

Benders 분해법을 이용한 스케줄링 및 강건한 생산계획

복진광 · 박선원[†]

한국과학기술원 화학공학과
(1998년 4월 28일 접수, 1998년 7월 3일 채택)

Batch Scheduling and Robust Production Planning using Benders Decomposition

Jin-Kwang Bok and Sunwon Park[†]

Dept. of Chemical Engineering, Korea Advanced Institute of Science and Technology
(Received 28 April 1998; accepted 3 July 1998)

요 약

본 연구에서는 화학공정에 관련된 의사결정 문제에서 계산의 효율성을 높이고자 benders 분해법을 이용한 모델을 제시하였다. 이 방법은 비선형 쌍대(雙對)알고리즘을 이용해 큰 규모의 문제를 SUB라는 primal 문제와 MASTER라는 master 문제의 두단계 문제로 분해하여 구성한다. 다루고자 한 문제의 대상 중 하나는 회분식공정에 있어서 설계와 스케줄링 문제를 동시에 고려한 문제이고, 또 다른 하나는 공정규모의 확대를 포함하는 장기 생산계획 문제이다. 단기 스케줄링 문제에서는 설계비용, 조업비용뿐 아니라 재고관리까지 고려해서 NPV를 최소화하는 문제로 구성된다. 생산계획 문제에서는 생산량 및 구입량뿐 아니라 각 공정규모의 확대까지 고려하고 있고, 미래의 수요예측이 불확실하기 때문에 이를 보완하고자 강건한 생산계획모델을 제시하고자 한다. 다른 두 가지 문제들은 결정변수들이 많기 때문에 문제의 규모가 크고, 이에 따라 효과적인 계산방법을 요구하는 문제들이었다. 제시된 모델의 효율성은 단기 스케줄링 문제와 장기 생산계획 문제를 다루는 예제들을 통해 입증하였다.

Abstract— Benders decomposition algorithm to improve the computation for the decision problems involved in chemical processes is addressed. Using nonlinear duality algorithm, a large scale problem is decomposed into two stage problems, SUB and MASTER, respectively. The target problems are short-term scheduling and design problems, and long-term planning problems including capacity expansions. In the first issue, the net present cost involving design cost and operation cost with consideration of inventory is minimized. The latter addresses robust models that are insensitive to future uncertainty. The problems presented in this paper are so computationally complex that enhanced algorithms have been required. The effectiveness of the improved algorithm is illustrated through examples relevant to scheduling and design problem, and long-term capacity expansion problem, respectively.

Key words: Design and Scheduling, Production Planning, Benders Decomposition

1. 서 론

70년대 이후 연속 공정이 주도해온 화학산업은 대부분이 대량생산을 위한 생산방식을 채택하고 있고 이에 따라 무분별한 생산확장으로 이어지기도 한다. 최근 들어 국제경기의 침체에서 비롯된 국내 산업으로의 여파는 점차 생산방식을 구매자의 주문에 따른 다품종 소량생산이나 혹은 정확한 수요예측을 통한 연속생산 방식으로의 변화를 시작하게 하였다. 이에 따라 회분식 공정에 대한 스케줄링 개념은 보다 실질적인 기법으로 우리에게 접근하고 있고, 관점을 긴 시간으로 보고 있는 장기 생산계획에 있어서도 보다 미래의 불확실한 요소에 대한 강건한 해를 요구하게 되었다. 화학공정에 있어 회

분식 공정은 정밀화학제품 등의 주로 소량, 고부가가치 상품에 의해 그 맥을 이어오고 있다. 근래에 들어서는 새로운 제품 생산을 위한 생산설비 구축단계에서도 이미 조업상황을 고려한 연구를 많이 엿볼 수 있다[1]. 이러한 의미에서 단순히 회분식 공정에 대한 스케줄링 연구문제는 확대되어 회분식 공정의 장치의 규모를 결정하는 디자인 문제와 함께 고려되고 있다. 스케줄링 문제에서는 주어진 조업정책(Zero-Wait, NIS, FIS, MIS 등) 생산방식[Single Product Campaign (SPC)과 Multiple Product Campaign(MPC)]에 의해 생산하고자 하는 생산물의 조업순서를 결정하게 되고 디자인 문제에서는 이런 조업이 가능하도록 각 장치의 생산능력(capacity)을 결정하게 된다.

연속생산의 방식인 경우는 여러 종류의 생산물을 만들기 보다 특정 생산물을 주력으로 만들어내는 것을 목적으로 하고 있다. 따라서 연속 생산계획에서 관심을 가져야 할 사항은 각 생산물의 단기 조업

[†]E-mail: swpark@convex.kaist.ac.kr

가이드인 스케줄링보다는 장기 생산계획에 그 초점을 맞추고 있다. 특히 관심을 두는 기간(horizon)에 따라 생산물의 양과 장치들의 shut-down을 결정하는 문제에서 장치들의 생산설비 확장(capacity expansion)을 결정하는 문제로 해석할 수 있다. 특히 이 분야에서는 미래의 상황에 대한 불확실성을 나타내고자 통계적인 개념을 이용하고 있다. 즉 특정기간 후의 예측되는 구매량에 대해 여러 가지 시나리오를 만들어서, 각 시나리오에 확률적인 양만을 고려해 생산계획량을 결정하게 되는 것이다. 특히, 예상되는 어떤 시나리오의 실현에 대해서도 거의 최적값을 제공할 수 있는 강건한 모델(robust model)에 대한 연구가 이루어져야 한다[2,3].

디자인을 고려한 회분식 공정의 스케줄링 문제인 경우는 두 가지를 동시에 고려하기 때문에 문제의 크기가 커져서 계산량이 많아지게 된다. 또한 연속공정의 생산계획 문제도 미래의 불특정한 요소들을 그 수만큼의 시나리오로 나타내야 하므로 문제를 풀기 위한 계산량이 많아지게 된다. 이에 본 연구에서는 이미 나와 있는 기존의 모델들이 실질적인 공정에 응용되기 위해 빠른 시간내에 해를 찾아낼 수 있도록 모델을 개선하고자 한다. Benders decomposition 알고리즘을 이용하여 큰 사이즈의 문제를 두 단계의 문제로 구성하여 반복계산을 통해서 최적해를 찾아내고자 한다.

제2절에서는 회분식 공정에서 스케줄링과 디자인을 고려한 문제와 장기생산계획문제를 기술하였고 3절에서는 benders decomposition algorithm에 대한 설명 및 모델 개선을, 그리고 4절에서는 예제를 통한 알고리즘의 효율성을 보이고자 한다.

2. 모델식

2-1. 회분식 공정

화학공정에서 회분식 공정의 비중은 상당하다. 제약(製藥)공정의 경우는 80%, 식품 및 음료생산 공정의 65%가 회분식 공정에 의해 이루어지고 있다[4]. 디자인과 스케줄링을 함께 고려하는 문제는 많은 논문에서 다루어져 왔다.

Moon 등[1]은 다품종 회분식 공정에 있어서 디자인과 스케줄링을 함께 고려하는 문제를 다루었다. 여기서는 주어진 주문량을 맞추기 위한 각 장치의 사이즈와 생산방식 및 생산순서를 결정하는 문제로 convex화된 MINLP를 제시하였다. Birewar와 Grossmann[5]이 제시한 문제에 대해 효과적인 zero-wait 스케줄링 알고리즘[6]을 이용하여 문제의 해를 개선하였다.

Voudouris와 Grossman[7]은 다품종 회분식 공정에 있어서 재고관리를 고려한 디자인 및 스케줄링 문제를 다루었다. 각 장치에 대해 가능한 사이즈를 선택하게 함으로써 비선형성을 해결하였고 목적함수에서도 단순한 디자인 비용뿐 아니라 조업에 관련된 재고관리 비용을 다룸으로써 전체 Net Present Value(NPV)를 최대화하고자 하였다. 이들은 이러한 알고리즘을 확대해 다목적 공정에 대해서도 다루었다[8].

본 연구에서 다루고자 하는 문제에서 설정한 가정은 다음과 같다.

- (a) 주문되는 모든 생산물은 같은 생산경로를 갖는다(multi-product plant).
- (b) 제품의 특성상 조업시 중간저장탱크에 저장되어선 안되고 한 공정이 끝난 즉시 다른 공정으로 이동해야 한다(zero-wait policy).
- (c) 각 장치에서의 조업시간은 장치의 크기와는 무관하게 일정하다.
- (d) 공장의 생산방안은 single product campaign(SPC)에 따른다.

각 생산물을 i , 각 장치를 j 라 하고 각 생산물에 대한 주문량이 Q_i 로 주어졌다고 가정하자. 각 주문량은 스케줄링 horizon H 안에 마쳐야 하고 이 기간 동안 생산은 일정한 주기 NC 를 갖는다고 하자. 각 주기에서의 생산량은 $q_i = Q_i/NC$ 로 생각할 수 있고 주기의 크기는 $P =$

H/NC 라고 할 수 있을 것이다. 주문량을 맞추기 위해 장치 j 가 필요하게 되고 장치의 크기는 부피 V_j 로 나타낼 수 있다. 그러면 주어진 크기요소(size factor, S_{ij})에 대해 각 생산물의 한 회분량 B_i 와 부피는 다음과 같은 관계를 갖는다.

$$V_j \geq S_{ij} B_i$$

그리고, 각 주기당 생산물의 생산회수는 다음과 같이 나타낼 수 있다.

$$n_i = q_i / B_i$$

목적함수로 하고자 하는 전체 현재 비용(Net Present Cost, NPC)는 다음과 같이 나타낼 수 있다.

$$NPC = Pc + Oc(1 - tx)(procoef) - (Pc/Ny)tx(procoef)$$

여기서 Pc 는 디자인에 필요한 비용으로 다음과 같이 계산된다.

$$Pc = \sum_j \alpha_j V_j^{\beta_j}$$

주기를 갖는 조업의 비용 Oc 는 다음과 같이 나타낼 수 있다[7].

$$Oc = \sum_i [\mu_i (Q_i/2)(P - T_i)] + miNC$$

여기서 μ_i 는 조업비용을, T_i 는 생산물 i 를 생산하는데 들어간 총 시간, NC 는 주기의 회수, 그리고 mi 는 그 주기마다 고려되는 비용을 나타낸다. tx 는 세율을 나타내고 $procoef$ 는 현재값 상수(present value coefficient)로 다음과 같이 계산된다.

$$procoef = \frac{(1 + in)^{Ny} - 1}{in(1 + in)^{Ny}}$$

Ny 는 공장의 life time이고 in 은 이자율을 나타낸다.

디자인 비용을 계산하는 식이 nonconvex한 식이 되므로 이를 convex화 하고자 다음과 같이 할 수 있다. 각 장치의 이용 가능한 부피(v_{js})가 주어졌다고 하고, 장치 j 가 크기 s 를 갖는다면 1, 나머지 경우는 0을 갖는 이진 변수 y_{js} 를 선언함으로써 convex하게 만들어 줄 수 있다[7].

또한 주기에 따른 생산 회수 식에서 나타난 비선형 성을 없애기 위해 Voudouris와 Grossman[7]은 다음과 같은 과정을 거쳤다. 우선 $q_i = Q_i/NC$, $V_j \geq S_{ij} B_i$ 그리고 $n_i = q_i/B_i$ 를 통합하여 다음과 같은 식으로 나타내었다.

$$n_i \geq \sum_s \left(\frac{q_i S_{ij}}{v_{js}} \right) y_{js}$$

$$\sum_s y_{js} = 1$$

여기서 비선형성을 없애기 위해 양수 연속변수인 e_{js} 를 추가해서 다음과 같이 선형화했다.

$$n_i \geq \sum_s \left(\frac{S_{ij} Q_i}{v_{js} H} \right) e_{js}$$

$$e_{js} \leq H y_{js}$$

$$P = \sum_s e_{js}$$

위에서 제시된 모델들을 종합하여 전체 모델을 다음과 같이 나타낼 수 있다.

$$\min NPC = Pc + Oc(1 - tx)(procoef) - (Pc/Ny)tx(procoef) \quad (1)$$

subject to

$$n_i \geq \sum_s \left(\frac{S_{ij} Q_i}{v_{js} H} \right) e_{js} \quad \forall i, j \quad (2)$$

$$e_{js} \leq H y_{js} \quad \forall i, s \quad (3)$$

$$P = \sum_s e_{js} \quad \forall j \quad (4)$$

$$Hq_i = P Q_i \quad \forall j \quad (5)$$

$$\sum_k NP_{ik} = n_i \quad \forall i \quad (6)$$

$$\sum_i NP_{ik} = n_k \quad \forall k \quad (7)$$

$$T_i = \left[n_i t_{im} + \left(\sum_k NP_{ik} S_{ikm} \right) \right] \quad \forall i \quad (8)$$

$$P \geq \sum_i \left[n_i t_{ij} + \left(\sum_k NP_{ik} S_{ikj} \right) \right] \quad \forall j \quad (9)$$

$$NP_{ii} = n_i - 1 \quad \forall i \quad (10)$$

$$H = \sum_{sv} sv h_{sv} \quad (11)$$

$$\sum_{sv} h_{sv} = P \quad (12)$$

$$h_{sv} \leq H r_{sv} \quad \forall sv \quad (13)$$

$$Pc = \sum_j \sum_s C_{js} y_{js} \quad (14)$$

$$Oc = \sum_i [\mu_i (Q_i / 2) (P - T_i)] + \text{minC} \quad (15)$$

$$\sum_s y_{js} = 1 \quad (16)$$

$$\sum_{sv} r_{sv} = 1 \quad (17)$$

$$q_i \geq 0 \quad \forall i, NP_{ik} \geq 0 \quad \forall i, k \quad (18)$$

$$P \geq 0 \quad (19)$$

$$n_i \in \text{integer} \quad \forall i \quad (20)$$

$$y_{js}, r_{sv} \in \{0, 1\} \quad (21)$$

2-2. 장기 생산계획 모델

본 연구에서 다루고자 하는 대상은 화학공정에서의 장기 생산계획 문제로서 생산량 및 원료 구입량을 결정할 뿐만 아니라 장치의 규모 확장(capacity expansion)이 포함된다. Sahinidis 등[9]은 규모 확장 문제로서 MILP 식을 제안했지만 미래의 값이 결정적(deterministic)이라는 가정을 두었다. 실제 장기 생산계획에서는 많은 상수 값들이 불확정하다고 보아야 한다는 게 일반적인 견해이다[2]. 대부분의 경우는 이를 고려하고자, 예측되는 주문량을 불확정하다고 보고 문제를 다루고 있다[10, 11]. 이에 따라 각 상수값이 취할 수 있는 경우에 대한 문제를 모두 풀어내야 하므로 알고리즘에 있어 효율성이 요구되었다. Two-stage algorithm은 이러한 이유에서 많은 관심을 불러 모았고 지금까지 활용되어오고 있다[2, 12, 13].

특정 시나리오가 실현될 것이라고 가정하고 공정의 규모확장이나 공정조업조건을 결정한다면 두 가지 문제가 발생할 수 있다. 우선, 가정한 시나리오가 실현된다면 다행히도 최적의 값을 구한 경우지만, 실제 시나리오와 어긋날 경우는 공정의 규모확장이나 조업으로 인한 기대값이 줄어들게 된다. 더군다나, 무분별한 공정규모 확장의 결과로 이어질 수 있으므로 이는 미래에 대한 계속적인 유지비용 측면에서 상당한 손실로 고려될 수 있다. 이러한 불확정적인 미래의 시나리오에 대해 언제나 최적에 가까우면서도 규모확장에 있어서 오차를 줄이고자 하는 것이 강건한 모델의 목표이다. 강건한 모델은

Markowitz[14]가 제안한 모델 framework으로 Malcolm과 Zenios[15]가 발전소에서의 전기량 용량을 확대하는 문제에 이용하였다. 모델의 robustness를 나타내주는 수치는 두 가지이다. 첫번째는 solution robustness로서, 각 시나리오에 대해 민감하지 않게 최적해에 가까운 해를 찾는 정도를 나타내고, 두번째 model robustness는 overcapacity나 lost sales가 없는 정도를 나타낸다.

강건한 모델을 수식으로 나타내면 다음과 같다.

$$\begin{aligned} &\text{maximize} && \Phi = E(\xi) - \lambda E[\text{dev}(\xi)^2] - \omega E(Z_s^2) \quad (\text{ROBUST}) \\ &\text{subject to} && g(x) \leq 0 \\ &&& h(x) = 0 \end{aligned}$$

여기서는 ξ 는 NPV를 나타내고, 이들의 값들의 기대값을 최대화하는 것만을 고려한게 아니라 각 시나리오가 실현되었을 때로부터의 차이 값인 $\text{dev}(\xi)^2$ 과 이로 인한 overcapacity와 undercapacity를 고려하게 된다.

본 연구에서는 이 강건한 모델을 화학공정의 장기 생산계획 문제에 이용하고자 한다. 위의 ROBUST 식을 장기 생산계획 문제에 적용하면 다음과 같이 나타낼 수 있다.

$$\begin{aligned} &\text{Maximize} && \text{Objective function} \\ &&& = \text{Expected NPV} - \text{Expected square of deviation of NPV} \\ &&& - \text{Expected square of excess capacity} \end{aligned}$$

$$\Phi = \sum_s p_s \xi_s - \lambda_s \sum_s p_s \left(\xi_s - \sum_{s'} p_{s'} \xi_{s'} \right)^2 - \omega \sum_s \sum_t \sum_i Z_{sit}^2 \quad (22)$$

subject to

$$\begin{aligned} \xi_s = & \sum_j \sum_l \sum_t (\gamma_{jlt} S_{sjlt} - \Gamma_{jlt} P_{sjlt}) - \sum_i \sum_t \delta_{it} W_{sit} \\ & - \sum_s \sum_t (\alpha_{it} Q E_{it} + \beta_{it} Y_{it}) \end{aligned} \quad (23)$$

$$Q_{it} = Q_{i,t-1} + Q E_{it} \quad \forall i, t \quad (24)$$

$$Y_{it} Q E_{it}^L \leq Q E_{it} \leq Y_{it} Q E_{it}^U \quad \forall i, t \quad (25)$$

$$P_{sjlt} \leq a_{sjlt}^U \quad \forall s, j, l, t \quad (26)$$

$$S_{sjlt} \leq d_{sjlt}^U \quad \forall s, j, l, t \quad (27)$$

$$W_{sit} = \mu_{ij} W_{sit} \quad \forall s, i, j, t \quad (28)$$

$$W_{sit} = \eta_{ij} W_{sit} \quad \forall s, i, j, t \quad (29)$$

$$\sum_l S_{sjlt} + \sum_i W_{sit} - \sum_l P_{sjlt} - \sum_i W_{sit} = 0 \quad \forall s, j, t \quad (30)$$

$$Z_{sit} = Q_{it} - W_{sit} \quad \forall s, i, t \quad (31)$$

$$Q E_{it}, Q_{it}, W_{sit}, P_{sjlt}, S_{sjlt}, Z_{sit} \geq 0 \quad (32)$$

$$Y_{it} \in \{0, 1\} \quad (33)$$

여기서 쓰인 문자들에 대한 설명이다.

(a) indices and sets

i=1, NP	Process
j=1, NC	Chemicals
l=1, NM	Market
s=1, NS	Scenario
t=1, NT	Time period

(b) Variables :

Y_{it}	: 0-1 variable to denote if the capacity of process i is expanded at time period t
P_{sjlt}	: Amount of product j purchased from market l at the beginning of time period t for scenario s
Q_{it}	: Total capacity of process i available in period t

- QE_{it} : Capacity expansion of process i to be installed in period t
 S_{jlt} : Amount of product j sold to market l at the beginning of time period t for scenario s
 W_{sit} : Operating level of process i at time period t for scenario s
 WI_{sit} : Amount of chemical j consumed by process i at time period t for scenario s
 WO_{sit} : Amount of chemical j produced by process i at time period t for scenario s
 Zp_{sit} : Excess capacity of process i at time period t for scenario s
 ξ_s : Net present value for scenario s
 (c) Parameters :
 α_{jlt}^u : Upper bound for purchase of chemical j from market l at time period t under scenario s
 d_{jlt}^u : Upper bound for sales of chemical j from market l at time period t under scenario s
 p_s : Probability of demand scenario s
 QE_{it}^L : Lower bound for the capacity expansion of process i at time period t
 QE_{it}^U : Upper bound for the capacity expansion of process i at time period t
 α_{it} : Unit cost for the capacity expansion of process i at time period t
 β_{it} : Fixed cost for the capacity expansion of process i at time period t
 γ_{jlt} : Price of sales of chemical j in market l during time period t
 Γ_{jlt} : Price of purchases of chemical j in market l during time period t
 δ_{it} : Unit operating cost for process i
 η_{ij} : Material balance coefficients for process i and input chemical j (zero if j is not an input to process i)
 λ : Penalty term for expected deviation of net present values
 μ_{ij} : Material balance coefficients for process i and output chemical j (zero if j is not an input to process i)
 ω : Penalty term for the excess capacities by different scenarios

식 (22)는 목적함수로서 NPV의 기대값에 각 시나리오가 실현될 경우 값들의 분산과 overcapacity되는 양을 고려한다. (23)은 NPV를 계산하는 식으로 생산물 판매이익에서 원료구입비용과 조업비용, 그리고 규모확장 비용을 뺀 값으로 나타낼 수 있다. 주어진 생산계획 모델은 전체 생산계획 horizon을 일정한 시간크기로 나누어서(discretizing) 각 의사결정을 고려하고 있다. (24)는 각 시간대에서 각 공정의 생산규모를 나타낸다. 전 시간대의 생산규모에 확장된 생산 규모를 더하게 된다. (25)에서는 생산규모 확장에 제한을 나타내는 식이다. Y_{it} 는 확장 여부를 나타내기 위한 이진 변수로 1을 취하면 확장을, 0을 취하면 비확장을 의미한다. 판매량과 구입량은 (26), (27)과 같이 나타낼 수 있고 (28), (29)는 (30)의 각 시나리오와, 물질과, 시간에 따른 물질수지식을 나타내기 위해 쓰이고 있다. (31)은 overcapacity를 구하기 위한 식이다.

3. Decomposition Algorithm

앞에서 언급한 바와 같이 스케줄링과 디자인을 함께 고려하는 문제나, 장기 생산계획 문제에서 가장 필요한 것 중의 하나는 큰 규모의 문제를 효과적으로 풀어낼 수 있는 알고리즘이다. 이에 본 연구에서는 기존의 모델을 두 단계의 문제로 만들어내는 decomposition

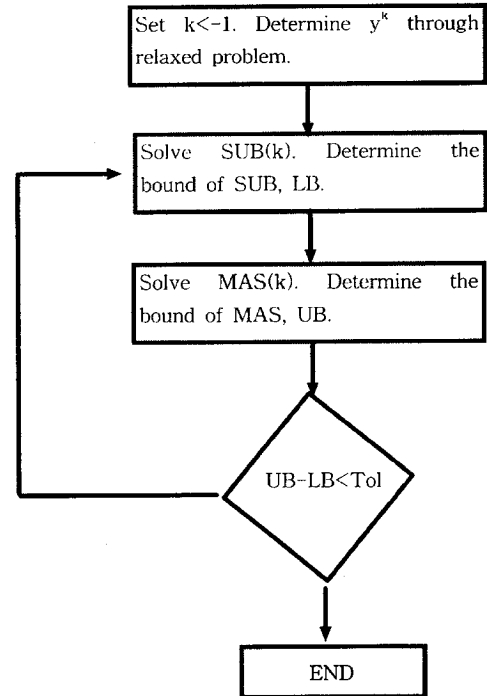


Fig. 1. Benders based decomposition algorithm.

알고리즘을 이용하고자 한다. Decomposition의 알고리즘은 benders decomposition(see APPENDIX)을 응용한 것으로 스케줄링 및 디자인 문제에서는 디자인과 스케줄링을 각각 고려하는 두 개의 MILP 문제로 나타내었고, 장기 생산계획 문제에 있어서는 규모확장과 이를 위한 생산관리를 각각 MINLP와 NLP로 나타내었다. 두 가지 모두 benders decomposition을 적용하기 위한 세 가지 조건을 만족했기 때문에 이를 응용할 수 있었다[2]. Benders decomposition 알고리즘을 통해 큰 규모의 문제는 SUB와 MASTER의 두 단계 문제로 나타낼 수 있다. 미리 각 변수들을 SUB 문제와 MASTER 문제의 변수로 나누어서, SUB는 MASTER의 변수의 projection을 통해, MASTER는 nonlinear duality를 이용해서 유도해낸다. 그리고 Fig. 1에서 나타난 반복계산을 통해 해를 구한다. SUB의 변수를 x , MASTER의 변수를 y 라고 할 때, 처음에는 relaxed된 문제를 통해 초기의 y 의 해를 구한다. 이 y 의 projection을 통해(상수로 간주해서) x 의 해를 구하게 된다. x 의 해는 다시 MASTER문제에서 상수로 취급되고, y 를 구하게 된다. 이번에는 MASTER의 해가 SUB의 해를 구하기 위해서 이용된다. 이러한 반복작업은 MASTER에서 제시하는 bound와 SUB에서 제시하는 bound의 차가 일정한 수준 이하일 때 끝이 나게 된다.

3-1. 스케줄링 및 디자인 문제

SUB 문제에서는 디자인에 관련된 결정을, MASTER 문제에서 결정된 디자인에 대한 재고관리 및 스케줄링을 통해 목적함수값을 최대화하고자 한다. SUB의 변수로는 P_c , y_j 를 MASTER의 변수로는 그 나머지 모두를 취하였다. 그러면 k 번째 iteration에서의 SUB의 문제는 다음과 같이 나타낼 수 있다.

$$\min \quad NPC = P_c + O_c^k(1 - tx)(procoef) - (P_c/Ny)tx(procoef) \quad (34)$$

subject to

$$e_{js}^k \leq Hy_{js} \quad \forall j, s \quad (35)$$

Constraints (14), (16)

$$y_{js} \in \{0, 1\} \quad (36)$$

유도된 k번째 iteration에서의 MASTER 문제는 다음과 같이 나타낼 수 있다.

$$\begin{aligned} \min \alpha \\ \alpha \geq Pc^k + Oc(1 - tx)(procoef) - (Pc^k / Ny)tx(procoef) \\ + \sum_j \sum_s \rho_{js} Cc_{js} - Hy_{js}^k \end{aligned} \quad (37)$$

subject to

Constraints (2), (4)-(13), (15)-(20)

$$r_{sv} \in \{0, 1\} \quad (38)$$

여기서 α 는 benders cut을 나타내고 ρ_{js} 는 식 (3)의 Lagrange multiplier의 값을 나타낸다.

3-2. 장기 생산계획

SUB 문제에서는 판매량, 구입량, 조업량(S_{jlt} , P_{jlt} , and W_{sit})을 결정하게 되고, MASTER문제에서는 규모확장과 관련된 변수들(Y_{it} , Q_{it} , and QE_{it})을 결정하게 된다. k번째 iteration에서의 SUB의 문제는 다음과 같이 표현된다.

Maximize (22)

subject to

$$\begin{aligned} \xi_s = \sum_j \sum_t (\gamma_{jlt} S_{jlt} - \Gamma_{jlt} P_{jlt}) - \sum_i \sum_t \delta_{it} W_{sit} \\ - \sum_s (\alpha_{it} QE_{it}^k + \beta_{it} Y_{it}^k) \end{aligned} \quad (39)$$

$$Zp_{sit} = Q_{it}^k - W_{sit} \quad (40)$$

Constraints (26)-(30)

$$W_{sit}, P_{jlt}, S_{jlt}, Zp_{sit} \geq 0 \quad (41)$$

k번째 iteration에서의 MASTER 문제는 다음과 같이 표현된다.

Maximize α

subject to

$$\begin{aligned} \alpha \leq \sum_s p_s \xi_s - \lambda_s \sum_s p_s \left(\xi_s - \sum_s p_s \xi_s \right)^2 - \omega \sum_s \sum_t \sum_t Zp_{sit}^2 \\ + \sum_s \sum_i \sum_t \rho_{sit}^k (W_{sit}^k - Q_{it}) \end{aligned} \quad (42)$$

$$\xi_s = \sum_j \sum_t (\gamma_{jlt} S_{jlt}^k - \Gamma_{jlt} P_{jlt}^k) - \sum_i \sum_t \delta_{it} W_{sit}^k$$

$$- \sum_s \sum_t (\alpha_{it} QE_{it} + \beta_{it} Y_{it}) \quad (43)$$

Constraints (24), (25), (33)

$$QE_{it}, Q_{it} \geq 0 \quad (44)$$

4. 예 제

본 절에서는 새롭게 제안된 모델식의 효율성을 입증하기 위하여, 회분식 공정의 스케줄링 및 디자인 문제, 장기 생산계획 문제의 예제를 다루어 보았다. 첫 번째 경우는 이미 논문에 발표된 회분식 공정 [1]을 대상으로 Voudouris와 Grossmann[7]과의 결과를 비교해 보았고, 두 번째의 경우는 강건한 모델의 효율성을 보이면서, 계산량이 줄어든다는 것이 보이고자 하였다. 수식의 모델링을 위해서는 GAMS [16]를 이용하였고, MIP의 solver로는 OSL[17], NLP의 solver로는 CONOPT[18]를 이용하였다.

4-1. 회분식 공정의 스케줄링 및 디자인

6종류의 생산물을 생산하는 회분식 공정을 설계하고자 한다. 각 생산물에 대한 요구량 및 각 장치에서의 가공시간은 Table 1에 나타나 있다. 수요량이 만족되어야 하는 horizon은 6000으로 주어지고, 각 장치마다 설계 가능한 용량은 {15000, 20000, 25000, 30000, 40000}으로 주어지고 있다. 각 장치에 대한 economic상수인 α , β 는 각각 250과 0.6으로 주어졌고, 공장의 life time은 10년, 세율은 45%, 이자율은 10%이다. 생산주기가 바뀔으로써 발생하는 준비 비용(mi)은 1로 하였다.

Table 2에서는 최적화의 결과를 보여주고 있다. Voudouris와 Grossmann[7]의 결과와 동일한 설계 및 조업조건을 제시하면서 빠른 시간에 해를 발견했다.

이번에는 생산물의 숫자와 장치의 숫자를 늘려가며 해를 비교하였다(Table 3). 생산물과 장치의 숫자가 각각 {6, 4}, {10, 10}, {15, 8}, 그리고 {20, 8}인 문제를 풀어서 계산시간을 나타낸 것이다. 문

Table 2. Optimization results of example 1

	0-1 variables		Continuous variables		Constraints		NPC	CPU [sec]
Voudouris	72		154		139		58E06	1.07
Benders-based decomposition	SUB	MAS	SUB	MAS	SUB	MAS	58E06	0.81
	20	52	22	133	26	114		

Table 3. Comparison of CPU for each process

	{Product no., Unit no}			
	{6, 4}	{10, 10}	{15, 8}	{20, 8}
Voudouris	1.07	3.7	10.3	17.1
Benders-based decomposition	0.81	2.1	5.6	7.3

Table 1. Processing time and demand for each product

Product	Processing times[hr]				Size factor[l/kg]				Q[kg]
	unit1	unit2	unit3	unit4	unit1	unit2	unit3	unit4	
A	6	2	4	1	2	3	2	6	300000
B	1	5	3	5	7	3	1	2	200000
C	2	7	3	7	1	4	3	2	400000
D	8	1	5	2	5	5	2	6	400000
E	4	1	2	2	1	6	3	2	100000
F	3	6	2	4	4	2	1	4	100000

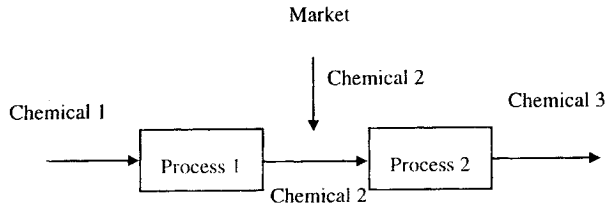


Fig. 2. Chemical processing network for the example.

제의 규모가 커짐에 따라 decomposition을 통한 알고리즘의 효율이 좋아지는 것을 볼 수 있다. 이는 실제 규모의 문제를 다루고자 할 때 발생하는 문제의 크기문제를 해결해 줄 수 있으리라 생각된다.

4-2. 생산계획 문제

Fig. 2는 다루고자 하는 화학공정 네트워크를 나타낸다. Chemical 1은 process 1을 위한 원료로 쓰이며 이를 통해서 Chemical 2가 나온다. Chemical 2는 Process 2의 원료 물질로 쓰이게 되며, Chemical 3이 최종 생산물로 나오게 된다. Chemical 3의 수요를 예측해서 Chemical 1의 구입량, Chemical 2의 생산량 및 구입량, Chemical 3의 생산량을 결정하게 된다. Chemical 3의 수요량은 쉽게 예측하기 어렵기 때문에 여러 가지 발생 가능한 사건들을 고려해 3가지의 시나리오로 나타낼 수 있다고 하자.

Fig. 3은 각 물질의 가격, 구입 가능량, 그리고 Chemical 3에 대한

수요량의 시나리오를 나타내고 있다. Table 4는 판매량의 상한(上限) 값(d_{ijt}^u), 구입량의 상한 값(a_{ijt}^u), 판매가격(γ_{ijt}^u), 조업비용(δ_{ijt}^u), 그리고 물질수지에 관련된 계수(η_{ij} , μ_{ij})를 나타내고 있다. 전체 생산계획 horizon을 12개의 균일한 discrete time으로 나타내었다.

Table 5는 최적화를 수행한 결과이다. 상단에서는 각 시나리오가 실현되었을 경우를 가정한 deterministic한 경우에, 그리고 하단에서는 강건한 생산계획 모델을 통한 NPV의 기대값, solution robustness, 그리고 model robustness를 나타내었다. Solution robustness로서는 각 NPV들의 표준편차로, model robustness로서는 overcapacity를 통해 나타내었다. 물론 예측했던 시나리오가 실현되었을 경우의 NPV는 강건한 모델의 값보다 나은 값을 제시하지만, 다른 시나리오가 실현되었을 경우는 그 값의 변동이 심하다. 반면 강건한 모델을 사용했을 경우는 시나리오에 상대적으로 덜 민감한 해를 찾았고, over capacity 또한 상당히 줄일 수 있었다. Table 5는 계산의 효율성을 보이고 있다. 계산량에 크게 영향을 미치는 이진변수는 MASTER의 변수로, 나머지 생산관리 변수들은 SUB 변수로 나눔으로써 계산에 효율성을 높였다. Bok 등[2]은 이를 이용해 한국 석유화학에서 발생하는 공정 네트워크를 대상으로 하였고, 큰 규모의 문제에 대해서 해를 찾아내는 것을 보였다.

5. 결 론

국내정기의 악화 및 세계시장의 불황은 화학공정 생산관리 시스템

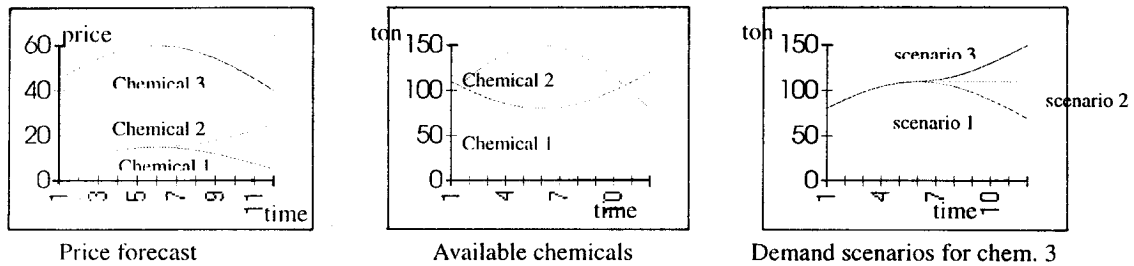


Fig. 3. Data for price, chemical amounts, and demand scenarios.

Table 4. Data for production planning

	Time Horizon											
	1	2	3	4	5	6	7	8	9	10	11	12
d_{11t}^u	80.35	90	98.2	104.6	108.6	110	108.6	104.6	98.2	90	80.35	70
d_{21t}^u	80.35	90	98.2	104.6	108.6	110	110	110	110	110	110	110
d_{31t}^u	80.35	90	98.2	104.6	108.6	110	111	115	121	130	140	150
a_{11t}^u	119.7	110	101.7	95.35	91.36	90	91.36	95.36	101.7	110	119.7	130
a_{21t}^u	98.12	115	129.5	140.6	147.6	150	147.6	140.6	129.5	115	98.12	80
a_{31t}^u	0	0	0	0	0	0	0	0	0	0	0	0
γ_{11t}^u	7.588	10	12.07	13.66	14.66	15	14.66	13.66	12.07	10	7.588	5.0
γ_{21t}^u	22.41	20	17.93	16.34	15.34	15	15.34	16.34	17.93	20	22.41	25
γ_{31t}^u	45.18	50	54.14	57.32	59.31	60	59.32	57.32	54.14	50	45.18	40
δ_{1t}^u	5	5	5	5	5	5	5	5	5	5	5	5
δ_{2t}^u	8	8	8	8	8	8	8	8	8	8	8	8
	Chemicals											
	1	2	3									
η_{1j}	1.1	0	0									
η_{2j}	0	1.05	0									
μ_{1j}	0	1	0									
μ_{2j}	0	0	1									

Table 5. Comparison of optimization results with deterministic model

		Expected NPV	Solution robustness	Model robustness
Deterministic	(Realized scenario)			
	1	30778.6	868.97	663.8
	2	2453.6	978.15	674.2
	3	32844.5	1335.38	791.2
Robust	(λ, ω)			
probability	(1.0, 0.1)	30368.7	0.58	199.06
(0.33, 0.33, 0.33)	(0.1, 0.1)	30369.2	6.35	199.15
	(0.1, 0.5)	28928.7	9.41	96.02
	(0.1, 1.0)	28481.2	587.74	71.59

Table 6. Computational statistics for the example

Type	0-1 variables	Continuous variables	Constraints	CPU(sec)	Iterations
MASTER	24	82	82	3.73	6
SUB	0	730	730	6.24	

템에 있어 더욱 효율적인 개선을 요구하고 있다. 주문에 의한 다품종 소량 생산방식이 점차적으로 중요해짐에 따라 공정설계를 할 때 조업정보를 고려하는 스케줄링 및 디자인 문제에 대해 많은 연구가 되어오고 있고, 연속조업에 있어서도 미래의 불확실성을 고려할 수 있는 모델에 대한 연구가 한창이다. 본 연구에서는 재고를 고려한 스케줄링 및 디자인 문제와 수요량의 예측이 여러 가지 시나리오로 나올 때 각각에 대해 최적에 가까운 해를 찾도록 새롭게 제안된 강건한 생산계획 모델을 다루었다. 이러한 문제들은 단순히 스케줄링 문제를 다루는 경우, 혹은 수요량의 특정 시나리오에 기반을 두고 푼 생산계획 문제보다 더욱 계산량이 많아지게 되었고 이를 해결할 수 있는 효율적인 알고리즘이 최우선으로 필요하게 되었다. 이에 본 연구에서는 benders 분해 알고리즘을 이용해서 모델을 구성함으로써 계산에 효율성을 높였다. 계산량이 큰 문제를 두 단계의 문제로 다시 구성, 변수들을 분류하였다. 스케줄링 및 디자인 문제에서는 디자인에 대한 문제와 스케줄링 재고관리를 위한 문제로, 장기생산계획 문제에서는 규모확장 문제와 나머지 공정관리 문제로 나누었다. 예제를 통해, 스케줄링 및 디자인 문제에서는 기존의 모델[7]에 대해 계산의 효율성을 보였고, 장기생산계획 문제에서는 solution robustness와 model robustness가 향상된 solution의 개선과 함께 계산량의 효율적 관리를 보였다. 이런 decomposition 알고리즘은 두 가지를 한꺼번에 고려해야 하는 복합적인 문제에 대해 모델이 요구조건을 만족한다면 아주 효율적으로 계산시간을 줄일 수 있을 것으로 기대된다.

감 사

본 연구는 한국과학재단 지정 우수 연구센터인 포항공대의 지능자동차 연구센터를 통한 과학재단의 부분적인 재정지원에 의해 수행되었으므로 이에 감사를 드립니다.

부 록

Benders 분해법

1. Formulation

Geoffrion[19]은 Benders[20]가 제안한 분해법을 일반화된 최적화 문제에 적용하기 위해 구조적이고 명확하게 정리하였다. 일반적인 최적화 문제가 다음과 같이 표현되고 밑의 세 조건을 만족한다고 하자.

$$\begin{aligned} \text{Min } f(x, y) \\ \text{s.t. } h(x, y) &= 0 \\ g(x, y) &\leq 0 \\ x \in X, y \in Y \end{aligned} \quad (a1)$$

그리고 다음의 3조건을 만족한다고 하자.

C1: X 은 nonempty, convex set이고 목적함수와 inequality constraints는 고정된(fixed) y 에 대해 convex, equality constraints는 선형(linear)이다.

C2: 집합 $Z_y = \{z : h(x, y) = 0, g(x, y) \leq z \text{ for some } x \in X\}$ 이 고정된 y 에 대해 닫혀있다.

C2: 고정된 y 가 다음과 같이 표현될 때,

$$V = \{y : h(x, y) = 0, g(x, y) \leq 0 \text{ for some } x \in X\}$$

다음의 한 가지가 만족된다:

(i) (a1)의 결과가 finite solution과 optimal multiplier vector를 갖는다.

(ii) (a1)이 unbounded이다.

그러면 (a1)은 다음과 같이 subproblem SUB와 master problem MAS로 decomposition해도 최적해가 보장된다[19].

2. Subproblem

Subproblem은 y 를 고정시켜서(y_k 로 표현: k 는 iteration counter) 다음과 같이 표현된다. 즉 k 번째의 subproblem은

$$\begin{aligned} \text{Min } f(x, y^k) \\ \text{s.t. } h(x, y^k) &= 0 \\ g(x, y^k) &\leq 0 \\ x \in X \end{aligned} \quad (a2)$$

(a2)는 (i) feasible primal한 경우와 (ii) infeasible primal한 경우로 구분되며 각각에 대해서 다음과 같은 작업을 수행한다.

Case (i): Primal problem k iteration에서 feasible하면, 해는 $x^k, f(x^k, y^k)$ 에 대한 값을 주며 이것은 전체 문제에 upper bound에 해당된다. 그리고 equalities와 inequalities의 각각에 대한 optimal multiplier vector λ^k, μ^k 를 제공하게 된다. 결과적으로, 이러한 정보를 통해 Lagrange function를 다음과 같이 구성할 수 있다.

$$L(x, y, \lambda, \mu) = f(x, y) + \lambda^T h(x, y) + \mu^T g(x, y)$$

Case (ii): Solver에 의해 infeasible하다고 판별이 되면, Fletcher와 Leyffer[21]가 제시한 general feasible problem을 고려하게 되는데 다음과 같이 나타내어진다.

$$\begin{aligned} \text{min } \sum_i \omega_i g_i^+(x, y^k), \omega_i \geq 0, i \in I' \\ \text{s.t. } h(x, y^k) &= 0 \\ g(x, y^k) &\leq 0 \\ g_i^+(x, y^k) &= \max[0, g_i(x, y^k)], i \in I' \end{aligned} \quad (a2)$$

I 는 feasible constraint를 I' 은 infeasible constraint를 나타낸다.

위 system은 equalities and inequalities에 대한 Lagrange multipliers 값을 구하게 된다. 따라서 k 번째 iteration의 infeasible subproblem에 대한 Lagrange function은 다음과 같이 정의된다.

$$\bar{L}(x, y, \lambda^k, \mu^k) = \bar{\lambda}^{kT} h(x, y) + \bar{\mu}^{kT} g(x, y)$$

3. Master problem

Master problem은 nonlinear duality theory를 이용해서 유도되는데 다음과 같은 단계를 거치게 된다.

Step1. Projection of (a1) onto the y-space

(a1)은 다음과 같이 쓸 수 있을 것이다.

$$\begin{aligned} \min_y \quad & f(x, y) \\ \text{s.t.} \quad & h(x, y) = 0 \\ & g(x, y) \leq 0 \\ & x \in X, y \in Y \end{aligned} \quad (a1)$$

여기서 $v(y)$ 를 다음과 같이 정의하고

$$\begin{aligned} v(y) = \inf_x \quad & f(x, y) \\ \text{s.t.} \quad & h(x, y) = 0 \\ & g(x, y) \leq 0 \\ & x \in X \end{aligned} \quad (a3)$$

집합 V 를 다음과 같이 정의한다면

$$V = \{y : h(x, y) = 0, g(x, y) \leq 0 \text{ for some } x \in X\}$$

(a3)은 다음과 같이 표현된다.

$$\begin{aligned} \min \quad & v(y) \\ \text{s.t.} \quad & y \in Y \cap Y \end{aligned}$$

Step 2. Dual representation of V

V 의 dual representation은 Geoffrions theorem[19]에 의해 나타내어질 수 있다.

Theorem 1. 조건 C1 and C2를 만족하는 y 가 집합 V 에 속한다는 조건과 다음의 식은 필요충분조건이다:

$$\begin{aligned} 0 &\geq \inf \bar{L}(x, y, \bar{\lambda}, \bar{\mu}), \forall \bar{\lambda}, \bar{\mu} \in \Lambda \\ \Lambda &= \{\bar{\lambda} \in \mathbb{R}^m, \bar{\mu} \in \mathbb{R}^p : \bar{\mu} \geq 0, \sum_i \bar{\mu}_i = 1\} \end{aligned} \quad (a6)$$

Step 3. Dual representation of $v(y)$

$v(y)$ 의 dual representation도 역시 Geoffrions theorem[19]에 의해 나타내어질 수 있다.

Theorem 2.

(4)는 다음과 같이 나타내어질 수 있다:

$$\begin{aligned} \sup_{\lambda, \mu \geq 0} \quad & \inf_{x \in X} L(x, y, \lambda, \mu) \\ y &\in Y \cap V \end{aligned} \quad (a7)$$

(a6)과 위의 Theorem 2를 이용하면, 결국 (a1)은 다음과 같이 유도

된다.

$$\begin{aligned} \min_y \quad & \mu_\theta \\ \text{s.t.} \quad & \mu_\theta \geq \min_{x \in X} L(x, y, \lambda, \mu) \\ & 0 \geq \min_{x \in X} \bar{L}(x, y, \bar{\lambda}, \bar{\mu}), \forall \bar{\lambda}, \bar{\mu} \in \Lambda \end{aligned} \quad (a8)$$

여기서 μ_θ 를 benders cut이라고 부른다.

(a8)식이 유도된 master problem이다.

참고문헌

1. Moon, S., Bok, J.-K. and Park, S.: *HWAHAK KONGHAK*, **35**, 599 (1997).
2. Bok, J.-K., Lee, H. and Park, S.: *Computers Chem. Engng.*, **22**, 1037(1998).
3. Lee, H.: Ph. D. Dissertation, KAIST, Taejon, Korea(1997).
4. Reeve, A.: *Process Eng.*, **73**, 33(1992).
5. Birewar, D. B. and Grossmann, I. E.: *Computers Chem. Engng.*, **13**, 141(1989).
6. Moon, S., Park, S. and Lee, W. K.: *Ind. Eng. Chem. Res.*, **35**(10), 3458(1996).
7. Voudouris, V. T. and Grossmann, I. E.: *Ind. Eng. Chem. Res.*, **32**, 1962(1993).
8. Voudouris, V. T. and Grossmann, I. E.: *Computers Chem. Engng.*, **20**, 1335(1996).
9. Sahinidis, N. V. and Grossmann, I. E.: *Computers Chem. Engng.*, **13**, 1049(1989).
10. Berman, O., Ganz, Z. and Wagner, J. M.: *Naval Research Logistics*, **41**, 545(1994).
11. Paraskevopoulos, D., Karakitsos, E. and Rustem, B.: *Management Science*, **37**, 787(1991).
12. Ierapetritou, M. G. and Pistikopoulos, E. N.: *Ind. Eng. Chem. Res.*, **33**, 1930(1994).
13. Liu, M. L. and Sahinidis, N. V.: *Ind. Eng. Chem. Res.*, **35**, 4154 (1996).
14. Markowitz, H. M.: "Stochastic Optimization Models in Finance", Academic press, New York(1975).
15. Malcolm, S. A. and Zenios, S. A.: *J. Opl Res. Soc.*, **45**, 1040(1994).
16. Brooke, A., Kendrick, D. and Meeraus, A.: In GAMS: Users Manual, Scientific Press, Redwood City, CA(1992).
17. IBM: "OSL(Optimization Subroutine Library) Guide and reference", release 2. Kingston, NY(1991).
18. Drud, A.: *Mathematical Programming*, **31**, 153(1985).
19. Geoffrion, A. M.: *J. Optim. Theory and Appl.*, **10**(4), 237(1972).
20. Benders, J. F.: *Number. Math.*, **4**, 238(1962).
21. Fletcher, R. and Leyffer, S.: *Math., Progr.*, **66**, 327(1994).