

a hybrid approach for jobshop scheduling with lot-streaming

김인수, 서운호

고려대학교 산업시스템정보공학과 {aJava, yoonhoseo}@korea.ac.kr

Abstract

The planning and scheduling is considered as an important function because it influence to the overall performance of production in manufacturing system for supporting global objectives. In this paper, propose the hybrid approach to solve the problem which considered alternative resources and sequences, a sequence-dependent setups and transportation times. The objective of this problem is minimize the makespan with AGV traveling time. A iterative two heuristic algorithm(PSO, SA) is proposed to obtain the sequence of operations and CPLEX package is used to obtain the optimal sub-lot sizes(OLS) when the sequence of operations and greedy-based AGV dispatching rules are fixed or given. we show that performance of iterative two heuristic algorithms and simple numerical experiment.

1. 서론

제조공급망에서 생산계획(planning)과 일정계획(scheduling)은 공장 내의 자원들을 제품생산 작업에 할당 하는 것과 제품의 전반적인 성능을 규정짓기 때문에, 포괄적인 목적들에 영향을 미치는 주된 역할을 한다. 한정된 자원들을 이용하여 생산 업무를 하는 경우 자원의 제약에 의해 마감일을 지키지 못해 손해를 얻는 경우가 있다. 이러한 경우에 제조사들은 대체 자원, 대체 라인, 아웃소싱을 이용하게 되며 이를 대비하기 위해 생산계획과 일정계획은 사전에 준비되어야 한다.

또한, 다중 설비시설 환경에서 각 시설은 드릴링, 밀링, 유연한 자원 모듈 등과 같은 자동화된 자원들을 갖추고 있고, 각 자원들은 각기 다른 용량을 갖으며 제한이 있어서 어떤 부품을 생산하는 시간이 서로 다르다. 대부분의 부품을 생산하는데 있어서 하나의 자원에서 모두 처리되지 않고 여러 공정을 걸쳐 제조되거나 여러 하위 부품을 조합하여 제품을 만들기 때문에 서로 다른 설비에 부품을 운반해야 한다. 보통 AGV(Automatic Guided Vehicles)와 같은 운송기계를 이용하게 되는데, AGV가

설비간에 운반하는데 걸리는 시간은 공정을 지연시키는 역할을 하기 때문에 일정계획에 목적을 달성하는데 영향을 끼치게 된다. 또한, AGV가 서로 다른 두 작업장 사이에서 부품을 운반할 때 한 번에 운반되는 로트를 적재단위(unit load), 적재단위에 포함된 부품의 개수를 적재단위 크기(unit load size)라 하는데, 이 크기는 AGV가 정해진 부품 크기만큼을 몇 번 운송해야 하는지 즉, 운송시간과 밀접한 관련이 있어서 물류 시스템에서 중요한 요소 중 하나로 여겨지고 있다.

기존의 일정계획을 개선하기 위한 기법인 LS(Lot Streaming)는 Reiter(1966)에 의해 처음 소개되었으며 공정들의 오버랩을 이용하여 오늘날 글로벌 경쟁시대에 대응하기 위하여 공정시간을 줄이기 위한방법으로 현재까지 많은 연구가 진행되어 왔다. 하지만 대부분 Flow shop에 대한 연구이며, open shop과 job shop에 대한 연구는 상대적으로 적은 연구가 진행되어 왔다. Stephane, Lasserre(1997)에서는 혼합정수계획법 모델을 설계하고 주어진 부분로트의 개수와 일정계획에 대해 최적의 부분로트 크기를 구하는 단계 그리고 전 단계에서 얻은 최적의 부분로트 크기를 이용하여 일정계획을 개선시키고, 일정 횟수의 해 개선이 없을 동안 반복하는 알고리즘을 제시하였다. Chinyao Low, et al. (2004)에서는 부분로트들의 크기가 같은 그리고 같지 않은 경우 그리고 LS가 일정계획이 시스템 성능에 미치는 영향을 분석하고 최적의 LS 일정계획을 이용한 총 비용과의 관계에 대해서 연구하였다. J. H. Chang, H. N. Chiu(2005)에서는 LS에 대해 크게 3개의 범주로 나누고, 7개의 하위 범주와 17개의 단계로 지금까지 연구해온 논문들에 대해 구분 짓고, 7개의 표기법을 정하여 관련 논문들의 특징을 분류 및 설명하였다. John Jackman(1991)은 Jobshop 환경에서 큐잉 네트워크를 이용하여 적재단위당 운송비용과 보유비용을 최소화하기 위한 적재단위크기를 결정하는 방법론을 제시하였다.

AGV 할당과 적재단위 크기에 대한 연구는 S.W. MOON, H. HWANG(1999)가 다중 제품, 다중 조립라인에서 AGV 할당규칙이 주어졌을 때, 경제적인 적재단위 크기를 결정하는 모델을 제시하고, 그 모델의 해 영역의 특징을 연구하여 근사해를 구하는 휴리스틱 방법을,

Benita M. Beamon, et al(1998)에서는 적재단위 크기와 차량대수를 동시에 고려하여 최적화 하는 모형을 제시하였다.

또한, W.J. Xia, Z.M. Wu(2006)는 최근 많은 연구가 진행되고 있는 PSO(Particle Swarm Optimization)를 이용하여 JSP(Jobshop Scheduling Problem)의 효과적인 해를 얻을 수 있는 방법을 제시하였다.

Lot Streaming을 이용하여 제품의 생산시간을 효율적으로 줄일 수 있다는 것은 이미 많은 관련 연구가 되어 있다. 하지만 부품의 운반수단인 AGV의 대수가 고정되어 있을 경우, 부분로트 개수, 부분로트 크기 그리고 AGV 할당 규칙에 따라 생산 시간이 어떻게 변화되는지 두 요소의 관계에 대해 연구해 보고자 한다.

2장에서는 다중 설비시설 환경에서의 문제에 대한 정의를 3장에서는 공정의 일정계획을 결정하기 위해 적용한 휴리스틱 알고리즘인 PSO와 SA를, 그리고 최적의 로트 크기를 결정하기 위한 ILOG사의 CPLEX 패키지를 이용한 근사해 도출방법(OLS)을 제안한다. 4장에서는 적용한 PSO의 성능과 본 연구에서 분석하고자 하는 실험과 결과를 보이고 분석을 통하여 5장에서 결론과 추후 연구에서의 개선사항 및 방향에 대해서 기술한다.

2. 문제 설명

보통 제조사들은 제한된 마감일을 지키지 않으면 큰 손해를 보기 마련이다. 이를 위해 보유하고 있는 설비들을 효율적으로 이용하여 최적의 성능을 원하고 있다. 다중 설비시설 환경에서 각 시설은 드릴링, 밀링, 유연한 자원 모듈 등과 같은 자동화된 자원들을 갖추고 있고, 각 자원들은 각기 다른 용량을 갖추고 제한이 있어서 어떤 부품이 생산되는 시간은 어느 설비에서 생산되느냐에 따라 시간이 다르기도 한다. 또한, 대부분의 제품을 생산하는데 있어서 하나의 설비에서 모두 생산되지 않고 여러 부품들을 서로 다른 공정을 걸쳐 제조하고 조립하여 제품을 만들기 때문에 설비에서 생산된 부품을 다른 설비로 운송하기 위해 운송기계를 이용하게 된다. 서로 다른 두 설비 간 운반할 때 걸리는 운송시간(Transportation time)은 생산된 부품의 다음 공정을 지연시키는 역할을 하기 때문에 일정계획의 목적을 달성하는데 영향을 끼치게 된다. 이 밖에도 각 설비별 생산시간, 제품별 선행관계가 있는 작업의 순서제약(precedence constraints), 순서기반 준비시간(sequence-dependence setup time), 대체공정순서(alternative sequences), 대체자원(alternative resources) 등이 있다.

보통 운반수단으로 이용되는 AGV는 적재단위 크기만큼의 로트를 싣고 운반하게 된다. 하

지만, 최대 적재단위 크기가 고정되어진 상태에서 항상 모든 운반 작업에서 적재단위 크기의 로트를 싣고 운반할 필요는 없을 것이다. 예를 들어, 적재단위 $v_k=5$ 그리고 로트 크기 $q_k=7$ 이라고 가정하자. 이러한 경우에 적재단위 개수는 최소 $2(= \lceil 7/5 \rceil)$ 에서 최대 $7(= \lfloor 7/1 \rfloor)$ 까지의 범위를 갖게 된다. 예를 들어 적재단위 개수가 최소인 2라고 가정하면 (2,5), (3,4) 그리고 (4,3)의 조합으로, 3인 경우는 (1,1,5), (1,2,4), (1,3,3), ..., (5,1,1)의 조합, 최대 크기 인 7인 경우에 (1,1,1,1,1,1,1)인 한 가지 방법으로 나타낼 수 있다. 주어진 로트크기와 최대 적재단위 크기를 고려하여 가능한 조합 중에서 제품 생산의 효율성을 최대화 하는 적재단위 개수와 적재단위 크기를 선택하여 결정할 수 있다. 이러한 경우 공정의 생산시간을 줄이기 위해 도입된 개념인 Lot Streaming 관점에서 보면 부분로트 크기가 최대 적재단위 크기보다 작아야 한다는 조건이 있는 Lot Streaming과 AGV를 고려한 문제라고 볼 수 있다.

공정들이 생산되는 순서와 생산될 설비 그리고 로트크기를 최적으로 찾아내는 것은 아주 작은 크기의 문제라 할지라도 상당한 시간을 요구한다. 따라서 본 연구에서는 로트크기가 주어졌을 경우 (i)생산순서 및 설비를 결정하는 부분(sec. 3.2, 3.3)과 결정된 생산순서 및 설비를 바탕으로 (ii)최적의 로트크기를 찾아내는 부분(sec. 3.4)으로 나누어 반복적으로 해를 찾는 알고리즘을 제안한다.

3. 해법 개발

본 연구에서 사용된 PSO는 1995년 J. Kennedy와 R. Eberhart에 의해 소개되었고, GA(Genetic Algorithm)와 유사하게 집단을 기반으로 해를 탐색하는 기법을 사용하고 있다. 그러나 유전적 연산인 교배나 돌연변이 기법을 사용하는 GA와 달리 PSO에서는 각 입자(particle)가 해를 탐색하면서 찾은 최선해($p(k)$), 그리고 무리(swarm)내에 있는 입자들이 찾은 전체 최선해(g)에 의해 결정되는 속력의 벡터 방향으로 위치를 이동해가며 해 공간을 탐색하는 방법을 사용하고 있다. 이 기법은 연산이 단순하며 계산 시간이 짧고 연속형 혹은 이산형 문제에 적용이 가능한 장점을 지니고 있다.([14])

본 연구에서 사용된 PSO는 의 목적으로는 최소의 총 처리시간을 갖으며 같은 시간에서는 보다 적은 AGV 이동시간을 갖는 해를 찾도록 하였다. 적재단위 크기의 개수에 따라 가능한 조합은 상당히 많은 수가 있다. 본 연구에서는 공정 순서의 근사 최적해를 찾아 주어진 공정순서, 선택된 기계, 그리고 할당된 AGV 순서를 바탕으로 ILOG사의 CPLEX 패키지를 이용하여 최적의 로트크기를 결정하였다.

3.1. 전체적인 알고리즘

알고리즘을 설명하기 앞서 전반적으로 사용되는 기호는 다음과 같이 정의한다.

기호 정의	
k, l	작업 번호
m, n	설비 번호
i, j	로트 번호
r, s	부분로트 번호
q_k	작업 k 에서의 로트 크기
n_k	작업 k 에서의 부품 수
(k, i, r)	작업 k 에서 공정 i 의 부분로트 r
v_k	작업 k 의 최대 적재단위 크기
m_k	작업 k 에서 부분로트의 개수
L_k, U_k	적재단위 개수의 하한, 상한
p_{kim}	설비 m 에서 로트 (k, i) 처리시간
t_{mn}	설비 m 에서 n 으로 이동하는 시간
s_{ij}	공정 i 와 처리되는 공정 j 사이의 준비시간
C_{kirm}	설비 m 에서 (k, i, r) 의 생산 종료시간
Z_{kr}	작업 k 의 r 번째 부분로트 크기
A_{ab}	a 번째 AGV의 b 번째 이동 시작시간
R	선행관계의 공정 쌍 (i, j) 집합 $R = [R_1, \dots, R_k]$
E	설비 m 에서 생산되는 공정 쌍 (i, j) 집합 $E = [E_1, \dots, E_m]$

본 연구에서 사용한 전체적인 알고리즘을 단계적으로 살펴보면 다음과 같다.

- 단계 1. : $L_k \leq m_k \leq U_k$ 인 m_k 를 결정한다.
- 단계 2. : 수식 (3.1)에 의해 초기 Z_{kr} 을 결정한다.
- 단계 3. : PSO 절차를 수행하여 공정의 순서와 설비선택을 한다. (sec 3.2)
- 단계 3.1 : SA 절차를 수행하여 공정의 순서를 개선시킨다. (sec 3.3)
일정 횟수 개선이 되지 않으면 단계 3.2로 간다.
- 단계 3.2 : 4, SA에서 얻은 최선해를 PSO 절차의 \vec{g} 로 변환하여 단계 3으로 간다.
이전 단계에서 구한 A_{ab}, R, E 를 이용하여 Cplex 패키지를 통해 Z_{kr} 을 업데이트 한다.
업데이트된 Z_{kr} 을 이용하여 절차
- 단계 5 : 3~3.1를 수행한 후 최선해가 개선되면 단계4. 그렇지 않으면 종료한다.

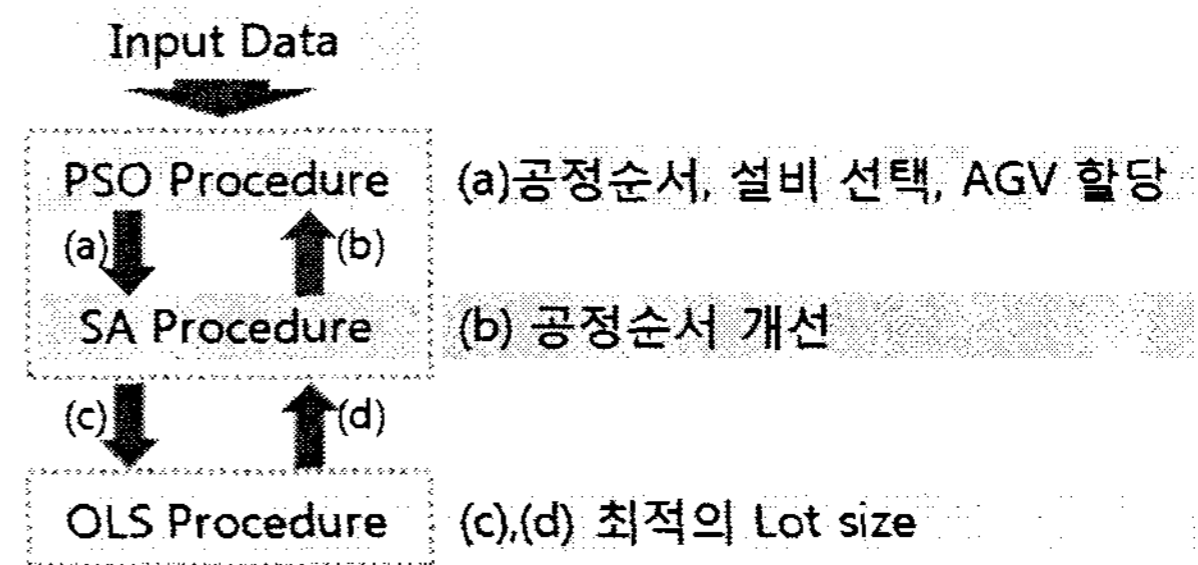
초기에 PSO 절차를 수행하기 위해서는 Z_{kr} 을 미리 결정해야만 한다. 따라서 m_k 개에 맞는

적당한 초기값을 구하기 위해서 (3.1) 수식을 이용하여 Z_{kr} 을 할당한다.

$$Z_{kr} = \left\lfloor \frac{q_k - \sum_{s=1}^{r-1} Z_{ks}}{n-r+1} \right\rfloor \quad (3.1)$$

for $\forall k, r = 1, 2, \dots, m_k$

또한, 단계별로 설명한 절차에서 주고 받은 정보를 그림으로 표현하면 <그림 1>과 같이 표현할 수 있다.



<그림 1> 절차별 정보의 흐름

<그림 1>에서 보듯이 알고리즘을 크게는 2개의 부분으로 나눌 수 있다. <PSO&SA>절차에서는 공정의 순서와 설비의 선택 그리고 각 공정후 다른 설비로 이동하기 위한 AGV의 할당을 담당하며 이를 바탕으로 <OLS(Optimize Lot Size)> 절차에서 최적의 로트크기를 결정하게 된다.

다음 3.2~3.4절에서 각 절차별 세부적으로 살펴보겠다.

3.2. PSO 절차

PSO 알고리즘에서는 입자의 움직임의 방향과 속도를 가지고 있는 속도벡터($\vec{v}(k)$)와 위치를 표현하는 위치벡터($\vec{x}(k)$)를 가지고 있다. 입자의 속도는 (3.2a)에 의해서 위치는 현위치에 속도벡터를 더해 결정하게 된다.

$$\vec{v}(k+1) = w \cdot \vec{v}(k) + rnd(0, c_1) \cdot (\vec{p}(k) - \vec{x}(k)) + rnd(0, c_2) \cdot (\vec{g} - \vec{x}(k)) \quad (3.2a)$$

$$\vec{x}(k+1) = \vec{x}(k) + \vec{v}(k+1) \quad (3.2b)$$

수식 (3.2a), (3.2b)에서 사용되는 변수는 다음과 같다.

$\vec{v}(k)$	k 번째 particle의 속도 벡터
$\vec{x}(k)$	k 번째 particle의 위치 벡터
$\vec{p}(k)$	k 번째 particle이 찾은 가장 좋은 위치 벡터
\vec{g}	지금까지 찾은 가장 좋은 위치 벡터
w	속도 벡터 자신에 곱하는 계수로써,

속도 벡터의 크기를 높이는 역할을 한다. 보통 0.9 ~ 1.2를 사용하며, 높으면 큰 속도를 가지고 해 공간을 광범위하게 탐색하며, 낮으면 local search에 가깝게 된다

c_1 : 현 속도벡터 $\bar{v}(k)$ 를 $\bar{p}(k)$ 로 향하도록 하는 가중치

c_2 : 현 속도벡터 $\bar{v}(k)$ 를 \bar{g} 로 향하도록 하는 가중치

c_1 과 c_2 에 의해 발생하는 난수의 평균을 1.0 이 되도록 각각 2.0을 사용한다.

PSO의 간략한 흐름도는 다음과 같다.

```

initialize swarm (particles)
while (  $N_{iter} < N_{max}$  )
{
 $N_{iter} \leftarrow N_{iter} + 1$ ;
Generate next swarm by equation 3.2a, 3.2b;
Evaluate swarm
{
find new  $\bar{p}(k)$ ,  $\bar{g}$ ;
Update  $\bar{g}$  of the swarm and  $\bar{p}(k)$  of each particle;
}
}

```

PSO만을 사용했을 경우, jobshop 문제와 같이 해 공간이 넓은 문제에 적합한 해를 찾아 내기 어렵다. 따라서 Xia.(2006)이 제시한 알고리즘인 PSO에서 결정한 공정의 순서와 선택된 기계에서 선택된 기계들을 기반으로 보다 더 좋은 공정들의 순서를 개선시킬 수 있는 SA절차를 소개한다.

3.3. SA 절차

SA(Simulated Annealing) 알고리즘은 Kirkpatrick et al.(1983)에 의해 제안되었으며, 조합최적문제를 해결하기 위해 사용되는 휴리스틱 해법중 하나이다. SA는 고체를 열탕에 넣어 녹인 고온의 액체상태에서는 분자들이 활발히 움직이며, 낮은 결정상태에서는 분자의 움직임이 적어지는 것을 응용한 것으로 높은 온도(T)에서는 넓은 해공간을 탐색하도록 해의 이동을 자유롭게 하고, 낮은 온도에서는 지역 최적해로 이동하도록 제한하여 좋은 해를 찾도록 유도하는 것이다.

SA에서 사용되는 변수 T_0 는 초기 온도이고, T_{cool} 은 온도 T 를 내려가게 하는 '쿨링'상수이다 그리고 T_{limit} 은 최저 온도 T 를 의미하며, 본 연구에서는 $T_0=0.997$, $T_{cool}=0.99$, $T_{limit}=0.001$ 로 하였다.

SA의 전체적인 흐름도는 다음과 같다.

Input

S : best solution of PSO

$T = T_0$

while ($T > T_{limit}$)

{

generate a neighbor solution S' from S;

{

compute fitness of S' ;

$\Delta \leftarrow f(S') - f(S)$;

if $\Delta < 0$ or $\exp(-\Delta/T) > \text{rnd}(0,1)$ **then**

$S \leftarrow S'$;

if $f(S') < f(S_{best})$ **then**

Update $S_{best} \leftarrow S'$;

end if

end if

}

$T \leftarrow T \times T_{cool}$;

}

3.4. OLS 절차

<PSO&SA> 절차에서 공정순서 및 설비 선택, 그리고 AGV할당순서를 결정한 후, 결정된 사항들에 대한 최적 로트크기를 결정하기 위해 자동적으로 IP(Integer Programming)을 생성하여 ILOG 사의 Cplex 패키지를 이용하여 값을 얻어낸다.

IP를 설명하기 전에 사용되는 기호와, 결정변수에 대해 다음과 같이 정의한다.

IP에서 사용되는 기호 정의

(a,b) a번째 AGV의 b번째 이동을 표현

$m_c(a,b)$ (a,b)의 현재 기계번호

$m_n(a,b)$ (a,b)의 다음 기계번호

$o_c(a,b)$ (a,b)할 때, 운반하는 공정 번호 (k,i,r)

$o_n(a,b)$ (a,b)할 때, 운반후 생산하는 공정 번호 (k,i,r)

결정변수 정의

C_{kirm} 설비 m에서 (k,i,r)의 생산 종료시간

Z_{kr} 작업 k의 r번째 부분로트 크기

A_{ab} a번째 AGV의 b번째 이동 시작시간

정의된 위 표현을 이용하여 수리적 모델을 다음과 같이 세울 수 있다.

Mathematical Model

Minimize C_{max}

s.t.

$$C_{max} \geq C_{kirm} \geq 0 \quad \text{for } \forall k, i, r, m \quad (1)$$

$$\sum_{r=1}^{m_k} z_{kr} = q_k \quad \text{for } \forall k \quad (2)$$

$$1 \leq z_{kr} \leq v_k \quad \text{for } \forall k, r \quad (3)$$

$$C_{kirm} + t_{mn} \leq C_{kjrn} - p_{kjin} \times Z_{kr} \quad \text{for } \forall k, r, m, n \text{ and } (i, j) \in R_k \quad (4)$$

$$C_{kirm} + s_{i,j} \leq C_{ljsm} - p_{lijm} \times Z_{lu} \quad \text{for } \forall k, l, r, s, m \text{ and } (i, j) \in E_m \quad (5)$$

$$A_{ab} \geq C_{kirm} \quad \text{for } \forall k, a, b, o_c(a, b) = (k, i, r), \text{ and } m_c(a, b) = m \quad (6)$$

$$A_{a(b+1)} - A_{ab} \geq t_{mn} + t_{nn'} \quad \text{for } \forall a, b, m_c(a, b) = m, m_n(a, b) = n, m_c(a, b+1) = n' \quad (7)$$

$$A_{ab} + t_{mn} \leq C_{kirn} - p_{kin} \times Z_{kr} \quad \text{for } \forall k, r, a, b, m_c(a, b) = m, m_n(a, b) = n, o_n(a, b) = (k, i, r) \quad (8)$$

위 정수계획법은 부분 로트의 크기에 대한 제약식 (2), (3)와 선행 공정시간에 대한 제약 (4), 같은 기계에서 생산되는 공정들에 대한 제약(5), 각 공정별 생산후에 AGV에 의해 이동할 때의 제약(6), AGV가 도착후에 생산되어야 하는 제약(8), AGV간의 시간 간격에 대한 제약(7)식에 의해 주어진 공정순서 및 기계 선택, 그리고 AGV 할당 순서가 주어졌을 때 최적의 부분로트 크기를 결정지을 수 있다.

4. Verification of solution method

<PSO&SA> 알고리즘을 이용하여 JSP의 benchmark 데이터들에 대한 성능을 평가한 결과이다.

[표 1]의 입력데이터는 [18]에서 얻을 수 있고, BKS(Best Known Solution)은 알려진 최적해 혹은 (LB, UB)이다. PSO&SA 는 5번의 반복횟수를 통해 얻은 가장 좋은 값이며, time(sec.)는 5번의 반복횟수동안 수행된 평균 걸리는 시간이다.

성능 평가 결과 작은 설비수의 문제에 대해서 대부분 최적해를 찾아냈으며 큰 입력에 대해서도 비교적 짧은 시간내에 좋은 해를 찾아내는 것을 볼 수 있다.

[표 1] JSP benchmark

Prob.	n/m	BKS	PSO & SA	time(sec)	Diff.(%)
abz5	10 / 10	1234	1234*	58.593	0.00%
abz6	10 / 10	943	943*	37.141	0.00%
abz7	20 / 15	656	706	1783.649	7.62%
abz8	20 / 15	(645,665)	724	1807.993	8.87%
abz9	20 / 15	(661,679)	754	1783.797	11.05%
ft06	6 / 6	55	55*	3.641	0.00%
ft10	10 / 10	930	941	82.750	1.18%
ft20	20 / 5	1165	1165*	96.078	0.00%
la01	10 / 5	666	666*	2.860	0.00%
la02	10 / 5	655	655*	3.198	0.00%
la03	10 / 5	597	597*	2.891	0.00%
la04	10 / 5	590	590*	2.969	0.00%
la05	10 / 5	593	593*	3.094	0.00%
la06	15 / 5	926	926*	12.843	0.00%
la07	15 / 5	890	890*	11.859	0.00%
la08	15 / 5	863	863*	11.391	0.00%
la09	15 / 5	951	951*	10.860	0.00%
la10	15 / 5	958	958*	10.656	0.00%
la11	20 / 5	1222	1222*	58.469	0.00%
la12	20 / 5	1039	1039*	55.750	0.00%
la13	20 / 5	1150	1150*	53.953	0.00%
la14	20 / 5	1292	1292*	60.656	0.00%
la15	20 / 5	1207	1207*	57.234	0.00%
la16	10 / 10	945	946	40.375	0.11%
la17	10 / 10	784	784*	41.938	0.00%
la18	10 / 10	848	848*	33.812	0.00%
la19	10 / 10	842	842*	45.907	0.00%
la20	10 / 10	902	902*	34.734	0.00%
la21	15 / 10	1046	1059	191.063	1.24%
la22	15 / 10	927	941	190.461	1.51%
la23	15 / 10	1032	1032*	190.305	0.00%
la24	15 / 10	935	958	180.586	2.46%
la25	15 / 10	977	1019	177.281	4.30%
la26	20 / 10	1218	1218*	472.313	0.00%
la27	20 / 10	1235	1292	495.797	4.62%
la28	20 / 10	1216	1238	517.211	1.81%
la29	20 / 10	1152	1272	498.211	10.42%
la30	20 / 10	1355	1355*	487.250	0.00%
la31	30 / 10	1784	1784*	2235.703	0.00%
la32	30 / 10	1850	1850*	2148.133	0.00%
la33	30 / 10	1719	1719*	2208.169	0.00%
la34	30 / 10	1721	1721*	2131.813	0.00%
la35	30 / 10	1888	1888*	2101.282	0.00%
la36	15 / 15	1268	1306	681.860	3.00%
la37	15 / 15	1397	1487	664.899	6.44%
la38	15 / 15	1196	1269	676.961	6.10%
la39	15 / 15	1233	1327	782.735	7.62%
la40	15 / 15	1222	1294	653.344	5.89%
orb01	10 / 10	1059	1092	80.813	3.12%
orb02	10 / 10	888	890	84.281	0.23%
orb03	10 / 10	1005	1033	81.204	2.79%
orb04	10 / 10	1005	1032	84.656	2.69%
orb05	10 / 10	887	913	80.922	2.93%
orb06	10 / 10	1010	1064	82.641	5.35%
orb07	10 / 10	397	401	84.313	1.01%
orb08	10 / 10	899	927	79.438	3.11%
orb09	10 / 10	934	952	83.516	1.93%
orb10	10 / 10	944	951	82.157	0.74%

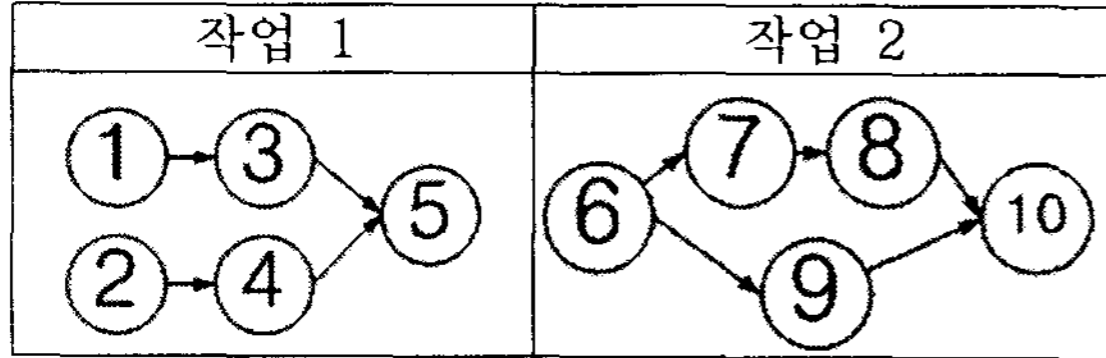
이러한 성능을 바탕으로 다음과 같은 간단한 예제를 통해 제시하는 알고리즘을 이용한 결

과를 살펴보도록 하자.

[표 2] 작업별 선행 공정

	로트 크기	최대 적재단위크기
작업 1	30	10
작업 2	30	10

[표 3] 작업별 선행 공정



[표 4] 설비간 이동시간

	1	2	3	4	5	6
1	0	5	6	8	4	3
2	5	0	7	6	3	5
3	6	7	0	8	5	6
4	8	6	8	0	5	6
5	4	3	5	5	0	7
6	3	5	6	6	7	0

[표 5] 설비에서 각 공정별 생산시간

	1	2	3	4	5	6	7	8	9	10
1	8	10	6	15	5	8	10	6	15	5
2	5	6	5	5	7	5	6	5	5	7
3	5	6	9	7	11	5	6	9	7	11
4	10	6	6	4	3	10	6	6	4	3
5	8	6	5	9	4	8	6	5	9	4
6	7	5	8	5	5	7	5	8	5	5

[표 6] 공정별 준비시간

	1	2	3	4	5	6	7	8	9	10
1	0	6	8	8	8	5	9	5	5	7
2	6	0	9	8	5	8	8	9	6	9
3	7	5	0	8	5	9	9	8	7	6
4	6	6	7	0	6	8	5	7	7	7
5	8	7	9	6	0	5	6	5	5	9
6	7	7	5	7	9	9	8	6	9	6
7	6	7	6	7	5	8	0	5	7	7
8	6	8	8	5	7	7	6	0	7	6
9	9	5	8	8	7	9	9	8	0	9
10	5	5	8	7	6	6	5	5	7	0

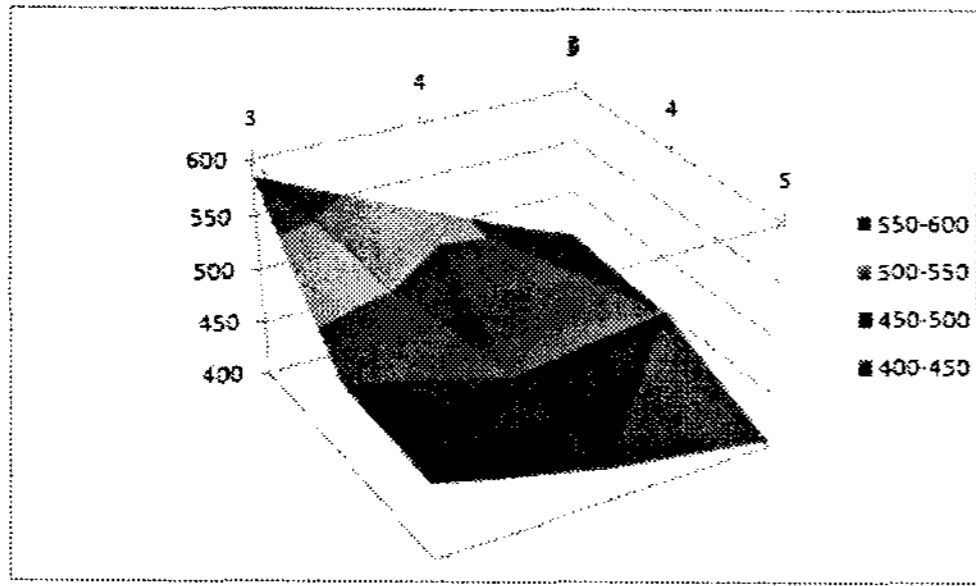
간단한 예제의 입력데이터에 대해 설명하면, 작업은 2개이며, 각 작업당 5개의 공정을 가지고 있고, 대체 공정 순서 및 대체 자원 이용이 가능한 유연한 환경이며, 예제 데이터에 대해 제시한 알고리즘을 수행한 결과는 다음과 같다. (a)는 PSO&SA를 이용하여 초기 구한 값으로 {총 처리시간/AGV이동시간}이며, (b)는 OLS 절차와 함께 개선되는 동안 반복실행한 결과이다.

[표 7] AGV가 무한대인 경우에 대한 결과

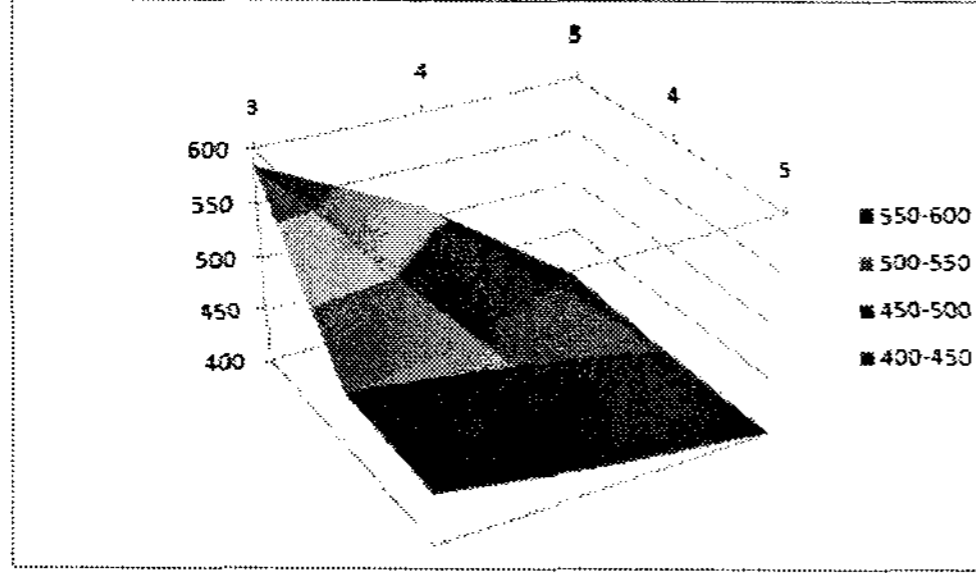
m_1	PSO&SA(a)	OLS(b)	최종 부분 로트 크기	반복 횟수
3	584	584	{10,10,10}	0
3	78	78	{10,10,10}	
3	470	451	{10,10,10}	2
4	138	135	{7,7,10,6}	
3	467	446	{10,10,10}	4
5	157	191	{6,8,8,3,5}	
4	518	511	{10,7,10,3}	2
3	141	130	{10,10,10}	
4	431	430	{3,7,10,10}	1
4	177	188	{8,6,8,8}	
4	429	419	{6,9,9,6}	1
5	189	192	{10,3,6,4}	
5	457	405	{1,10,7,2,10}	5
3	170	185	{10,10,10}	
5	450	397	{8,1,10,5,6}	6
4	186	192	{6,6,10,8}	
5	405	397	{4,7,7,7,5}	1
5	216	205	{1,9,9,2,9}	

[표 8] AGV가 1대인 경우에 대한 결과

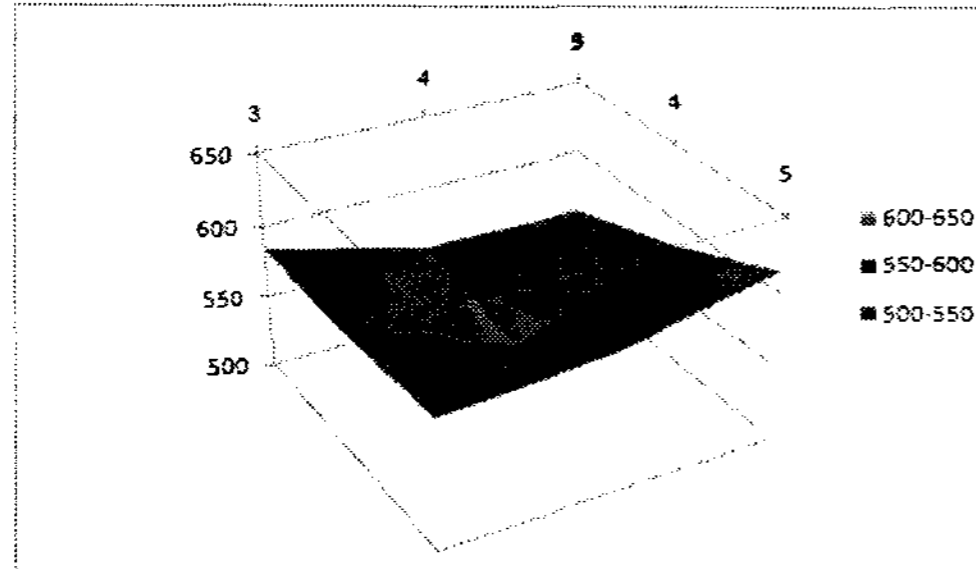
m_1	PSO&SA(a)	OLS(b)	부분로트 크기	반복 횟수
3	584	584	{10,10,10}	0
3	123	119	{10,10,10}	
3	581	496	{10,10,10}	2
4	156	123	{8,3,10,9}	
3	587	479	{10,10,10}	4
5	206	193	{4,9,3,10,4}	
4	557	495	{7,6,7,10}	3
3	163	125	{10,10,10}	
4	540	476	{8,6,10,6}	5
4	185	125	{7,5,8,10}	
4	593	493	{10,10,1,9}	4
5	226	218	{1,8,8,4,9}	
5	556	487	{6,3,6,5,10}	5
3	216	205	{10,10,10}	
5	581	514	{8,4,7,6,5}	2
4	118	233	{7,8,7,8}	
5	615	584	{9,10,2,1,8}	3
5	274	266	{6,10,8,5,1}	



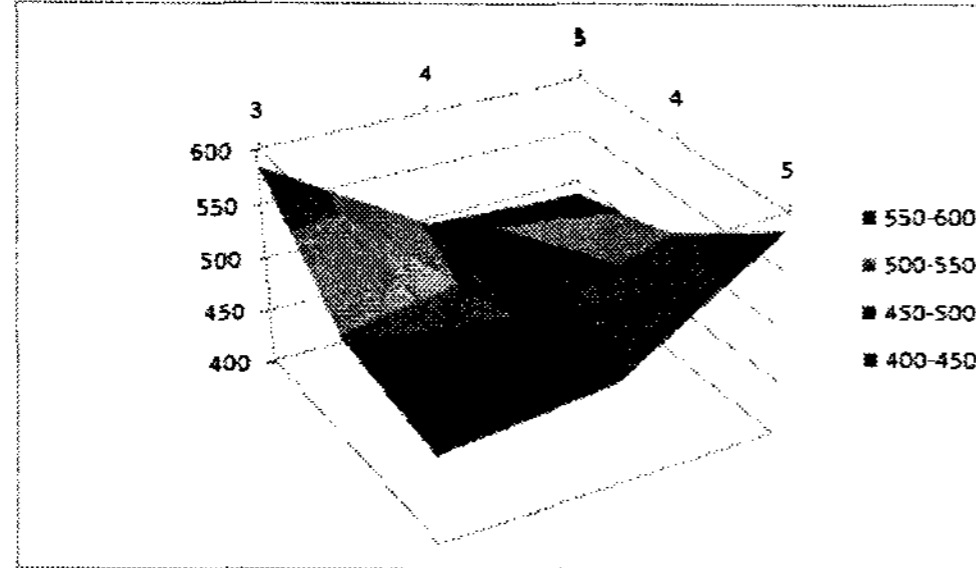
<그림 2> AGV가 무한대일 때, PSO&SA의 C_{max}



<그림 3> AGV가 무한대일 때, OLS의 C_{max}



<그림 4> AGV가 1대일 때, PSO&SA의 C_{max}



<그림 5> AGV가 1대일 때, OLS의 C_{max}

5. Conclusions

[표 7], [표 8]에서 보듯이, 부분로트의 변화가 없는 부분로트 개수가 {3,3}인 실험에서는 OLS 절차 후 총 처리시간의 감소가 보이지 않았다. 하지만, 그 외에 경우에는 적게는 1번에서 많게는 6번까지 개선되는 효과를 볼 수 있다. 이것은 초기 PSO&SA 절차에서 얻은 공정의 순서 및 선택된 기계의 영향을 크게 받는 것을 보여주며, 대체로 성능이 향상되었음을 볼 수 있다.

본 접근 방식은 다양한 작업(job)이 있을 경우, 접근하기가 까다로운 점이 있다. 따라서 추후 본 연구를 발전시켜 작업의 공정수, 로트 크기 등의 정보를 분석 및 특징을 파악하여 부

분로트를 몇 개 혹은 더 많이 만들어야 하는지의 판단을 할 수 있는 기준을 만들어야 할 것이다.

6. References

- [1] Baker, K. R., **Introduction to Sequencing and Scheduling**, John Wiley & Sons Inc., New York, Ch.2-8, 1974.
- [2] Stanley Reiter, **A System for managing Job-Shop Production**, *The Journal of Business*, Vol. 39, No. 3, 1966, pp. 371-393
- [3] Stephane D.P.; J. B. Lasserre, **Lot Streaming in Job-shop Scheduling**, *Operation Research*, Vol. 45, No. 4, 1997, pp. 584-595
- [4] Chinyao Low; Chih-Ming Hsu; Kai-I Huang, **Benefits of lot splitting in job-shop scheduling**, *Int. Adv. Manuf. Technol.*, (2004) 24: 773-780
- [5] JEN HUEI CHANG; HUAN NENG CHUI, **A comprehensive review of lot streaming**, *International Journal of Production Research*, Vol. 43, No. 8, 2005, pp. 1515-1536
- [6] Martin Feldmann; Dirk Biskup, **On lot streaming with multiple products**, *Discussion Paper No. 542*, 2005
- [7] John Jackman, **Unit Load size Determination Based on Economic Move Quantity**, *The Journal of the Operational Research Society*, Vol. 42, No. 8, 1991, pp. 703-711
- [8] SEONG WOO MOON, HARK HWANG, **Determination of unit load sizes of AGV in multi-product multi-line assembly production systems**, *International Journal of Production Research*, Vol. 37, No. 15, 1999, pp. 3565-3581
- [9] Benita M. Beamon, Ajit N. Deshpande, **A Mathematical Programming Approach to Simultaneous Unit-Load and Fleet-Size Optimization in Material handling Systems Design**, *International Journal of Advanced Manufacturing Technology*, Vol.14, No.11, 1998, pp. 858-863
- [10] B. M. Beamon; A. N. Deshpande, **Mathematical Programming Approach to simultaneous Unit-Load and Fleet-Size Optimisation in Material Handling Systems Design**, *International Advanced Manufacturing Technology*, 14:858-864
- [11] Seo Yoonho, **Model Development for Machining Process Sequencing and Machine Tool Selection**, *Journal of Korean institute of Industrial Engineers*, Vol. 21, No. 3, 1995, pp. 329-343
- [12] Facilities planning 3rd, JHON WILLEY & SONS, INC.
- [13] D Biskup; M Feldmann, **Lot streaming with variable sublots: an integer programming formulation**, *Journal of the Operation Research Society*, 2006, 57, 296-303
- [14] 유명련, 김현철, **PSO 탐색과정 가시화 툴**, 2002
- [15] Maurice Clerc, **Particle Swarm Optimization**, ISTE, 2006
- [16] Xia, Wei-jun ; Wu, Zhi-ming , **A hybrid particle swarm optimization approach for the job-shop scheduling problem**, *International journal of advanced manufacturing technology*, Vol. 29 no. 3/4, 2006, pp.360-366
- [17] <http://www.ilog.com/>
- [18] <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/jobshopinfo.html>