

명령어 기반 고속선형계획법 인터페이스 설계 및 구현†

김상국¹ · 안재근² · 임성목¹ · 박순달¹

¹서울대학교 산업공학과 / ²한경대학교 컴퓨터공학과

The Design and Development of Command-Based User Interface for High Speed Linear Programming Solving System

Sang-Gook Kim¹ · Jae-Geun Ahn² · Sung-Mook Lim¹ · Soon-Dal Park¹

This paper deals with the design and the development of user interface for a high speed linear programming solving system. The general users' educational and practical requirements are analyzed and the functionalities that should be implemented for the requirements are designed. The user interface of the system consists of various carefully designed built-in commands that perform the functionalities. The names of the built-in commands are chosen to enhance the users' convenience. We also address the various factors that should be considered in the design of the linear programming solvers when integrated with the user interface. As one of user-friendly environments, data input system that can understand user-written arithmetic form is added to our user interface system. Finally, we compare our system with other user interface systems.

1. 서론

기존의 사용되고 있는 선형계획법 프로그램인 GINO나 LINDO(Schrage)는 명령어를 기반으로 한 DOS용 프로그램으로서 교육용으로 많이 사용되어 왔으며, DOS 환경에서 사용되면서 프로그램 자체의 통합환경을 가진 프로그램으로 Turbo-Simplex, STORM 등과 같은 프로그램이 있다. 이런 프로그램들은 선형계획법 이외에도 2차계획법, 일정관리나 재고관리, 생산관리, 수요예측 등 여러 가지 해법에 대한 솔루션을 제공하는 장점을 가지고 있다. 하지만 이러한 프로그램들은 교육용이란 목적으로 개발되었기 때문에 입력할 문제의 크기와 컴퓨터의 메모리에 많은 영향을 받는다(설동렬, 1995). 이에 비해 상용 패키지인 CPLEX(CPLEX Optimization Inc., 1996)와 같은 소프트웨어는 선형계획법, 혼합정수계획법, 비선형계획법 등 여러 가지의 해법을 제공하면서 문제 입력자료의 규모가 아주 큰 것도 소화해 낼 수 있는 소프트웨어로, 실무에서 문제의 규모가 큰 자료의 입력이 효과적으로 수행될 수 있는 장점을 가지고 있다.

이외에도 내부점(interior point) 해법을 이용한 선형계획법 프로그램인 LIPSOL, HOPDM, BPMPD, LOQO, PCX의 프로그램과 단체법(simplex method) 해법을 이용한 LP SOLVE, MINOS, SOPLEX 등의 프로그램은 빠른 계산속도와 해의 정확성을 목적으로 많이 연구 개발되어 왔다. 현재 공개용 선형계획법 프로그램인 LPAKO(1999)나 LPABO(1999) 그리고 LPASO(1999)도 이와 같은 목적으로 연구 개발되어 왔으며 위에서 설명한 프로그램들과 비교하여 계산속도나 규모가 큰 문제에 대한 해의 정확성에 대해서도 뒤지지 않는 프로그램으로 발전해 왔다. 하지만 이러한 목적을 가지고 개발된 프로그램은 사용자가 쉽고 편하게 사용할 수 있는 프로그램으로서 갖추어야 할 사용자 편의성이 고려되지 않은 소프트웨어였다.

본 연구에서는 선형계획법 상용 패키지나 교육용 소프트웨어처럼 사용자 편의성을 고려한 프로그램 개발의 필요성을 인식하고 이를 목적으로 사용자에게 편의성을 줄 수 있는, 명령어를 기반으로 한 프로그램을 개발하였다. 서두에서 언급한 상용 패키지나 교육용 선형계획법 소프트웨어 대부분은 유닉스 환경과 DOS 환경에서 사용되었기 때문에 선형계획법 문제에 대한 자료를 입력받고, 입력받은 문제를 풀며, 이렇게 해서

† 본 연구는 한국과학재단의 특정기초연구과제(과제번호 98-0200-07-01-2)의 지원을 받았다.

나온 결과를 확인하는 일련의 모든 과정들이 명령어를 기반으로 사용자 인터페이스를 형성하여 왔다. 최근에 와서는 window 환경을 기반으로 한 프로그램들이 만들어지고 있는 것 또한 사실이다. 본 연구에서는 유닉스 환경에서 사용할 수 있는 소프트웨어를 개발하기 위해 명령어를 기반으로 한 사용자 인터페이스를 형성하도록 하였다. 또 명령어를 개발하는 데 있어서 사용자가 이해하기 쉬운 명령어를 선택하는 것은 아주 중요한 사항이 되었고, 소프트웨어에서 요구되는 기능들이 무엇 인지를 파악하는 것 또한 중요한 일이었다. 본 연구에서 개발한 프로그램의 명령어 설계는 사용자가 이해하기 쉽고 편하게 사용할 수 있는 용어를 선택하였고, 기능설계에서는 문제의 자료입력을 쉽게 하고 입력된 문제의 정보나 입력된 문제를 풀고 나온 결과의 정보들을 사용자가 원하는 부분만 쉽게 보여줄 수 있는 기능에 주안점을 두었으며, 이 밖에 부가기능에도 많은 중점을 두었다.

먼저 본 연구에서는 선형계획법 소프트웨어에 있어서 자료의 입력과 출력, 사용자와의 커뮤니케이션 등 사용자 인터페이스를 향상시키기 위한 방안을 모색하여 구현하였다. 특히 사용자 인터페이스를 향상시키는 방안 가운데 하나로 자료를 입력하는 데 있어서 사용자가 화면상에서 직접 선형계획문제의 형태를 수식형태(arithmetic form)와 유사하게 입력하여 문제를 풀 수 있도록 하는 구문처리기를 개발하였다. 또한 부가기능 중에 입력받은 문제를 풀고 나서 제약식을 추가하거나 변수를 수정하는 기능, warm start 기능 등 몇 가지 기능들을 개발함으로써 편의성을 향상시켰다.

2. 선형계획법 사용자 인터페이스 개발

2.1 선형계획법 해법 프로그램의 종류와 특징

선형계획 문제를 위한 해법 프로그램으로는 공개용 프로그램인 단체법 프로그램 LPAKO와 내부점법 프로그램인 LPABO, 그리고 두 가지 방법을 혼합한 방법인 LPASO를 선택하였다. 단체법 프로그램인 LPAKO는 사용자가 직접 조정 가능한 해법의 종류로 원단체법과 쌍단체법이 있으며, 파라미터로는 사전처리, 평가전략, LU분해 갱신방법, 규모화, LU자료구조, 초기해 설정방법, 국면문제 생성방법, 퇴화방지방법 등이 있다. 내부점법 프로그램인 LPABO는 해법의 종류로 원아핀규모화, 쌍대아핀규모화, 원쌍대아핀규모화, 원장벽, 쌍대장벽, 원쌍대장벽, 예측지수정자방법, 비가능예측지수정자방법, 다중중심수정(multiple centrality correction)을 이용한 비가능예측지수정자방법이 있고 파라미터로는 분해방법(factorization method), 순서화방법, 사전처리, 규모화, 개선폭 결정방법, 종료조건 등이 있다. 단체법과 내부점 방법을 혼합 사용한 프로그램인 LPASO는 처음에 내부점 방법으로 문제를 풀다 최적해가 정점 최적해에 가까워지면 단체법으로 바꿔 문제를 풀도록 하는

crossover 방법을 구현한 프로그램이다. 또한 이 프로그램은 순수한 단체법 방법과 내부점 방법으로도 문제를 풀 수 있도록 되어 있어 사용자가 이런 해법을 직접 조정가능하며, 파라미터로는 crossover의 시기를 결정하는 규칙이 있다.

2.2 인터페이스 설계

2.2.1 사용자의 편의성을 고려한 기능설계

기존의 선형계획법 프로그램은 고유의 모형화 및 사용자 편의성을 강조하여 개발되었다. 이런 프로그램들은 입출력 방식이 패키지별로 각각 다르기 때문에 입력정보의 파일보관 가능성과 문제의 형태변환 등을 사전에 충분히 고려해야 한다(Yurkiewicz, 1988). 이들 패키지에서 사용하는 대표적인 입력형태로는 대형 패키지에서 많이 사용하는 MPS 파일 입력방식과 소형에서 주로 입력하는 대화형 입력, 그리고 스프레드 시트(spread sheet) 입력방식 등을 들 수 있다(Sharda, 1992).

사용자 인터페이스 프로그램은 교육용 목적 또한 가져야 하기 때문에 사용자가 해법을 이해하는 데 도움을 주며 사용이 편리해야 한다. 이런 목적에 적합한 프로그램을 개발하기 위해 입출력 양식의 다양화를 고려하였다. 먼저 입력 방식의 편의성을 위해서는 MPS 형식과 열단위 형식을 입력받는 파일입력방식과 화면상에서 선형계획법 문제의 수식을 입력받는 수식형태방식을 사용하였다. 파일입력방식은 프로그램의 수행도중 발생할 수 있는 입력오류나 입력정보의 간편한 수정을 위해 사용하였으며, 수식형태 입력방식은 선형계획법 문제의 형식과 거의 유사하게 입력함으로써 선형계획법을 처음 배우는 사용자도 쉽게 문제를 입력함으로써 교육적인 효과를 높일 것으로 기대할 수 있었다. 출력방식으로는 파일출력방식과 화면출력방식을 사용하였다. 파일출력방식은 첫째로 해의 결과 및 대략적인 수행절차를 확인하는 것과 둘째로 자세한 수행절차를 확인할 수 있도록 하는 파일출력방식을 사용하였다. 하지만 이러한 파일출력방식은 선형계획법에 대한 전문적인 지식을 가지지 않은 사용자는 이해하기 어려운 내용들을 포함하고 있는 것이 사실이다. 예를 들면 구조변수 외에 여유변수가 무엇을 의미하는지를 알지 못할 것이다. 이러한 이유 때문에 사용자가 필요로 하는 결과만 선별적으로 출력해 주는 화면출력방식이 요구되었다. 또한 교육용의 목적으로서 부가적인 정보를 화면에 출력해 주는 기능 또한 필요로 하였다. 예를 들면 다양한 파일입력방식으로 입력된 문제는 이 문제가 어떤 수식형태를 가지는지 사용자는 궁금해 할 수 있을 것이다. 이러한 정보를 출력해 줌으로써 사용자가 선형계획법을 이해하는데 많은 도움이 될 거라 기대되었다. 이처럼 사용자 편의성을 고려한 기능을 설계하면서 입력과 출력의 기능에 많은 중점을 두었다. 설계되어야 할 전반적인 기능들은 다음 <표 1>과 같다.

2.2.2 시스템 설계

본 연구에서 개발한 사용자 인터페이스 시스템 설계에서 자

표 1. 사용자 편의성을 고려한 기능들

기능 분류	내용
도움말 기능	명령어에 대한 도움말
문제 입력	파일데이터를 입력 데이터의 화면입력
문제 풀이	해법에 따라 문제 풀이
입·출력정보, 부가정보 확인	원하는 입·출력 결과 확인
파라미터 설정	파라미터를 설정
부가 기능	초기해 데이터를 입력 제약식추가, 변수추가 운영체제 명령어 실행
프로그램 종료	프로그램 종료

로 입력은 파일입력방식과 수식입력방식을 이용하여 문제를 입력받도록 하였는데, 파일입력방식은 MPS 형식과 열단위 형식을 입력받고 수식입력방식은 선형계획법 문제형태와 유사하게 입력받도록 하였다. 사용자가 조정 가능한 파라미터는 각각의 해법 프로그램별로 존재하는데 LPAKO는 사전처리, 평가전략, LU분해 개선방법, 규모화, LU자료구조, 초기해 설정방법, 국면문제 생성방법, 퇴화방지방법이 있고, LPABO는 분해방법, 순서화방법, 사전처리, 규모화, 개선폭 결정방법, 종료조건 등이 있으며 LPASO는 crossover 규칙이 있다. 이러한 파라미터 변경방법을 이용하여 사용자가 선택한 다양한 방법으로 문제를 풀 수 있도록 하여 프로그램의 유연성을 증대시키도록 하였다. 문제풀이는 각각의 해법 프로그램별로 다양한 해법들이 존재하는데 이러한 해법들을 이용하여 문제를 풀 수 있도록 하여 교육적인 효과를 증대시키려 하였다. LPAKO는 원단체법과 쌍대단체법의 해법이 존재하고, LPABO는 원아핀규모화, 쌍대아핀규모화, 원쌍대아핀규모화, 원장벽, 쌍대장벽, 원쌍대장벽, 예측자수정자방법, 비가능예측자수정자방법, 다중중심수정을 이용한 비가능예측자수정자방법들이 존재한다. LPASO는 단체법방법과 내부점 방법 그리고 이를 혼합하여 사

용하도록 하는 crossover 방법이 존재한다. 마지막으로 결과출력은 파일출력과 화면출력을 이용하였는데 파일출력은 해의 결과 및 대략적인 수행절차를 확인하는 것과 자세한 수행 절차를 확인할 수 있도록 두 가지로 분류하여 출력 가능하도록 하였으며, 화면출력은 사용자가 필요로 하는 결과만을 선택적으로 출력하도록 하여 사용자 편의성을 증대시킬 수 있었다. 선형계획법의 사용자 인터페이스의 프로그램 실행과정은 <그림 1>과 같으며 선형계획법 인터페이스 시스템 구조도는 <그림 2>와 같다.

2.2.3 명령어 개발

사용자가 처음 이 소프트웨어를 사용하게 될 때 쉽게 이해하고 사용할 수 있도록 명령어를 선택하고 결정해야 한다. 일반적인 소프트웨어들이 사용하는 명령어들 중에서 보편적이고 공통적으로 사용하는 명령어를 선택하는 방법은 사용자들이 혼동하지 않고 아주 쉽게 이용할 수 있는 좋은 방법 중 하나라고 볼 수 있다.

선형계획법 프로그램의 수행과정에서 가장 중요한 것은 기본적으로 문제를 입력하는 것과 문제를 풀어 해를 구하고, 그렇게 해서 나온 결과 중에서 사용자가 원하는 정보를 쉽게 보여줄 수 있는 기능이 필수적이다. 이러한 주요 기능을 구현하는 명령어를 포함하여, 다른 명령어들에 대한 도움말을 제공하는 명령어 또한 아주 중요한 역할을 수행한다. 이런 이유로 프로그램을 처음 실행하면 사용자가 이 소프트웨어를 쉽게 사용하기 위해 도움말을 줄 수 있는 도움말 기능 명령어를 사용하도록 하는 메시지를 보여주는 것도 좋은 방법이라 할 수 있다. 본 연구에서도 이를 적용하여 사용자에게 이런 메시지를 보여주도록 하였으며 어떤 명령어를 실행하고 나면 다음 단계에 사용 가능한 명령어들을 나열해 줌으로써 사용자가 혼동하지 않고 쉽게 이용할 수 있도록 하였다. 또한 도움말 기능을 강화하여 각각의 명령어에 대한 자세한 설명을 해 줌으로써 사용자가 쉽게 사용하도록 하였다.

선형계획법 인터페이스에서 사용되는 명령어는 다음 <표 2>와 같이 정의하였다.

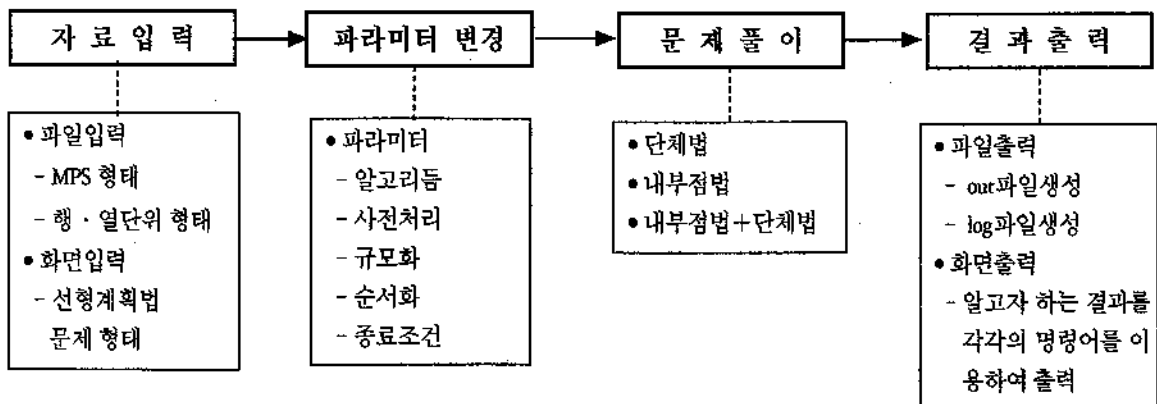


그림 1. 선형계획법 사용자 인터페이스의 프로그램 실행과정.

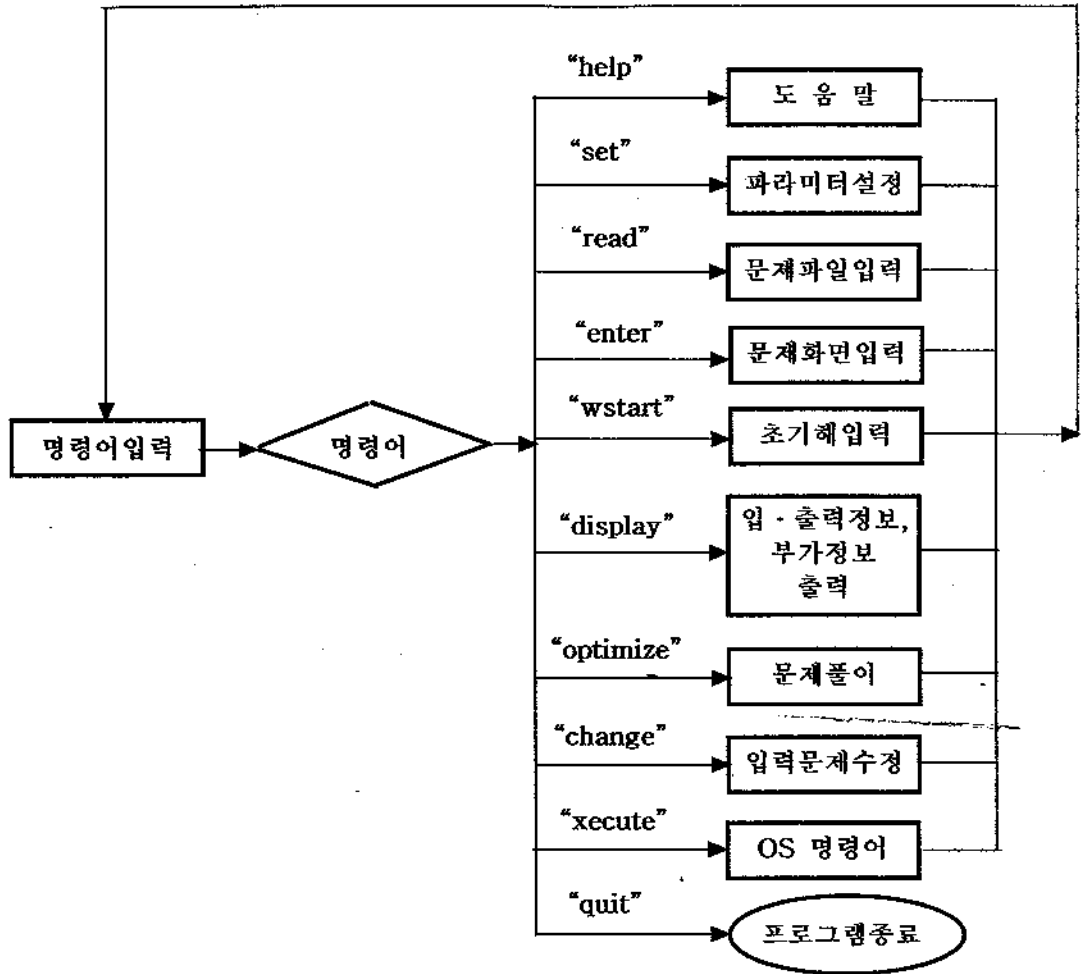


그림 2. 선형계획법 인터페이스 시스템 구조도.

표 2. 선형계획법 인터페이스의 명령어

명령어	간단한 설명
help	명령어에 대한 사용설명과 정보를 제공한다.
set	파라미터를 설정한다.
read	열단위 파일형태, MPS 파일형태의 문제를 입력받는다.
enter	화면상에서 선형계획문제의 형태로 문제를 입력받는다.
wstart	초기해 파일을 입력받는다.
display	입·출력 정보와 부가정보를 출력한다.
optimize	단체법·내부점 방법으로 문제를 푼다. (LPAKO, LPABO, LPASO)
baropt opt baropt stop	내부점법+단체법으로 문제를 푼다. (LPASO) 내부점법으로 문제를 푼다. (LPASO)
change	제약식을 추가하거나 변수의 계수를 수정한다.
xecute	선형계획법 프로그램을 실행중이면서 운영체제 명령어를 사용하도록 한다.
quit	선형계획법 프로그램을 종료한다.

(1) 도움말 가능 명령어

많은 소프트웨어들이 공통적으로 'help'라는 명령어를 사용하는데, 이 사용자 인터페이스 프로그램에서도 'help'라는 명령어를 사용한다. 이 명령어는 각각의 명령어에 대한 간단한 설명과 자세한 설명을 해주는 부분으로 대부분의 소프트웨어에서 상당히 중요하게 다루는 부분 중의 하나이다. 사용자 매뉴얼 등에서 자세한 설명이 되어있겠지만 사용자가 소프트웨어를 사용하면서 바로 편하고 쉽게 이용할 수 있는 명령어라 할 수 있다. 이러한 점 때문에 도움말 명령어는 가급적이면 예제를 중심으로 설명하는 것이 중요하다고 할 수 있다.

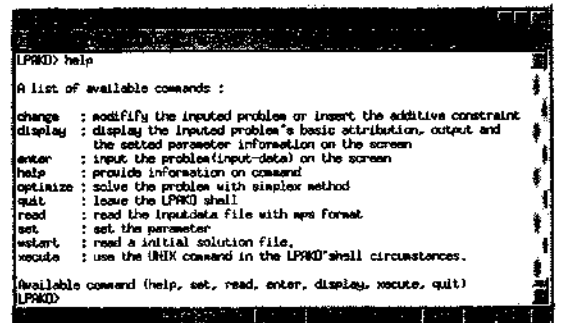


그림 3. 도움말 가능 명령어의 예(1).

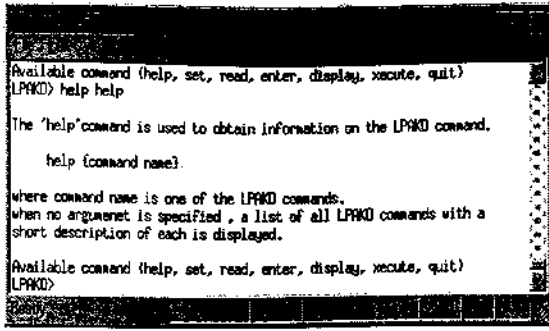


그림 4. 도움말 기능 명령어의 예(2).

(2) 문제입력 명령어

문제를 입력하는 방식으로 디스크에 저장되어 있는 데이터 파일을 읽어들이는 방식과 사용자가 직접 화면상에서 편집하며 선형계획법 문제를 입력하는 방식이 있다. 이를 간단히 말해 파일입력방식과 화면입력방식이라 한다. 파일입력방식으로는 MPS 파일을 읽어들이는 형식과 열단위 파일을 읽어들이는 형식이 있다. MPS 파일 형태는 IBM이 발표한 이래 많은 상용 프로그램들이 지원하고 있는 표준적인 파일형태이고 열단위파일형태는 제약식의 행렬을 열 위주로 비영요소만을 입력하도록 하는 파일형태이다. 화면입력방식은 수식형태를 그대로 화면상에 입력하는 방식이다. 이 방식은 구문처리를 이용하여 화면상에서 입력받은 데이터를 내부적으로 MPS 형태의 파일을 생성해 낸다. 여기서 사용자가 입력한 선형계획법 문제를 다른 선형계획법 소프트웨어에서도 사용할 수 있는 파일형태로 만드는 것은 나름대로 유용할 수가 있으므로 상용 소프트웨어에서 많이 이용되는 MPS 형태의 파일인 이런 파일을 저장하는 것도 필요로 한다. 그렇기 때문에 본 연구에서는 파라미터에 따라서 사용자가 화면-입력한 문제를 MPS 형태의 파일로 저장할 수도 있고, 하지 않을 수도 있도록 하였다. 여기서 사용자가 알아야 할 사항은 MPS 형태의 문제를 풀 때 모든 상용 소프트웨어들이 목적식의 형태가 최소화(MIN) 문제를 표

준으로 규정하고 풀기 때문에 사용자가 입력한 문제의 목적식 형태가 최대화(MAX) 문제일 때는 최소화 문제로 바꾼 상태에서 입력해야 제대로 된 MPS 형태의 파일이 생성될 수 있다. 이때의 목적함수식 값은 (-1)을 곱한 값이 원래 목적함수의 값이란 사실은 자명하다.

(3) 문제풀이 명령어

입력받은 문제를 풀게 하는 명령어로서 단체법 프로그램인 LPAKO와 내부점 방법 프로그램인 LPABO는 각각의 해법으로 문제를 풀지만, 단체법과 내부점 방법을 혼합한 방법인 LPASO에서는 단체법 방법과 내부점 방법 그리고 처음에 내부점 방법으로 문제를 풀다가 최적기저를 crossover시켜서 단체법으로 푸는 방법 이 세 가지 방법을 선택적으로 사용하여 해를 구하도록 하였다.

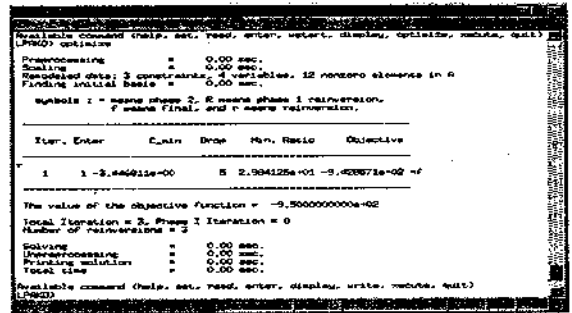


그림 7. 문제풀이 명령어의 예.

(4) 결과확인 명령어

이 명령어는 입력된 문제의 정보나 수행된 결과에 대한 정보, 그리고 부가정보들을 사용자가 필요로 하는 결과만 선택적으로 화면상에 출력하도록 하는 명령어로 이러한 명령어를 개발함으로써 사용자의 편의성을 증대시키는 데 많은 기여를 하게 되었다. 결과확인 명령어에 대한 세부적인 내용은 <표 3>과 같다.

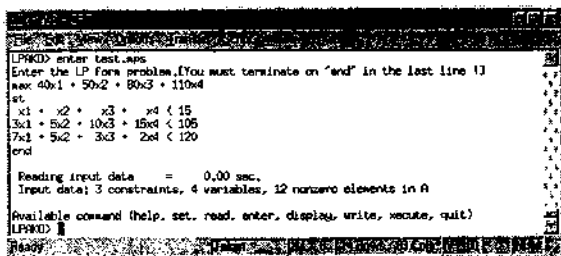


그림 5. 화면입력의 예.

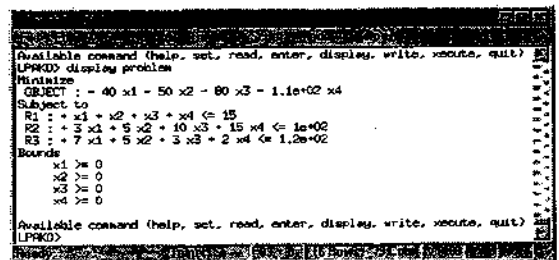


그림 8. 'problem'의 예.

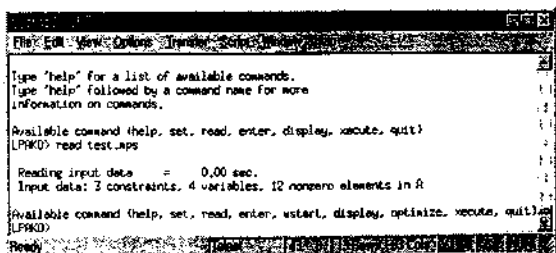


그림 6. 파일입력의 예.

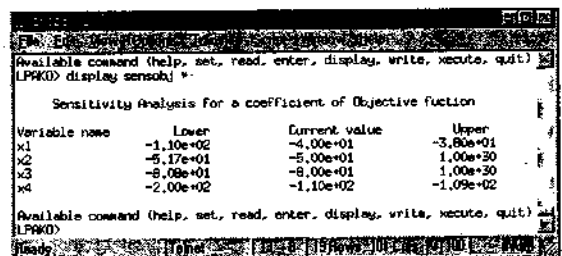


그림 9. 'sensobj'의 예.

표 3. 결과 확인 명령어와 기능

option name	기능 설명
bounds	입력된 문제의 변수한계를 보여준다.
dual	쌍대변수값을 보여준다.
constraints	입력된 문제의 제약식을 보여준다.
input	입력된 문제의 정보를 보여준다.
iteration	반복횟수를 보여준다.
name	변수 또는 제약식의 이름을 보여준다.
nzerob	B^{-1} 있는 비영요소의 개수를 보여준다.
objective	목적함수의 최적해를 보여준다.
parameter	현재 설정된 파라미터의 정보를 보여준다.
problem	입력된 문제를 선형계획법 문제 형태로 보여준다.
reduced	할인가를 보여준다.
sensobj	목적함수의 계수에 대한 감도분석의 결과를 보여준다.
sensrhs	우변상수에 대한 감도분석의 결과를 보여준다.
slacks	여유변수의 값을 보여준다.
solution	구조변수의 값을 보여준다.
stars	통계적인 정보를 보여준다.
time	문제를 푸는 데 걸린 모든 시간의 정보를 보여준다.
variable	입력된 문제의 제약식 매트릭스의 열을 보여준다.

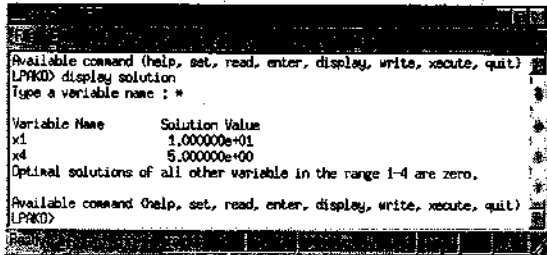


그림 10. 'solution'의 예.

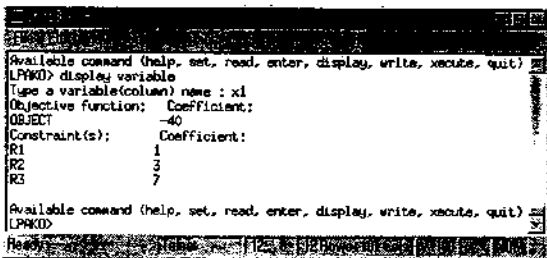


그림 11. 'variable'의 예.

(5) 파라미터 설정 명령어

단체법에서 사용하는 파라미터는 알고리즘 선택, 사전처리의 규칙, 결과출력에 대한 내용, 감도분석의 유무와 대안해 출력의 유무, 규모화의 방법, MPS 파일 입력시 사용하는 hashing 방법, 국면1 수행 방법, 초기기저 구성방법, 퇴화방지기법, 기

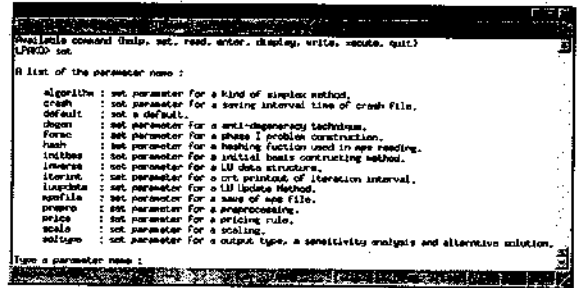


그림 12. 파라미터 설정 명령어 (단체법의 경우).

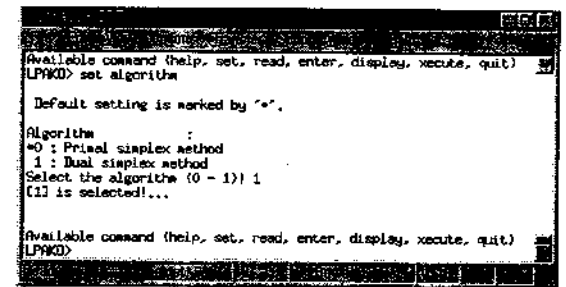


그림 13. 파라미터 설정 예 (단체법의 경우).

저역행렬의 자료구조, crash 파일의 저장시간 간격, 화면출력의 반복횟수를 조절하는 파라미터가 존재하고, 내부점 방법에서 사용하는 파라미터는 위에서 설명한 파라미터 외에도 순서화 방법, 분해방법, 개선폭의 규칙, 종료조건 등을 조절하는 파라미터가 존재한다. 이와 같은 파라미터들을 사용자가 변경할 수 있도록 하여 프로그램을 더욱 유연하게 사용할 수 있게 하였다.

(6) 부가기능 명령어

부가명령어로는 사용자가 만든 초기해 파일을 읽어들이 warm start를 실행할 수 있는 명령어나 현재의 프로그램을 실행 중에 운영체제 명령어를 사용할 수 있도록 하는 명령어, 그리고 입력된 문제에 변수나 제약식을 추가하거나 변수의 계수를 수정할 수 있는 명령어를 개발함으로써 사용자 편의성을 증대 시키는데 많은 역점을 두었다.

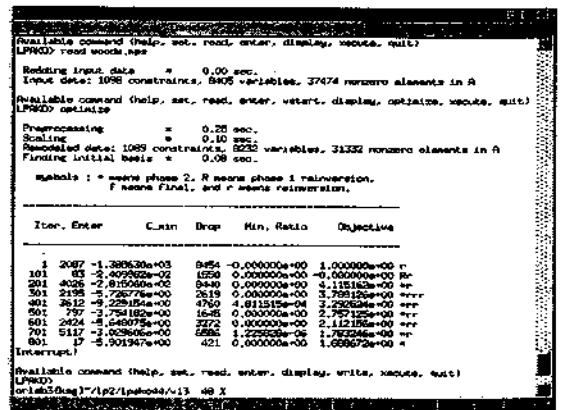


그림 14. warm start 예: 임의의 crash 상황.

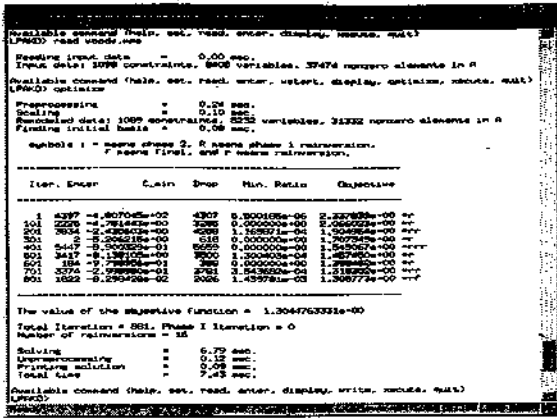


그림 15. warm start 예: 재실행한 경우.

위의 예제문제는 총 반복횟수가 1400여 회 정도 되는데 프로그램 정지(crash) 이후에 프로그램을 재실행하여 같은 문제를 풀었을 경우 계산시간과 반복횟수가 감소되었음을 볼 수가 있다.

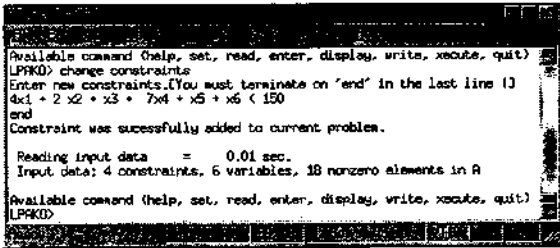


그림 16. 제약식 추가의 예.

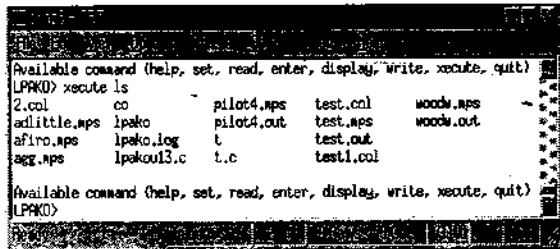


그림 17. 운영체제 명령어의 사용 예.

(7) 프로그램 종료 명령어

선형계획법 소프트웨어인 프로그램을 종료시키고자 할 때 사용하는 명령어로 모든 프로그램이 필수적으로 가지고 있는 명령어이다.

3. 화면에서의 구문처리기 설계 및 구현

선형계획법 문제의 자료를 입력하는 방법에는 프롬프트(prompt) 방식, 파일방식, 스프레드시트 방식, 수식형태 방식 등 여러 가지가 있을 수 있다(Sharda, 1992). 수식형태의 방식은 문제의 구조를 한눈에 알아볼 수 있을 뿐 아니라 선형계획법을

처음 배우는 초보자의 경우 선형계획법 문제 표현형태와 거의 비슷하게 자료를 입력하기 때문에, 쉽게 입력방법을 익힐 수 있어서 교육용 선형계획법 자료입력방법으로 유용하다. 따라서 본 연구에서는 사용자가 수식형태로 선형계획법 문제를 입력하면 입력된 내용을 MPS 형태의 파일로 변환시켜 문제를 풀 수 있도록 하기 위해 구문처리기를 개발하였다.

3.1 구문처리기의 기능 설계

수식형태 입력은 일반적인 선형계획법 문제의 정형화된 형태를 따른다. 따라서 다음과 같은 순서로 문제를 입력한다.

- [단계 1] 목적함수의 형태(max or min)를 입력한다.
- [단계 2] 목적함수식을 입력한다.
- [단계 3] 제약식 기호(st or subject to)를 입력한다.
- [단계 4] 제약식들을 입력한다.
- [단계 5] 종료기호(end)를 입력한다.

목적함수의 형태가 가장 먼저 입력된다. 일반적인 표현방법대로 최대화 문제일 경우에는 “max”를 입력하고, 최소화 문제일 경우에는 “min”을 입력한다.

목적함수의 형태를 정의한 다음에 목적함수식을 입력한다. 목적함수식은 여러 개의 항이 덧셈이나 뺄셈 기호로 연결된다. 하나의 항은 계수와 변수로 이루어진다. 계수와 변수 사이는 띄어쓰기도 하고 붙여쓰기도 한다. 구문처리기의 수식 형태에서도 이러한 규칙이 그대로 적용되었으며, 단 변수이름은 반드시 알파벳으로 시작되어야 하고 변수이름에 공백문자가 섞이면 안 되도록 하였다. 변수 이름이 숫자로 시작될 경우에 계수와 변수를 구별하는 것이 불가능해지기 때문에 반드시 알파벳으로 시작되도록 정하였다. 일반적으로 제약식에 대한 기호로 “st” 또는 “subject to”를 사용한다. 구문처리기에서는 제약식을 입력할 때에는 “st” 또는 “subject to”로 제약식 입력이 시작되었음을 알려 주도록 하였다. 선형계획법 문제의 제약식에서 부등호는 ‘≤’ 또는 ‘≥’의 형태를 가지게 되는데, 이를 실제로 입력하기가 불편하여 ‘≤’는 ‘<’로 대신하였고, ‘≥’는 ‘>’로 대신하였다. 모든 제약식은 반드시 ‘<’, ‘>’, ‘=’ 가운데 한 가지 형태를 취해야 한다. 제약식을 입력하는 요령은 목적함수식을 입력할 때와 같고, 다만 제약식 형태와 우변상수값이 입력되는 것만 다르다. 제약식 형태가 입력되는 것으로 제약식이 끝났음을 인식하기 때문에 제약식이 길어질 경우에는 여러 줄에 걸쳐서 입력할 수도 있다. 다만 한 줄에 여러 개의 제약식을 기술할 수는 없도록 하였다. 모든 제약식의 입력이 끝나면 “end”를 자료의 맨 끝에 써 주어서 선형계획법 문제 자료의 모든 입력이 완료되었음을 표시해 주어야 한다.

다음 <그림 18>은 선형계획법 문제를 구문처리기에서 사용하는 수식형태로 표현한 예이다.

선형계획법 문제	수식형태 입력 내용
Max $40x_1 + 50x_2 + 80x_3 + 110x_4$ s.t. $x_1 + x_2 + x_3 + x_4 \leq 15$ $3x_1 + 5x_2 + 10x_3 + 15x_4 \leq 105$ $7x_1 + 5x_2 + 3x_3 + 2x_4 \leq 120$ $x_j \geq 0, j = 1, \dots, 4$	max $40x_1 + 50x_2 + 80x_3 + 110x_4$ st $x_1 + x_2 + x_3 + x_4 < 15$ $3x_1 + 5x_2 + 10x_3 + 15x_4 < 105$ $7x_1 + 5x_2 + 3x_3 + 2x_4 < 120$ end

그림 18. 구문처리기에서 지원하는 수식형태의 예.

위의 예에서 보는 바와 같이 계수가 1일 경우에는 계수는 생략하고 변수 이름만을 쓸 수 있도록 하였다.

3.2 구문처리의 시스템 설계

화면 입력시 수식형태로 자료를 입력하면 구문처리가 그 내용을 해석해서 선형계획법에서 사용하는 MPS 자료 형태로 바꾼다. 다음 <그림 19>는 구문처리의 작동원리를 보여 준다.

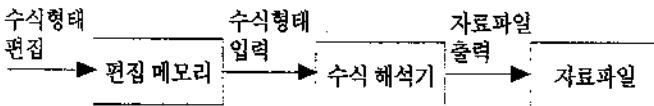


그림 19. 구문처리의 작동원리.

수식형태로 편집된 내용이 저장되어 있는 편집메모리에서 한 줄씩 구문처리의 임시 버퍼에 읽어들이 다음 구문처리가 해석을 수행한다. 전체 내용을 모두 해석한 다음에는 그 결과를 MPS 형태 파일로 출력한다.

- [단계 1] 편집 메모리로부터 수식형태의 문제를 입력받는다. 편집 메모리에 문제를 입력받는다. 이때 문제는 구문처리기에서 정의된 문법을 따라서 입력해야 한다.
- [단계 2] 편집메모리에 있는 내용을 해석한다. 구문처리는 편집메모리에서 한 줄씩 내용을 읽어들이어서 문제를 만들어 간다. 목적함수가 첫 줄에 들어가고 제약식이 그 다음에 들어간다. 마지막 줄까지 해석이 끝나면 각 열에서 비영요소의 순서를 행 번호가 작은 순서대로 정렬한다.
- [단계 3] 파일로 문제를 출력한다. MPS 형태의 파일로 출력된다. 그리고 사용자는 현재 구문처리가 만들어 낸 MPS 결과파일을 파라미터를 이용하여 생성할 수도, 하지 않을 수도 있다.

3.3 구문처리를 이용한 입력데이터 생성기능 구현

이 기능은 사용자가 화면상에서 수식형태의 선형계획법 문

제를 입력할 때 이 문제에 대한 데이터를 MPS 형태의 파일로 생성가능하게 함으로써 비전문적인 사용자가 이 실험데이터를 이용하여 다른 선형계획법 프로그램에서도 적용가능하도록 함으로써 선형계획법을 이해하는 데 많은 도움을 주도록 하였다. 이런 기능은 사용자를 위한 편의성 차원에서도 많은 도움이 될 수 있으리라 생각한다.

이런 기능을 그림을 보면서 설명하면, 먼저 사용자가 화면 상에서 수식형태의 선형계획법 문제를 입력하기 전에 MPS 형태의 파일로 저장할 것인가를 먼저 결정해야 한다.

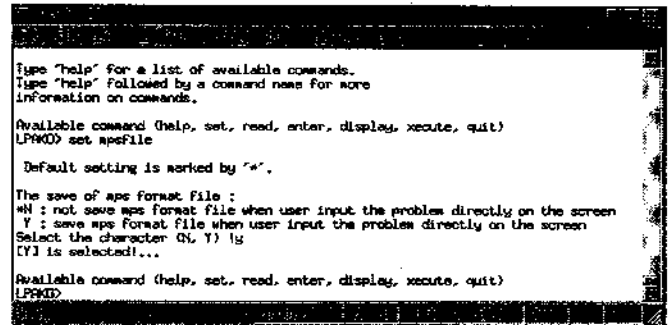


그림 20. MPS 형태의 파일로 저장유무를 설정.

<그림 20>처럼 'set'이라는 파라미터 설정 명령어를 이용하여 MPS 파일저장 파라미터를 설정한다. 그런 다음 <그림 21> 처럼 수식형태의 문제를 입력한다. 그러면 <그림 22>처럼 파일명.mps 형태의 새로운 입력데이터가 생성되는 것을 볼 수 있다. 이렇게 생성된 데이터는 세계적인 표준으로 많이 사용되고 있는 MPS 형태의 파일로 생성된다. MPS 형태의 파일

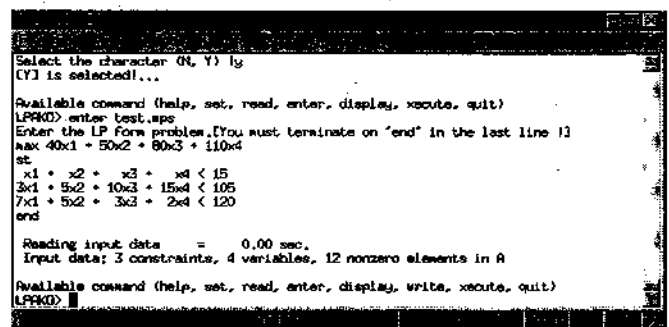


그림 21. 수식형태 선형계획법 문제를 입력.

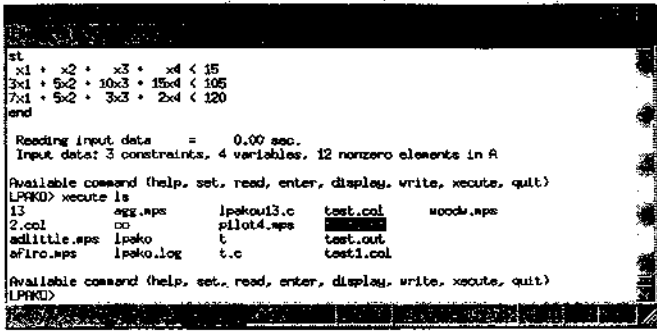


그림 22. 새로운 형태의 입력데이터 생성(test.mps).

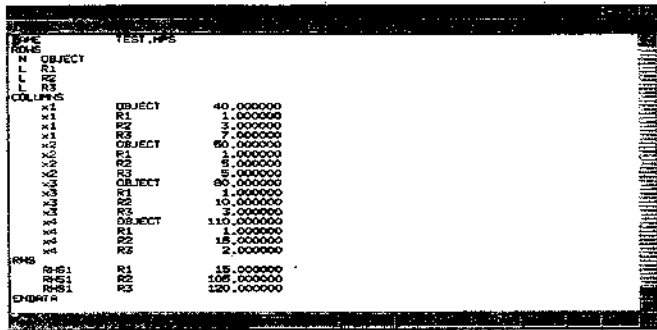


그림 23. vi 에디터로 본 MPS 형태의 파일(test.mps).

내용은 <그림 23>과 같다.

4. 인터페이스 프로그램과 타시스템과의 비교분석

4.1 프로그램 메뉴의 비교

다음 <표 4>는 본 논문에서 개발한 인터페이스 프로그램과 외부상용프로그램인 CPLEX(CPLEX Optimization Inc., 1996)와 교육용 프로그램인 LINDO(Schrage)의 기능별 명령어에 대한 유무를 비교 분석한 자료이다. 이 자료에 의하면 선형계획법이

표 4. 타시스템과의 기능 비교

종류	CPLEX	LINDO	인터페이스 프로그램
도움말	○	○	○
문제입력	○	○	○
문제풀이	○	○	○
입력정보 확인	○	○	○
출력정보 확인	○	○	○
파라미터 설정	○	○	○
입력문제 수정	○	○	○
warm start	○	×	○
운영체제 명령어 사용	○	×	○
메뉴 개수	9	7	9

가져야 하는 기본적인 기능인 문제 입·출력기능 등은 공통적으로 모두가 가지고 있다. 한 가지 주목할 만한 것은 Warm start 기능으로써 상용프로그램인 CPLEX도 구현되어 있지만, 본 연구에서는 이와 같은 기능을 구현하였을 뿐 아니라 단체법의 경우 구조변수에 대한 초기의 기저, 비기저의 정보나 내부점방법의 경우 원(primal) 문제에 대한 초기의 가능 또는 비가능해를 초기해로 사용하여 문제를 풀 수 있도록 하는 기능 또한 구현하였다.

4.2 명령어의 개수 비교

다음 <표 5>는 마찬가지로 본 연구에서 개발한 인터페이스 프로그램과 외부시스템과의 명령어 개수에 대한 비교를 한 자료이다. 명령어는 외부 상용프로그램과 거의 같은 수준의 명령어를 개발하였으며, 문제풀이 명령어에서는 단체법을 이용하여 문제를 풀 수 있는 명령어와 내부점방법을 이용하여 문제를 풀 수 있는 명령어, 그리고 두 가지 방법을 혼합한 방법의 명령어가 있는데 인터페이스 프로그램에서는 내부점방법의 경우 앞부분의 2절 2.2에서 설명하였던 원아핀규모화방법, 쌍대아핀규모화방법, 원쌍대아핀규모화방법, 원장벽법, 쌍대장벽법, 원쌍대장벽법, 예측지수정자방법, 비가능예측지수정자방법, 다중중심수정을 이용한 비가능예측지수정자방법의 알고리즘을 이용하여 문제를 풀 수 있도록 함으로써 내부점방법을 이해하는데 많은 도움을 줄 수 있도록 하였다.

표 5. 타시스템과의 명령어 개수 비교

종류	CPLEX	LINDO	인터페이스 프로그램
도움말	1	1	1
문제입력	2	3	3
문제풀이	3	1	3
입력정보 확인	8	5	8
출력정보 확인	8	4	9
파라미터 설정	단체법	10	15
	내부점		11
	기타		0
입력문제 수정	8	6	3
운영체제 명령어 사용	1	0	1
명령어 개수	57	30	54

4.3 편의성 향상

인터페이스 프로그램을 실행하여 운영할 때 현재상태에서 사용 가능한 명령어(Available command)를 나열하여 사용자에게 알려줌으로써 혼동을 피하도록 하였으며, 만일 현 상태에서 사용가능하지 않은 명령어를 사용할 때 그에 따른 메시지

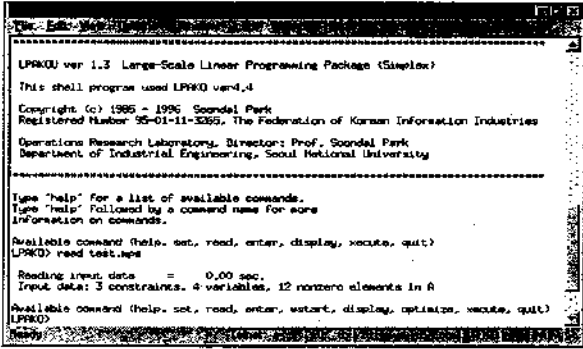


그림 24. 사용 가능한 명령어를 나타내는 예.

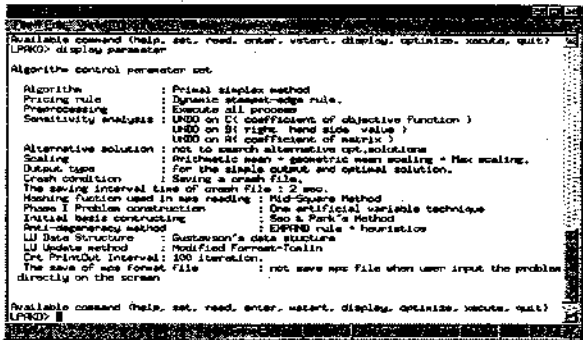


그림 25. 설정된 파라미터들을 보여주는 예.

를 보여주도록 하였다. 또 선형계획법 프로그램은 많은 파라미터들이 존재하는데 이러한 파라미터 정보를 모두 알려주도록 함으로써 현재 설정된 파라미터들을 알 수 있도록 하였다. 또한 파라미터를 설정할 때 사용자가 변경하기 싫은 파라미터는 바로 엔터키를 치면 현재 설정된 채 변하지 않게 함으로써 사용자가 쉽게 이용할 수 있도록 하였다. 다음 <그림 26>은 사용자가 알고리즘의 종류를 변경하려고 하다가 알고리즘을 변경하지 않고 현재 설정된 알고리즘을 그대로 사용하려고 바로 엔터키를 치므로써 현재 설정된 알고리즘이 변경되지 않았다는 메시지와 함께 바로 명령어를 입력받을 수 있는 프롬프트 상태로 돌아가는 예를 보여주고 있다. 이처럼 모든 파라미터를 설정함에 있어서 사용자가 변경하기 싫은 파라미터는 곧 바로 엔터키만을 치므로써 파라미터를 변경하지 않고 종료될 수 있도록 하였다.

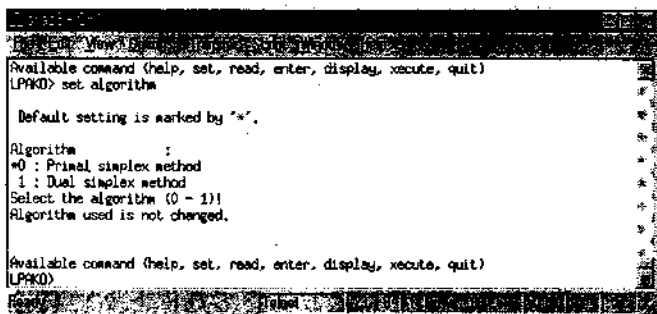


그림 26. 파라미터 설정의 예.

4.4 전문가와 비전문가를 위한 기능 구현

4.4.1 비전문가를 위한 기능 구현

선형계획법을 새롭게 배우는 학생들이나 경영학도들과 같은 비전문가를 위한 기능으로 문제입력기능 중 화면상에서 직접 수식형태의 문제를 입력가능하게 함으로써 쉽게 사용할 수 있도록 하였으며, 결과를 쉽게 이해할 수 있도록 사용자가 원하는 결과만을 출력가능하도록 하였다. 이렇게 함으로써 사용자가 필요하지 않은 정보는 출력하지 않아 혼동을 피하도록 하였다. 특히 결과 출력 명령어에서 bounds, constraints, input, iteration, name, objective, problem, solution, time, variable과 같은 명령어를 그 주요한 예로 볼 수 있으며, 이런 모든 기능에 대한 그림들은 서두에서 설명하였던 2.2를 참조하기 바란다.

4.4.2 전문가를 위한 기능 구현

선형계획법을 실무에서 사용할 때 어떤 특별한 상황으로 인해서 프로그램이 다운되는 상황과 그 문제에 대한 초기정보를 알고 있을 때 이 정보를 이용하여 문제를 풀 수 있도록 하는 Warm start 기능을 구현하였다. 이 Warm start 기능은 프로그램이 다운되는 상황과 문제의 초기정보를 알고 있을 때 문제를 효율적으로 풀 수 있도록 하는 것이다. 먼저 프로그램이 다운되는 경우를 그림과 함께 설명한다. <그림 26>은 woodw.mps 라는 문제를 풀다가 프로그램이 다운되는 상황을 연출한 것이다.

이렇게 되면 <그림 27>에서와 같이 파일명.xxx(여기서의 예는 woodw.xxx)라는 크래쉬(Crash) 파일이 생성되는 것을 볼 수가 있다. 이 크래쉬 파일은 문제를 끝까지 풀지 못하고 프로그램이 종료되었을 때 다시 똑같은 문제를 입력받아 풀게 되는 경우, 크래쉬 파일의 유무를 검사한 후에 존재하게 되면 이 파일을 이용하여 앞에 풀었던 정보를 버리지 않고 프로그램이 다운되었던 상황 이후부터 문제를 풀게 함으로써 프로그램을 효율화시켰다. 이런 기능은 전문가들을 위한 기능으로 문제의 규모가 큰 대형문제를 풀 때 사용하면 아주 효율적으로 프로그램을 사용할 수 있다.

원래 이 woodw.mps 파일의 문제를 풀면 같은 프로그램을 이용하여 풀었을 경우 반복횟수가 1400여 회 만에 종료된다. 또한 문제의 초기정보를 이용하여 문제를 푸는 경우는 단체법의 경우 초기정보를 기저정보와 비기저정보만을 이용하여 문제를 푸는 게 가능하고, 내부점 방법의 경우 초기정보는 선형계획법 문제의 임의의 구조변수 값을 이용하여 문제를 풀 수 있도록 하는 기능이다. 이런 초기해를 이용하여 문제를 푸는 예는 생략하였다. 또 한 가지 전문가를 위한 기능 중의 하나는 결과출력에서 쌍대변수값을 알려주는 dual 명령어를 비롯하여, nzerob, parameter, reduced, sensobj, sensrhs, slacks, stats 등의 명령어이다. 이런 각각의 명령어는 서두에서 설명한 2.2를 참조하기 바라며, 마지막으로 문제를 입력하고 나서 입력된 문제를 수정할 수 있는 명령어를 개발한 것이다. 이 명령어를 개발함으로써 입력된 문제에 제약식을 추가하거나, 제약식의 계수를

```

Available command (help, set, read, enter, display, xecute, quit)
LPAKO> read woodw.mps

Reading input data      = 0.00 sec.
Input data: 1098 constraints, 8405 variables, 37474 nonzero elements in A

Available command (help, set, read, enter, wstart, display, optimize, xecute, quit)
LPAKO> opt

Preprocessing          = 0.23 sec.
Scaling                = 0.10 sec.
Remodeled data: 1089 constraints, 8232 variables, 31332 nonzero elements in A
Finding initial basis  = 0.07 sec.

symbols : * means phase 2, R means phase 1 reinversion,
          f means final, and r means reinversion.

-----
Iter. Enter      C_min Drop   Min. Ratio   Objective
-----
  1  2087 -1.385630e+03  8454 -0.000000e+00  1.000000e+00 r
101   83 -2.409982e-02  1550 0.000000e+00 -0.000000e+00 Rr
201  4026 -2.815060e+02  8440 0.000000e+00  4.115162e+00 *r
301  2195 -5.726776e+00  2619 0.000000e+00  3.789126e+00 *rrr
401  3612 -9.229154e+00  4760 4.811515e-04  3.292524e+00 *rr
501   797 -3.754182e+00  1645 0.000000e+00  2.757125e+00 *rr
601  2424 -5.648075e+00  3272 0.000000e+00  2.112156e+00 *rr
701  5117 -3.029606e+00  6586 1.225828e-06  1.783246e+00 *r
801   17 -5.901947e+00   421 0.000000e+00  1.688672e+00 *r
901  5554 -8.309599e-01  5766 1.430104e-03  1.542202e+00 *rr
1001 3613 -7.160277e+00  3842 7.510439e-05  1.418751e+00 *r
Interrupt!

Available command (help, set, read, enter, display, write, xecute, quit)
LPAKO> quit
orlab3{ksg3~/lp2/lpako44/v13 24 %
    
```

그림 27. 임의로 프로그램 다운상황을 연출.

```

orlab3{ksg3~/lp2/lpako44/v13 19 % ls
13*      agg.mps      lpakout3.c  test.col    woodw.out
2.col    co*           pilot4.mps  test.mps    woodw.xxx
adlittle.mps lpako*         t*         test1.col
afiro.mps  lpako.log     t.c        woodw.mps
orlab3{ksg3~/lp2/lpako44/v13 20 %
    
```

그림 28. 프로그램이 다운되었을 때 크래쉬 파일이 생성(예: woodw.xxx).

수정하고, 우변상수의 값을 수정하여 문제를 다시 풀 수 있게 하였다. 이런 명령어의 개발은 프로그램을 더욱더 효율적으로 사용할 수 있도록 하였다.

<그림 29>는 문제를 입력한 후에 제약식을 추가로 입력한 예를 보여주고 있으며, 이렇게 입력된 문제를 풀면 <그림 30>과 같은 결과를 출력한다.

또한 제약식의 계수 수정이나 우변상수의 값을 수정하는 예는 생략하기로 한다.

5. 결론

본 연구에서는 명령어를 기반으로 한 선형계획법 프로그램을

설계하고 개발하였다. 명령어 기반 선형계획법 소프트웨어는 문제자료입력기능, 문제풀이기능, 문제의 원하는 결과를 출력하는 기능 외에 부가적인 기능들을 갖고 있다. 문제 자료입력기능은 MPS 파일과 열단위 형태 파일을 입력받을 수 있도록 하는 파일형태 입력방식과 사용자가 화면상에서 선형계획법 문제와 비슷한 형태로 입력할 수 있는 수식형태 입력방식을 사용하였다. 수식형태 입력방식을 사용하기 위해 구문처리를 개발하였으며, 이것은 사용자가 입력한 수식형태의 입력정보를 MPS 파일 형태로 변환시키는 기능을 가지고 있다. 문제풀이 기능에서는 단체법, 내부점방법 그리고 내부점방법과 단체법을 함께 사용하는 crossover 방법 등 각각의 solver에 따라서 문제를 풀 수 있도록 하였다. 결과 출력기능에서는 파일출력

```

Available command (help, set, read, enter, wstart, display, optimize, xecute, quit)
LPAKD> opt

Preprocessing      =    0.24 sec.
Scaling            =    0.10 sec.
Remodeled data: 1089 constraints, 8232 variables, 31332 nonzero elements in A
Finding initial basis =    0.07 sec.

symbols : * means phase 2, R means phase 1 reinversion,
          f means final, and r means reinversion.

-----
Iter. Enter      C_min Drop   Min. Ratio   Objective
-----
  1  5126 -2.778020e+01   5311  0.000000e+00  1.687416e+00 *r
101  120  -1.087817e+00    742  0.000000e+00  1.534269e+00 *r
201  5551 -8.703419e-01   5111  1.258001e-04  1.452581e+00 *r
301  722  -3.102427e-01   1358  0.000000e+00  1.395902e+00 *rrr
401  1884 -3.896969e-01   1680  0.000000e+00  1.329909e+00 *rr
501  1803 -9.349825e-02   1933  3.161608e-03  1.310400e+00 *rr
601   68 -2.400445e-02    270  2.372047e-02  1.304728e+00 *f

-----

The value of the objective function =  1.3044763331e+00

Total Iteration = 618, Phase I Iteration = 0
Number of reinversions = 13

Solving           =    5.15 sec.
Unpreprocessing   =    0.12 sec.
Printing solution =    0.09 sec.
Total time        =    5.77 sec.

Available command (help, set, read, enter, display, write, xecute, quit)
LPAKD>

```

그림 29. 같은 문제를 다시 풀었을 때 크래쉬 파일의 존재를 확인하고 이 정보를 이용하여 프로그램이 다운되었던 상황 이후부터 문제를 풀어내는 예.

```

01: a13 - LF
Available command (help, set, read, enter, display, xecute, quit)
LPAKD> enter test.mps
Enter the LP form problem.[You must terminate in "end" in the last line !]
max 40x1 + 50x2 + 80x3 + 110x4
st
  x1 + x2 + x3 + x4 < 15
 3x1 + 5x2 + 10x3 + 15x4 < 105
 7x1 + 5x2 + 3x3 + 2x4 < 120
end

Reading input data =    0.00 sec.
Input data: 3 constraints, 4 variables, 12 nonzero elements in A

Available command (help, set, read, enter, display, write, xecute, quit)
LPAKD> change constraint
Enter new constraints.[You must terminate on "end" in the last line !]
4x1 + 2x2 + 3x3 + 10x4 + 5x5 + 12x6 < 150
end
Constraint was successfully added to current problem.

Reading input data =    0.02 sec.
Input data: 4 constraints, 6 variables, 18 nonzero elements in A

Available command (help, set, read, enter, display, write, xecute, quit)
LPAKD>

```

그림 30. 입력된 문제에 제약식을 추가로 입력한 예.

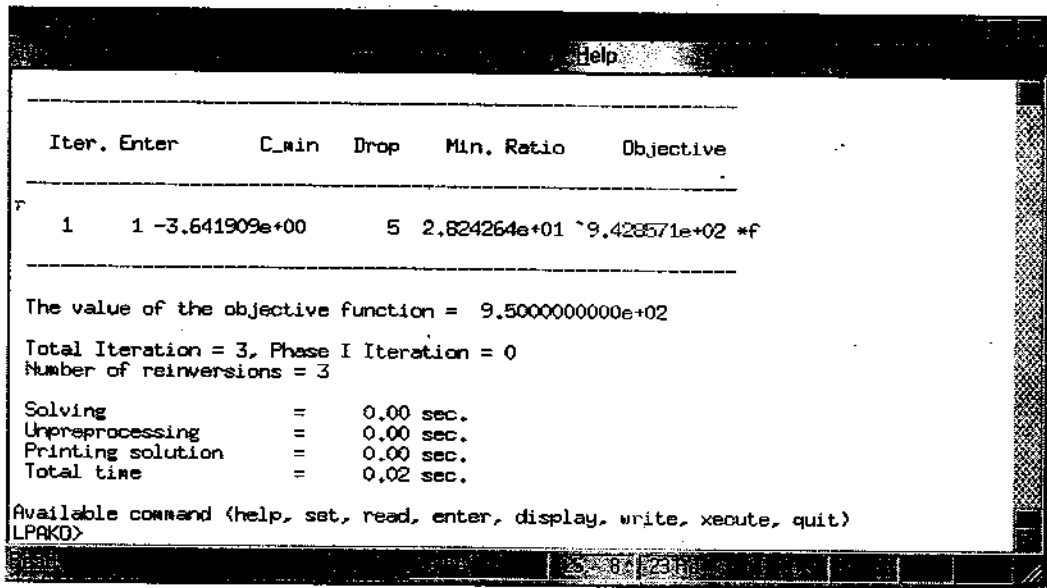


그림 31. 제약식을 추가로 입력한 후에 결과 출력.

과 화면출력을 하도록 하였다. 파일출력은 해의 결과 및 대략적인 수행절차를 확인하는 것과 자세한 수행절차를 확인할 수 있도록 두 가지로 분류하여 출력 가능하도록 하였으며, 화면 출력은 사용자가 필요로 하는 결과만을 선택적으로 출력하도록 하여 사용자 편의성을 증대시켰다. 부가적인 기능은 사용자가 임의의 초기해를 알고 있을 때 이 정보를 이용하여 문제를 풀 수 있도록 하는 warm start 기능을 구현하였으며, 입력된 문제를 풀고 나서 또 다른 제약식을 추가하거나 변수의 계수를 수정하여 문제를 풀 수 있도록 하는 기능도 구현함으로써 사용자 편의성을 증대시켰다. 마지막으로 선형계획법 소프트웨어를 사용하면서 운영체제 명령어를 사용할 수 있도록 하여 프로그램을 종료시키지 않고도 운영체제 명령어를 사용 가능하도록 하였다.

참고문헌

설동렬 (1995), 다양한 운영체제에서 선형계획법 통합환경 소프트웨어의 개발에 관한 연구, 서울대학교 공학석사 학위논문.
 CPLEX Optimization Inc (1996). CPLEX ver 4.0 사용자 매뉴얼.
 LPABO Ver 5.5 사용자 매뉴얼(1999), 서울대학교 산업공학과 경영과학 연구실.
 LPAKO Ver 4.4 사용자 매뉴얼(1999), 서울대학교 산업공학과 경영과학 연구실.
 LPASO Ver 2.5 사용자 매뉴얼(1999), 서울대학교 산업공학과 경영과학 연구실.
 Schrage, L., *Linear, Integer, and Quadratic Programming with LINDO의 사용자 매뉴얼*, 2nd ed.
 Sharda, R. (1992), Linear Programming Software for Personal Computer:1992 Survey, *OR/MS today*, 44-60.
 Yurkiewicz, J. (1988), Educational Operations Research Software:A Review, *Interface* 18, 59-76.



김상국
 한양대학교 산업공학과 학사
 서울대학교 산업공학과 석사
 현재: 현대정보기술
 관심분야: 선형계획법, 소프트웨어 사용자 인터페이스 개발



임성묵
 서울대학교 산업공학과 학사
 서울대학교 산업공학과 석사
 현재: 서울대학교 산업공학과 박사과정
 관심분야: 컴퓨터 활용, 선형계획법 및 조합 최적화



안재근
 서울대학교 산업공학과 학사
 서울대학교 산업공학과 석사
 서울대학교 산업공학과 박사
 현재: 한경대학교 컴퓨터공학과 교수
 관심분야: 네트워크 최적화 알고리즘, 선형계획법, 컴퓨터 소프트웨어 개발



박순달
 미국 University of Cincinnati 이학박사
 현재: 서울대학교 산업공학과 교수
 관심분야: Deterministic OR, 컴퓨터 활용