Journal of the Korean Institute of Industrial Engineers
Vol. 34, No. 4, pp. 408-413, December 2008.

주문 집약 문제에 대한 휴리스틱 기법

박종호[1†] · 임경국[1] · 최봉하[2]

[1]포항공과대학교 산업경영공학부 / [2]한국에너지기술연구원 정책연구센터

The Heuristic Approach to the Order Consolidation Problem

Jongho Park[1] · Kyungkuk Lim[1] · Bongha Choi[2]

[1]Department of Industrial and Management Engineering, POSTECH
[2]Energy Policy Research Center, Korea Institute of Energy Research (KIER)

We consider the batch processing of orders where either whole or part of a single order or a specific pair of different orders may be grouped in a batch within a fixed capacity. Our objective is to maximize the total number of batches filled up to the batch size. In this paper, we study the Level-2 problem where at most 2 kinds of orders can be grouped in a batch. This problem is known to be NP-hard and Max SNP-hard. So we develop heuristic algorithm and evaluate the performance of the algorithm.

Keywords: Order Consolidation, Batch Processing, Heuristic Algorithm

1. 서 론

철강이나 화학 산업 등의 소재 산업에서는 소비자들의 다양한 주문들을 집약하여 규격화된 배치(batch)로 생산한다. 주어진 배치에서 배치 크기만큼 가득 채워 생산한다고 가정하면 이런 배치의 수가 많으면 많을수록 생산 공정의 효율이 높다고 할 수 있다. 이렇게 다양한 주문들을 규격화된 배치단위로 묶어서 생산하는 것을 *주문 집약(order consolidation)* 이라 하고 실제로 철강 산업의 연주 공정(continuous casting)에서 아주 중요한 문제로 대두된다(L. Tang, 2001, S. Y. Chang, 2000). 연주 공정은 액체 상태인 용강을 연속 주조기(continuous caster)를 통과시켜 중간 소재인 슬라브(slab)로 만드는 과정이다. 이 공정에서 슬라브를 생산할 때에는 고객들의 모든 주문을 만족시켜 주는 것이 아니라 철강 등급과 크기에 따라 집약하여 규격화된 제품을 만들게 된다. 왜냐하면 고객들이 원하는 주문들 중에서 규격 차이가 크지 않는 것들을 하나로 묶어서 같은 배치로 생산하면 생산 효율성을 높일 수 있기 때문이다.

주문 집약 문제는 배치의 구성에 따라 *단일 배치(homogeneous batch)*와 *혼성 배치(heterogeneous batch)*로 구분된다. 하나의 주문으로만 이루어진 경우를 *단일 배치*라 부르고 두 가지 이상의 주문으로 이루어진 경우를 *혼성 배치*라 부른다. 일반적으로 생산방식과 공정제약 등에 의하여 한 배치로 묶을 수 있는 최대 주문 개수가 제한될 수 있는데 최대 허용 주문 개수를 ***B*** 라고 할 때 우리는 이 문제를 *레벨-**B** 문제(Level-**B** problem)*라 한다. 이 논문에서는 레벨-2 문제(Level-2 problem)만 다룬다.

주문 집약 문제(order consolidation) 중에서 배치 크기가 2이고 각 주문의 수량이 모두 1인 경우는 Maximum Cardinality Matching 문제로 볼 수 있다. 이 문제의 경우는 Micali *et al.*가 최적 알고리즘을 개발하였다(S.Micali, 1980). Hwang *et al.*은 단일 배치와 두 가지 주문으로 이루어진 혼성 배치를 사용하여 배치의 개수를 최대화하는 문제를 다루었다(H.C. Hwang, 2005). 그들은 이 문제가 NP-hard이며 다항시간근사해법군(PTAS)까지 불가능한 Max SNP-hard라 증명하였다. 그리고 그들은 그래프 *G*가 나무(tree) 구조를 가지는 경우에 대해서 최적 알고리즘을 개발하였다. Lee *et al.*은 배치 개수를 최소화하는 문제에 대해서 다루었다. 그들은 이 문제가 일반적으로 NP-

이 논문은 교육인적자원부의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(연구 과제명 : 3차년도 미래형 기계 국고지원금, 과제번호 : 4.0002619.02).
† 연락저자 : 박종호, 790-784 경북 포항시 남구 효자동 산 31 번지 포항공과대학교, 산업경영공학과, Fax : 054-279-2870,
E-mail : xcross@postech.ac.kr
2008년 1월 12일 접수; 2008년 5월 22일 수정본 접수; 2008년 6월 3일 게재 확정.

hard이지만 구간 그래프(interval graph)에 한해서는 최적 알고리즘이 존재한다고 증명하였다(Kangbok Lee, 2004). 이번 연구에서는 Hwang *et al*. (H.C. Hwang, 2005)의 결과를 바탕으로 레벨-2 문제의 휴리스틱 알고리즘을 개발하고 임의의 데이터로 실험 및 평가를 하였다.

본 논문의 구성은 다음과 같다. 제 2절에서는 문제를 수리적으로 정의하였다. 제 3절에서는 tree 구조에서는 최적해가 존재함을 보이고 이를 이용하여 우리 문제의 휴리스틱 알고리즘 제안하고 이에 대한 성능 평가를 하였다. 마지막으로 제 4절에서는 이 논문의 의의 및 추후 연구 방향을 제시하도록 한다.

2. 문제정의

레벨-2 문제는 무향 그래프(undirected graph) $G = (V, E)$으로 모델링이 가능하다. 여기서 V는 유한개의 점들의 집합이고 E는 모서리(edge)들의 집합이다. 점 $v \in V$ 는 주문을 나타내며 각 점의 가중치 $w(v)$는 각 주문의 주문량을 나타낸다. 만약 주문 u와 주문 v가 집약 가능하다면 모서리 $e = (u, v) \in E$으로 표현하고 모서리 $e \in E$에 대한 혼성 배치는 이차원 벡터 $x_e = (x_e(u), x_e(v))$로 표현한다. 여기에서 $x_e(u)$와 $x_e(v)$는 가중치 $w(u)$와 $w(v)$의 일부분으로 $x_e(u) + x_e(v) = \lambda$을 만족한다.

각 점 v의 단일 배치를 표현하기 위한 변수를 y_v라 하고, 모서리 e의 혼성 배치를 표현하기 위한 이진 변수를 y_e라 하자. 마지막으로 각 점 v가 가지는 모든 모서리의 집합을 집합 $E(v)$로 표현하면 우리 문제는 다음과 같은 수학적 모델로 표현할 수 있다(H. C. Hwang, 2005).

$$
\begin{aligned}
\textit{Maximize} \quad & \sum_{v \in V} y_v + \sum_{e \in E} y_e \\
\textit{subject to} \quad & \lambda y_e - x_e(u) - x_e(v) = 0 && \text{for } e = (u,v) \in E \\
& \lambda y_v + \sum_{e \in E(V)} x_e(v) \le w(v) && \text{for } v \in V \\
& x_e(v) \ge 0 && \text{for } v \in V \text{ and } e \in E(v) \\
& y_v \ge 0 \text{ 정수} && \text{for } v \in V \\
& y_e \in \{0, 1\} && \text{for } e \in E
\end{aligned}
$$

Lemma 1 : 두 점 사이에 두 개의 혼성 배치가 존재한다고 가정하면 혼성 배치가 최대 한 개인 배치로 만들 수 있다.

Proof. Hwang *et al*. (H. C. Hwang, 2005) □

3. 휴리스틱

3.1 레벨 2 문제에서 최적 나무(Optimal-Tree)

그래프 G가 나무(tree) 구조를 가지는 특별한 경우에는 최적 알고리즘인 *TreeGroup* 알고리즘이 존재한다(H. C. Hwang, 2005). 다음 Lemma 2는 여러 최적해들 중에서 나무 구조를 가지는 최적해가 반드시 존재함을 보여준다.

Lemma 2 : 사이클(cycle)이 있는 혼성 배치가 있다면 사이클이 없고 배치 수가 같은 배치로 만들 수 있다.

Proof : 혼성 배치의 최적해가 n개의 모서리로 이루어진 사이클 $C = <(v_1, v_2), (v_2, v_3), \cdots (v_n, v_1)>$을 가지고 있다고 가정하자. $p(v_i)$, $q(v_i)$와 θ를 다음과 같이 정의하자.

$$
\begin{aligned}
p(v_i) &= x_{(v_i, v_{i+1})}(v_i) && \text{if} && i < n \\
&= x_{(v_i, v_1)}(v_i) && \text{else} && i = n \\
q(v_i) &= x_{(v_{i-1}, v_i)}(v_i) && \text{if} && i > 1 \\
&= x_{(v_n, v_i)}(v_i) && \text{else} && i = 1 \\
\theta &= \min\{q(v_1), q(v_2), \cdots, q(v_n)\}
\end{aligned}
$$

Lemma 1에 의하여

$$
\begin{aligned}
& x_{(v_i, v_{i+1})}(v_i) + x_{(v_i, v_{i+1})}(v_{i+1}) = p(v_i) + q(v_{i+1}) = \lambda \\
& \qquad \text{for} \quad i < n \\
& x_{(v_i, v_1)}(v_i) + x_{(v_i, v_1)}(v_1) = p(v_i) + q(v_1) = \lambda \\
& \qquad \text{for} \quad i = n
\end{aligned}
$$

을 얻을 수 있다. 일반성을 잃지 않고 $\theta = q(v_k)$ $(1 \le k \le n)$라고 가정하고 $p'(v_i) = p(v_i) + \theta$와 $q'(v_i) = q(v_i) - \theta$로 놓으면 사이클 C를 제거한 최적해를 구할 수 있다. 이 때 새로운 가능해는 다음과 같다.

$$
\begin{aligned}
& x'_{(v_{i-1}, v_i)}(v_{i-1}) = p'(v_i), \quad x'_{(v_{i-1}, v_i)}(v_i) = q'(v_i) \\
& \qquad \text{for} \quad i > 1,\ i \ne k \text{ and } i \ne n \\
& x'_{(v_n, v_1)}(v_n) = p'(v_n) \qquad x'_{(v_n, v_1)}(v_1) = q'(v_1) \\
& \qquad \text{for} \quad i = n \\
& x'_{v_{j-1}} = p'(v_{j-1}) \quad \text{for} \quad i = k
\end{aligned}
$$

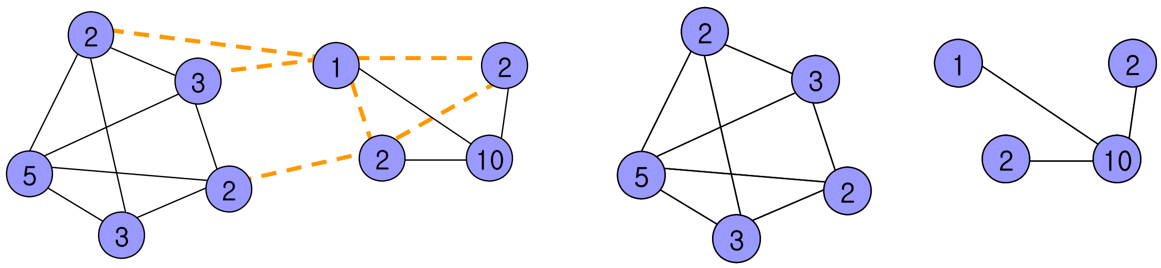
그러면 $x(v_{k-1}) < \lambda$이기 때문에 배치의 개수는 변화가 없고 새로운 가능해는 사이클 C가 제거된 최적해이다. □

Lemma 2에 의하면 일반적인 그래프 G에 대해서도 최적 숲(optimal forest)이 항상 존재함을 알 수 있다. 따라서 나무 구조와 유사한 *근사최적나무(near-optimal tree)*를 찾는다면 *TreeGroup* 알고리즘을 적용하여 좋은 해를 얻을 수 있을 것이다(<**Figure 1**> 참조).

근사최적나무를 얻기 위한 절차는 다음과 같다.

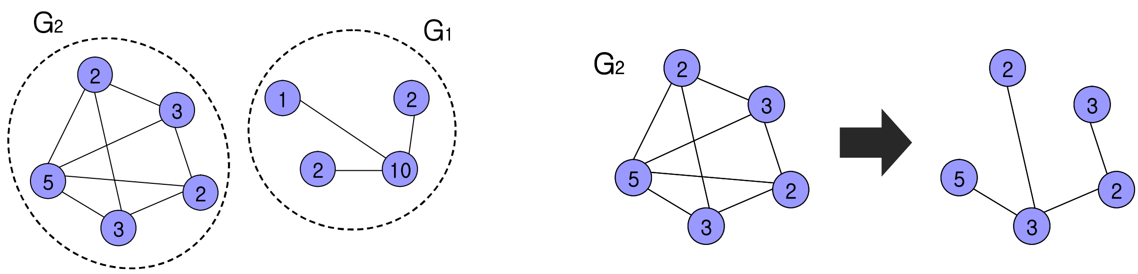
Step 1 : Remove $e = (u, v) \in E$ such that $w(u) + w(v) < \lambda$

Step 2 : Identity all connected components $(G_1, G_2, \cdots, G_k)$ from G.



(a) Given graph G, where dotted line represents $w(u)+w(v) < \lambda$

(b) Remove $e=(u, v)$ such that $w(u)+w(v) < \lambda$



(c) Identity all connected components (G_1, G_2)

(d) Find near-optimal tree in G_2.

Figure 1. Procedure to obtain near-optimal tree

Step 3 : For each component, solve for G_k.

1) If G_k is tree, done

2) else, find near-optimal tree in G_k.

3.2 레벨 2 문제에 대한 직관

근사최적나무를 찾기 위해 우리는 직관에 의한 2가지 관찰을 제안한다.

관찰 1 : 최적나무(Optimal tree)에서 가중치가 큰 점에는 인접 모서리(incident edge)가 많을 것이고 가중치가 작은 점에서는 인접 모서리가 적을 것이다.

관찰 2 : 각 점은 최대 구성 가능한 혼성 배치 개수의 상한을 가진다. 따라서 최적 나무에서 각 점에 인접한 모서리의 개수는 이 상한을 넘지 못 한다.

관찰 2의 상한 값을 $B(u)$라 하고 이것을 찾기 위한 방법은 <**Figure 2**>와 같다.

$A(u)=\{e_1, e_2, \cdots, e_k\}$을 $u \in V$의 인접모서리집합(incident edge-set)이라 하자. 여기에서 $e_i=(u, v_i)$이고 $w(v_1) \geq w(v_2) \geq \cdots \geq w(v_k)$라 가정한다. 혼성 배치를 최대로 만들기 위해 가중치가 큰 점들부터 인접한 점들을 확인한다. 가중치가 $w(v_i) \geq \lambda$라면 $w(u)$의 가중치 하나를 이용하여 $x_e=(x_e(u), x_e(v_i))=(1, \lambda-1)$인 혼성 배치를 만들고 $w(v_i) < \lambda$이고 $w(u) \geq (\lambda-w(v_i))$인 경우는 $x_e=(x_e(u), x_e(v_i))=(\lambda-w(v_i), w(v_i))$인 혼성 배치를 만든다. 만약 남은 가중치가

Procedure Max-Likely(G) for $G(V, E)$

begin
  Let $Q(u)$ be low likelihood incident edge-set of $u \in V$ for all $u \in V$
begin
  $Q(u) := \varnothing$
  $B(u) := 0$
  Let $A(u)$ be incident edge-set of $u \in V$
  $A(u)= e_1, e_2, \cdots, e_k$ with $w(e_1) \geq w(e_2) \geq \cdots \geq w(e_k)$
    where $e_i=(u, v_i)$
  $Tw := w(u)$
  $i := 1$
  for $i := 1$ **to** k **begin**
    if $w(v_i) \geq \lambda$, **then** $Tw := Tw-1$;
    else $Tw := Tw-(\lambda-w(v_i))$;
    if $Tw<0$, **then** $Q(u) := Q(u) \bigcup \{(u, v_i)\}$;
    else $B(u)$++;
    i++; **end**
end
end

Figure 2. Procedure Max-Likely(G)

$w(u) < \lambda-w(v_j)$ $(i=j)$라면 u를 이용해서는 혼성 배치를 만들 수 없으므로 $B(u)$는 $j-1$이 된다. 그리고 $Q(u)$를 u의 로우 라이클리후드 인접모서리집합(low likelihood incident edge-set)이라 한다면 $Q(u)$는 모서리 $(u, v_j), \cdots, (u, v_k)$의 집합이 되며 이 모서리들은 그래프 G에서 제거시킨다. 이와

같은 방법을 모든 점들에 대해서 실행하게 된다. Max-Likely를 수행한 후에는 나타나는 그래프 G'은 숲(forest)로 나타날 수 있으며, 이 숲에 각각의 성분(component)에 대해서 Maximum Spanning Tree 알고리즘을 수행하면 각 성분들에 대한 나무(tree)들을 얻을 수 있다. 이 각각의 나무들은 다음 절에서 설명하는 Connecting Procedure에 의해 하나의 근사최적나무를 구성하게 되며, 최종적으로 이 나무에 Tree Group 알고리즘을 적용하여 문제를 푼다.

3.3 레벨 2 문제의 휴리스틱 알고리즘

근사최적나무를 만드는 휴리스틱 알고리즘을 다음과 같이 제안한다.

Step 1 : Max-Likely Procedure
Step 2 : Maximum Spanning Tree
Step 3 : Connecting Procedure

Step 1에서 제거된 모서리위에 혼성배치가 존재할 수 있기 때문에 나무를 만들기 위해서는 각 성분을 연결해야 한다. 이때, 각 성분들은 하나의 점이거나 모서리를 갖는 부분 나무(sub-tree)이다. 모든 점은 인접한 점들 중 하나와 결합하여 적어도 한 개의 혼성 배치를 만들 수 있다. 그러나 현재 단계에서는 하나의 점에서는 인접 모서리를 가질 수 없다. 그래서 우리는 한 개의 점을 고려한 후 모서리를 가지고 있는 부분 나무에 관해서 Connecting Procedure를 고려한다.

모서리 $e=(u,v)\in E$의 달림수(degree)를 $\deg(e)=\deg(u)+\deg(v)$라 정의하자. 모서리를 선택하는 방법으로는 가장 큰 가중치를 가지는 모서리를 선택하는 경우, 가장 작은 가중치를 가지는 모서리를 선택하는 경우, 가장 큰 달림수를 가지는 모서리를 선택하는 경우와 가장 작은 달림수를 가지는 모서리를 선택하는 경우가 존재한다. 만약 선택에 있어서 동일한 경우가 일어난다면 다른 기준(가중치나 달림수)으로 선택할 수 있기 때문에 총 8가지가 존재한다. 그런데 우리 결과에 의하면 선택 기준이 동일한 경우에는 가장 먼저 가장 작은 달림수와 가장 작은 가중치를 생각하는 것이 좋았다.

다음 단계에서 <**Figure 4**>와 같은 절차를 고려한다.

Step 4 : Tree-Group Algorithm

Step 3을 통해 근사최적나무를 만들 수 있기 때문에 Tree-Group 알고리즘을 실행할 수 있다. 그래서 전체적인 레벨-2 문제에 대한 휴리스틱 알고리즘은 다음과 같다.

Heuristic Algorithm for Level-2 problem
begin
Max-Likely(G)
Maximum Spanning Tree(G)
Connecting Procedure(G)
Tree Group(G)
end

이 휴리스틱 알고리즘에서 Step 1과 Step 2와 Step 3의 각각의 시간복잡도(time complexity)는 $O(|E|)$이며 Step 4의 시간복

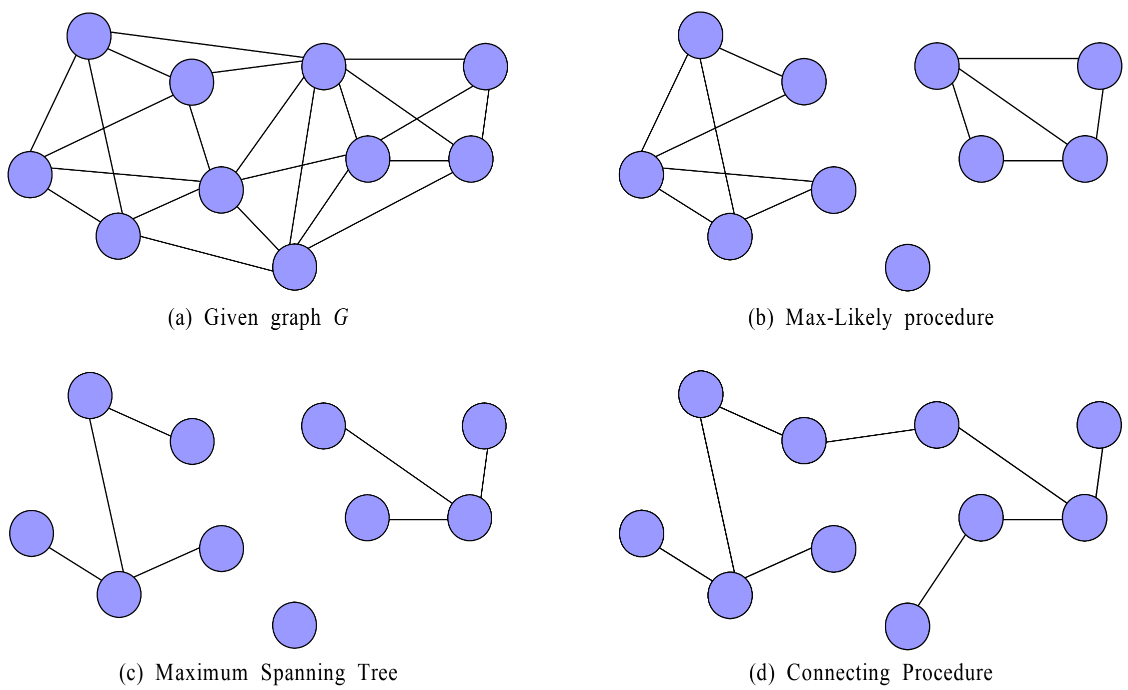


Figure 3. Heuristic algorithm to construct near-optimal tree

```
Connecting Procedure for G(V, E)
begin
  if |E| < |V|-1
     then begin
        Q(G) is set of removed edges from G
        for all u∈V such that deg(u) = 0 do
           Q(u) is set of removed incident edges
                from u
           Let e_s = (u, v_s) be selected edge in Q(u)
           Let A(u) be the incident edge-set of u
           by 1) smallest degree 2) smallest weight
           E := E ∪ {e_s}
           Q(u) := Q(u) - {e_s}, Q(v_s) := Q(v_s) - {e_s}
           A(u) := A(u) ∪ {e_s}, A(v_s) := A(v_s) ∪ {e_s}
           Q(G) := Q(G) - {e_s}
     end
end
begin
  if |E| < |V|-1
    then begin
       S(G) be set of edges from G, S(G) := ∅
       H_i is component of G
       Q(H_i, H_j) be set of removed edges between H_i
              and H_j, i < j
    For all Q(H_i, H_j) do
       Select one e = (u_i, v_j) by 1) smallest degree 2)
              smallest weight
       S(G) := S(G) ∪ {e}
    end
    Sort S(G) by 1) smallest degree 2) smallest weight
    Let S(G) be e_1, e_2, ..., e_l, e_k = (u_p, v_q)
    k := 1
    while (|E| < |V|-1)
       if e_k do not make cycle then
          E(G) := E(G) ∪ {e_k}
          Q(u_p) := Q(u_p) - {e_k}, Q(v_q) := Q(v_q) - {e_k}
          A(u_p) := A(u_p) ∪ {e_k}, A(v_q) := A(v_q) ∪ {e_k}
          Q(G) := Q(G) - {e_k}
       k := k+1
    end
  end
end
```

Figure 4. Connecting Procedure

잡도는 $O(|V|\log|V|)$이다(H. C. Hwang, 2005). $|E|<|V|^2$이기 때문에 이 휴리스틱의 시간복잡도는 $O(|V|^2)$이라 말할 수 있다.

3.4 휴리스틱 알고리즘의 평가

우리가 제안한 휴리스틱 알고리즘을 평가하기 위하여 데이터를 무작위로 생성하여 실험을 실시하였다. 이번 실험을 비교 분석하기 위해서 그룹핑비(grouping-ratio)를 다음과 같이 정의하여 사용하였다.

$$grouping-ratio = \frac{number\ of\ batches\ by\ heuristic}{optimal\ number\ of\ batches}$$

레벨-2 문제에 대한 휴리스틱 알고리즘의 connecting procedure에서 선분을 선택하는 방법은 총 8가지가 존재하지만 이번 실험에서는 한 가지를 더 추가했다. 즉, Max-Likely Procedure를 생략한 상태에서 실험을 하였다.

이번 실험은 세 가지 조건하에 실행하였다. 조건 A는 점의 수가 100, 점의 평균 가중치가 50이며 모서리 밀도(edge density)가 50%이고 조건 B는 노드 수가 50, 노드 평균 가중치가 100이며 모서리 밀도가 75%이다. 그리고 조건 C는 노드 수가 150, 노드 평균 가중치가 200이며 모서리 밀도가 25%이다. 여기에서 모서리 밀도는 임의로 생성한 데이터의 두 점 사이에 모서리를 만들 확률이다. 그리고 모든 배치 크기는 100으로 하였고 모서리를 선정할 때에는 SD(Smallest Degree), BD(Biggest Degree), SW(Smallest Weight)와 BW(Bigger Weight)의 기준으로 30번씩 반복하여 실험하였다. <**Table 1**>은 각 실험에 관한 그룹핑비를 나타낸 것이다.

Table 1. Grouping-ratio for every policy of connecting procedure

Policy	조건 A	조건 B	조건 C
SD-BW	91.7	92.4	96.5
SD-SW	94.5	94.2	96.5
BD-BW	82.4	90.2	96.5
BD-SW	82.2	90.2	96.5
SW-SD	92.1	93.8	96.5
SW-BD	91.4	93.6	96.5
BW-SD	82.7	90.2	96.5
BW-BD	82.8	90.2	96.5
Only MaxST	81.9	90.0	96.5

조건 C에서 그룹핑비가 모두 같은데 이는 모든 모서리가 연결되어 있기 때문이다. 그리고 실험 결과를 살펴보면 SD-SW (smallest degree-smallest weight) 기준이 가장 좋다는 것을 알 수 있다. 그래서 레벨-2 문제에 대한 휴리스틱 알고리즘에서 Connecting Procedure에서는 SD-SW기준에 대해서 실험하였다.

레벨-2 문제의 휴리스틱을 평가하기 위하여 배치 크기는 100으로 하여 실험하였다. 이번 실험은 <**Table 2**>처럼 3가지 변수를 가지고 하였다.

Table 2. Condition for each variable

Number of nodes	50, 100, 150, 200
Average weight of node	50, 100, 200
Edge density	25%, 50%, 75%

각 실험은 총 30번씩 실시하여 <**Table 3**>과 같은 결과를 얻었다.

<**Table 3**>을 보면 모서리 밀도가 높을수록 그룹핑비가 떨어지는 것을 알 수 있다. 이는 그래프에서 많은 모서리들이 있을 때 근사최적나무를 찾기 힘들기 때문이다. 그러나 점들의 수와 점들의 평균 가중치의 경우는 그룹핑비 사이에 특정한 관계를 찾기 힘들었다.

Table 3. Grouping-ratio for every condition in Level-2 problem

Number of Node	Avg. Weight of node	Edge Density		
		25%	50%	75%
50	50	95.6	94.4	94.0
	100	95.4	93.4	94.2
	200	98.9	97.5	95.3
100	50	96.4	94.5	92.8
	100	95.2	93.3	92.7
	200	97.9	94.6	92.2
150	50	96.1	93.7	93.6
	100	94.9	94.8	91.8
	200	96.5	93.6	92.2
200	50	95.5	94.6	93.3
	100	95.1	93.2	91.3
	200	95.6	92.7	90.8

4. 결론 및 추후연구

이번 연구에서는 하나의 배치에 묶여서 처리될 수 있는 주문의 수를 최대 B개로 제한하는 문제를 레벨-B 문제라 정의한 후 레벨-2 문제에 대해 다루었다. 레벨-2 문제는 NP-hard이며, MAX SNP-hard로 알려져 있다(H. C. Hwang, 2005). 그래서 우리는 이 문제를 효율적으로 풀 수 있는 휴리스틱 알고리즘을 개발하였고, 이를 테스트하기 위해 임의로 만든 데이터들로 성능을 실험하였다. <Table 3>을 살펴보면 그룹핑비가 90% 이상임을 볼 수 있는데 이로 인해서 제안한 알고리즘이 효율적이라 할 수 있다.

하지만 이번 실험은 임의로 생성한 데이터로 한 결과일 뿐이다. 앞으로 실생활에서 얻은 데이터를 이 알고리즘에 적용해 보는 것과 알고리즘에 대한 worst case 분석을 통하여 더 좋은 알고리즘 개발을 하는 것이 필요하다. 그리고 이번 연구에서는 레벨-2 문제에 대해서만 다루었지만 B가 3이상인 레벨-B 문제에 대한 연구도 필요하다.

참고문헌

Bongha, Choi, S. Y. Chang, and June-Ho Chang (2007), The heuristic approach to the order consolidation problem, Proceedings of the Conference of Korean Institute of Industrial Engineers.

Hwang, H. C. and S. Y. Chang (2005), Order consolidation for batch processing, *Journal of Combinatorial Optimization*, **9**, 121-138.

June-Ho Chang and S. Y. Chang (2006), Minimum batch cover for efficient order consolidation, Proceedings of the Joint Conference of Korean Institute of Industrial Engineers and Korean Operations Research and Management Science Society.

Kangbok, Lee, S. Y. Chang, and Y. S. Hong (2004), Continuous slab caster scheduling and interval graph, *Production Planning and Control*, **15**, 495-501.

L. Tang, J. Liu, A. Ring, and Z. Yang (2001), A review of planning and scheduling systems and methods for integrated steel production, *European Journal of Operational Research*, **133**, 1-20.

S. Micali and V. V. Vazirani (1980), An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs, *in Proc. Twenty-first Annual Symposium on the Foundations of Computer Science*, 17-27.

S. Y. Chang, M. R. Chang, and Y. S. Hong (2000), A lot grouping algorithm for a continuous slab caster in an integrated steel mill, *Production Planning and Control*, **11**, 363-368.

Y. H. Min, S. P. Hong, S. W. Kwon, B. C. Choi, and S. Y. Chang (2007), Maximum batch cover for order consolidation, Proceedings of the Conference of Korean Operations Research and Management Science Society.