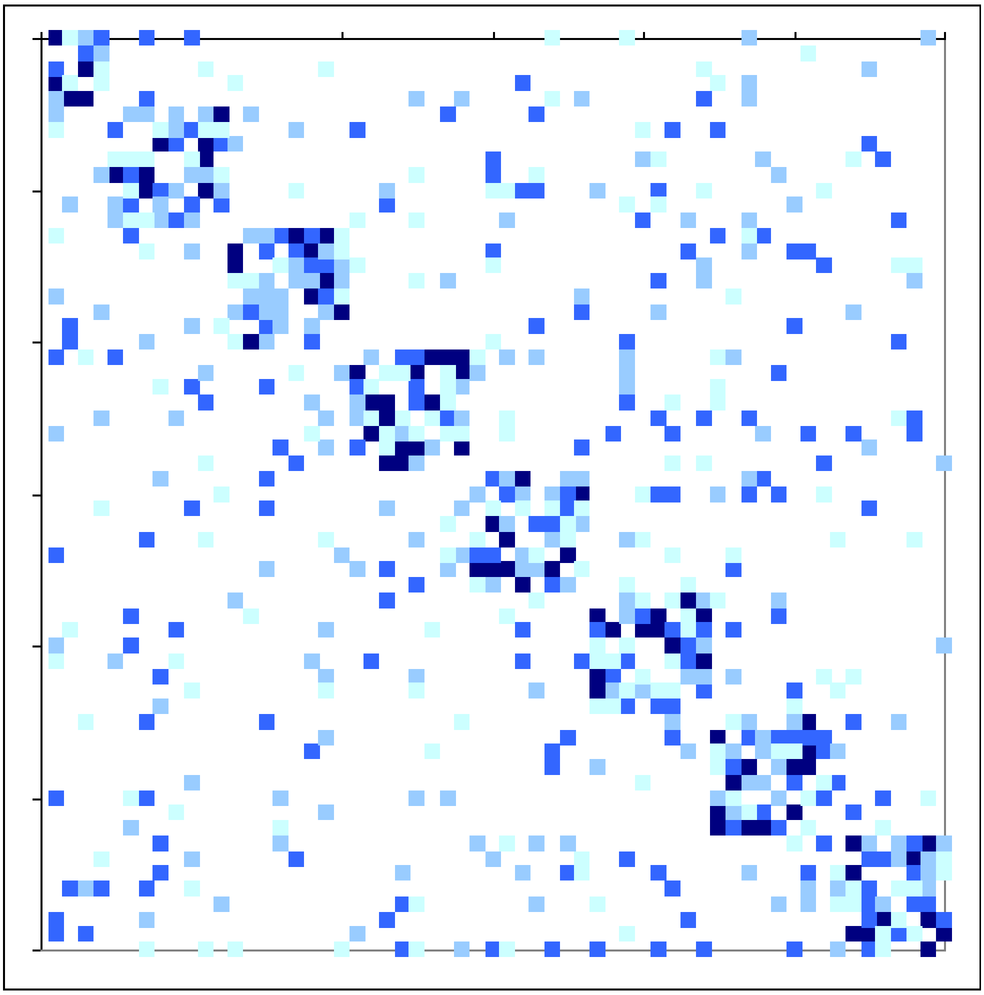

Figure 3. Flow matrix similar to practical cases.

동량 데이터가 최적해를 도출하기 용이한 경우에 근사할수록 작업 균등을 위한 제한 조건은 큰 의미를 가지지 못한다.

참고문헌

Bhuyan, J. N, Raghavan, V. V., and Elayavalli, V. K. (1991), Genetic Algorithm for Clustering with an Ordered Representation, *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, Los Altos, CA, 408-415.

Davis, L. (1985), Applying Adaptive Algorithms to Epistatic Domains, *Proceedings of the International Joint Conference on Artificial Intelligence*, 162-164.

Goldberg, D. E. and Lingle, R. (1985), Alleles, Loci, and the TSP, *Proceedings of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, Hillsdale, NJ, 154-159.

Groover, M. P. and Zimmers, E. W. (1984), CAD/CAM: Computer Aided Design and Manufacturing, Englewood Cliffs: Prentice Hall.

Gupta, Y. P., Gupta, M. C., Kumar, A, Sundrum, C. (1995), Minimizing total intercell and intracell moves in cellular manufacturing: A genetic algorithm approach, *International Journal of Computer Integrated Manufacturing*, **6**(2), 92-101.

Harhalakis, G., Proth, J. M., Xie, X. L. (1990), Manufacturing cell design using simulated annealing: An industrial application, Journal of Intelligent Manufacturing, **1**, 185-191.

Jones, D. R. and Bertramo, M. A. (1991), Solving Partitioning Problems with Genetic Algorithms, *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, Los Altos, CA, 442-449.

Kaufman, L. and Rousseeuw, P. J. (1990), *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, Inc.

MacQueen, J. (1967), Some Methods for Classification and Analysis of Multivariate Observations, *Proceedings on 5th Berkeley Symposium*, 281-297.

Oliver, I. M., Smith, D. J., and Holland, J. R. C. (1987), A Study of Permutation Crossover Operators on the traveling Salesman Problem, *Proceedings of the Second International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, Hillsdale, NJ, 224-230.

Plaquin, M. and Pierreval, H. (2000), Cell formation using evolutionary algorithms with certain constraints, *International Journal of Production Economics*, **64**, 267-278.

Zolfaghari. S. and Liang, M. (2002), Comparative study of simulated annealing, genetic algorithms and tabu search for solving binary and comprehensive machine-grouping problems, *International Journal of Production Research*, **40**(9), 2141-2158.