효율적 주문집약을 위한 단위작업 편성
Minimum batch cover for efficient order consolidation

장준호, 장수영

포항공과대학교 산업경영공학과(junho@postech.ac.kr, syc@postech.ac.kr)

초록

소정의 조건을 만족하는 서로 다른 주문들의 일부 혹은 전부가 동일 배치에 집약하여 처리할 수 있을 때, 최소개수의 작업단위로 모든 주문을 처리하는 문제를 다룬다. 무방향 그라프($G=(V,E,w(v))$)와 작업 단위 크기 λ로 표현되는 이 문제는 엔피하드(NP-hard)일 뿐 아니라 맥스에스엔피하드(MAX SNP-hard)임을 보인다. 또한, 그라프가 나무(Tree)구조일 때에 대한 $O(|V|\log|V|)$ 계산 복잡도를 갖는 최적 알고리듬을 제시한다.

1. Introduction

철강, 화학 산업과 같은 소재 산업에서의 생산은 대부분 일정한 수량의 배치로 이루어진다. 이러한 배치 생산 공정에서는 주어진 다양한 주문들을 비슷한 성질의 주문들을 집약하여 가능한 작은 수의 배치들로 만족시키는 것이 생산 공정의 효율성을 높이는데 중요한 요인이 된다.

이러한 문제들은 실제 철강산업의 연주공정에서 찾아볼 수 있다. 연주공정에서는 생산과 재고관리의 효율성을 위해서 철강의 등급과 크기에 따라 몇 개의 표준 슬라브들로 생산이 이루어진다. 이 과정에서 고객들의 다양한 주문들을 철강 등급과 크기에 따라서 집약하는 문제가 발생된다(Chang et al., 2000, Tang et al., 2001).

이러한 문제들은 무방향 그라프(Undirected Graph) $G=(V,E,w(v))$를 이용해서 일반적으로 모형화 될 수 있다. $G=(V,E,w(v))$는 유한개의 node들의 집합인 V, edge들의 집합인 E, 그리고 각 node $v\in V$들의 weight 함수 $w(v)$로 이루어진다. 각 node $v\in V$는 각 주문에 대응하며, $w(v)$는 각 주문의 수량을 의미한다. Edge $e=(u,v)\in E$는 두 개의 개별 주문이 집약되어 하나의 배치로 생산이 가능할 경우 정의한다. 배치는 배치의 구성에 따라 하나의 주문으로만 이루어진 homogeneous 배치와 edge에 의해서 상호 연결된 2개 이상의 주문으로 이루어진 heterogeneous 배치로 구분할 수 있다. 또한 배치는 배치가 구성된 양에 따라 배치 최대 크기 λ만큼 구성된 complete 배치와 λ보다 작게 구성된 incomplete 배치로 구분된다.

이 문제는 목적식에 따라 배치 개수를 최대화하는 문제와 최소화하는 문제로 나눌 수 있다. 최대화 문제의 경우 homogeneous 배치와 2개의 주문으로 이루어진 heterogeneous 배치만을 이용하여 complete 배치 개수를 최대화하는 제한된 문제가 NP-hard이며 또한 다항시간근사해법군(PTAS)까지 불가능한 MAX-SNP-hard 문제이다. 하지만 G가 tree구조를 가지는 경우 최적 알고리듬이 존재한다(Hwang and Chang., 2005). 최소화 문제의 경우 일반적인 경우 NP-hard문

제이나 G가 interval 그라프일 때에 한해 최적 알고리듬이 존재한다(Lee et al., 2004).

본 논문에서는 homogeneous 배치와 2개의 주문으로 이루어진 heterogeneous 배치만을 이용하여 주문의 수량을 커버하면서 배치의 수를 최소화하는 문제를 다룬다. 우리 문제는 배치의 크기 λ가 2이고 각 주문의 수량이 모두 1인 경우 Maximum Cardinality Matching문제로 볼 수 있다. 이 경우 최적 알고리듬이 알려져 있다 (Micali and Vazirani, 1980 and Papadimitriou and Steiglitz, 1988). 하지만 불행히 일반적인 문제는 NP-hard문제이며 이에 대한 논의는 3절에서 보다 자세히 다룰 것이다.

본 논문의 구성은 다음과 같다. 2절에서는 문제를 수리적으로 정의하고 3절에서 우리 문제가 NP-hard, MAX-SNP-hard문제임을 보인다. 4절에서는 tree구조를 가지는 최적해가 반드시 존재함을 보이고, 5절에서 G가 tree 구조를 가질 때 최적 알고리듬을 제안한다. 마지막으로 6절에서는 본 논문의 의의와 추후 연구 방향을 제시하고자 한다.

2. Problem Definition

Node v의 homogeneous 배치가 구성된 양을 나타내는 변수는 x_v로 edge $e=(u,v)$의 heterogeneous 배치가 구성된 양은 2차원 vector $y_e=(y_e(u),y_e(v))$로 각각 정의하자. 여기서 $y_e(u)$는 node u에 해당하는 주문량을 $y_e(v)$는 node v에 해당하는 주문량을 의미한다. $E(v)$는 각 node v의 인접한 edge들의 집합이다. Node v의 degree는 $d(v)$로 표현한다. 그러면 우리 문제는 다음과 같은 수리 모형으로 나타낼 수 있다.

Minimum Batch Cover (BC)

$$\textbf{Min} \quad \sum_{v\in V}\lceil x_v/\lambda\rceil+\sum_{e\in E}\lceil y_e(u)+y_e(v)/\lambda\rceil$$

$$\textbf{s.t.} \quad x_v+\sum_{e\in E(v)} y_e(v)=w(v),\ \forall v\in V.$$

$$x_v, y_e(v)\ge 0,\ \forall v\in V$$

다음 property는 최적해에서 각 edge당 2개 이상의 heterogeneous 배치는 고려하지 않아도 됨을 준다.

Property 1. $^{\forall}e=(u,v)\in E$ 에 대해서 $0\le y_e(u)+y_e(v)\le\lambda$ 을 만족시키는 최적해 $<x_v^*,y_v^*;v\in V,e\in E>$ 가 반드시 존재한다.

Proof. 위의 property가 거짓이라고 가정하자. $<x_v^*,y_v^*;v\in V,e\in E>$를 $G(V,E,w(v))$ 의 최적해라고 한다면 $y_e^*(p)+y_e^*(q)>\lambda$를 만족시키는 $(p,q)\in E$가 반드시 존재한다.

Case 1. $y_{(p,q)}^*(p)\ge\lambda$ or $y_{(p,q)}^*(q)\ge\lambda$.

일반성을 잃지 않고 우리는 $y_{(p,q)}^*(p)\ge\lambda$ 임을 가정할 수 있다. $y_{(p,q)}^*(p)$와 $y_{(p,q)}^*(q)$를 다음과 같이 정의하자. $y_{(p,q)}^*(p)=k_1\lambda+r_1$, $y_{(p,q)}^*(q)=k_2\lambda+r_2$. 이 때 k_1은 양의 정수, k_2는 음이 아닌 정수이고 $0\le r_1,r_2<\lambda$이다. 위와 같은 상황에서 우리는 새로운 가능해 $<x_v',y_e';v\in V,e\in E>$를 다음과 같이 만들 수 있다.

$$x_p'=x_p^*+k_1\lambda,\ x_q'=x_q^*+k_2\lambda$$

$$y_{(p,q)}'(p)=r_1,\ y_{(p,q)}'(q)=r_2$$

$$x_v'=x_v^*,\ ^{\forall}v\in V \text{ except } p,\ q$$

$$y_e'=y_e^*,\ ^{\forall}e\in E \text{ except } (p,q)$$

p, q를 제외한 모든 node $v\in V$에서의 배

치와 (p,q)를 제외한 모든 edge $e \in E$에서의 배치들은 변화가 없기 때문에 우리는 p, q, (p,q)의 배치수만 고려하면 된다. $< x_v^{'}, y_e^{'}; v \in V, e \in E >$에서 p, q의 homogeneous 배치는 $< x_v^{*}, y_v^{*}; v \in V, e \in E >$보다 각각 k_1, k_2만큼 증가하였으나 (p,q)의 heterogeneous 배치가 $k_1 + k_2$만큼 증가하였기 때문에 $< x_v^{*}, y_v^{*}; v \in V, e \in E >$와 $< x_v^{'}, y_e^{'}; v \in V, e \in E >$의 배치 수는 결국 같다. 따라서 $< x_v^{'}, y_e^{'}; v \in V, e \in E >$는 최적해이다.

Case2. $y^*_{(p,q)}(p),\ y^*_{(p,q)}(q) < \lambda$

$< x_v^{*}, y_v^{*}; v \in V, e \in E >$의 (p,q)에서 구성된 2개 heterogeneous 배치를 이용해서 p에서 한 개의 추가 homogeneous 배치를 구성하고, (p,q)에서는 1개의 heterogeneous 배치를 구성하는 가능해 $< x_v^{'}, y_e^{'}; v \in V, e \in E >$를 다음과 같이 만들 수 있다.

$$x_p^{'} = x_p^{*} \text{ and } x_q^{'} = x_q^{*} + \lambda$$

$$y^{'}_{(p,q)}(p) = y^{*}_{(p,q)}(p)$$

$$y^{'}_{(p,q)}(p) = y^{*}_{(p,q)}(p) + y^{*}_{(p,q)}(q) - \lambda$$

$x_v^{'} = x_v^{*}$, $^{\forall} v \in V$ except p, q
$y_e^{'} = y_e^{*}$, $^{\forall} e \in E$ except (p,q)
$< x_v^{'}, y_e^{'}; v \in V, e \in E >$와 $< x_v^{*}, y_v^{*}; v \in V, e \in E >$의 배치 개수는 변하지 않았기 때문에 $< x_v^{'}, y_e^{'}; v \in V, e \in E >$도 최적해라고 말할 수 있다. 따라서 이 property는 만족한다.□

3. Hardness and Nonapproximability.

이번 절에서 우리는 잘 알려진 NP-Complete문제인 평면 그라프에서 최대 degree가 3을 넘지 않는 node cover문제가 우리 문제의 제한된 경우로 변환됨을 보임으로써 우리 문제가 NP-hard임을 증명하고자 한다. 하지만 우리 문제는 최소화 문제이기 때문에 우리 문제의 결정문제 버전이 필요하다. 우리 문제의 결정문제 버전은 주문들을 집약하여 특정한 정수 K보다 작거나 같은 수의 배치들로 모든 주문 수량을 만족할 수 있는가를 묻는 것이다.

Node Cover in planar graphs with maximum degree 3 (NC-3)

Instance: A planar graph $G' = (V', E')$ with no node degree exceeding 3, positive integer $K' \le |V'|$.

Question: Is there a node cover of size K' or less for G', i.e. a subset $\overline{V} \subseteq V'$ with $|\overline{V}| \le K'$ such that for each edge $e = (u,v) \in E'$ at least one of u and v belongs to $\overline{V}$?

.**Theorem 1** NC-3은 다항시간에 우리 문제, BC의 결정문제 버전으로 변환된다.

Proof. NC-3의 instance $G' = (V', E')$가 주어졌을 때 배치 크기 $\lambda = 4$인 우리 문제를 만들어보자. G'의 각 edge $e' \in E'$에 대해서 1개의 edge node $v_{e'}$를 만들고 그것의 weight, $w(v_{e'}) = 1$이라고 정한다. 그리고 G'의 각 node $v_i^{'} \in V'$에 대해서 $d(v_i^{'}) + 1$개의 node들로 구성된 star-component $S(v_i^{'})$를 Fig. 1과 같이 정의한다. node의 원호 안의 숫자는 각 node의 weight을 의미한다. 이러한 방법으로 BC의 instance $G(V, E, w(v))$를 만든다. $K = K' + 2|E'|$라 정의한다.

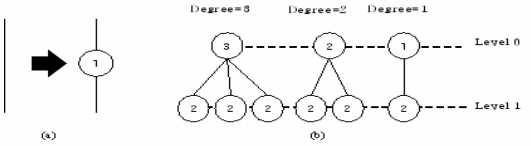

Fig. 1. (a) Transformation of $e' \in E'$ (b) Star component $S(v_i^{'})$

$S(v_i')$ 에서 level-0에 해당하는 node들을 뿌리 node, level-1에 해당하는 node를 리프 node라하고 각각 v_{il_k}', v_{il_k}' 라고 표시한다.

먼저, VC-3의 가능해가 BC의 가능해로 변환됨을 보이겠다. $\overline{V} \subseteq V'$ 가 $|\overline{V}| \le K'$ 를 만족하는 G' 의 가능해라고 하자. 그러면 대응하는 G 에 대해서 다음과 같이 배치들을 구성한다. $v_i' \notin \overline{V}$ 인 node에 대응하는 G 의 star-component들의 모든 리프-node들에 대해 $y_{(v_{ir}', v_{il_k}')}(v_{ir}') = 1$, $y_{(v_{ir}', v_{il_k}')}(v_{il_k}') = 3$ 와 같은 heterogeneous 배치를 구성한다. 그리고 $v_i' \in \overline{V}$ 인 node들에 대응하는 G 의 star-component들에 대해서 다음과 같이 배치를 구성한다.

$x_{v_{ir}'} = d(v_{ir}')$, $y_{(v_{il_k}', v_{ij}')}(v_{il_k}') = 3$, and

$$\begin{cases} y_{(v_{il_k}', v_{ij}')}(v_{ij}') = 1, \text{ if } v_{ij}' \text{ is not covered by other batch.} \\ y_{(v_{il_k}', v_{ij}')}(v_{ij}') = 0, \text{if } v_{ij}' \text{ is covered by other batch.} \end{cases}$$

where $k = 1,..., d(v_i')$. $\overline{V}$ 는 V3-C의 가능해이기 때문에 BC의 모든 edge-node는 반드시 인접한 star-component들의 리프 node와 heterogeneous 배치를 구성하게 된다. 따라서 위의 방법으로 만들어진 배치들은 모든 $v \in V$ 들을 커버한다. 또한 만들어진 배치수의 총합은 $|\overline{V}| + 2|E'|$ 로 $|\overline{V}| \le K'$ 이기 때문에 $K = K' + 2|E|$ 보다 작다.

역으로 G 의 가능해가 G' 의 가능해로 변환됨을 알아보자. S 를 배치 개수가 K 보다 작은 G 의 가능해라고 하자. 만약 edge-node $v_{e'}$ 들 중 incomplete 배치로 구성된 것이 있다면 우리는 edge-node $v_{e'}$ 가 인접한 star-component의 리프 node와 complete homogeneous 배치로 구성되면서 S 보다 배치수가 많지 않은 가능해 S' 을 만들 수 있다. 따라서 모든 edge-node $v_{(p',q')} \in S'$ 는 인접한 star-component인 $S(p')$ 나 $S(q')$ 와 함께 배치를 구성되어야 한다. 따라서 모든 star-component들을 두 개의 집합으로 D 와 $\overline{D}$ 로 나뉘어 진다. 여기서 D 는 최소한 한 개 이상의 리프 node가 인근 edge-node와 함께 heterogeneous 배치를 구성한 star-component들의 집합이다. 반면 $\overline{D}$ 는 단 한 개의 리프 node도 인근 edge-node와 함께 heterogeneous 배치를 구성하지 않은 star-component들의 집합이다. 그러면 우리는 D 을 이용해 $\overline{V}'$ 를 다음과 같이 정의할 수 있다. $\overline{V}' = \{v' \in V' \mid S(v') \in D\}$. 모든 edge-node들이 D 의 star-component들과 함께 배치를 구성하였기 때문에 $\overline{V}'$ 는 G' 의 모든 edge를 cover한다. 한 개의 star-component의 weight 합은 $d(v') * \lambda$ 이고 $\lambda = 4$ 이기 때문에 star-component들의 weight 총합은 $8|E|$ 이기 때문에 S' 의 배치수 $2|E| + |D|$ 이다. $2|E| + |D|$ 는 K 보다 작기 때문에 $|D| = |\overline{V}'|$ 는 K' 보다 작다. 따라서 $\overline{V}'$ 는 G' 의 가능해이다.□

어떤 문제가 MAX SNP-hard임 을 보이기 위해서는 알려진 MAX SNP-Complete문제가 증명하고자 하는 문제로 선형변환(L-reduction)이 됨을 보여야 한다. 본 논문에서는 Papadimitriou and Yannakakis (1991)에 의해 MAX SNP-Complete 문제라고 증명된 Maximum Node Cover-B문제로부터 논의를 진행하겠다.

Node Cover-B (NC-B)

Instance: A graph $G' = (V', E')$ with degrees bounded by B.

Question: Find the smallest node cover G', i.e. a smallest subset $\overline{V} \subseteq V'$ such that for each edge $e=(u,v) \in E'$ at least one of u and v belongs to $\overline{V}$?

Theorem 2 NC-B 문제는 우리 문제인 BC로 선형변환 된다.

Proof. 먼저 NC-B의 an instance G'를 우리 문제의 instance G로 변환 시키자. 이것은 앞서 Theorem 1의 변환 과정과 거의 같다. 다만 배치 크기는 $\lambda = B$로 star-component의 weight는 Fig.3.과 같이 변화하였다.

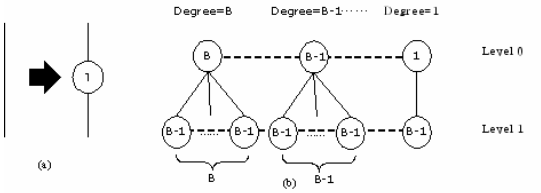

Fig.3. (a) Transformation of $e' \in E'$ (b) Star component $S(v_i')$

선형변환의 첫 조건 만족여부를 논의하기 위해 최적해의 배치 개수를 알아보자. 먼저 $B*OPT(G') \geq |E|$ 임은 자명하다. 또한 star-component weight 합은 $d(v')*\lambda$ 이고 주변 edge-node들의 weight을 모두 합쳐도 λ를 넘지 않기 때문에 $OPT(G) = 2|E| + OPT(G')$ 이다.

$$\begin{aligned} OPT(G) &= 2|E| + OPT(G') \\ &\leq 2B*\mathrm{OPT(G')} + \mathrm{OPT(G')} \\ &= (2B+1)*OPT(G') \end{aligned}$$

따라서 선형변환의 첫 조건이 만족되었다.

이제 우리는 선형변환의 두 째 조건에 대해서 논의하겠다. $\overline{G}'$을 VC-3의 가능해라고 하고 $f(\overline{G}')$을 $\overline{G}'$의 목적값이라고 하자. 같은 방법으로 $\overline{G}$를 $\overline{G}'$를 통해 $\overline{G}$얻은 BC의 가능해라고 하고, $f(\overline{G})$를 이런 의 목적값이라고 하자. 그러면 우리는 다음과 같은 식들을 얻을 수 있다.

$$f(\overline{G}) \geq 2|E| + f(\overline{G}'). \text{since } OPT(G) = 2|E| + OPT(G')$$

$$f(\overline{G}) - OPT(G) \geq f(\overline{G}') - OPT(G')$$

따라서 선형변환의 두 째 조건도 만족한다. 따라서 우리 문제 BC는 MAX-SNP-hard 문제이고 다항시간근사해법군은 존재하지 않는다. □

Theorem1에서 알 수 있듯이 우리 문제의 최적 알고리듬에 대해서 논의하는 것은 무의미한 일이다. 게다가 Theorem2에 의해 우리 문제의 다항시간근사해법군을 찾는 것도 매우 어려운 문제임을 확인할 수 있다.

4. Properties of Optimal Solutions

주어진 instance G에 대해서 여러 최적해가 존재할 수 있다. 다음 lemma는 여러 최적해들 중에서 tree 구조를 가지는 최적해가 반드시 존재함을 보인다.

Lemma 1. tree 구조를 가지는 최적해가 반드시 존재한다.

Proof. tree 구조를 가지는 최적해가 존재하지 않는다고 가정하자. 그러면 어떤 최적해도 최소한 하나의 cycle이 존재한다. 일반성을 잃지 않고 최적해가 k개의 edge로 이루어진 cycle $C = <(u_1,u_2),(u_2,u_3)\Lambda\Lambda,(u_k,u_1)>$를 가지고 있다고 가정하자. $p(u_i)$, $n(u_i)$, 그리고 θ를 다음과 같이 정의하자.

$$\begin{aligned} p(u_i) &= y_{(u_i,u_{i+1})}(u_i), \text{ if } i < k \\ &= y_{(u_i,u_1)}(u_i), \text{ if } i = k \\ n(u_i) &= y_{(u_{i-1},u_i)}(u_i), \text{ if } i > 1 \\ &= y_{(u_k,u_i)}(u_i), \text{ if } i = 1 \end{aligned}$$

$$\theta = \min_i \{n(u_i)\}, \quad i = 1,\Lambda\Lambda,n$$

Property 1에 의해 다음과 같은 식을 얻는다.

$$\begin{aligned} & y_{(u_i,u_{i+1})}(u_i) + y_{(u_i,u_{i+1})}(u_{i+1}) \\ &\quad = p(u_i) + n(u_{i+1}) \leq \lambda \quad \text{for } i < k \end{aligned}$$

and $y_{(u_i,u_1)}(u_i) + y_{(u_i,u_1)}(u_1)$

$= p(u_i) + n(u_1) \le \lambda$ for $i = k$.

일반성을 잃지 않고 $\theta = n(u_j)$ 라고 가정하자. θ를 이용하여 cycle C를 제거한 최적해를 구할 수 있다. 우선 모든 i에 대해서 $p'(u_i) = p(u_i) + \theta$, $n'(u_i) = n(u_i) - \theta$ 로 놓는다. 이런 $p'(u_i)$와 $n'(u_i)$를 이용하여 다음과 같이 새로운 가능해를 만든다.

For all $i > 1$, if $i \neq j$ and $i \neq k$

$y'_{(u_{i-1},u_i)}(u_{i-1}) = p'(u_i)$, $y'_{(u_{i-1},u_i)}(u_i) = n'(u_i)$,

else if $i = k$,

$y'_{(u_k,u_1)}(u_k) = p'(u_k)$, $y'_{(u_k,u_1)}(u_1) = n'(u_1)$,

else $i = j$

$x'_{u_{j-1}} = p'(u_{j-1})$, since $n'(u_j) = 0$.

그러면 $x(u_{j-1}) < \lambda$ 이기 때문에 배치의 개수는 변화가 없다. 따라서 새로운 가능해는 cycle C가 제거된 최적해이다.

Lemma1로 인해 우리는 일반적인 그라프 G에서도 최적해는 tree 구조를 가짐을 알 수 있다. 따라서 G가 tree일 때 최적해를 찾는 것이 일반적인 문제를 해결하는데 의미를 지닌다.

5. Optimal Algorithms of Tree

Tree에서 최적 알고리듬을 구하기 위해 먼저 2개의 property를 제안한다. 주어진 그라프 $G = (V, E, w(v))$가 tree일 때 G를 $T = T(V, E, w(V))$ 로 표시하자. $\pi(v)$ 는 node v에 이웃한 node라고 정의하자.

Property 2. $T(V, E, w(v))$ 에서 degree=1이고 weight이 λ보다 큰 모든 node $p \in V$ 는 최적해에서 한 개 이상의 complete 배치가 구성된다.

Proof. 2가지 case를 생각해보자.

case1). $w(p) > 2\lambda$ **.**

$d(p) = 1$이기 때문에, Property 1에 의해 자명하다.

case2). $\lambda \le w(p) < 2\lambda$ **.**

Property가 거짓이라고 가정하자. 그러면 최적해에서 한 개의 homogeneous 배치와 한 개의 heterogeneous 배치가 구성된 $d(p) = 1$ 인 node p가 반드시 존재한다. 우리는 새로운 가능해 $< x'_v, y'_e; v \in V, e \in E >$ 를 아래와 같이 만들 수 있다.

$x'_p = \lambda$, $y'_{(p,\pi(p))}(p) = x^*_p + y^*_{(p,\pi(p))}(p) - \lambda$,

$y'_{(p,\pi(p))}(\pi(p)) = y^*_{(p,\pi(p))}(\pi(p))$.

$x'_p = x^*_p$, for $\forall v \in V$ except p

$y'_e = y^*_e$ for $\forall e \in E$ except $(p, \pi(p))$.

새로운 가능해는 최적해와 배치 개수가 같거나 작다. 따라서 새로운 가능해는 최적해이다. 따라서 property가 성립한다.

Property2로 인해 우리는 tree에서 최적해를 찾을 때 degree가 1이고 weight가 λ보다 큰 node의 경우 가능한 많은 complete homogeneous 배치를 구성할 수 있다. 따라서 앞으로의 논의에서는 degree가 1인 node의 weight는 λ보다 항상 작다고 가정한다.

Degree가 1인 node $v \in V$ 에 대해서 $\pi(v)$와 heterogeneous 배치를 구성할 때 사용할 수 있는 최대 weight는 $\overline{\delta_v}$, 최적해 $< x^*_v, y^*_e; v \in V, e \in E >$ 에서 쓰인 weight는 δ^*_v 라 쓰고 다음과 같이 정의한다.

$\overline{\delta_v} = \lambda - w(v)$, if $w(\pi(v)) \ge \lambda - w(v)$

$= w(\pi(v))$, otherwise.

$\delta^*_v = y^*_{(v,\pi(v))}(v)$.

$\overline{\delta_v}$ 와 δ^*_v를 이용해서 우리의 분석에서 유용하게 쓰일 그라프에서 $\pi(v)$ 를 대신하게 될 4개의 node를 $\overline{\pi}(v)$, $\underline{\pi}(v)$, $\overline{\mu}(v)$, 그리고

$\underline{\mu}(v)$ 라 하면 그들의 weight 함수는 다음과 같다.

$w(\overline{\pi}(v)) = \overline{\delta_v}$, $w(\underline{\pi}(v)) = w(\pi(v)) - \overline{\delta_v}$,

$w(\overline{\mu}(v)) = \delta_v^*$, $w(\underline{\mu}(v)) = w(\pi(v)) - \delta_v^*$.

새로운 node $\overline{\pi}(v)$ 와 $\underline{\pi}(v)$ 를 이용하여 $T(V,E,w(v))$ 를 둘로 분할한 sub-tree $T(\overline{\pi}(v)) = T(V(\overline{\pi}(v)), E(\overline{\pi}(v)), w(v))$ 와 $T(\underline{\pi}(v)) = T(V(\underline{\pi}(v)), E(\underline{\pi}(v)), w(v))$ 를 정의한다.. (see Fig. 4.)

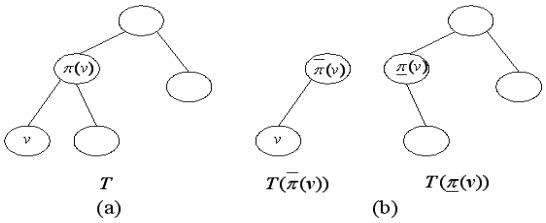

Fig. 4. (a)Given tree, T (b)$T(\overline{\pi}(v))$ and $T(\underline{\pi}(v))$

위의 경우와 같이 $\overline{\mu}(v)$ 와 $\underline{\mu}(v)$ 를 이용해서 $T(\overline{\mu}(v)) = T(V(\overline{\mu}(v)), E(\overline{\mu}(v)), w(v))$ 와 $T(\underline{\mu}(v)) = T(V(\underline{\mu}(v)), E(\underline{\mu}(v)), w(v))$ 를 정의할 수 있다. $V(\overline{\pi}(v))$ ={ v , $\pi(v)$ }이고 $E(\overline{\pi}(v))$ ={ $(v, \overline{\pi}(v))$ }이다. $V(\underline{\pi}(v))$ 는 v 를 제외하고 $\pi(v)$ 대신 $\underline{\pi}(v)$ 이 포함된 모든 node들의 집합이며 $E(\underline{\pi}(v))$ 는 $T(\underline{\pi}(v))$ 에서 정의된 모든 edge들의 집합이다. 이와 비슷하게 nodes $\overline{\mu}(v)$ 와 $\underline{\mu}(v)$ 를 이용하여 우리는 $T(\overline{\mu}(v))$ 와 $T(\underline{\mu}(v))$ 를 각각 정의할 수 있다.

우리는 이렇게 정의된 $T(\overline{\pi}(v))$ 와 $T(\underline{\pi}(v))$ 의 최적해를 결합하여 $T(V,E,w(v))$ 의 가능해를 만들 수 있다.

$$OPT(T(\overline{\pi}(v))) + OPT(T(\underline{\pi}(v))) > OPT(T) \text{ - (1)}$$

반면 $T(\overline{\mu}(v))$ 와 $T(\underline{\mu}(v))$ 는 최적해를 기반으로 $T(V,E,w(v))$ 를 분할하였기 때문에 식(2)를 얻을 수 있다.

$$OPT(T(\overline{\mu}(v))) + OPT(T(\underline{\mu}(v))) = OPT(T) \text{ - (2)}$$

다음 Property는 최적 알고리듬을 위한 분할 정복 접근의 기초를 제공한다.

Property 3. 주어진 $T(V,E,w(v))$ 에 대해서 degree가 1인 모든 node $p \in V$ 에 대해서 다음의 식이 성립한다.

$$OPT(T(\overline{\pi}(p))) + OPT(T(\underline{\pi}(p))) = OPT(T).$$

Proof. Property1과 $\overline{\delta_p}$ 의 정의에 의해서

$$\overline{\delta_p} \geq \delta_p^* \quad \text{- (3)}$$

이다. 또한 $w(\underline{\pi}(v)) = w(\pi(v)) - \overline{\delta_v}$ 이고 $w(\underline{\mu}(v)) = w(\pi(v)) - \delta_v^*$ 이기 때문에 $w(\underline{\pi}(v)) \leq w(\underline{\mu}(v))$ 이다.

$$OPT(T(\underline{\pi}(p))) \leq OPT(T(\underline{\mu}(p))) \quad \text{-(4)}$$

$T(\overline{\pi}(p))$ 와 $T(\underline{\pi}(v))$ 의 각각의 weight의 합은 정의에 의해 λ 보다 작거나 같다. 따라서,

$$OPT(T(\overline{\pi}(p))) = OPT(T(\overline{\mu}(p))) = 1 \quad \text{-(5)}$$

식(4)와 식(5)를 합하면

$$OPT(T(\overline{\pi}(p))) + OPT(T(\underline{\pi}(p))) \leq OPT(T(\overline{\mu}(p))) + OPT(T(\underline{\mu}(p))) \text{ -(6)}$$

을 얻을 수 있다. 그리고 식(2)에 의해

$$OPT(T(\overline{\pi}(p))) + OPT(T(\underline{\pi}(p))) \leq OPT(T) \quad \text{-(7)}$$

를 도출하게 된다. 식(7)과 식(1)에 의해 property가 성립함을 확인할 수 있다.□

Property 2와 Property 3을 기반으로 본 논문에서는 최적 알고리듬을 제안하고자 한다. 알고리듬은 Fig.5에 나와있다

Batch on Tree Algorithm for $T(V,E,w(v))$
Begin
Let $V := V$, $x_v = 0$, $y_e = (0,0)$ **for all** $v \in V, e \in E$;
Sort V **by ascending order using degree of node;**

```
While V ≠ 0 begin
    v := first node in V ;
    if d(v) = 0 begin
        x_v = w(v) ;
    end
    else d(v) = 1 begin
        x_v = ⌊w(v) / λ⌋ * λ ;
        y_(v,π(v))(v) := w(v) − x_v ;
        if w(π(v)) > λ − y_(v,π(v))(v) begin
            y_(v,π(v))(π(v)) := λ − y_(v,π(v))(v) ;
            w(π(v)) := w(π(v)) − y_(v,π(v))(π(v)) ;
        end
        else begin
            y_(v,π(v))(π(v)) := w(π(v)) ;
            w(π(v)) := 0 ;
        end
        d(π(v)) := d(π(v)) − 1 ;
        E = E − {(v, π(v))} ;
        Reorder π(v) using d(π(v)) ;
    end
    V = V − {v} ;
end
End
```

Fig. 5. Batch on Tree Algorithm

이 과정들은 $\pi(v)$ 를 재정렬하는 과정은 $O(\log|V|)$ 이 소요되고 나머지 과정은 상수시간에 실행 가능 하다. 이 과정이 $|V|$ 번 반복되기 때문에 알고리듬의 전체 실행시간은 $O(|V|\log|V|)$ 이다. 또한 V 를 정렬하는 시간도 $O(|V|\log|V|)$ 이면 가능하기 때문에 다음 theorem이 성립하게 된다.

Theorem 3. The Batch on Tree Algorithm 알고리듬의 시간복잡도는 $O(|V|\log|V|)$ 로 제한된다..

6. Conclusion

우리는 배치 커버 문제가 NP-hard문제일 뿐만 아니라 MAX-SNP-hard문제임을 보였다. 그리고 그라프가 tree인 특별한 경우에 최적 알고리듬을 개발하였다. 추후 연구방향으로써 휴리스틱 알고리듬의 개발 및 worst case bound 분석 및 tree이외의 다른 특수한 그라프에서 최적 알고리듬 개발에 대한 연구가 필요하다.

References

[1] S.Y. Chang, M.R. Chang, and Y.S. Hong, “A lot grouping algorithm for a continuous slab caster in an integrated steel mill,” *Production Planning & Control*, vol. 11, pp. 363-368, 2000.

[2] L. Tang, J. Liu, A. Ring, and Z. Yang, “A review of planning and scheduling systems and methods for integrated steel production,” *European Journal of Operational Research*, vol. 133, pp. 1-20, 2001.

[3] Kangbok Lee, S.Y. Chang and Y.S. Hong, “Continuous slab caster scheduling and interval graphs,” *Production Planning & Control* vol. 15, pp. 495-501, 2004.

[4] H.C. Hwang, S.Y. Chang, “Order consolidation for batch processing,” *Journal of Combinatorial Optimization*, vol. 9, pp. 121-138, 2005

[5] S. Micali and V.V. Vazirani, “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*, 1980, pp. 17–27.

[6] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*, Dover Publications: Mineola, NY, 1988.

[7] C.H. Papadimitriou and M. Yannakakis, “Optimization, approximation and complexity classes,” *Journal of Computer and System Sciences*, vol. 43, pp. 425–440, 1991.

[8] M. R. Garey and D. S. Johnson, “the rectilinear Steiner tree problem is NP-complete,” *SIAM J. APPL. MATH*, vol. 32, pp. 826-834, 1977.