

# 대규모 최적화 문제의 일반화된 교차 분할 알고리즘과 응용†

최경현 · 곽호만

한양대학교 산업공학과

## Generalized Cross Decomposition Algorithm for Large Scale Optimization Problems with Applications

Gyunghyun Choi · Ho-Mahn Kwak

In this paper, we propose a new convex combination weight rule for the cross decomposition method which is known to be one of the most reliable and promising strategies for the large scale optimization problems. It is called generalized cross decomposition, a modification of linear mean value cross decomposition for specially structured linear programming problems. This scheme puts more weights on the recent subproblem solutions other than the average. With this strategy, we are having more room for selecting convex combination weights depending on the problem structure and the convergence behavior, and then, we may choose a rule for either faster convergence for getting quick bounds or more accurate solution. Also, we can improve the slow end-tail behavior by using some combined rules. Also, we provide some computational test results that show the superiority of this strategy to the mean value cross decomposition in computational time and the quality of bounds.

### 1. 서론

분할 방법(Decomposition Method)은 대규모 최적화 문제를 위한 다수준 기법(Multi-Level Technique)이며, 문제의 특수한 구조를 이용한다.

대규모 최적화 문제들에서는 거의 항상 특수한 구조가 발견되는데, 이는 대개 시간이나 공간적으로 독립적인 하부단위들을 연결하는 것으로부터 기인한다. 이러한 특수한 구조를 이용하는 특수화된 알고리즘을 개발함으로써, 범용 알고리즘에 비해 계산효율과 필요로 하는 컴퓨터 메모리의 감소 측면에서 상당한 이득을 기대할 수 있다. 이러한 방법은 규모가 매우 큰 문제에 있어서는 필수적이며, 시간과 저장공간상의 제약으로 인해 다른 방법으로는 해를 구할 수 없는 경우가 많다.

분할 방법은 문제를 몇 개의 문제들로 나누어서 풀기 때문에, 각각의 문제들에 대해 지금까지 알려진 가장 좋은 해법을

적용할 수 있다. 따라서 가공되지 않은 단수준 방법(Single-Level Method)에 비해 유연하고 깔끔한 해법이 가능하다. 원래의 문제가 경제적인 성질의 것인 경우, 하위문제와 마스터 문제 사이의 정보의 흐름은 때로 유용한 경제적 해석을 제공하기도 한다.

지난 수십 년 간 많은 종류의 구조화된 문제들이 확인되었으며, 이들에 대한 수많은 알고리즘들이 개발되어왔다. 이전에도 상당한 노력들이 있어왔지만, 이러한 시기는 1960년에 Dantzig-Wolfe 분할 원리가 발표되면서 본격화되었다.

60년대 후반과 70년에 개발된 많은 분할 알고리즘들은 Dantzig-Wolfe (1960) 분할 알고리즘이나 Benders (1962) 분할 알고리즘의 특수한 경우나 그 변형으로 밝혀졌는데, 이들은 각각 문제의 쌍대 구조와 원시 구조를 이용한다. 따라서, Dantzig-Wolfe 분할 알고리즘과 Benders 분할 알고리즘을 각각 쌍대 분할 알고리즘과 원시 분할 알고리즘이라고 하기도 한다. 이들은 순수한 선형계획법 문제에 적용되었을 때, 쌍대 관계를 이

† 이 논문은 1997년 한양대학교 교내 연구비 지원에 의하여 연구되었음.

룬다는 것이 알려져 있다.

비교적 최근의 분해방법인 교차 분할 알고리즘은 Van Roy (1983)에 의해 발표되었는데, 이는 Benders 분할 알고리즘과 Dantzig-Wolfe 분할 알고리즘의 장점을 결합한다. 이 방법의 주된 아이디어는 원시 마스터 문제와 쌍대 하위문제, 그리고 쌍대 마스터 문제와 원시 하위문제 사이의 밀접한 관계를 이용하는 것이다.

가장 최근의 분해방법인 평균값 교차 분할 알고리즘은 Holmberg (1992)에 의해 발표되었다. 이는 Kornai-Liptak 알고리즘에서 영감을 얻었다. Kornai-Liptak 알고리즘은 우변 할당을 이용하며, Block-Angular 구조를 갖는 문제에 적용되는 Benders 분할 알고리즘의 단순화된 방법으로 분류된다.

교차 분할 알고리즘에서의 하위문제 단계는 일시적으로 매우 빠른 수렴을 제공할 수도 있지만, 전혀 수렴하지 않을 수도 있고, 이 경우에 마스터 문제를 이용해야만 한다. 한편, Kornai-Liptak 알고리즘은 매우 느린 점근적 수렴을 갖지만, 마스터 문제를 이용하지 않는다. Aardal and Ari (1990)는 Kornai-Liptak 알고리즘이 교차 분할 알고리즘의 수정으로 볼 수 있음을 보였다. 이들은 Kornai-Liptak 알고리즘에서 생성된 하위문제와, Block Angular 구조를 갖는 선형계획법 문제에서 교차 분할 알고리즘을 적용하여 얻은 하위문제 사이의 유사함을 보였다. 사실상, 이들간의 유일한 차이점은, 교차 분할 알고리즘에서는 마지막으로 얻은 해를 이용하는 반면에, Kornai-Liptak 알고리즘에서는 지금까지 구한 해의 가중화 된 정보를 이용하는 것이다. 따라서, 교차 분할 알고리즘에서의 원시와 쌍대 하위문제의 해에 같은 가중치를 주면, Kornai-Liptak 마스터 문제와 하위문제와 같아지며, 이는 수렴하는 일련의 해를 산출한다.

이 연구에서는 평균 대신에 최근에 구한 해에 더 많은 가중치를 주는 일반화된 볼록 조합 가중치 규칙(Convex-Combination Weight Rule)을 이용하는 것을 제시한다. 이는 평균값 교차 분할 알고리즘의 일반화이며, 따라서 일반화된 교차 분할 알고리즘이라고 하기로 한다. 최근의 해가 최적해에 더 가까울 것으로 기대할 수 있으므로, 이러한 전략은 합리적이며, 수렴속도를 가속화시킬 것이 기대된다. 이와 같은 방법의 또 다른 장점은, 문제의 구조와 수렴 행동에 따라서 볼록 조합 가중치를 선택할 더 많은 여지를 갖게 된다는 것이다. 그럼으로써, 더 빨리 상한이나 하한을 구하기 위한 빠른 수렴을 위한 규칙이나, 더 정확한 해를 구하기 위한 규칙을 선택할 수 있다. 또한, 복잡한 규칙을 이용함으로써 느린 End-Tail 현상을 개선시킬 수도 있다.

이 논문의 구성은 다음과 같다. 2장에서는 대규모 최적화 문제의 모형과 기존의 분할 기법 이론에 대하여 서술하고, 3장에서는 이 연구에서 제안하는 일반화된 교차 분할 알고리즘을 설명하고, 그 수렴성에 대하여 논하였다. 4장에서는 이를 생산-구매 결정문제에 적용하여 평균값 교차 분할 알고리즘의 결과와 비교한 수치실험의 결과를 제시한다.

## 2. 분할기법 이론

### 2.1 대규모 최적화 문제 모형

다음의 대규모 선형계획법 문제를 고려하자.

$$(P) \ z^* = \min \ c'x + d'y$$

$$\text{s.t.} \ A_1x + B_1y \leq b_1 \quad (1)$$

$$A_2x \leq b_2 \quad (2)$$

$$x \geq 0 \quad (3)$$

$$y \in Y = \{y : B_2y \leq b_3\} \quad (4)$$

제약식 (1)과 (2)에 대응되는 쌍대 변수를 각각  $u$ 와  $v$ 로 나타내고, 원시 최적해를  $(x^*, y^*)$ 로 나타내고, 쌍대 최적해를  $(u^*, v^*)$ 로 나타내자. 제약식 (2)와 (3)은 유계인 가능영역을 형성한다고 가정하고,  $Y$ 는 유한 다면체(Polytope)이며, 유한개의 극점  $y_l, l \in P_Y$ 를 가진다고 가정한다. 만약  $Y$ 가 유한이 아닌 경우, 충분히 큰 값의 상한을 이용한다. 이때,  $P_Y$ 는  $Y$ 의 극점들의 색인 집합(Index Set)이다. 또한, 각  $y \in Y$ 에 대하여, 제약식 (2), (3), (4)를 만족하는  $x$ 가 존재한다고 가정한다. 일반적으로 인공변수(Artificial Variable)를 이용하여 이 가정을 만족시킬 수 있게 만들 수 있으며, 이 경우 충분히 큰 값의 상한을 이용하여 유한 다면체의 가정도 동시에 만족하게 할 수 있다.

그리고,  $y$ 는 고정되었을 경우 나머지 문제가 풀기 쉬워지는 변수라 가정하고, 제약식 (1)은 Lagrangian 이완 방법 하에서 이완시키면 풀기 쉬워지는 어려운 제약식이라고 가정한다.

문제를 다음과 같은 방법으로  $x$ 와  $y$ 로 분리한다.  $y$ 를 임의의  $\bar{y} \in Y$ 에 고정시켰을 때, 다음의 원시 하위문제(PS : Primal Subproblem)를 얻는다.

$$(PS) \ h(\bar{y}) = \min \ c'x + d'\bar{y}$$

$$\text{s.t.} \ A_1x \leq b_1 - B_1\bar{y}$$

$$A_2x \leq b_2$$

$$x \geq 0$$

모든  $\bar{y} \in Y$ 에 대하여  $h(\bar{y})$ 는 원시문제 P의 임의의 가능해에 대한 목적함수값이므로  $h(\bar{y}) \geq z^*$ 이다. 주어진 임의의  $\bar{y} \in Y$ 에 의하여  $x$ 의 가능해 집합을

$$X(\bar{y}) = \{x : A_1x \leq b_1 - B_1\bar{y}, A_2x \leq b_2, x \geq 0\}$$

으로 나타내면, 가정에 의해, (PS)는 어떤  $\bar{y} \in Y$ 에 대해서도 가능해를 갖는다. 즉,  $\bar{y} \in Y$ 에 대해서  $X(\bar{y}) \neq \emptyset$ 이다. 원시 하위문제 PS의 선형계획법 쌍대문제(Linear Programming Dual Problem)는 다음과 같이 주어지며, 이를 원시 하위쌍대(PSd : Primal Subproblem Dual)문제라 하자.

(PSd)

$$h(\bar{y}) = \max d^t \bar{y} + (B_1 \bar{y} - b_1)^t u - b_2^t v$$

$$\text{s.t. } -A_1^t u - A_2^t v \leq c$$

$$u, v \geq 0$$

이때  $u$  와  $v$  는 (PS)의 각각의 제약식에 해당되는 쌍대 변수이다.  $U$ 가 (PSd)의 가능영역(Feasible Region)이라고 하면,

$$U = \{(u, v) : -A_1^t u - A_2^t v \leq c, u \geq 0, v \geq 0\}$$

이고,  $U$ 는  $y$ 에 독립적이다. 또한,  $U$ 는 유계인 다면체(Polytope)이며 유한 개의 극점( $u_k, v_k$ )를 갖는다. 따라서  $U$ 의 모든 원소는 이들 극점들의 볼록 조합으로 나타낼 수 있다.  $P_U$ 를  $U$ 의 극점들의 색인 집합이라고 하면,

$$U = \{(u, v) : u = \sum_{k \in P_U} \lambda_k u_k, v = \sum_{k \in P_U} \lambda_k v_k,$$

$$\sum_{k \in P_U} \lambda_k = 1, \lambda_k \geq 0 \forall k \in P_U\}$$

으로 표현될 수 있다.

이를 이용하여 원시문제 P의 Benders 마스터 문제를 정의하면,

$$z^* = \min q_1$$

$$\text{s.t. } q_1 \geq d^t y + (B_1 y - b_1)^t u_k - b_2^t v_k, \forall k \in P_U$$

$$y \in Y$$

으로 나타난다. Benders의 마스터 문제는  $U$ 의 모든 극점들이 제약식에 포함되어 있기 때문에 제약식의 수가  $P_U$ 의 원소의 개수만큼 많으며, 이를 한꺼번에 고려하는 해법은 실용성이 없다. 따라서 극점의 일부 즉, 색인집합  $P_U$ 의 부분집합  $P_U'$ 을 이용하여 제한(축소)된 Benders 마스터 문제를 정의하며, 이를 원시 마스터(Primal Master) 문제라고 부른다.

(PM)  $z_{PM} = \min q_1$

$$\text{s.t. } q_1 \geq d^t y + (B_1 y - b_1)^t u_k - b_2^t v_k \quad \forall k \in P_U'$$

$$y \in Y$$

여기서  $P_U' \subseteq P_U$ 이다.

$z_{PM}$ 은 원문제의 이완문제의 최적해이므로,  $z_{PM} \leq z^*$ 가 된다.

유사한 방법으로 원시문제 P의 쌍대문제의 마스터 문제를 정의할 수 있다. 먼저 원시문제 P의 첫번째 제약식 (1)을 Lagrangian 쌍대변수  $u$ 를 이용하여 쌍대 문제를 정의하면,

(D)  $z^* = \max g(u)$

$$\text{s.t. } u \geq 0$$

으로 표현되며, 여기서 임의의 주어진  $\bar{u} \geq 0$ 에 대하여, 쌍대 하위문제는 다음과 같이 주어진다.

(DS)  $g(\bar{u}) = \min c^t x + d^t y + \bar{u}^t (A_1 x + B_1 y - b_1)$

$$\text{s.t. } A_2 x \leq b_2$$

$$x \geq 0, y \in Y = \{y : B_2 y \leq b_3\}$$

원시문제 P의 제약식 (2)와 (4)가 서로 독립적이므로, 쌍대 하위문제(DS)에서도 변수  $x, y$ 에 대하여 분리 가능(Separable)하다. 모든 쌍대 가능해  $\bar{u} \geq 0$ 에 대하여,  $g(\bar{u}) \leq z^*$ 이다. (DS)의 가능해의 집합을  $X = \{(x, y) : A_2 x \leq b_2, x \geq 0, y \in Y\}$ 라 하면, 이것은  $\bar{u}$ 에 독립적인 유한 다면체이고, 따라서 유한개의 극점( $x_k, y_k$ ),  $k \in P_X$ 를 갖는다. 따라서,  $X$ 를 다음과 같이 쓸 수 있다.

$$X = \{(x, y) : x = \sum_{k \in P_X} \lambda_k x_k, y = \sum_{k \in P_X} \lambda_k y_k,$$

$$\sum_{k \in P_X} \lambda_k = 1, \lambda_k \geq 0 \forall k \in P_X\}$$

따라서, 모든  $\bar{u} \geq 0$ 에 대해서,  $g(\bar{u})$ 를 다음과 같이 표현할 수 있다.

$$g(\bar{u}) = \min_{(x, y) \in X} c^t x + d^t y + \bar{u}^t (A_1 x + B_1 y - b_1)$$

$$= \min_{k \in P_X} c^t x_k + d^t y_k + \bar{u}^t (A_1 x_k + B_1 y_k - b_1) \quad (5)$$

앞 절에서와 같이 (5)를 이용하여 쌍대 문제 (D)를, Dantzig-Wolfe의 쌍대 마스터(Dual Master) 문제를 다음과 같이 정의한다.

$z^* = \max q_2$

$$\text{s.t. } q_2 \leq c^t x_k + d^t y_k$$

$$+ \bar{u}^t (A_1 x_k + B_1 y_k - b_1) \quad \forall k \in P_X$$

$$u \geq 0$$

또한  $P_X$ 의 부분집합  $P_X'$ 을 이용하여 제한(축소)된 쌍대 마스터 문제 (DM)을 다음과 같이 정의한다.

(DM)  $z_{DM} = \max q_2$

$$\text{s.t. } q_2 \leq c^t x_k + d^t y_k$$

$$+ \bar{u}^t (A_1 x_k + B_1 y_k - b_1) \quad \forall k \in P_X'$$

$$u \geq 0$$

여기서  $P_X' \subseteq P_X$ 이다.

(DM)은 쌍대 문제 (D)의 이완된 문제이므로, 분명히  $z_{DM} \geq z^*$ 이다.

## 2.2 Dantzig-Wolfe와 Benders의 분할 알고리즘

Dantzig-Wolfe 분할 알고리즘은 쌍대 하위문제(DS)와 Dantzig-Wolfe 마스터 문제(DM)을 반복하여 푼다. 마스터 문제는  $\bar{u}$

를 산출하는데, 이것은 하위문제에서 이용된다. 하위문제를 풀 때마다  $X$ 의 새로운 극점을 얻게 되는데, 이는 마스터 문제에 포함시킬 절단(Cut)을 생성시키는데 이용된다. 따라서 매 반복마다 절단의 수는 증가하게 된다. 하위문제에서 생성되는 절단은  $\bar{u}$ 에서 최소인 절단, 즉 가장 위반되는 절단이다. 이러한 방식으로 필요한 절단을 생성시키면서, 알고리즘은 최악의 경우 모든 절단들을 생성시킨 후에 유한 번만에 끝나게 되며, 이때 완전(exact) 최적해를 얻게 된다.

모든  $\bar{u} \geq 0$ 에 대하여  $g(\bar{u}) \leq z^*$ 이고, 모든  $P_X' \subseteq P_X$ 에 대하여  $z_{DM} \geq z^*$ 이므로, 매 반복마다 우리는  $z^*$ 의 하한과 상한을 얻게 된다. 마스터 문제에 계속해서 절단을 추가해 나가므로,  $z_{DM}$ 은 매 반복마다 개선된다. 필요한 모든 절단들을 알게 되면  $z_{DM} = z^*$ 가 되며, 마스터 문제는 해  $u = u^*$ 를 산출하고, 이때  $g(\bar{u}) = z^*$ 가 되어  $z^*$ 에 대한 상한과 하한이 같아지게 된다.

Benders 분할 알고리즘은 Benders 하위문제(PSd) (또는 (PS))와 Benders 마스터 문제 (PM)을 반복하여 푼다. 마스터 문제는  $\bar{y}$ 를 산출하는데, 이것은 하위문제에서 이용된다. 하위문제를 풀 때마다  $U$ 의 새로운 극점을 얻게 되는데, 이는 마스터 문제에 포함시킬 Benders 절단을 생성시키는데 이용된다. 따라서 매 반복마다 절단의 수는 증가하게 된다. 하위문제에서 생성되는 절단은  $\bar{y}$ 에서 최대인 절단, 즉 가장 위반되는 절단이다. 이러한 방식으로 필요한 절단을 생성시키면서, 알고리즘은 최악의 경우 모든 절단들을 생성시킨 후에 유한 번만에 끝나게 되며, 이때 완전 최적해를 얻게 된다.

모든  $\bar{y} \in Y$ 에 대하여  $h(\bar{y}) \geq z^*$ 이고, 모든  $P_U' \subseteq P_U$ 에 대하여  $z_{PM} \leq z^*$ 이므로, 매 반복마다 우리는  $z^*$ 의 하한과 상한을 얻게 된다. 마스터 문제에 계속해서 절단을 추가해 나가므로,  $z_{PM}$ 은 매 반복마다 개선된다. 필요한 모든 절단들을 알게 되면  $z_{PM} = z^*$ 가 되며, 마스터 문제는 해  $y = y^*$ 를 산출하고, 이때  $h(\bar{y}) = z^*$ 가 되어  $z^*$ 에 대한 상한과 하한이 같아지게 된다.

### 2.3 교차 분할 알고리즘과 평균값 교차분할 알고리즘

교차 분할 알고리즘(Cross Decomposition Algorithm)은 주로 하위문제 단계로 이루어져 있으며, 이는 원시 하위문제(PSd)와 쌍대 하위문제(DS) 사이를 반복한다. (DS)는 (PSd)에 입력할  $\bar{y}$ 와,  $z^*$ 에 대한 하한을 생성시킨다. (PSd)는 (DS)에 입력할  $\bar{u}$ 와,  $z^*$ 에 대한 상한을 생성시킨다. 하위문제들은 또한 원시와 쌍대 마스터 문제, (PM), (DM)에 대한 절단들을 생성시킨다. 매 반복에서 우리는 또한  $\bar{y}$ 에 대한 원시 수렴 테스트와  $\bar{u}$ 에 대한 쌍대 수렴 테스트를 이용하여, 이들 중 하나가 실패하면, 마스터 문제 (PM) 또는 (DM)을 풀도록 한다. 그 후에 다시 하위문제 단계로 돌아오게 된다.

평균값 교차 분할 알고리즘(MVCD : Mean Value Cross Decom-

position Algorithm)은 다음과 같이 원시 하위문제 (PSd)와 쌍대 하위문제 (DS) 사이를 반복한다.

처음에는 초기 쌍대해  $u_1 \geq 0$ 을 선택하고, 반복 횟수  $k$ 를 1로 둔다.

반복  $k$ 에서는 (DS)에 대한 입력을 다음과 같이 구한다.

$$\bar{u}_k = \sum_{i=1}^k \frac{1}{k} u_i = \frac{1}{k} u_k + \frac{k-1}{k} \bar{u}_{k-1} \quad (6)$$

여기서  $u_i$ 은 반복  $i$ 에서의 (PSd)의 해의  $u$  부분이다. 그러므로,  $\bar{u}_k$ 는 이전에 얻은 모든 쌍대해  $u$ 들의 평균이다. 분명히  $\bar{u}_k \geq 0$ 이다.

(DS)를 풀어서 이때의 최적 목적식값  $g(\bar{u}_k)$ 와 최적해  $y = y_k$ 를 구한다. 이때,  $g(\bar{u}_k)$ 는 원문제의 하한이 된다.

반복  $k$ 에서의 (PSd)의 입력은 다음과 같이 구한다.

$$\bar{y}_k = \sum_{i=1}^k \frac{1}{k} y_i = \frac{1}{k} y_k + \frac{k-1}{k} \bar{y}_{k-1} \quad (7)$$

$y_i$ 은 반복  $i$ 에서의 (DS)의 해의  $y$  부분이다. 그러므로,  $\bar{y}_k$ 는 전에 구한 모든 원시해  $y$ 들의 평균이다.  $Y$ 가 볼록 집합이므로,  $\bar{y}_k \in Y$ 이다.

(PSd)를 풀어서 이때의 최적 목적식값  $h(\bar{y}_k)$ 와 최적해  $u = u_k$ 를 구한다. 이때,  $h(\bar{y}_k)$ 는 원문제의 상한이 된다.

$z^*$ 에 대한 상한값과 하한값이 충분히 가까워질 때까지 (DS)와 (PSd)를 푸는 과정을 반복한다. 이 알고리즘은 점근적인 수렴성을 갖는다고 알려져 있다.

## 3. 일반화된 교차 분할 알고리즘

### 3.1 일반화된 교차 분할 알고리즘

일반화된 교차분할 알고리즘(GCD : Generalized Cross Decomposition)의 하위문제들과 알고리즘 과정은, 평균값 교차 분할 알고리즘(MVCD)과 동일하다. 차이점은, 하위문제들에 대한 입력으로 이전에 얻은 모든 해들의 평균 대신에 좀 더 일반화된 볼록 조합을 이용한다는 것이다. 즉, 반복  $k$ 에서는, (6)과 (7) 대신에 (8)과 (9)를 하위문제들의 입력으로 이용한다.

$$\bar{u}_k = \sum_{l=1}^k \lambda_{kl} u_l, \quad \sum_{l=1}^k \lambda_{kl} = 1, \quad \lambda_{kl} \geq 0 \quad \forall l \quad (8)$$

$$\bar{y}_k = \sum_{l=1}^k \lambda_{kl} y_l, \quad \sum_{l=1}^k \lambda_{kl} = 1, \quad \lambda_{kl} \geq 0 \quad \forall l \quad (9)$$

따라서, 평균값 교차 분할 알고리즘은 일반화된 교차 분할 알고리즘의 특수한 경우가 된다.

일반화된 교차 분할 알고리즘의 단계들을 요약하면 다음과 같다.

[일반화된 교차 분할 알고리즘의 단계]

[초기화 단계]

적당한  $\varepsilon > 0$ 과 초기 쌍대해  $u_l \geq 0$ 를 선택한다.  
반복횟수를  $k=1$ 로 둔다.

[단계 1] 반복  $k$ 에서, 다음을 계산한다.

$$\bar{u}_k = \sum_{l=1}^k \mu_{kl} u_l,$$

이때,  $\sum_{l=1}^k \mu_{kl} = 1, \mu_{kl} \geq 0 \quad \forall l=1, \dots, k$

DS( $\bar{u}_k$ )를 풀어서 최적 목적식값  $g(\bar{u}_k)$ 를 구하고, 이때의 최적해를  $y = y_k$ 라 하자.  $g(\bar{u}_k)$ 는 원문제의 최적 목적식값의 하한이다.

[단계 2] 다음을 계산한다.

$$\bar{y}_k = \sum_{l=1}^k \mu_{kl} y_l,$$

이때,  $\sum_{l=1}^k \mu_{kl} = 1, \mu_{kl} \geq 0 \quad \forall l=1, \dots, k$

PS( $\bar{y}_k$ )을 풀어서 최적 목적식값  $h(\bar{y}_k)$ 를 구하고, 이때의 최적해를  $u = u_k$ 라 하자.  $h(\bar{y}_k)$ 는 원문제의 최적 목적식값의 상한이다.

[단계 3] 다음의 수렴 테스트를 수행한다.

$$\left| \min_k h(\bar{y}_k) - \max_k g(\bar{u}_k) \right| \leq \varepsilon$$

만족되면 현재의 해가 최적해이고, 알고리즘을 종료한다.  
만일 실패하면, 반복횟수  $k$ 를 1 증가시키고, 단계 1로 가서 반복한다.

### 3.2 일반화된 교차 분할 알고리즘의 수렴성 정리

이 절에서는 일반화된 교차 분할 알고리즘의 수렴성 이론을 제시한다. 이를 위해 PS의 목적식값의 극한값  $\bar{h}$ 와 DS의 목적식값의 극한값  $\bar{g}$ 가 같은 점으로 수렴함을 보일 것이다. 그러면 이 알고리즘은 점근적인 수렴성을 가진다는 것이 증명된다. 증명에 이용되는 정리들과 그 증명은 다음과 같다.

[정리 1]

$y_l$ 이 유계영역 내에 존재하고,  $\sum_{l=1}^k \lambda_{kl} = 1$ 이고, 모든  $l=1, \dots, k$ 에 대하여  $\lambda_{kl} \geq 0$ 일 때,  $\bar{y}_k$ 는  $\bar{y}_k = \sum_{l=1}^k \lambda_{kl} y_l$ 으로 정의되고,  $\lim_{k \rightarrow \infty} \lambda_{kk} = 0$ 일 때,  $\{\bar{y}_k\}$ 는 어떤 점  $y^*$ 으로 수렴하는

수열이다.

[증명]

$$\begin{aligned} & \| \bar{y}_k - \bar{y}_{k-1} \| \\ &= \left\| \sum_{l=1}^k \lambda_{kl} y_l + \sum_{l=1}^{k-1} \lambda_{k-1,l} y_l \right\| \\ &= \left\| \sum_{l=1}^{k-1} (\lambda_{kl} - \lambda_{k-1,l}) y_l + \lambda_{kk} y_k \right\| \\ &\leq \left\| \sum_{l=1}^{k-1} (\lambda_{kl} - \lambda_{k-1,l}) y_l \right\| + \lambda_{kk} \| y_k \| \end{aligned}$$

$M$ 을 다음과 같이 정의하자.

$$M = \text{maximum of } \| y_l \|, \quad l=1, 2, \dots, k.$$

그러면, 다음과 같은 부등식을 얻는다.

$$\| \bar{y}_k - \bar{y}_{k-1} \| \leq \sum_{l=1}^{k-1} |\lambda_{kl} - \lambda_{k-1,l}| M + \lambda_{kk} M$$

이때,

$$\begin{aligned} & \sum_{l=1}^{k-1} |\lambda_{kl} - \lambda_{k-1,l}| \\ &= |(\lambda_{k1} + \dots + \lambda_{kk-1}) - (\lambda_{k-1,1} + \dots + \lambda_{k-1,k-1})| \\ &= |(1 - \lambda_{kk}) - 1| = |-\lambda_{kk}| = \lambda_{kk} \end{aligned}$$

이므로,

$$\| \bar{y}_k - \bar{y}_{k-1} \| \leq M \lambda_{kk} + M \lambda_{kk} = 2M \lambda_{kk}$$

이다.

$y_k$ 가 유계 영역 내의 점이므로,  $M$ 은 유한한 수이다. 따라서,  $\lambda_{kk} \rightarrow 0$ 이면, 충분히 작은  $\varepsilon > 0$ 에 대하여,

$$\| \bar{y}_k - \bar{y}_{k-1} \| \leq 2M \lambda_{kk} \leq \varepsilon \text{ 이다.}$$

그러므로,  $\lim_{k \rightarrow \infty} \lambda_{kk} = 0$ 이면, 점  $\bar{y}_k$ 의 수열은 어떤 점  $y^*$ 로 수렴한다. □

[정리 2]

임의의 실수열  $\{z_k\}$ 가  $\lim_{k \rightarrow \infty} z_k = a$ 이고, 블록 조합 가중치  $\mu_{kl} \geq 0, l=1, \dots, k$ 가

$$\text{maximum}_{l=1, \dots, k} \{ \mu_{kl} \} = \mu_{k \max} \leq \frac{\alpha}{k}, \quad \alpha \text{는 상수}$$

인 성질을 만족할 때,

$$\lim_{k \rightarrow \infty} \sum_{l=1}^k \mu_{kl} z_l = a$$

이다.

[증명]

$$\begin{aligned} \sum_{l=1}^k \mu_{kl} z_l - a &= \sum_{l=1}^k \mu_{kl} (z_l - a) \\ &\leq \sum_{l=1}^k |\mu_{kl} (z_l - a)| \\ &= \sum_{l=1}^k \mu_{kl} |z_l - a| \\ &\leq \sum_{l=1}^k \mu_{k \max} |z_l - a| \\ &= \mu_{k \max} \sum_{l=1}^k |z_l - a| \end{aligned} \tag{10}$$

그런데,  $\mu_{k \max} \leq \frac{\alpha}{k}$  이므로,

$$\mu_{k \max} \sum_{l=1}^k |z_l - a| \leq \frac{\alpha}{k} \sum_{l=1}^k |z_l - a| \tag{11}$$

이다.

그런데,  $z_k$ 가  $a$ 로 수렴하므로, 임의의  $\epsilon > 0$ 에 대하여  $K_1$ 이 존재하여,  $k \geq K_1$ 인 모든  $k$ 에 대해  $|z_k - a| \leq \frac{\epsilon}{2\alpha}$ 이 성립된다. 또한,  $\frac{1}{K_2} \sum_{l=1}^{K_1} |z_l - a| \leq \frac{\epsilon}{2\alpha}$ 를 만족하는  $K_2$ 가 존재한다.

$K = \max \{ K_1, K_2 \}$ 라 하면,  $k \geq K$ 인 모든  $k$ 에 대하여 다음이 성립한다.

$$\begin{aligned} &\frac{1}{k} \sum_{l=1}^k |z_l - a| \\ &= \frac{1}{k} \left\{ \sum_{l=1}^{K_1} |z_l - a| + \sum_{l=K_1+1}^k |z_l - a| \right\} \\ &\leq \frac{K_2}{k} \frac{\epsilon}{2\alpha} + \frac{k - K_1}{k} \frac{\epsilon}{2\alpha} \leq \frac{\epsilon}{2\alpha} + \frac{\epsilon}{2\alpha} = \frac{\epsilon}{\alpha} \end{aligned}$$

따라서, 모든  $\epsilon > 0$ 에 대하여,  $k \geq K$ 인 모든  $k$ 에 대해

$$\frac{\alpha}{k} \sum_{l=1}^k |z_l - a| \leq \epsilon \tag{12}$$

이다. 그러므로, 식 (10), (11), (12)에 의하여, 모든  $\epsilon > 0$ 에 대하여,  $k \geq K$ 인 모든  $k$ 에 대해

$$\sum_{l=1}^k \mu_{kl} z_l - a < \epsilon$$

이다. □

정리 2를 만족하는 볼록 조합 가중치는 정리 1에서의 조건인  $\lim_{k \rightarrow \infty} \mu_{kk} = 0$ 도 만족하게 된다.

[정리 3]

볼록 조합 가중치가

$$\text{maximum}_{l=1, \dots, k} \{ \mu_{kl} \} = \mu_{k \max} \leq \frac{\alpha}{k}, \quad \alpha \text{는 상수}$$

인 성질을 만족할 때, 일반화된 교차분할 알고리즘에서 발생하는 수열  $\{g(\bar{u}_k)\}$ 와  $\{h(\bar{y}_k)\}$ 는 수렴하는 수열이고 그 수렴 점은 같다. 즉,

$$\lim_{k \rightarrow \infty} g(\bar{u}_k) = \bar{g} = \bar{h} = \lim_{k \rightarrow \infty} h(\bar{y}_k)$$

이다.

[증명]

정리 1과  $g, h$ 의 연속성에 의하여, 원시 하위문제와 쌍대 하위문제의 목적식값의 수열은 각각

$$\lim_{k \rightarrow \infty} h(\bar{y}_k) = h(y^*) = \bar{h}$$

$$\lim_{k \rightarrow \infty} g(\bar{u}_k) = g(u^*) = \bar{g}$$

이다.

$u_k$ 는 PS  $(\bar{y}_k)$ 의 최적해의  $u$ 부분일 때,  $h^*(\bar{y}_k, u_k) = h(\bar{y}_k)$ 으로 정의하면, 각각의  $l = 1, \dots, k$ 에 대하여 양변에  $\mu_{kl}$ 을 곱하여 합하면,

$$\sum_{l=1}^k \mu_{kl} h(\bar{y}_l) = \sum_{l=1}^k \mu_{kl} h^*(\bar{y}_l, u_l)$$

이 되며, 정리 2에 의하여

$$\begin{aligned} \lim_{k \rightarrow \infty} \sum_{l=1}^k \mu_{kl} h(\bar{y}_l) &= \bar{h} \\ &= \lim_{k \rightarrow \infty} \sum_{l=1}^k \mu_{kl} h^*(\bar{y}_l, u_l) \end{aligned}$$

이 된다. 또한,

$$\lim_{k \rightarrow \infty} (h^*(\bar{y}_k, u_k) - h^*(y^*, u_k)) = 0$$

이며, 정리 2에 의해,

$$\lim_{k \rightarrow \infty} \sum_{l=1}^k \mu_{kl} (h^*(\bar{y}_l, u_l) - h^*(y^*, u_l)) = 0$$

이고,

$$\begin{aligned} \lim_{k \rightarrow \infty} \sum_{l=1}^k \mu_{kl} h^*(\bar{y}_l, u_l) \\ = \lim_{k \rightarrow \infty} \sum_{l=1}^k \mu_{kl} h^*(y^*, u_l) \end{aligned}$$

이다. 따라서,

$$\begin{aligned} \bar{h} &= \lim_{k \rightarrow \infty} \sum_{l=1}^k \mu_{kl} h^*(\bar{y}_l, u_l) \\ &= \lim_{k \rightarrow \infty} \sum_{l=1}^k \mu_{kl} h^*(y^*, u_l) \end{aligned}$$

이다.

그런데,  $h^*(\bar{y}_k, u_k)$ 는  $u_k$ 의 오목 함수(Concave Function)이므로,

$$\begin{aligned} \bar{h} &= \lim_{k \rightarrow \infty} \sum_{l=1}^k \mu_{kl} h^*(y^*, u_l) \\ &\leq \lim_{k \rightarrow \infty} h^*(y^*, \sum_{l=1}^k \mu_{kl} u_l) \\ &= \lim_{k \rightarrow \infty} h^*(y^*, \bar{u}_k) = h^*(y^*, u^*) \end{aligned} \quad (13)$$

DS에 대해서도 유사하게,

$y_k$ 는  $DS(\bar{u}_k)$ 의 최적해의  $y$ 부분일 때,  $g^*(y_k, \bar{u}_k) = g(\bar{u}_k)$ 으로 정의하면, 각각의  $k$ 에 대하여  $g(\bar{u}_k) = g^*(y_k, \bar{u}_k)$ 으로 볼 수 있으며,

$$\sum_{l=1}^k \lambda_{kl} g(\bar{u}_l) = \sum_{l=1}^k \lambda_{kl} g^*(y_l, \bar{u}_l)$$

이 되며, 정리 2에 의하여

$$\begin{aligned} \lim_{k \rightarrow \infty} \sum_{l=1}^k \lambda_{kl} g(\bar{u}_l) &= \bar{g} \\ &= \lim_{k \rightarrow \infty} \sum_{l=1}^k \lambda_{kl} g^*(y_l, \bar{u}_l) \end{aligned}$$

이 된다.

여기서도 PS에서와 같이 정리 2를 이용하면,

$$\begin{aligned} \bar{g} &= \lim_{k \rightarrow \infty} \sum_{l=1}^k \lambda_{kl} g^*(y_l, \bar{u}_l) \\ &= \lim_{k \rightarrow \infty} \sum_{l=1}^k \lambda_{kl} g^*(y_l, u^*) \end{aligned}$$

이다.

그런데,  $g^*(y_k, \bar{u}_k)$ 는  $y_k$ 의 선형함수이므로,

$$\begin{aligned} \bar{g} &= \lim_{k \rightarrow \infty} \sum_{l=1}^k \lambda_{kl} g^*(y_l, u^*) \\ &= \lim_{k \rightarrow \infty} g^*(\sum_{l=1}^k \lambda_{kl} y_l, u^*) \\ &= \lim_{k \rightarrow \infty} g^*(\bar{y}_k, u^*) = g^*(y^*, u^*) \end{aligned} \quad (14)$$

여기서,

$$\begin{aligned} h(y^*, u^*) &= \left[ \begin{array}{l} \max -b_2^t v + \{d^t y^* + (B_1 y^* - b_1)^t u^*\} \\ \text{s.t. } -A_2^t v \leq c + A_1^t u^* \\ v \geq 0 \end{array} \right] \\ g(y^*, u^*) &= \left[ \begin{array}{l} \min (c^t + A_1^t u^*)x + d^t y^* + (B_1 y^* - b_1)^t u^* \\ \text{s.t. } A_2 x \leq b_2 \\ x \geq 0 \end{array} \right] \end{aligned}$$

$h^*(y^*, u^*)$ 와  $g^*(y^*, u^*)$ 는 쌍대관계에 있는 선형계획법 문제로서, 쌍대 차(Duality Gap)가 없으므로, 동일한 목적식값을 가진다. 따라서,

$$h^*(y^*, u^*) = g^*(y^*, u^*) \quad (15)$$

가 된다.

(13), (14), (15)에 의해,

$$\bar{h} \leq h^*(y^*, u^*) = g^*(y^*, u^*) = \bar{g}$$

그런데,  $\bar{h}$ 는 원문제의 상한인 원시 한계의 극한이고,  $\bar{g}$ 는 원문제의 하한인 쌍대 한계의 극한이므로, 원문제의 최적 목적함수값  $z^*$ 에 대하여,

$$\bar{h} \geq z^* \geq \bar{g}$$

이다. 따라서,  $\bar{h} = \bar{g} = z^*$ 이다. □

위의 정리에 의해, 반복  $k$ 에서의 블록 조합 가중치가

$$\text{maximum}_{l=1, \dots, k} \{\mu_{kl}\} = \mu_{k \max} \leq \frac{\alpha}{k}, \quad \alpha \text{는 상수}$$

인 성질을 만족할 때, 일반화된 교차 분할 알고리즘은 점근적인 수렴성을 갖는다.

#### 4. 수치예제 및 실험

이 절에서는 일반화된 교차 분할 알고리즘(GCD)과 평균값 교차 분할 알고리즘(MVCD)을 비교하기 위해, 고도로 구조화된 선형계획법 문제인 생산-구매 결정문제(PPP: Produce or Purchase Problem)에 두 알고리즘을 적용한 실험결과를 제시한다.

##### 4.1 생산-구매 결정문제

생산-구매 결정문제는  $n$ 개의 공장에서  $m$ 가지의 제품을 고정된 비율로 생산하는 상황을 나타낸다. 각 공장에서는 또한 서로 다른 제품을 구매할 수도 있다. 각 공장  $j$ 는 반드시 각 제품  $i$ 의 주어진 수요  $d_{ij}$ 를 만족시켜야만 하며, 다른 제품의 여분은 처리할 수 있다.

공장  $j$ 에서의 생산 프로세스의 가동시간은  $y_j$ (시간/일)로 나타낸다. 공장  $j$ 에서 생산 프로세스를 하루에 한 시간 가동할 때 제품  $i$ 의 생산량은  $e_{ij}$ 이고, 비용은  $f_j$ 이다. 시장에서 구입 가능한 제품  $i$ 의 양은  $b_i$ 로 나타낸다. 제품  $i$  한 단위를 구입해서 공장  $j$ 로 운반하는 비용은  $c_{ij}$ 이다. 변수  $x_{ij}$ 는 공장  $j$ 로 운반되는 구입한 제품  $i$ 의 양이다. 마지막으로, 제품  $i$ 를 구입해서 공장  $j$ 로 운반할 수 있는 양의 상한은  $s_{ij}$ 로 나타낸다. 공장의 생산능

력은 수요를 충족시킬 수 있을 만큼 충분하며, 시장에도 수요를 충족시킬 수 있을 만큼 충분한 제품이 있다고 가정한다. 목적함수는 총비용을 최소화하는 것이다. 생산-구매 결정문제의 수리적 모형은 다음과 같이 주어진다.

(PPP)

$$\begin{aligned} \min \quad & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n f_j y_j \\ \text{s.t.} \quad & x_{ij} + e_{ij} y_j \geq d_{ij} \quad i=1, \dots, m, j=1, \dots, n \quad (16) \\ & \sum_{j=1}^n x_{ij} \leq b_i \quad i=1, \dots, m \quad (17) \\ & x_{ij} \leq s_{ij} \quad i=1, \dots, m, j=1, \dots, n \quad (18) \\ & x_{ij} \geq 0 \quad i=1, \dots, m, j=1, \dots, n \\ & 0 \leq y_j \leq 24 \quad j=1, \dots, n \end{aligned}$$

여기서, 제약식 (16), (17), (18)에 대응되는 쌍대 변수를 각각  $u, v, w$  라 하자.

$y$ 를  $\bar{y} \in Y = \{y : 0 \leq y_j \leq 24\}$ 에 고정하였을 때, 가능해  $x$ 가 존재한다고 가정한다. 위의 LP 문제에서의 변수의 수는  $(m+1)n$ 이고, 제약식의 수는  $(n+1)m$ 이다. 제약식 행렬은 0인 원소가 매우 많고, 매우 구조화되어 있다. 쌍대 하위문제 (DS)와 원시 하위문제(PS) 양쪽 모두 여러 개의 매우 쉽게 풀 수 있는 문제들로 분리될 수 있다.

Holmberg(1992)는 평균값 교차 분할 알고리즘과, Subgradient Optimization의 몇 가지 변형들, Dantzig- Wolfe 분할 알고리즘, Benders 분할 알고리즘, 교차 분할 알고리즘, 평균값 교차 분할 알고리즘과 교차분할 알고리즘의 하위문제 단계의 조합을 생산-구매 결정문제에 적용하여 비교하였다. 지금까지 알려진 방법들 중에서는 평균값 교차 분할 알고리즘이 생산-구매 결정문제에 가장 효율적인 방법이라고 알려져 있다. 특히, 평균값 교차 분할 알고리즘이 산출하는 하한값은 다른 방법들이 산출하는 하한보다 훨씬 나은 값을 제공하였다. 따라서 이 연구에서는 평균값 교차 분할 알고리즘과 일반화된 교차 분할 알고리즘만을 비교하였다.

#### 4.2 하위문제 및 해법

생산-구매 결정문제에서 생성되는 하위문제들은 다음과 같다.

(PS)는 원시 변수  $y$ 를 고정시켜서 얻고, (DS)는 제약식 (16)을 Lagrangian 이완시켜서 얻는다.

(PS)

$$\begin{aligned} \max \quad & \sum_{i=1}^m \sum_{j=1}^n (d_{ij} - e_{ij} \bar{y}_j) u_{ij} - \sum_{i=1}^m b_i v_i \\ & - \sum_{i=1}^m \sum_{j=1}^n s_{ij} w_{ij} + \sum_{j=1}^n f_j y_j \end{aligned}$$

$$\begin{aligned} \text{s.t.} \quad & u_{ij} - v_i - w_{ij} \leq c_{ij}, \quad i=1, \dots, m, j=1, \dots, n \\ & u_{ij} \geq 0, \quad v_i \geq 0, \quad w_{ij} \geq 0 \quad \forall i, j \end{aligned}$$

(DS)

$$\begin{aligned} \min \quad & \sum_{i=1}^m \sum_{j=1}^n (c_{ij} - \bar{u}_{ij}) x_{ij} \\ & + \sum_{j=1}^n (f_j - (\sum_{i=1}^m \bar{u}_{ij} e_{ij})) y_j + \sum_{i=1}^m \sum_{j=1}^n \bar{u}_{ij} d_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^n x_{ij} \leq b_i \quad i=1, \dots, m \\ & x_{ij} \leq s_{ij} \quad i=1, \dots, m, j=1, \dots, n \\ & x_{ij} \geq 0 \quad i=1, \dots, m, j=1, \dots, n \\ & 0 \leq y_j \leq 24 \quad j=1, \dots, n \end{aligned}$$

위의 하위문제들은 다음과 같은 방법으로 매우 쉽게 풀 수 있는 문제들이다.

(PS)는 다음과 같은 방법으로 풀 수 있다.

만일 (PS)를  $u$ 만의 문제라고 가정하고,  $v_i = w_{ij} = 0$ 으로 두면, 위의 문제는  $u$ 의 목적함수 계수가 비음인 것부터 큰 순서로  $u$ 에 가능한 한 최대를 할당하는 방법으로 쉽게 풀 수 있는 문제가 된다.  $v$ 나  $w$ 의 증가는 목적함수값을 감소시키기는 하지만,  $u$ 의 값을 증가시킬 수 있게 해준다.  $v$ 나  $w$ 가 한 단위 증가하면  $u$ 가 한 단위 증가할 수 있으므로,  $v$ 나  $w$ 가 한 단위 증가함으로써 감소하는 목적식값보다 이로 인해  $u$ 가 한 단위 증가함으로써 증가하는 목적식값이 더 크게 된다면,  $v$ 나  $w$ 를 증가시키고자 할 것이다.

$v_i$ 가 한 단위 증가함으로써 감소하는 목적식값은  $b_i$ 이고, 이 때 모든  $j$ 에 대하여  $u_{ij}$ 가 각각 한 단위씩 증가할 수 있는데, 이로써 증가할 수 있는 목적식값은  $\sum_{j=1}^n (d_{ij} - e_{ij} \bar{y}_j)$ 이다. 따라서,  $b_i \leq \sum_{j=1}^n (d_{ij} - e_{ij} \bar{y}_j)$ 이면  $v_i$ 를 증가시키게 된다.

$w_{ij}$ 가 한 단위 증가함으로써 감소하는 목적식값은  $s_{ij}$ 이고, 이 때  $u_{ij}$ 가 한 단위 증가할 수 있는데, 이로써 증가할 수 있는 목적식값은  $(d_{ij} - e_{ij} \bar{y}_j)$ 이다. 따라서,  $s_{ij} \leq d_{ij} - e_{ij} \bar{y}_j$ 이면  $w_{ij}$ 를 증가시키게 된다.

그런데,  $y = \bar{y} \in Y = \{y : 0 \leq y_j \leq 24\}$ 에 고정하였을 때, 가능해  $x$ 가 존재한다고 가정하였으므로,  $y=0$ 인 경우에도 가능해  $x$ 가 존재하여야 한다.

즉,  $y=0$ 인 경우,

$$\begin{aligned} \sum_{j=1}^n x_{ij} &\leq b_i \quad \forall i \\ x_{ij} &\geq d_{ij}, \quad 0 \leq x_{ij} \leq s_{ij} \quad \forall i, j \end{aligned}$$

를 만족하는  $x$ 가 존재하여야 한다. 위의 부등식으로부터 다음과 같은 부등식을 유추할 수 있다.



$$b_i \geq \sum_{j=1}^n d_{ij} \quad \forall i, \quad s_{ij} \geq d_{ij} \quad \forall i, j$$

$e_{ij}$ 와  $\bar{y}_j$ 는 비음이므로,

$$b_i \geq \sum_{j=1}^n d_{ij} \geq \sum_{j=1}^n (d_{ij} - e_{ij} \bar{y}_j)$$

이고,

$u_{ij} \geq d_{ij} \geq e_{ij} y_j$ 이다. 따라서,  $v$ 와  $w$ 를 증가시키지 않을 것이다. 그러므로 (PS)는  $u$ 에 관한 매우 풀기 쉬운 문제가 된다.

(DS)는  $x$ 와  $y$ 에 관한 문제로 분리 가능하며, 각각의 문제는 매우 풀기 쉬운 형태이다. 비양(Nonpositive)인 목적함수 계수를 갖는 변수에 대하여 작은 순서대로 가능한 만큼 값을 할당하고 비음(Nonnegative)인 목적함수 계수를 갖는 변수는 0을 할당하는 방법으로  $x$ 와  $y$ 에 관한 각각의 문제를 풀 수 있다.

### 4.3 수치실험 결과

이 테스트에서는 최근에 얻은 해에 더 많은 가중치를 부여하는 다음과 같은 블록 조합 가중치를 이용하였다.

$$\mu_{kl} = \lambda_{kl} = \frac{l}{\sum_{i=1}^k i}, \quad l=1, \dots, k \text{ (반복 } k \text{ 일 때)}$$

최근의 해가 그 이전에 구한 해들보다는 최적해에 더 가까울 것으로 기대할 수 있으므로, 이러한 전략은 합리적이며, 수렴속도를 가속화시킬 것이 기대된다. 이와 같은 가중치를 이용함으로써 얻을 수 있는 또 한가지 장점은, 이전에 얻은 모든 해들을 저장할 필요 없이, 바로 전 단계의 정보만을 이용할 수 있다는 것이다.  $k$ 번째 반복에서는 다음과 같은 방법으로 하위 문제의 입력값을 구할 수 있다.

$$R_k = \sum_{i=1}^k i = \frac{k(k+1)}{2} \text{ 이라 정의하면,}$$

$$\begin{aligned} \bar{y}_k &= \sum_{i=1}^k \frac{l}{R_k} y_i \\ &= \sum_{i=1}^k \frac{l \cdot R_{k-1}}{R_k \cdot R_{k-1}} y_i \\ &= \sum_{i=1}^{k-1} \frac{l}{R_k} y_i + \frac{k}{R_k} y_k \\ &= \sum_{i=1}^{k-1} \left( \frac{R_{k-1}}{R_k} \right) \frac{l}{R_{k-1}} y_i + \frac{k}{R_k} y_k \\ &= \left( \frac{R_{k-1}}{R_k} \right) \bar{y}_{k-1} + \frac{k}{R_k} y_k \\ &= \frac{(k-1)k}{2} \bar{y}_{k-1} + \frac{k}{k(k+1)} y_k \\ &= \frac{k-1}{k+1} \bar{y}_{k-1} + \frac{k+1}{2} y_k \end{aligned}$$

즉,  $k$ 번째 반복에서는, 이와 같이 이전단계의 하위 문제의 입력값인  $\bar{y}_{k-1}$ 만을 이용하여  $\bar{y}_k$ 를 구할 수 있다. 이는 평균값 교차 분할 알고리즘의 장점인 컴퓨터 메모리의 효율적인 이용을 그대로 유지할 수 있게 해준다.

최근의 값에 더 많은 가중치를 주는 방법으로, 예측이론에서의 지수평활법에 이용되는 가중치가 있는데, 이는 마지막의 값에 주는 가중치가 일정하고, 이전의 값일수록 지수적으로 감소하는 가중치를 주는 방법이다. 그러나 이러한 가중치는 일반화된 교차 분할 알고리즘에 이용될 수 없다. 왜냐하면, 정리 1에서,  $\lim_{k \rightarrow \infty} \lambda_{kk} = \lim_{k \rightarrow \infty} \mu_{kk} = 0$ 인 블록 조합 가중치여야만 해들의 블록 조합이 어떤 점으로 수렴해 가므로, 마지막의 값에 주는 가중치는  $k$ 가 증가할수록 0으로 수렴해야만 한다. 그러나, 지수평활법에 이용되는 가중치에서는 이러한 조건이 만족되지 않기 때문에, 알고리즘의 수렴성이 보장되지 않는다.

이 알고리즘 시험은 펜티엄 III 350 MHz PC에서 수행되었으며, 소스코드는 C++로 작성하였다. 테스트 문제는 다섯 가지 크기의 서로 다른 문제들을 다음과 같은 방법으로 생성하였다. 먼저, 임의로  $c_{ij}, f_i, e_{ij}, d_{ij}$ 를 생성시키고,  $y = \bar{y} \in Y = \{y: 0 \leq y_j \leq 24\}$ 에 고정하였을 때, 가능해  $x$ 가 존재한다는 가정을 만족시키기 위해,  $s_{ij}$ 는  $d_{ij}$  이상의 값으로 임의로 생성시키고,  $b_i$ 는  $\sum_{j=1}^n d_{ij}$  이상의 값으로 임의로 생성시킨다.

테스트의 수행은 상한과 하한의 상대오차

$$\left| \frac{\min_k h(\bar{y}_k) - \max_k g(\bar{u}_k)}{\min_k h(\bar{y}_k)} \right|$$

가 0.01보다 작아질 때까지 쌍대와 원시 하위 문제 사이를 반복하도록 하였으며, 최대 반복 횟수는 500번으로 설정하였다. 초기 쌍대해는 동일하게 0으로 설정하였다.

테스트의 결과는 일반화된 블록 조합 가중치를 이용하는 전략이 평균을 이용하는 평균값 교차 분할 알고리즘에 비교해 계산시간과 반복횟수 면에서 훨씬 나은 결과를 산출함을 보여준다. 평균적으로, 일반화된 교차 분할 알고리즘은 평균값 교차 분할 알고리즘에 비해 계산시간 면에서 45% 이내, 반복횟수 면에서 40% 이내에 설정한 상대오차 이내로 수렴하였다. 테스트 결과의 일부를 <표 1>에 제시하였다. 표에서, 반복수는 알고리즘이 종료할 때까지의 반복(Iteration) 횟수이다.

<그림 1>은 일반화된 교차 분할 알고리즘(GCD)과 평균값 교차 분할 알고리즘(MVCD)의 성능을 비교하기 위하여, 평균값 교차 분할 알고리즘의 계산시간과 반복횟수를 1로 보았을 때, 일반화된 교차 분할 알고리즘의 결과를 비율로 나타낸 것이다.

<그림 1>에서 보는 것처럼, 일반화된 교차 분할 알고리즘은 평균값 교차 분할 알고리즘의 성능의 40~75% 정도의 향상 가능하다.

<그림 2>에서는 두 알고리즘이 수렴해 가는 모습을 보여준

표 1. MVCD와 GCD의 성능비교

문제 크기 (변수 수, 제약식수)	문제번호	MVCD			GCD		
		반복수	상대오차	계산시간(초)	반복수	상대오차	계산시간(초)
(615, 640)	1	103	0.00991098	0.17	26	0.00943016	0.05
	2	109	0.00998146	0.22	50	0.00904056	0.11
	3	103	0.0098916	0.16	27	0.00955252	0.06
(1020, 1050)	4	108	0.00994481	0.5	40	0.00987269	0.17
	5	128	0.00986815	0.55	59	0.00944859	0.33
	6	120	0.00986452	0.49	40	0.00989501	0.17
(8190, 8190)	7	108	0.00997938	5.16	46	0.00980274	2.41
	8	117	0.0099502	5.54	45	0.00973374	2.42
	9	117	0.00999385	5.33	49	0.00997798	2.42
(2020, 2010)	10	123	0.00987465	0.99	60	0.00962814	0.55
	11	126	0.00992713	1.04	54	0.00996125	0.55
	12	109	0.00982074	0.82	36	0.00998972	0.39
(13590, 13650)	13	115	0.00995338	8.96	41	0.0098558	3.63
	14	109	0.00996893	8.51	41	0.00999982	3.68
	15	112	0.00995677	8.73	48	0.00966834	4.18

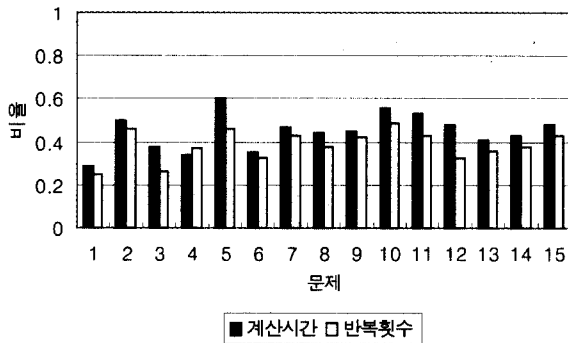


그림 1. MVCD 대비 GCD의 계산시간 및 반복횟수 비율.

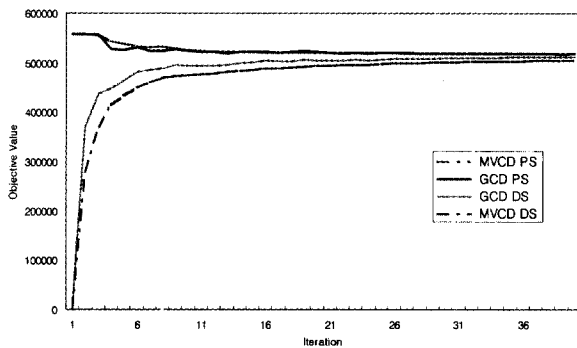


그림 2. MVCD와 GCD의 수렴성 비교.

다. 평균값 교차 분할 알고리즘은 평균값 교차 분할 알고리즘s 보다 빨리 수렴함을 볼 수 있으며, 상한보다 하한에서의 개선이 더욱 두드러짐을 볼 수 있다. 또한, 느린 End-Tail 현상이 두

알고리즘 모두에서 관찰되고 있다.

### 5. 결론 및 추후 연구 과제

이 연구에서는 일반화된 블록 조합 가중치를 이용하는 경우의 수렴성 이론을 제시하였다. 특히, 최근의 해에 더 많은 가중치를 줌으로써 수렴속도를 가속화시킬 수 있으며 평균을 이용하는 경우에 비해 더 나은 상한과 하한을 얻을 수 있음을 확인하였다. 특히, 하한의 개선이 더 두드러졌다. 느린 End-Tail 현상은 두 알고리즘 모두에서 매우 유사한 형태로 관찰된다.

추후 연구과제로는, 각 응용 분야에서 문제의 구조와 수렴 행동에 적합한 블록 조합 가중치의 개발에 관한 연구가 필요하며, 선형계획법 문제 이외의 문제로 일반화된 교차 분할 알고리즘을 확장시키기 위한 연구도 필요할 것이다.

### 참고문헌

Aardal, K. and Ari, A. (1990), On the resemblance between the Kornai-Liptak and cross decomposition techniques for block-angular linear programs, *European Journal of Operational Research*, 46, 393-398.  
 Benders, J. F. (1962), Partitioning procedures for solving mixed-variables programming problems, *Numerische Mathematik*, 4, 238-252.  
 Dantzig, G. B. and Wolfe, P. (1960), Decomposition principle for linear programs, *Operations Research*, 8, 101-111.  
 Holmberg, K. (1990), On the convergence of cross decomposition, *Mathematical programming*, 47, 269-296.  
 Holmberg, K. (1992), Linear mean value cross decomposition: A generalization of the Kornai-Liptak method, *European Journal of*

- Operational Research*, 62, 55-73.
- Kornai, J. and Liptak, Th. (1965), Two level planning, *Econometrica*, 33, 141-169.
- Van Roy, T. J. (1983), Cross decomposition for mixed integer programming, *Mathematical Programming*, 25, 46-63.
- Van Roy, T. J. (1986), A cross decomposition algorithm for capacitated facility location, *Operations Research*, 34, 145-163.
- Vlahos, K. (1992), Mean value cross decomposition for non-linear convex problems – Application to the stochastic power plant emissions problem, *TIMS/ORSA Joint National Meeting*, Orlando, U.S.A..