다종 S/W 적용에 의한 선형계획법 연구 (An Applied Technique of Linear Programming Using Multi-Softwares)

한계섭[1]

I. 서론

선형계획법(Linear Programming: L.P.)은 경영의 최적화 규모 설정, 생산계획, 수송, 분배, 재고조정, 직무배치, 예산책정, 증권투자 등 의사결정 활동에 많이 적용되고 있다. 최근에는 정보통신에 있어서 정보통신망 조정, 데이터 관리, 전송 및 통신망 설계, 이동통신과 위성통신의 효율적인 송수신망 설정, 주파수 및 시분할 정보 배정, 전자상거래(Electronic Commerce) 등 정보통신산업에도 많이 적용되고 있다. 제 2차 대전후 선형계획법은 이론적으로 복잡하고 과학적인 문제해결기법으로 지대한 관심 속에 개발되어 각광을 받고 있다. 1960년대부터 컴퓨터의 덕택에 복잡한 선형계획 문제도 쉽게 해결할 수 있는 모형으로 발전되어 초기의 수작업에 의한 최적해 도출 방법에 비해서 획기적인 진전을 거듭하면서 컴퓨터 전용 모형처럼 되고있다. 그러나 초기에는 대형 컴퓨터에 의한 기저변수(Basic Variable)의 해만을 도출하였으나 여유변수(Slack Variables)와 초과변수(Surplus Variables)의 해를 구하기는 쉽지 않아 대부분 수작업에 의한 결과 해석에만 의존해 왔다.

선형계획에서 변수에 대한 해의 값은 수학적인 견지에서 필요충분한 해답으로서 만족되어도 기업 경영의 측면에서는 일부분의 해에 불과한 것이다. 즉, 선형계획 모형에서 제약조건을 만족시키는 부등식의 변수 해는 목적함수 값을 최대화 또는 최소화하는 어느 시점에 있어서 모형의 정태적인 해(Static Solution)일 뿐이다. 따라서 L.P.를 기업이나 경제활동으로의 적용목적은 단순한 어느 시점에서의 과학적인 해의 값을 추구하는 것이 아니고 제한된 자원을 최대로 활용했을 때의 해값을 중심으로 시간적인 전후관계와 상황적인 상하관계를 고려, 앞으로의 전망에 대한 방책을 찾기 위한 것이다. 그러므로 주어진 L.P. 모델의 정적인 최적해 값보다는 전후, 상하, 좌우 등의 인접한 상황에서의 동태적인 추정치(Dynamic Inference Value)가 더욱더 중요시되고 강조되어야 할 사항이다. L.P.의 경제적 처리과정을 고려하는 것이 수학적인 처리와는 달리 기업활동에 대한 L.P.를 올바르게 적용하는 목표인 것이다.

또한 이와 같은 경제적 관점을 고려한 L.P.의 응용에 있어서 동태적인 해석을 중점으로

1) 동아대학교 경영정보학과 교수

기업경영이나 정보통신에서 그 해법을 도출할 필요가 있는 것이다. 특히 최근의 PC 발전은 1990년대 이전의 슈퍼 컴퓨터보다도 더 편리하게 L.P.의 해석을 도와줄 수 있는 능력을 갖추고 있다. 수치를 문자화시키고 수치계산에만 본원적인 목표를 두었던 컴퓨터가 최근에는 멀티미디어적인 인간활동 지원도구로 활용되는 것처럼 각 분야에서 인간의 사회활동에 응용되면서 정태적인 내용을 기반으로 동태적 문제해결을 가능케 하고 있다.

컴퓨터를 통한 동태적 L.P.해를 얻기 위해서는 여러 가지 L.P.전용 소프트웨어(S/W)를 이용할 수도 있으나 대부분의 L.P. 전용 S/W들이 고정적 정태중심 해만을 다룰 수 있다. 따라서 본 연구에서는 수학 전용 소프트웨어인 "mathematica"와 통계학 전용 소프트웨어인 "SAS"를 이용하여 소기의 동태적 L.P. 해법을 제시했다.

Ⅱ. L.P. 해석의 절차와 처리

1. L.P.의 수학적 전제와 논리적 구조

L.P.를 실질적이고 성공적이며 보다 효율적으로 경영에 적용하기 위해서는 다음 두 가지의 조건이 전제된다.

첫째는 인간에 의한 생산활동이나 경제활동은 모두 선형적이거나 이와 유사한 것이므로 L.P.를 인간활동과정에 적용하여도 문제가 있었다.

둘째로는 L.P.에 유사한 문제들은 문제의 해를 도출하는데 있어서 아주 유익하고 단순한 알고리즘(Algorithm)을 가지고 있으며 컴퓨터의 해를 작성함에 있어 단순화 해야한다. L.P.는 경제학자와 게임이론가들에게 중요한 것이다. 비록 대부분의 경제모형이 비선형일지라도 교환, 생산 그리고 자본 축적의 선형모형들은 아주 중요한 교훈적 기능을 제공한다. 더욱이 Simplex의 알고리즘과 쌍대이론, 그리고 민감도 분석의 비선형 최적화 방법 등을 터득하고 경제분석의 중심을 이해하는데 이들 기법은 크게 기여하고 있다. 게임이론에서 Zero Sum Game의 해는 L.P.문제이고 L.P.의 Minmax 이론은 기본적인 L.P.의 쌍대이론이다. Non-zero Sum 게임의 효율적인 해는 보완적인 Pivot Algorithm이라 불리는 Simplex Algorithm의 변형을 사용한다. 상호 협동적인 게임이론은 L.P.이론과 L.P.기법의 적용을 효과적으로 전개하는데는 별로 효과가 없다는 평도 받고 있다.

최근 선택적 내부점(Alternative Interior Point) 계산 방식이라는 새로운 L.P.해법의 발견에도 불구하고 종전의 Simplex 방법과 Simplex 변형방법이 L.P. 문제해법으로서 계속 보편적이고 실질적인 실용화 방식으로 적용되고 있음을 주시할 필요가 있다.

Simplex 방법은 단계적으로 한 단계씩 높여가면서 산술계산을 아주 단순하고 직관적인 개념에 기초를 두어 풀이하는 기법이다. 이렇게 단순한 방법이면서도 전통적인 프로그램 언어를 적절한 수학적인 공식으로 기술하여 실행시키는 가장 본원적(원초적)인 처리기법 때문에 연산기법의 순수한 단순성이 자주 감추어지는 경향이 있다. 그러나 지금까지 대부분의 Simplex 방법은 산술적인 절차에 의해 정적인 상태에서의 고정된 변수 처리 결과 값을 얻을 수 있게 되어 있다. 다시 말하면 Simplex 기법은 수학적인 기술절차를 보다 구체

화해서 경영과학적인 해석이 가능토록 하고 있는 기법인 것이다.

그러나 Simplex에 의한 L.P. 문제 해의 도출에는 여러 가지 제약이 있다. 목적함수 값의 최대화나 최소화의 방향에 따라 제약조건의 논리적 값과 수학적인 값과의 차이가 생긴다. 즉 최대화 문제에서는 논리식의 좌변 값이 우변 값보다 작다는 것이며 최소화 문제에서는 그와 반대이다. 또한 좌변과 우변 값이 서로 같을 때(등호일 때)는 여유변수와 초과변수의 개념을 제외시키는 점도 해를 찾는 필요 착안사항이다. 양변을 등호로 변경시키면 비록 여유변수나 초과변수가 달라져도 논리적인 해법에서는 문제가 없으나 수학적인 처리에서는 변수 수가 완전히 달라져 해의 수에 변동이 생긴다.

즉 수학적인 해를 고려하면 쌍대모형(Duality Model)과 민감도 분석(Sensuality Analysis)에서 컴퓨터 처리과정에 많은 변동이 발생한다. 이런 조건을 고려하여 제작된 소프트웨어들도 더러는 있지만 변수 수가 많아질 때 처리능력에 많은 제한을 받게 된다. 최근에 발표된 L.P. 전용 S/W에서 이런 점을 착안하여 제작된 것들도 변수 수가 100개를 넘을 때는 거의 손을 쓸 수 없는 것이 대부분이다. 따라서 실질적인 기업경영의 환경에서는 이런 점들 때문에 L.P.해를 위한 전용 S/W에서는 다양한 의사결정변수에 대처하지 못하는 편이다. 그러나 수학과 통계학 전용 S/W에서는 순수한 학문적 측면에서 수학이론을 기반으로 작성된 것이므로 다양하고 다수의 변수로 된 Simplex도 수학적인 이론 도입으로 어떠한 L.P.전용 S/W보다 강력하게 그 해를 도출 할 수 있다.

대표적인 수학 S/W로는 Mathematica와 Gaus, 통계학 S/W로는 SAS를 들 수 있다. 물론 몇가지 스프레드시트로서 L.P.의 기초적 해법을 적용한 MS-Excel, Lotus-123, Quatro 등의 패키지도 있다. L.P. 전용 S/W로는 Manager, Quantitative Business Analysis, Storm, L.P. Solution 등도 볼 수 있다. 그러나 대부분의 L.P. 전용 S/W들은 일정한 상황을 전제로 한 고정적이고 정태적인 분석 기법으로 정착시켜 놓아 해석에 있어 제약성이 많다. 또한 가상적인 전제가 없고 변동되는 상황을 연속적으로 찾을 수 있는 기회가 없어서 반복적으로 단편화하는 절차만을 기술하며 전후관계나 상황적 변화에 대응하는 방안 검토가 지엽적으로만 이루어진다.

본 연구에서는 이러한 상황을 감안하여 원시코드가 공개되어 있어 S/W의 일부를 가감하거나 방향 전환을 통해 L.P.의 해법을 상황에 부응하도록 조작해서 처리할 수 있는 "Mathematica"와 비록 원시코드는 공개되지 않았지만 L.P. 해법의 기간이 되는 역행렬(Inverse Matrix)을 효과적으로 조작함으로서 연속적이고 동태적인 여러 가지 해를 얻을 수 있는 SAS의 IML(Interactive Matrix Language)을 중심으로 L.P. 해 도출을 수학적 기법을 L.P.로 전환하는 절차를 응용하는 과정을 제시하고자 한다.

2. L.P.의 일반 구성

L.P.에서의 반복적인 절차를 처리하면서 경제모형의 크기에 따라 기복이 많은 산업연관분석 등은 변수가 400개를 넘을 경우에 PC의 처리능력의 한계에 부딪친다. 그러나 SAS의 IML에서는 행과 열의 수가 400여개가 넘어도 몇 개로 분리해서 처리한 후 통합시키는 역행렬 기능이 월등해서 그 절차는 다소 복잡하지만 많은 변수의 해를 동태적으로 찾을

수 있다. 또한 Mathematica에서는 변수의 수에 제한을 두지 않고 여유변수나 초과변수의 처리가 용이함을 알 수 있으며 지금까지의 정태적인 방법에서의 수학적인 절차와는 색다른 내용을 세부적으로 응용할 수 있게 해준다. Mathematica에서는 Simplex 방법의 해를 위한 패키지로 Constrained Max(or Min)이라는 프로그램 코드가 내포되어 있다. 따라서 목적함수와 제약식의 조건을 수학적인 표현으로 입력시키면 그에 대한 해를 자세하게 제공하며 또한 해의 내용을 중간 단계 없이 신속하게 제공해 준다.

L.P.는 특성상 수학적이고 논리적인 절차와 동태적인 해를 통한 타당성을 입증하는 해석이 요구되며 이를 염두에 두고 방향을 결정해야 한다. L.P.에서는 의사결정변수, 목적함수 그리고 시스템의 제약조건식을 정의하여 L.P. 모형이 실제상황에 적용되도록 구성해야 한다. L.P.의 일반구조와 응용분야의 모형들의 구성요소를 보면 다음과 같다.

1) 의사결정변수(Deterministic variable)

활동수준이나 창출하고자 하는 단위량 등을 변수로 결정하는 것으로 일반구조에서 n개의 결정변수로 나타낸다.

$x_1, x_2, \cdots, x_j, \cdots, x_n$등으로 표시하고 x_1은 활동 1의 양, x_2는 활동 2의 양, … x_j는 활동 j의 양, … x_n은 활동 n의 양등으로 표시된다.

2) 목적함수(Objective Function)

목적함수는 시스템의 목적달성을 최적화하기 위하여 모형상의 각 의사결정변수(x_j)와 그 공헌계수(c_j)를 곱한 것의 합으로 표시되며 의사결정자가 달성하고자 하는 목적에 따라 최대화 또는 최소화로 구분한다. L.P.에서의 목적함수의 일반구조는

최대화(혹은 최소화) $Z = c_1x_1 + c_2x_2 + \cdots + c_jx_j + \cdots + c_nx_n$

여기서 Z : 목적함수의 값

c_j = 활동j(j=1, 2, 3, …, n)의 단위당 공헌도

3) 시스템 제약조건

L.P. 모형의 제약조건식은 제한된 자원의 양으로 사용가능한 여러 가지(1~m가지)의 주어진 양 b_i(i=1, 2, …, n)로 정의하고 j(j=1, 2, …, n) 활동의 한 단위에 소요되는 i번째 자원의 소모량을 a_{ij}로 정의하면 제약조건식의 일반구조는 다음과 같은 선형계획법의 최대화 표준형으로 표시된다.

$$
\begin{array}{l}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1j}x_j + \cdots + a_{1n}x_n \le b_1 \\
\vdots \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2j}x_j + \cdots + a_{2n}x_n \le b_2 \\
\vdots \\
a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ij}x_j + \cdots + a_{i}x_n \le b_i \\
\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mj}x_j + \cdots + a_{mn}x_n \le b_m \\
x_1,\ x_2, \cdots\ x_j,\ \cdots\ x_n \ge 0
\end{array}
\qquad - (1\text{-}1)
$$

물론 위의 식에서 제약조건이 ≤로만 표시되어 있으나 최대화와 최소화 또는 문제의 내용에 따라 논리적 부등식은 변환되어, ≥ 또는 등호인 =로 그 형태 제약조건이 달라질 수 있다. 따라서 L.P. 모형의 일반구조를 형태에 따라 일반식으로 나타내면 다음 같이 요약된다.

최대화(최소화) $Z = c_1x_1 + c_2x_2 + \cdots + c_jx_j + \cdots + c_nx_n$

제약조건(Subject to):

$$
\begin{array}{l}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1j}x_j + \cdots + a_{1n}x_n\ (\le, =, \ge)\ b_1 \\
\vdots \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2j}x_j + \cdots + a_{2n}x_n\ (\le, =, \ge)\ b_2 \\
\vdots \\
a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ij}x_j + \cdots + a_{in}x_n\ (\le, =, \ge)\ b_i \\
\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mj}x_j + \cdots + a_{mn}x_n\ (\le, =, \ge)\ b_m \\
x_1,\ x_2, \cdots\ x_j,\ \cdots\ x_n \ge 0
\end{array}
\qquad - (1\text{-}2)
$$

위의 일반식을 $\sum$ 기호로 표시하여 정리하면

$$
Max(Min) \qquad Z = \sum_{j=1}^{n} c_j x_j
$$

$$
Subject\ to: \quad \sum_{j=1}^{n} a_{ij}x_{ij}\ (\le, =, \ge)\ b_i \qquad - (1\text{-}3)
$$

$$
(i = 1,\ 2,\ \cdots,\ m)
$$

여기서 c_j, a_{ij}, b_i의 값은 L.P. 문제에서 매개변수로서 단위공헌률, 기술계수 및 제약된 자원의 량으로서 의사결정변수의 계수 등이다.

위의 일반식에서 보는 바와 같이 주어진 조건하에서 의사결정변수 값을 구하는 것은 변수의 수에 관계없이 해의 도출이 가능할 것으로 예상된다. 그러나 L.P.의 실제 조건은 목적함수 값의 최대화를 위한 해를 구할 때, 여유변수 수가 의사결정변수의 수와 같아지거나 유사하게 되어 의사결정변수 수가 n개일 때 전체변수 수는 2n개가 된다. 그리고 최소화 문제에 있어서 초과변수와 인공변수(Artificial Variable)의 수를 합쳐 전체변수의 수는 2n과 3n사이에 있게 된다. L.P.의 해를 Simplex 방법에 의해 처리할 때 구해야 할 실제 변수 수는 의사결정변수의 2배 또는 3배가 되어 현재와 같은 Simplex 연산법으로는 결정변수 수가 400개가 넘는 산업연관 표와 같은 알고리즘구조에서 가능하지만 컴퓨터 처리능력 제한으로 슈퍼 컴퓨터의 경우에서조차도 해의 도출이 어렵다. 특히 이들 전체변수가

1000개가 넘어갈 때 역행렬 계수(Coefficient of inverse matrix)는 극히 어려운 연산과정을 거쳐야만 되며 일상적인 연산 방법으로는 거의 초기단계 처리조차 힘들게 되어진다. 이런 점을 고려할 때 현재까지 개발된 각종 L.P. 전용 S/W 패키지로는 문제의 해법을 거의 찾을 수가 없다. Simplex를 통한 L.P.의 민감도 분석의 경우, 전치(轉置: Transformation)되는 행렬의 수가 많을 때, 연산처리는 거의 기대할 수 없다. 그러나 다종 S/W응용으로 L.P.해를 도출할 때는 비교적 의외로 쉽게 처리되어 명료한 결과를 얻게 된다.

III. Mathematica에 의한 L.P. 해의 도출

Mathematica package는 L.P.의 해를 구하기 위한 절차와 방법은 그 능력이 아주 미약하다. 단지 간단한 L.P. 문제만 전용 프로그램으로 해결할 수 있다. 변수의 수가 많을 때 여유변수 및 인공변수는 더욱 많아져 그 몇 개의 모듈과 별도의 프로그램으로 구성된 S/W로는 접근조차 힘든다. 이런 점을 보완하기 위해서는 현재 Mathematica 패키지에 들어 있는 프로그램의 내용을 일부 수정 보완함으로서 복잡한 절차도 쉽게 처리된다. Mathematica에서 처리하는 간단한 Simplex 해법절차를 예를 들어 소개하면 다음과 같다.

먼저 Mathematica에서 수학계산에 있어서 최소화와 또는 최대화 문제의 제약식으로서 Simplex 해를 구하면,

$$Min \qquad Z = x_1 + 3x_2 + 7x_3$$

Subject to:

$$\begin{aligned} x_1 - 3x_2 &\le 7 \\ 2x_1 + 3x_3 &\ge 5 \\ x_1 + x_2 + x3 &< 10 \end{aligned} \qquad - (1\text{-}4)$$

인 경우

In[1] *Constrained Max* $[17x_1 - 20x_2 + 18x_3, \{x_1 - x_2 + x_3 < 10, x_1 < 5, x_1 + x_3 > 20\}, \{x_1, x_2, x_3\}]$

Out[1] *Constrained Max* $\{160, \{x_1 \to 0, x_2 \to 10, x_3 \to 20\}\}$

In[2] *Constrained Min* $[Z = x_1 + 3x_2 + 7x_3, \{x_1 - 3x_2 < 7, 2x_1 + 3x_3 \ge 5, x_1 + x_2 + x_3 < 10\}, \{x_1, x_2, x_3\}]$

Out[2] $\{\frac{5}{2}, \{x_1 \to \frac{5}{2}, x_2 \to 0, x_3 \to 0\}\}$

또한 L.P.에서 주어진 식을 행렬 계수를 이용하여 Mathematica에서 해결할 경우는 다음과 같은 식으로 된다.

만약 $Min \quad Z = 2x_1 - 3x_2$

$$
\begin{aligned}
S.T: \ & x_1 + x_2 < 10 \\
& x_3 - x_2 > 2 \\
& x_1 > 1
\end{aligned}
\qquad - (1\text{-}5)
$$

일 때

$IN[3] : = LinearProgrammming[\{2, -3\}, \{\{-1, -1\}, \{-1, -1\}, \{1, 0\}\}, \{-10, 2, 1\}]$

$OUT[3] : = \{6, 3\}$으로 나타난다.

위와 같이 L.P.의 문제를 풀었으나 결과는 결정변수의 정태값만 산출되었으며 의사결정을 위한 동태적 판단자료인 여유변수와 초과변수에 관한 자료값은 전혀 나타나지 않아 공공조직이나 기업에서 필요로 하는 해답은 찾을 수가 없다.

이와 같은 패키지를 L.P.에 적용하기 위해서는 Simplex와 Sensitivity Algorithm의 개발이 요구되며 이를 위해 공개된 기존의 패키지내의 프로그램 일부를 보완함으로서 다음과 같이 처리할 수 있다. 변수 수를 단순하게 하여 펜티엄 PC를 만드는 조립공장의 경우, 예를 들어 실제 문제를 다음과 같은 모형으로 나타내면 <표 1>과 같다.

<표 1> 제품 모형 내용 요소

	펜티엄 a	펜티엄 b	펜티엄 c	자원
제 1공정	2	2	1	30
제 2공정	1	2	3	25
제 3공정	2	1	1	20
순 이 익	3	1	3	

위 표의 모형에서 3가지 펜티엄 PC를 a, b, c라 하고 3개 공정에서 주어진 자원으로 조립할 때 순이익을 최대로 하기 위한 처리과정은

목적함수 $Max \quad Z = 3x[a] + x[b] + 3x[c]$

Subject to :

$$
\begin{aligned}
constraints = \ & \{2x[a] + 2x[b] + x[c] <= 30, \\
& x[a] + 2x[b] + 3x[c] \leq 25, \\
& 2x[a] + x[b] + x[c] \leq 20\} \\
& \qquad x_i \geq 0
\end{aligned}
\qquad - (1\text{-}6)
$$

위의 제약식에서 부등식을 등식으로 만들려면 식이 최대화 문제이므로 여유변수를 도입하여 표준형(Standard form)으로 만들 필요가 있다. 편의상 제1, 2, 3 공정에 대한 여유변수를 각각 s_1, s_2, s_3 으로 부호화하면,

$In[4]$ $StandardForm[constraints\_,\ labels\_List]\ //=$
$Transpose@\{labels, constraints\}/\{i\_, lhs\_ \le rhs\_\}:> s[i] == rhs - lhs$
$ln[5]$ $StandardForm[constraints\{t, l, m\}]$

으로 되어 여유변수 값에 대한 해석이 가능하다. 또한 제약식의 변동에 대한 대체값을 변환시킴으로서 변동되는 동태적 해를 간단하게 얻을 수 있다. 일단 위와 같은 절차에 따라 보완해 놓으면 요구되는 생산계획이나 예산집행, 작업의 흐름 등에 대한 선행계획 문제해를 변동되는 대로 그 내용을 분석할 수 있게 된다.

최초의 L.P. 해를 찾아내는 과정에서 기준열(Pivot Column)과 기준행(Pivot Row) 및 기준 요소(Pivot Element)는 기저변수와 비기저변수간의 한계대체률로서 결정되지만 그림자 가격(Shadow Price)을 나타내는 여유변수나 초과변수는 최적해를 결정하는 과정에서 그 파급 효과로 나타나게 된다. 이를 위해 프로그램을 보완 수정하면, 펜티엄a의 경우 다음과 같은 결과를 얻게 된다.

$In[6]$ $Revised\ Pivot\ Row\ =\ Roots[PivotRow, X[a]] := ExpandAll$
$In[7]$ $ShadowPrice\ [objective\_, Var\_] :=$
$- Coefficient\ [objective, Var]$

따라서 일반적인 L.P. 해를 Mathematica에 의해서 구하고자 할 때는 패키지 내부의 프로그램을 검토하여 자기가 요구하는 조건에 맞도록 보강함으로서 수학적인 개념을 경제적 · 경영적 내용으로 전환할 수 있게 된다.

일반적인 L.P. 모형의 해를 구하는 최대화 문제의 과정에서는 다음과 같은 일반식을 응용하여 얻을 수 있다.

$In[8]$ $objective\ =\ a_1 x[1] + a_2 x[2] + a_3 x[3] + \cdots\cdots + a_n x[n]\ ;$

Subject to :

$$\begin{aligned}
&constraints\ = \\
&\{\ k_{11}x[1] + k_{12}x[2] + k_{13}x[3] + \cdots + k_{1n}x[n](\le, =, \ge)\ b_1, \\
&\ \ k_{12}x[1] + k_{22}x[2] + k_{23}x[3] + \cdots + k_{2n}x[n](\le, =, \ge)\ b_2, \\
&\ \ \vdots \qquad\quad \vdots \qquad\quad \vdots \qquad\qquad\quad \vdots \qquad\qquad\qquad \vdots \\
&\ \ k_{1m}x[1] + k_{2m}x[2] + k_{3m}x[3] + \cdots + k_{mn}x[n](\le, =, \ge) b_m\ \}\ ;
\end{aligned} \qquad \text{- (1-7)}$$

L.P. 문제의 해를 추적하는데 있어서 일반 부등식이나 등식의 해법과 크게 다른 점은 해의 결과에 대한 해석에서 L.P. 식의 해는 가능한 모든 해영역(Feasible Solution Area)에서의 유동적인 점들이며 그 자체 모두가 해답이고 앞으로의 전망에 대한 융통성값이라는 점에서 수학적인 해값 자체와 큰 차이가 있다.

따라서 L.P.에서의 한 점에 있는 해값보다는 여러 점에 걸친 해값이 보다 더 큰 역할을 하는 이유는 제한된 자원의 잠재적 능력을 표시하는 그림자 가격(Shadow Price)과 민감도 분석의 핵심요소이기 때문이다. 그러나 그림자 가격이나 민감도 분석의 대상인 L.P. 문제의 세부적인 변동에 대한 최적해의 반응성을 예측하는 것이 무엇보다도 중요하다. L.P. 문제의 최적해에 대한 민감도는 연역적인 방법을 이용하여 Simplex의 최종해의 표에서 구할 수도 있다. 수학적인 부등식을 푸는 방법으로는 민감도나 그림자 가격을 찾는 L.P.모형의 해를 찾는데는 한계가 있다. L.P.의 해는 제한식의 모든 범위를 결정함으로서 최종해표를 구체적으로 해부할 수 있는 가능서이 나타나게 되는 것이다. 그리고 쌍대성이라고 불리는 L.P. 문제와 연관된 해도 새로운 의미를 부여하게 되는 요인이어서 구체적으로 고려해야 할 대상이다. Mathematica를 이용한 민감도를 찾는 기법을 살펴보면,

<< Linear Programming 이라는 패키지에서

profit[or cost] $= c_1x_1 + c_2x_2 + \cdots + c_nx_n$ 으로 목적함수를 제시하고

$$\text{constraints} = \begin{array}{l} \{a_1x_1[s_1] + a_2x_2[s_2] + a_nx_n[s_n] \\ \;b_1x_1[s_1] + b_2x_2[s_2] + b_nx_n[s_n] \\ \;\;\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ \;k_1x_1[s_1] + k_2x_2[s_2] + k_nx_n[s_n]\}\ ; \end{array} \qquad - (1\text{-}8)$$

로서 제약조건을 제시할 때 다음과 같은 내용의 Mathematica 명령어로 기저변수값과 비기저변수값(여유변수 또는 초과변수)를 생성시켜 최적해의 값과 민감도 값을 찾을 수 있다.

In[9] *FinalTableau* = *LP*[*profit*, *StandardForm*[*constraints*, {s_1, s_2, s_3}]]

In[10] *BasicVariables*[*FinalTableau*] → 기저변수값으로 최적해점 값 산출 명령

In[11] *NonBasicVariables*[*FinalTableau*] → 비기저변수값으로 여유변수나 초과변수의 민감도 분석값 산출 명령

즉, 기저변수값은 제약조건식의 좌변 값이며 변동에 대한 최적해 값이고 비기저변수 값은 제약조건식의 우변상수 값으로 여유변수나 초과변수의 민감도를 측정할 수 있게 해주는 값이다. 따라서 Mathematica에 의한 L.P.해에서는 기업경영에서의 의사결정을 할 수 있는 비기저변수 값을 별도로 측정할 수 있어 정책결정방안을 동태적 그림자가격 등 다른 수학적 방법에서 찾을 수 없는 내용들도 추출할 수 있다. 또한 우변상수 값의 민감도를 보다 더 구체적이고 동태적인 방법으로 측정하기 위해 필요한 여러 가지 내용을 다음과 같은 명령으로 파악하게 된다.

In[12] *ShadowPrices*[*FinalTableau*, {$s_1, s_2, s_3, \cdots, s_n$}]

이런 방식으로 그림자 가격을 산출하여 문제를 수학적인 방식으로만 해석함이 없이 가상적인 변화와 매개변수의 통계적 추정 변동내용으로 민감도를 전환, 평가할 수 있게 해준다. 선형계획법에 의해 해석되는 그림자 가격은 자원의 경제적 한계변동을 측정하는데 적용할 수 있다. 이는 사용되어지는 제한된 자원을 가장 효율적으로 활용할 수 있는 방안을 제시해 주는 것이다. 이런 관점에서 볼 때 그림자 가격은 오직 비기저변수(Nonbasic Variable)값으로만 나타나, 기저변수(Basic Variable)의 그림자 가격은 항상 "0"임을 알 수 있다.

이런 내용들을 Mathematica의 원시코드로 조정하면 L.P.를 통하여 얻을 수 있는 각종 값들은 오히려 전용 L.P. 패키지에서 산출하는 고정적인 내용보다도 기업경영의 입장에서 더욱 다양한 의사 결정 요인을 분석할 수 있게 해준다. 예를 들면 주어진 L.P.문제에서 그림자 가격을 찾고자 할 때 최종 단계인 최적해의 분석단계에서,

$$ShadowPrices[\,FinalTableau, \{S[a], S[b], S[c], \cdots, S[n]\}]$$

(여기서 S[a], S[b], Ss[c], ⋯, S[n]은 여유변수나 초과변수값이다.)

으로 표시, 복잡하고 다양한 동태적 분석도구로 이용된다. 앞에서 처리한 문제는 주로 L.P.의 최대화 문제나 최소화 문제에 대한 기저변수의 최적해 값이 확정된 상태이다. 한계값이 변동상태인 비기저변수의 한계값을 변동에 대한 의사결정 자료로 어떠한 L.P. 소프트웨어보다 가변적인 적용이 가능하다. 특수한 경우의 L.P. 해법에서 문제의 본질을 Simplex 기법의 일반적인 원칙으로 해결하기가 곤란하다. 특히 단위공헌률과 주어진 해값의 총이익과의 차이를 나타내는 해당변수가 기저변수로 도입될 때 발생하는 단위당 순이익(혹은 순비용) 증가효과에 복수개의 값을 가질 때 필요한 해를 찾기는 더욱 어렵다. 그러나 이같은 L.P.에 있어서도 Mathematica의 수학적인 해법을 도입하면 해의 도출과정을 의사결정자의 자의대로 관찰할 수 있어 동태적 경로 분석이 용이하다. L.P.의 퇴행문제(Degeneracy Problem)에 Mathematica를 이용하여 처리하는 내용을 보면, 목적함수와 일반 제약식들의 입력은 앞의 식들과 같으며 최적해는 아래와 같다.

먼저 복수의 도입변수나 복수의 방출변수가 존재하는 퇴행문제의 제약식을 DFContraints 라 하면

```
In[13]  DFConstraints = StandardForm[constraints]
        Vars = Union@@Variables/@DFConstraints
        Solution = ConstrainedMax[Objective, SFConstraints, Vars]
```

이때의 해의 값이 모두 "0"이거나 또는 일부는 실수이고 일부는 "0"일 때 "0" 값을 분석하는 식은,

$zeroVars = Cases[Solution[[2]], Rule[x\_, 0] \to x]$
$nonZerovVars = Complement[Vars, ZeroVars]$
$Solve[DFconstraints, \{x[1], s[1], s[2]\}]$
이 된다.

만약 k개의 기저저변수에 대한 퇴행문제를 해결하기 위해서는 k개의 Subsets를 활용하면 이산 수학적 개념을 도입, 'Discrete MathCombination'를 이용하면 퇴행의 변수에 대한 최적값을 찾는다.

$In[14]$ $KSubsets\ [1\_List, 0 \cdots k\_Integer?Positive\ ] :=$
$Join[Map[(Prepend[\#, First[1]]) \ \& \ KSubsets\ [Rest[1], k-1]],$
$Ksubsets\ [Rest[1], k]]$

$In[15]$ $KSzerovars = KSubsets\ [ZeroVars, n]$
Zero의 수에 따라 n=1, 2, 3, …, k라 하면

$In[16]$ $FindBasic[DFConstraints, nonzerovars, KSzerovars]$

이런 식으로 Mathematica의 절차를 끝내고 일반 L.P.와 같이 FinalTableau [Objective, Constraints]를 통하여 퇴행문제도 단계적 절차로 최적해를 구할 수 있다. 그러므로 Mathematica는 순수 수학의 해법을 위한 소프트웨어이지만, 이를 경영과학적인 측면에서 활용하기에 따라 각종 의사결정 변수를 찾을 수 있다. 특히 자연어(Natural Language)를 사용할 수 있어 적합한 해의 도출에 있어서, 어떠한 L.P. 전용 소프트웨어보다 효과적이고 다양하게 적용될 수 있다.

IV. SAS에 의한 L.P.해의 도출

SAS(Statistical Analysis Systems)는 통계학을 위한 소프트웨어이다. 자체 소프트웨어 패키지에 L.P. 해를 위한 전용 패키지를 포함하고 있지만 효과적인 수학적인 처리절차는 응용하지 않고 있다. 그러나 L.P.문제 이전에 행렬(Matrix), 그 중에서 역행렬(Inverse Matrix)에 관한 수학적인 내용을 L.P. 해법에 적용하면 어떠한 L.P. 전용 소프트웨어보다 탁월한 해의 도출효과를 얻을 수 있다.

1. SAS에 의한 Simplex 처리

Mathematica에 의한 L.P. 문제 처리는 공개된 원시코드를 조작함으로써 최초의 L.P. 모델에서 주어진 제약식 그래프를 이용하여 각종 의사 결정 변수 값들을 효과적으로 얻을 수 있음을 보았다. SAS에서는 Mathematica와는 다소 다르게 최대화, 최소화 문제의 성격

과 제약조건의 조건에 따라 행렬(Matrix)과 벡터(Vector)조합의 형태로 변형시켜 동태적인 Simplex 값을 얻게 된다. SAS 소프트웨어에서 취급하는 L.P. 해에 관한 목표는 SAS 자체가 통계학적인 문제 해결에 초점을 두고 제작된 것이어서 회귀모델(Regression Model)작성과 그 해를 위한 언어구조로 되어있다. 그리고 SAS에서의 일반선형모형(General Linear Model)의 구성은 분산분석처리를 위한 모형변수나 지표변수로 작성된다. L.P. 문제를 Simplex로 처리하기 위해서는 제약식의 구성을 부등식의 논리조건에서 모두 등식의 수학적인 조건 표시로 전환시켜야 함은 이미 앞에서 살펴보았다. 즉 선형 부등식 관계의 제약조건들을 일반선형 방정식 관계로 전환 재작성해야 한다. 그 결과 목적 함수 값이 최대화를 위한 조건이면 제약식들은 여유변수를 추가 시켜야 하며 최소화의 경우는 초과변수들을 추가시켜 아래와 같은 일반 선형 방정식으로 작성한다.

목적함수 $Max(Min) \quad Z = a_1x_1 + a_2x_2 + \cdots + a_nx_n$

Subject to :

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + s_1 + s_2 + \cdots + s_n &= b_1 \\ a_{12}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + s_1 + s_2 + \cdots + s_n &= b_2 \\ \vdots \qquad \vdots \qquad\quad \vdots \qquad\quad \vdots \qquad\quad &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + s_1 + s_2 + \cdots + s_n &= b_m \end{aligned} \qquad \text{- (2-1)}$$

$$\text{단, } x_j \geq 0, \ j = 1, 2, \cdots, n$$

윗식(2-1)은 앞절의 (1-1)식과 (1-2)식의 L.P. 모형을 Simplex화하여 최대치와 최소치를 구하기 위한 일반식이다. 이 식에서 $s_1, s_{2,}, \cdots, s_n$은 여유변수를 사용할 수 있고 초과변수로도 사용할 수 있다.

위의 Simplex모형의 제약식을 행렬로 만들기 위해 좌변값을 A, 우변 값을 B로 하면

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots a_{1n} \\ a_{21} & a_{22} & \cdots a_{2n} \\ \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots a_{mn} \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad S = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \qquad \text{- (2-2)}$$

로 구성된다.

최대화 문제에 있어서는 $AX \leq B, \ X \geq 0$인 조건에서 $Max \ Z = CX$이다.

최소화 문제는 $AX \geq B, \ X \geq 0$인 조건에서 $Min \ Z = CX$된다.

이들 제약조건식으로 표시하여 Simplex의 일반식으로 만들면 다음과 같다.

$$\begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} x_1 + \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} x_2 + \cdots + \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix} + \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & s_m \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

$x_j \geq 0,\ j=1,2,\cdots,n,\cdots,n+m$인 조건에서

$$\text{Max(Min)}\quad Z=c_1x_1+c_2x_2+\cdots+c_nx_n+0x_{n+1}+\cdots+0x_{n+m} \qquad -\ (2\text{-}3)$$

(2-2)식과 (2-3)식을 이용하면

$[A+S]X=B$이므로

$X=\dfrac{B}{[A+S]}$ 가 되는데,

$$S=\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}_n \text{ 인 단위행렬이므로 } S=I_n \qquad -\ (2\text{-}4)$$

$X=\dfrac{B}{A+I_n}$ 로 되어 $X=[A+I_n]^{-1}$, $B=[A^{-1}+I_n]B$, $X=A^{-1}B+I_nB$이다.

$$-\ (2\text{-}5)$$

$(\because\ [I_n]^{-1}=I_n)$

이들 행렬값을 SAS의 IML(Interactive Matrix Language)로 표현하면

$X=\text{Solve}(A,B)$;

$X=A^{-1}B+I_nB$, $X=\text{Inv}(A)\times B+I_nB$가 된다.

X의 해는 행렬 A의 역행렬에다 B의 값을 각각 곱한 값에 I_nB 즉, S_nB로 대체된다. 그런데 S_n의 값을 X의 값이 최적해인때 즉 기저해 값이 0일 때, S_n 값이 최적이되므로 X=0이 되어야 한다. 따라서 $I_nB=-A^{-1}B$, $I_n=-A^{-1}$일 때 최적이다. $I_n=S_n$이므로 $S_n=-[A]^{-1}$로 구할 수 있다.

이는 기저변수(또는 초과변수)값은 A의 역행렬 값에 음의 부호(-)를 부가함으로써 얻어진다. 그 결과 A와 $S_n(I_n)$이 모두 정방행렬이고 S_n은 단위행렬로 되어 어떠한 행렬에서도 그 해를 구할 수 있다. 이들 해를 구하는 절차를 SAS의 IML로 표시하면 다음과 같다.

```
PROC IML ;
        DATA AA1 ;
        USE DATA = A ;             → (2-2)식의 A행렬값을 사용함.
        Read all into A ;          → 행렬 A의 각 요소값 전체로 A이라는 행렬
                                     요소값으로 변환.
        Print A ;                  → A값을 print
        USE DATA = Sn ;
```

```
Read All into Sn ;              → Slack 또는 Surplus 변수값을 행렬로 나타
                                  냄
Print Sn                        → Sn을 출력
In = Sn(n)                      → n개의 단위행렬로 작성
X = InV(A + In)*B               → X 값의 도출
Sn = -InV(A)                    → Sn(여유 또는 초과변수값) 도출
```

이처럼 X 와 S_n 값은 n의 크기에 구애됨이 없이 SAS IML에서 간단한 명령과 연산식으로 소정의 해를 각각 얻을 수 있다. 또한 여유변수나 초과변수 값의 동태적인 내용을 분석하고자 할 때는 Do 문장을 이용 반복되는 변동값을 찾아 분석할 수 있다. 이를 위한 IML 절차와 알고리즘을 SAS 언어로 표시하면 다음과 같다.

```
PROC IML ;
        DATA AA1 ;
        USE Data = A ;
        Read All into A ;
        DATA AA2 ;
        USE Data = Sn ;
        Read All into Sn ;
        DATA AA3 ; set AA1 AA2 ;
        In = Sn(n) ;
        X = INV(A+Sn)*B ;
        Sn = -INV(A) ;
        Do DATA ;
           Read NEXT ;
           IF Sn = n then STOP ;
        END ;
        Print ;                          → 민감도 분석 내용
        Run ;
        Quit ;
```

SAS를 통한 L.P.문제를 쌍대모형으로 전환시키기 위해서는 쌍대모형이 되는 조건인 본원문제(Primal Problem)의 최적해가 존재한다는 전제가 있어야 한다. 이때 쌍대문제(Dual Problem)의 최적해도 존재하고 또 이 두 문제의 최적값(Optimal Value)은 서로 같게 될 것이다. 이들 본원문제와 쌍대문제는 각 행렬의 전치행렬에 불과하므로 쌍대문제는 단지 본원문제를 SAS의 IML에서 Proc Transpose[Primal Problem];이라는 명령으로 해를 얻게 된다. 또한 쌍대문제의 민감도 분석에 있어서도 한계값 또는 단위당 증감효과 측정

으로 이루어질 수 있으므로 본원 문제에서와 마찬가지로 반복적인 한계량 변화를 측정함으로서 Mathematica에서 구한 민감도를 측정할 수 있다. 쌍대문제에 대한 해와 민감도 분석 내용을 IML로 핵심절차와 알고리즘을 표시하면 다음과 같다.

위의 SAS의 IML Program은 단위당 민감도를 측정해 준다. 그리고 쌍대모형은 본원문제의 최소화를 위한 목적함수로 구성되었을 때 초과변수와 인공변수의 처리가 복잡함을 피할 수 있다. 그 결과 최소화 문제를 최대화 문제로 전환시킴으로서 행렬 연산을 대폭 단축시켜 주는 이점을 가지고 있다. 특히 본원문제에 있어서 변수가 많은 경우는 쌍대모형으로 전치시킴으로서 연산효율을 크게 향상시키는 역할을 한다. 이외에도 SAS에서는 역행렬 계수의 변동을 이용하여 각종 단위공헌률의 변동, 기술계수의 전환에 의한 새로운 경영전략 수립, 제한적 자원인 우변상수값의 활용의 대한 최적방안 및 최적 재고량과 물류 유통의 단위개념과 한계 개념을 도입하여 활용할 수 있게 해준다.

```
Proc IML ;
        Data BB1 ;
        Use Data = A ;
        Read All into A ;
        Data BB2 ;
        Use Data = Sn ;
        Read All into Sn ;
        Data BB3 ;
        Set BB1 BB2 ;
        In = Sn(n)
        X = INV(A + Sn)*B
        Sn = -INV(A)
        run ;
        Data BB4 ; Set BB3 ;
        Proc Transpose data BB3 out = BB4 ;           → 쌍대모형문제
        run ;
        Do Data = BB4 ;
        Read NEXT ;
        IF Sn = n then Stop ;
        End ;
        Print ;                                       → 민감도 분석내용
        Run ;
        Quit ;
```

이처럼 기업경영이나 조직의 운용에 있어서 가장 많이 사용하는 관리기법으로서의 선형계획 기법은 제한적인 소프트웨어 개발찾기 보다는 기존의 소프트웨어를 활용함으로서 폭이 더 넓고 깊이 있는 분석과 수학적 개념을 경영학적인 개념으로 전환시키는 역할을 맡을 수 있다. 최근 계속적으로 첨단화되고 있는 각종 개발도구의 출현으로 더 좋은 기구들이 출현되겠지만 누구나 일반화시켜 손쉽게 또 가까이 할 수 있는 이들 기존 소프트웨어를 활용하면 L.P.문제는 물론이고 이보다 더 복잡하고 어려운 의사결정문제도 관리자의 의도대로 해답을 도출할 수 있다.

V. 기타 소프트웨어에 의한 L.P.해의 도출

Mathematica와 SAS에 의한 L.P.문제를 처리하는 방법과 알고리즘을 살펴보았다. L.P.문제 해를 얻기 위한 전용 소프트웨어도 많이 나왔지만 그들의 한계성과 제약성 때문에 상당부분 소기의 해답을 얻기는 힘들다. Mathematica나 SAS 소프트웨어나 L.P.해법 전용 패키지와는 달리 행렬과 역행렬, 단위행렬, 고유 값, 전치행렬, 대각행렬 등을 이용하여 사무자동화적인 측면에서 보통 일반 스프레드시트인 MS-Excel과 Lotus-123, Qutro를 알아보자.

Excel과 Lotus-123은 처음부터 표의 계산, 도표, 통계적 처리 등을 위해 개발되다가 점점 수학적, 논리적 개념을 소프트웨어에 포함시켜 간단한 행렬, L.P.처리를 할 수 있게 강화하고 있다. 특히 통계함수, 검색함수, 데이터베이스, 수치연산, 문자데이터, 재무, 삼각함수, 정보함수 및 행렬 연산함수를 별도로 조직, 매크로 처리까지도 하는 함수 레퍼런스(Function Reference)를 자체 포함하여 L.P. 문제해결까지도 추진하고 있다. 그러나 방대한 각종 업무처리 기능을 강화하다보니 L.P.문제 해결은 극히 지엽적이고 부분적으로만 처리할 수 있다. 이는 교육을 위한 교실에서의 초보적 L.P.문제를 다룰 수 있는 수준이며 L.P.명령도 불과 10여개를 주작성축으로 이용함으로서 제한적 문제 해결에만 적용된다.

그러나 L.P.에 입문하거나 L.P.의 각종 변수의 정태적 처리를 하고자 할 때는 상당부분 Excel과 Lotus-123으로도 해결할 수 있다. L.P.는 우리사회에서 발생하는 각종 문제해결에 있어서 큰 기여를 하고 있으며 기업에서 발생하는 경영문제의 의사결정과정에 도움이 될 수 있는 계량적 참고자료를 제공하는데 큰 공헌을 제공하고 있으므로 이에 종사하는 요원들은 이런 점에서 MS-Excel이나 Lotus-123을 활용할 수 있는 수준으로 L.P.문제 전반은 물론이고 Simplex 기법과 같은 복잡한 처리도 보다 시각적이며 감각적인 수준에서 그 내용과 의사결정을 위한 분석에서 쉽게 이해할 수 있어야겠다. 그러한 소프트웨어 개발이 L.P.전문가가 Mathematica나 SAS에서 보다 구체적인 분석 사항을 찾아 의사결정 변수로 찾아내듯 할 수 있도록 대중적 소프트웨어 개발에 착안할 수 있을 것이다.

VI. 결론

선형계획법은 경영과학의 핵심으로서 일반 기업활동에서는 물론이고 생산공정, 제품수주, 자재관리, 조직관리, 공공업무 처리 및 행사, 군의 전략 전술 시행 등 의사결정변수 조정에 있어 중추적인 계량적 문제해결의 도구로 오랫동안 선두자리를 지켜오고 있다. 이에 대한 그 동안의 많은 변동과 방법 개선 그리고 처리능력 제고를 위한 각양각색의 전진적 시도가 도모되고 있으나 아직까지는 고전적인 테두리에서 벗어나지 못하고 있다. 더구나 선형계획법에 의한 문제해결의 각종 방법들이 컴퓨터를 통한 계산 형식으로 전환되고, 전용 소프트웨어가 개발되어, 쉽고 간단하게 처리되는 방안이 활용되고 있다. 그러나 이들 L.P. 전용 소프트웨어의 대부분은 아직도 정태적 분석 방법에서 해답을 찾는 차원을 벗어나지 못하고 있다.

본 연구에서는 이러한 정태적 해법에서 벗어나, 새로운 환경과 상황변동에 대처할 수 있는 동태적 변동요인을 그때그때의 위치와 시기에 맞게 문제의 해를 얻을 수 있는 몇 가지 방안을 제시했다. 이들 방법은 기존의 선형계획법 전용 패키지를 이용하면서도 동태적 분석을 위한 내용을 얻고자 할 때도 얻지 못한채 선형계획법에 의한 최적해의 고정값만 도출하고 만다. 단지 전용 패키지에서와 같은 변수처리의 결과값만 정태적 확정치로서 나타날 뿐이다. 그 결과, 고정된 값 하나로만 의사결정을 해야 하는 모순을 갖고 있다. 따라서 여유변수나 초과변수의 단위당 한계치(Unit Marginal Value)에 의해 결정되는 그림자 가격의 유발계수 측정은 또 다른 절차를 반복해야 찾아 볼 수 있다.

제한된 자원의 활용성과에 의해 나타나는 생산유발계수 측정은 고정된 선형계획 해법으로는 거의 그 최적해를 찾을 수가 없다. 그러나 경영의사결정자의 입장에서는 어떤 환경이나 어떤 시각에서도 주어진 선형계획법의 모형을 통해, 보다 과학적인 방법으로 최적방안을 도출하려고 노력할 것이다. 본 연구에서 제시한 Mathematica를 이용하게 되면 동태적인 내용요소를 체계적으로 도출하고 그때의 환경과 상황을 제시함으로서 보다 세부적인 결과를 얻을 수 있다.

L.P.해의 Mathematica 응용 방법에서 변수의 변화에 대한 시계열적 내용은 변동되는 조건에 따라 서로 다르게 그 해를 나타낸다. 이는 고정된 L.P. 패키지에 비해 보다 더 구체적이므로 요구되는 조건을 자유롭게 조작할 수 있어 소요되는 결과를 쉽게 획득할 수 있다. 그러므로 Simplex 해를 통한 최적해 값에서 그림자 가격의 해석을 가급적 Mathematica의 처리 기법 등을 이용하면 의사결정에 큰 도움을 줄 수 있음을 본 연구에서 제시했다.

선형계획법에서 최적해는 기저변수와 비기저변수 또는 기저변수 상호간의 한계대체율(Marginal-rates of Substitution)에 의해 결정된다. 이는 변수간의 유발계수와 기술계수 및 전후방 연관계수(Forward and Backward Linkage Effects)의 관계로 도출된다. 이는 제약된 자원의 최종 한 단위가 변동하면 그에 따라 다른 변수의 변동에 어느정도 영향을 가져오는가를 표시한다. 이러한 관계는 선형계획법에서 각 기술계수와, 기술계수에 의한 여유 및 초과변수의 매개변수, 그리고 제한된 자원간의 행렬로 나타난다. 단위개념에 의한 기술계수의 변동은 각 요소 행렬과 역행렬 계수로 산출되기 때문에 L.P. 해의 소프트웨어에서는 고정적 해조차 찾기가 어렵다.

특히 행과 열의 수가 많은 행렬과 역행렬 계수를 찾으려는 현행의 L.P. 전용 소프트웨어로서 최적해 도출은 극히 제한적이다. 그러나 SAS에서는 어떠한 역행렬 값도 구하는데 간단한 절차로 최적치 도출이 가능하여 실질적인 L.P. 해의 도구로서 이들 소프트웨어 활용이 적극 권장된다. 간단하고 시험적으로 사용하기 위한 문제는 각종 선형계획법 전용 소프트웨어를 이용해도 그 가치를 충분히 발휘할 수 있다. 또한 최근에 스프레드시트로 개발되어 각종 유익한 소프트웨어의 장점들을 도입하여 제작한 MS-Excel, Lotus-123, Quatro 등도 L.P. 문제를 해결하기 위한 기법들을 활용함으로서 대중적인 최적해 획득 수단이 되고 있다. 그러나 이들 기능은 아주 초보적이고 원론적인 단계에서 응용되고 있다. 이들 패키지들도 L.P.에 의한 의사결정의 중요성이 지속적으로 인식되면 점점 유익한 도구가 될 것으로 전망된다.

본 연구에서도 이들 스프레드시트로 제작된 도구를 이용하여 L.P. 문제들을 처리해 보았으나 제약조건이 너무 많아 소기의 해를 얻을 수는 없었다. 이런 결과를 바탕으로 볼 때 경영활동에 있어서 의사결정의 중요성을 감안, 다양한 변수 변동을 최적해 도출에 적극 활용하기 위해서, 현존하는 각종 L.P. 소프트웨어를 활용하면 큰 효과가 기대된다. 또한 보다 더 차원 높은 결정변수의 해 값들을 얻을 수 있는 본 연구에서 제시한 다종의 소프트웨어인 mathematica와 SAS를 선형계획 모델의 최적해 산출에 적용하면 경영활동전반에서 결정변수값을 통하여 상황을 판단하는데 큰 도움이 될 것이다.

참고문헌

- Stephen Wolfram, Mathematica, 2nd ed., Addison-Wesley, New York, 1991.
- T. W. Gray & J. Glynn, Exploring Mathematics with Mathematica, Addison-Wesley, New York, 1991.
- W. T. Shaw & J. Tigg, Applied Mathematica, Addison-Wesley, New York, 1994.
- Roman Maeder, Programming in Mathematica, Addison-Wesley, New York, 1991.
- R. D. Skeel & J. B. Keiper, Elementary Numerical Computing with Mathematica, McGraw-Hill, New York, 1993.
- P. Boyland, A. Chandra, J. Keiper, Gwide to Standard, Mathematica Packages, 3rd ed., Wolfman Research Inc., Illonis, 1993.
- Nancy Blachman, Mathematica: A Practical Approach, Prentice-Hall, New Jersey, 1992.
- K. A. Ross & C. R. B Wright, Discrete Mathematics, 3rd ed., Prentice-Hall, New Jersey, 1992.
- Steven Skiena, Implementing Discrete Mathematics, Addison-Wesley, Redwood City, CA, 1990.
- Hal R. Varian, Editor, Economic and Financial Modeling with Mathematica, Springer-Verlog, New York, 1993.
- Ramon C. Little & R. J. Freund, SAS system for Linear Models, 3rd ed., SAS Institute Inc., Cary NC, 1991.
- SAS Institute, SAS/IML Software, SAS Inc., Cary NC, 1994.
- Paul E. Spector, SAS Programming for Researchers and Social Scientists, SAGE Publications, London, 1993.
- R. P. Cody & J. K. Smith, Applied Statistics and the SAS Programming Language, 3rd ed., North-Holland, Amsterdam, 1991.
- R. Aster & R. Seidman, Professional SAS Programming Secrets, McGraw-Hill, New York, 1991.
- Lotus Development, Reference Manual, JA Part NC, 1993.
- Microsoft, Excel User's Guide, Microsoft Corp., 1991.
- Microsoft, Excel Function, Microsoft Corp., 1993.
- SAS Institute, SAS/OR User's Guide, SAS Inc., Cary NC, 1993.
- SAS Institute, SAS/IML User's Guide, SAS Inc., Cary NC, 1993.
- Hillier Lieberman, Introduction to Operations Research, Holden-Day, Oakland CA, 1992.

- Saul I. Gass, Linear Programming, 2nd ed., McGraw-Hill, New York, 1988.
- SAS Institute, SAS Procedure Guide, SAS Inc., Cary NC, 1992.

An Applied Technique of Linear Programming Using Multi-Software

Han Kay Seob[1)]

Linear programming has become an important tool in decision-making of modern business management. This remarkable growth can be traced to the pioneering efforts of many individuals and research organizations. The popular using of personal computers make it very easy to process those complicated linear programming models. Furthermore advanced linear programming software packages assist us to solve L.P. models without any difficult process. Even though the advanced L.P. professional packages, the needs of more detailed deterministic elements for business decisions have forced us to apply dynamic approaches for more reasonable solutions. For the purpose of these problems applying to the "Mathematica" packages which is composed of mathematic tools, the simplex processes show us the flexible and dynamic decision elements included to any other professional linear programming tools.
Especially we need proper dynamic variables to analyze the shadow prices step by step. And applying SAS(Statistical Analysis System) packages to the L.P. problems, it is also one of the best way to get good solution. On the way trying to the other L.P. packages which are prepared for Spreadsheets i.e., MS-Excel, Lotus-123, Quatro etc. can be applied to linear programming models. But they are not so much useful for the problems. Calculating simplex tableau is an important method to interpret L.P. format for the optimal solution. In this paper we find out that the more detailed and efficient techniques to interpret useful software of mathematica and SAS for business decision making of linear programming. So it needs to apply more dynamic technique of using of Mathematica and SAS multiple software to get more efficient deterministic factors for the sophiscated L.P. solutions.

1) Dong-A University, Dept of MIS