C언어를 사용한 최적설계 통합코드

Development of Integrated Software for Optimum Design

임 오 강* · 조 헌** · 김 성 태*** · 이 병 우****
Lim, O-Kaung Cho, Heon Kim, Sung-Tae Lee, Byung-Woo

요 약

본 연구에서는 Boland C언어를 사용하여 최적설계 수행에 필요한 작업을 쉽게 수행할 수 있는 전처리기와 후처리기를 개발하였다. 전처리기는 선택된 최적설계 알고리즘에 필요한 입력 데이터나 서브루틴 작성을 도와준다. 후처리기에서는 반복 계산중에 생성된 수치값을 그래프로 도시해 줌으로써 문제에 대한 전반적인 파악이 가능하도록 하였다. 수치예제는 선형문제와 삼부재 구조물에 대해서 제시하였다.

Abstract

A graphics system for optimum design(GOD) was developed for the various optimization programs. It is composed of a preprocessor and a postprocessor using the methods of pull-down and pop-up menus. The preprocessor of GOD system helps the designer to make a input file or a subprogram according to a selected optimization program. The postprocessor of the system display the numerical results generated during the iterative numerical analysis processes graphically in the graphic mode. Numerical examples as a mathematical linear problem and a 3-bar truss structure are presented to explain the use of GOD system. The system was programmed in one of the computer programming languages, Borland C.

Keywords : optimum design, integrated software, preprocessor, postprocessor, boland C

1. 서 론

최적화는 응용수학의 한 분야로서 수십 년 동안 많은 연구가 수행되어 왔으며 최근에는 대규모의 공학 설계의 응용에 관한 연구가 계속되고 있다. 최근 컴퓨터 기술과 수치 계산법의 발달로 수치적 최적화 방법이 광범위하게 개발되었다. 컴퓨터는 공학 설계 문제에서 다량의 계산과 자료를 빠르고, 정확하게 처리하는 것을 가능하게 해주었다. 한편, 최적화 기법을 적용할 수 있는 상용 소프트

* 정회원 · 부산대학교 기계공학과, 기계기술연구소 교수
** 국방과학연구소, 연구원
*** 현대자동차 조선해양연구소 진동소음연구실, 연구원
**** 정회원 · 동의공전 자동차과, 전임강사

• 이 논문에 대한 토론을 1996년 12월 31일까지 본 학회에 보내주시면 1997년 6월호에 그 결과를 게재하겠습니다.

웨어의 개발[1,2]에도 불구하고, 사용자는 최적설계 시 많은 어려움에 직면한다. 상용 프로그램을 이용한 최적설계 적용시 프로그램에 대한 전반적인 이해뿐만 아니라 각 프로그램마다 복잡한 입력값을 요구한다. 즉, 각 알고리즘에 관한 명확한 이해 없이는 이들 알고리즘들을 사용하는 데 여러 가지 어려움을 겪을 수 있다. 이 외에도 사용자가 각 알고리즘으로부터 계산된 수치 결과만으로 문제의 특성을 파악하거나 결과의 타당성 여부를 판단하는 것은 쉽지 않다.

최근 최적설계 분야에서는 최적화 알고리즘 자체의 개발을 위한 노력뿐만 아니라 사용자가 편리하게 사용할 수 있도록 사용자 환경 개발을 위한 노력도 병행되고 있다. 그러나 기존의 연구는 특정 최적화 알고리즘만을 이용할 수 있도록 작업환경을 구성하거나[3,4] 특정 구조물의 최적설계를 위한 작업환경을 개발[5]하는데 치중하였다.

일반적인 설계 최적화 소프트웨어는 알고리즘에 따라서 설계변수의 수, 제약 조건식의 수, 목적함수와 제약함수, 목적함수와 제약함수의 경사도를 입력해야 한다. 또한, 각기 다른 알고리즘에 필요한 매개변수도 각 알고리즘에 맞게 입력할 수 있어야 한다. 본 연구에서는 위에서 언급한 문제점을 개선하고자 최적화 알고리즘을 적용하는 작업환경을 다음과 같이 개선하였다. 첫째, 문제의 정식화 및 데이터 입력을 용이하도록 통합환경을 구성하였다. 둘째, 문제의 특성에 따라 적절한 최적설계 알고리즘을 선택할 수 있도록 통합환경 내에 다양한 알고리즘들을 구현하였다. 셋째, 문제에 대한 컴파일 및 실행 등을 통합환경 내에서 할 수 있도록 하여 사용자에게 편의를 제공하였다. 마지막으로, 각 알고리즘들에 의해 해석된 수치 결과들을 사용자가 비교.검토가 용이하도록 계산과정 중의 수치 결과를 통합환경 내에서 그래프나 도표로 가시화 하였다.

본 연구의 통합환경은 IBM호환 PC에서 볼랜드 C(boland C)로 프로그램 하였으며, 크게 문제 입력 및 실행을 위한 전처리기(preprocessor)와 결과를 표나 그래프 등으로 출력하기 위한 후처리기(postprocessor)로 구성하였다. 통합환경에서 최적설계 알고리즘은 선형 계획법과 비선형 계획법을 이렇게 해서 구성된 통합환경은 선형 계획 문제와 부재 3개로 구성된 트러스에 대해서 적용하였다.

2. C 언어를 이용한 그래픽

2.1 통합환경 구성

본 연구에서는 통합환경을 구성하기 위해 풀다운(pull down)과 팝업(pop up) 메뉴를 이용하였다[6]. 풀다운과 팝업 메뉴방식은 선택이 이루어지면 현재의 화면 위에 새로운 메뉴가 겹쳐져 나타나게 되고 선택이 이루어진 후에는 화면이 원래 상태로 되돌아간다. 풀다운과 팝업 메뉴의 차이점은 팝업 메뉴에는 종속 메뉴(submenu) 없이 오직 하나의 메뉴만이 존재하는 데 반해서 풀다운 메뉴에는 종속 메뉴가 존재한다. 풀다운과 팝업 메뉴를 이용하여 목적하는 통합환경을 구성하기 위해서는 필연적으로 컴퓨터의 화면 제어를 필요로 하게 된다.

화면 제어를 위해 비디오 램(Video RAM)에 접근하는 방식에는 ROM-BIOS를 이용하는 방법과 직접접근방식(direct access method)의 두 가지가 있다.[7] ROM-BIOS 함수 이용 방법은 INT 10H부터 INT 1AH까지를 사용하는 것이다. 이 방법은 물리적인 화면 표시 장치에 관계없이 IBM 호환의 ROM-BIOS를 가지고 있다면 작동하므로 프로그램의 이식성을 높여 준다. 그러나, 메뉴의 크기가 작다면 이 방법은 매우 빠른 실행 속도를 보이나 메뉴의 크기가 커지거나 깊어지는 경우에는 전체 프로그램의 성능이 떨어져 화면으로의 출력 속도가 느려진다. 본 연구의 통합환경은 여러 가지 기능을 갖추기 위해 메뉴 속에 다른 메뉴가 존재하여 비교적 빠른 화면 제어가 요구되므로 이 방법은 적합하지 않다. 그러므로 두 번째 방법인 리프래시 버퍼(refresh buffer)에 직접 접근하는 방식으로 통합환경을 구성하였다. 이 방식은 비디오 램을 직접 제어하므로 빠른 실행 속도를 보장한다. 하드웨어 의존적이라는 단점을 가지고 있으나 현재 대부분의 소프트웨어가 이 방식을 채택하

고 있으므로 호환성에도 문제가 되지 않는다.

2.2 그래프 구성

본 절에서는 수치 자료를 사용자가 한 눈에 그 특성을 쉽게 파악할 수 있도록 그래프로 출력하는 방법에 관하여 기술하였다.[8,9] 그래프 출력을 위해서는 자료를 읽어 들이는 부분, 그래프의 좌표축 형태를 결정하는 부분, 각 수치들이 컴퓨터 화면에 표시될 수 있도록 크기를 조정하는 자료 정규화 부분과 정규화된 수치 자료로 그래프를 그리는 부분 등이 필요하다.

자료 입력부에서는 그래프로 나타낼 수치의 전체 크기를 읽어 들인다. 이를 바탕으로 수치 자료를 저장하기 위한 변수들의 기억 장소를 확보한다. 확보된 기억 장소에 그래프로 출력할 수치들을 읽어 들인다.

좌표축 형태 결정부에서는 입력된 수치 자료의 크기에 적합한 좌표축의 눈금 간격과 각 눈금을 나타내는 치수 형식을 결정한다. 각 좌표축에 표시할 눈금의 최대 개수는 사용자에 의해 주어지고 이 범위 내에서 적절한 눈금의 개수가 프로그램 내에서 결정된다. 각 눈금을 나타내는 치수 형식은 주어진 수치가 정수형인 경우는 정수 형태 그대로 하고 실수형이나 지수형인 경우는 수치 자료의 최대값과 최소값에 상용 로그를 취하여 소수점 좌우의 자리 수를 알아내고 이것이 표기될 위치에 적절하도록 실수형이나 지수형으로 결정한다.

자료 정규화부는 그래프로 나타낼 수치들을 컴퓨터 화면에 표시될 수 있도록 일정한 비율로 크기를 조정하는 부분이다. 컴퓨터 화면 좌표는 정수 좌표계로 각 그래픽 카드마다 크기가 정해져 있다. 본 연구에서는 VGA 640×480 그래픽 카드를 사용하였다. 여기서 화면 좌표의 크기 640은 수평 방향 픽셀 개수이고 480은 수직 방향의 픽셀 개수를 나타낸다. 특정 수치들을 화면에 그래프로 나타내기 위해서는 이 값들이 화면 좌표계 내의 특정 범위에 있어야 한다. 그러나 실제의 수치들은 임의의 범위를 가질 수 있으므로 화면에 표시할 수 있는 좌표값으로 비례적으로 바꾸어 주는 것이 필요한 데 이를 정규화(normalize)라 한다. 정규화를 위해서는 특정한 비율 즉, 정규화 비율을 실제의 수치 자료에 곱해 주어야 한다. 정규화 비율을 구하기 위해서는 식 (1)과 같이 수치 자료의 최대, 최소값의 차이를 산출하여 이것으로 정규화 하고자 하는 화면 좌표 범위를 나누어주었다.

$$\text{정규화 비율} = (\text{그래프를 위한 화면 좌표 범위}) \times (\text{최대값} - \text{최소값}) \quad (1)$$

이렇게 하여 얻어지는 정규화된 수치는 다음 식과 같다.

$$\text{정규화된 수치} = \text{실제의 수치} \times \text{정규화 비율} \quad (2)$$

파일로부터 읽어 들여진 수치들은 정수형뿐만 아니라 실수형 및 지수형 등이므로 정규화 과정을 거쳐서 이 값들은 특정 범위의 정수형으로 변환된다. 이렇게 정규화된 수치는 화면 좌표계의 좌표값을 나타내며 그래프를 그릴 때 직접 사용된다.

그림 1은 본 연구에서 그래프 구성을 위해 사용된 모듈의 흐름도이다. 그림에서 Initiate Screen 부는 스크린 모드를 정의하고, 그래프와 배경 색깔을 선정한다. Find Max & Min부에서는 입력

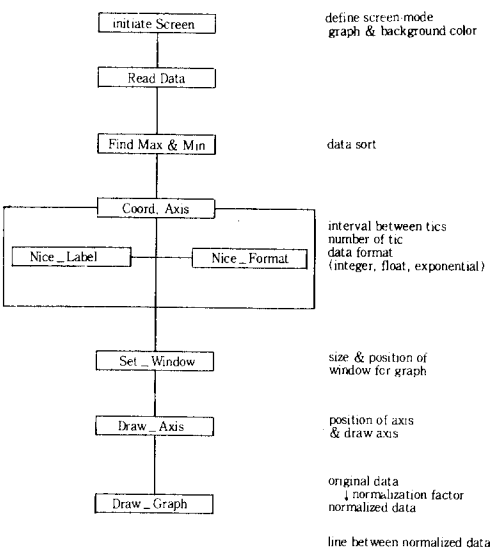

그림 1 그래프 구성을 위한 흐름도

된 데이터를 고려하여 데이터 소팅을 하게 하였다. 후처리기에서 Draw-Graph부는 그래프를 그릴 때 적절한 크기를 가지도록 데이터를 정규화해서 사용하도록 하였다.

3. GOD 시스템 구조

C언어를 이용해 구성된 통합환경을 그래픽 최적설계(graphics for optimum design)라는 뜻으로 편의상 GOD 시스템이라 명한다. GOD 시스템의 흐름도가 그림 2에 나타나 있다. 흐름도는 문제 입력을 위한 부분인 전처리기와 결과를 가시화 하기 위한 부분인 후처리기, 도움 기능 등으로 구성하였다. 프로그램을 실행했을 때 초기 화면은 그림 3에 나타내었는 데 전처리기, 후처리기, 전체 시스템의 사용에 관한 도움 기능 등을 선택할 수 있도록 구성되어 있다.

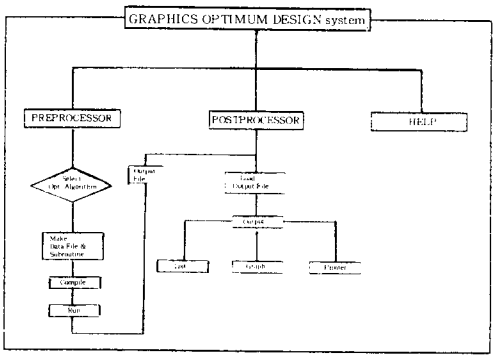

그림 2 GOD 시스템 흐름도

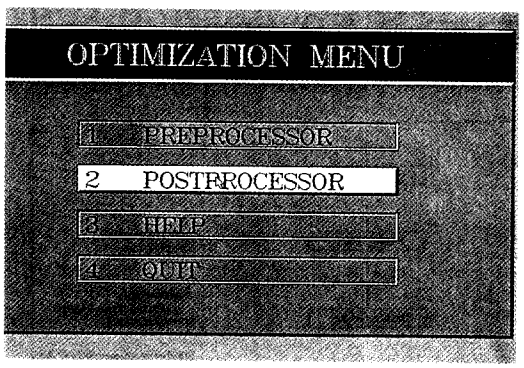

그림 3 GOD 시스템 초기화면

전처리기의 주된 기능은 사용자들이 여러 가지 최적화 알고리즘들을 사용할 수 있도록 되어있다. 일반적인 설계 최적화 소프트웨어는 알고리즘에 따라서 설계변수의 수, 제약 조건식의 수, 목적함수와 제약함수, 목적함수와 제약함수의 경사도를 입력해야 한다. 또한, 각기 다른 알고리즘에 필요한 매개변수도 각 알고리즘에 맞게 입력할 수 있어야 한다. 전처리기에서는 사용자가문제에 적합한 최적설계 알고리즘을 선택하여 각 알고리즘에 필요한 입력 파일 및 부프로그램의 작성, 프로그램의 수정, 컴파일 등의 작업을 할 때 통합환경의 도움을 받아 데이터를 입력하면 자동적으로 실행이 되도록 하였다.

후처리기의 주된 기능은 전처리 과정을 거쳐서 만들어진 방대한 양의 수치 해석 결과를 한 눈에 알아볼 수 있도록 GOD 시스템 내에서 그래프로 가시화 하는 것이다. 즉, 수치 결과보다는 그래프를 통해 문제 특성 파악과 최적설계 결과 검토를 더 용이하게 하였다.

3.1 전처리기 구조 및 기능

전처리기의 전체적 구성은 그림 4에 나타내었다.

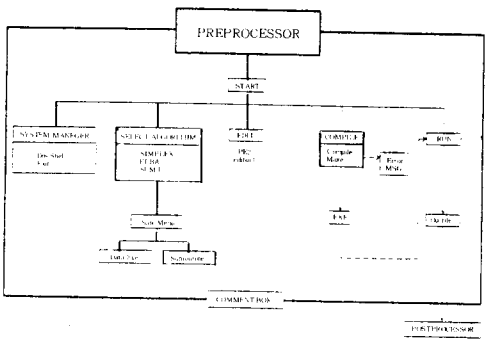

그림 4 전처리기 흐름도

그림 4에서 시스템 관리자(system manager)는 작업 환경을 GOD 시스템 환경과 도스(DOS) 환경으로 서로 전환하는 기능과 작업 종료 기능을 가지고 있다.

알고리즘 선택(select algorithm)부에서는

GOD 시스템에서 제공하는 심플렉스(simplex)법, PLBA(pshenichny-lim-belegundu-arora), SUMT(sequential unconstrained minimization technique) 등의 최적화 알고리즘들을 문제의 특성에 따라 선택할 수 있다. 이들 중 임의의 알고리즘을 선택하면 사이드 메뉴 박스(side menu box)에 여러 가지 메뉴가 나타나게 된다. 사이드 메뉴는 각 알고리즘마다 필요한 입력 파일이나 부프로그램 등을 요구한다. 이 때 사용자는 사이드 메뉴의 지시에 따라 각 알고리즘에 필요한 입력 파일과 부프로그램 등을 작성할 수 있다. 그러나, 각 알고리즘은 실행에 필요한 입력 데이터 및 매개변수들이 각기 다르므로 입력자료 작성에 실수가 생기지 않도록 해야한다. 본 소프트웨어에서는 데이터 입력창에 데이터를 입력하면 자동적으로 실행이 될 수 있도록 하였다. 이들 알고리즘에 대해서는 4장에서 설명하였다.

그림 5는 알고리즘 선택부에서 SUMT를 선택했을 때 SUMT를 실행하기 위해 필요한 파일들을 작성하도록 사이드 메뉴가 나타나고, 이 때 입력 파일을 만들기 위한 사이드 메뉴를 선택했을 때 나타나는 화면이다.

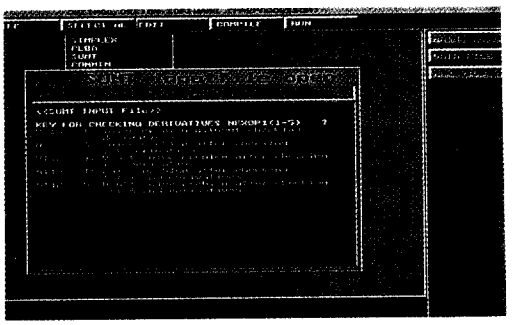

그림 5 문제 입력을 위한 사이드 메뉴

편집(edit)부는 문제 입력시 오류로 인하여 입력 파일이나 부프로그램의 일부를 수정하는 경우 상용 프로그램인 PE2 편집기로 작업 환경을 전환하도록 구성되었다. 편집 환경에서 작업이 끝나면 자동적으로 GOD 시스템으로 작업 환경을 전환하도록 하였다.

컴파일(compile)부에서는 알고리즘 선택부에서 선택되어진 최적화 알고리즘의 목적 파일과 그 알고리즘에 필요한 부프로그램들을 링크 하여 실행 파일을 만든다. 컴파일시 에러가 발생한다면 MSG 파일로 자동 저장되도록 하였다. 그림 3, 5는 앞 과정에서 만든 부프로그램을 이용하여 실행 파일을 만드는 예를 보여주고 있다.

실행(run)부에서는 컴파일부에서 만들어진 최적설계 실행 파일을 실행한다. 이 때 만들어진 최적설계 결과 파일들은 후처리기에서 사용할 수 있도록 파일로 저장된다.

3.2 후처리기 구조 및 기능

후처리기의 전체적 구성은 그림 6의 흐름도에 나타나 있다. 파일 관리자(file manager)는 전처리 과정에서 만들어진 최적설계 결과 파일을 후처리 과정으로 불러오기를 행한다. 불러오기를 선택하면 GOD 시스템은 사용자에게 불러오기가 가능한 최적설계 결과 파일들의 나열을 보여준다. 나열된 파일들 이외의 다른 파일 이름을 입력하면 경보음과 함께 다시 파일 이름을 입력할 것을 요구한다. 파일 관리부에서 파일 불러오기 기능 이외에도 전처리 과정에서와 마찬가지로 작업환경 전환 기능과 작업 종료 기능을 가지고 있다.

리스트(list)부에서는 파일 관리부에서 불러들여진 파일을 텍스트 화면에 수치 나열로 출력한다. 사용자는 이 과정에서 최적화 반복 과정 각 단계에서 발생하는 설계변수값, 목적함수값, 제약함수값 등의 수치 결과들을 볼 수 있도록 하였다.

그래프(graph)부에서는 최적화 반복 과정 각 단계에서 발생하는 수치 결과들을 그래프로 가시화 한다. GOD 시스템은 다양한 종류의 그래프를 제공하며 각 알고리즘마다 선택 가능한 그래프는 사이드 메뉴에 나타난다. 사용자가 이들 중 원하는 그래프를 사이드 메뉴에서 키보드나 마우스로 선택하면 그래프가 나타난다.

마지막으로 프린트(print)부에서는 화면상에 나타난 그래프를 출력하기 위해 그래프의 크기, 위치 등을 지정한다.

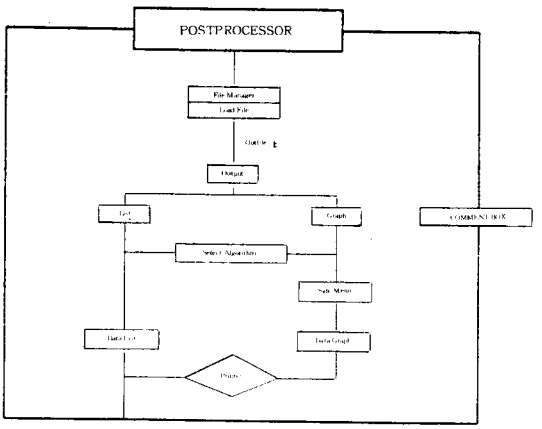

그림 6 후처리기 흐름도

4. 시스템이 제공하는 최적화 알고리즘

최적설계는 설계변수를 선정하여 적절한 제약조건들을 만족시키면서 목적함수를 최소화하는 것이다. 최적설계 문제를 풀기 위해서는 우선 문제의 정식화 과정이 요구된다. 문제에 대한 정식화가 이루어진 뒤 설계자는 문제에 적합한 최적화 알고리즘을 선택한다. 최적화 알고리즘을 사용하기 위해서는 무엇보다도 그 알고리즘의 특성을 파악하는 것이 중요하다. 최적설계 이론의 발달로 지금까지 많은 최적화 알고리즘이 개발되어 왔다. 최적화 알고리즘은 크게 선형 계획법과 비선형 계획법으로 분류된다.[10]

선형 계획법은 목적함수와 제약함수가 설계변수에 관하여 선형함수인 최적설계 문제를 풀 때 사용된다.[11] 그 중에서 대표적인 방법으로 심플렉스(Simplex)법을 선정하였는 데, 제약조건은 등호 제약조건, 부등호 제약조건(≤, ≥형) 모두가 가능하다.

비선형계획법은 목적함수와 제약함수가 설계변수에 관하여 비선형 함수인 최적설계 문제를 풀 때 사용된다. 비선형 계획법에는 원래 문제를 직접적으로 푸는 기본법(primal method)과 제약조건 문제를 비제약조건 문제로 바꾸어 푸는 변환법(transformation method)이 있다. PLBA(Pshenichny-Lim-Belegundu-Arora)는 기본법에 속하는 알고리즘으로 반복이차 계획법에서 헤시안 행렬(hessian matrix) 수정과정과 잠재집합방책(active set strategy)을 채택한 알고리즘이다.[12]

SUMT(Sequential Unconstrained Minimization Technique)는 비선형 문제를 풀기 위한 변환법으로 기본 원리는 벌칙매개변수(Penalty Parameter)를 이용하여 목적함수와 제약함수를 포함하는 복합함수를 구성하여 이 복합함수를 최소화하는 문제로 정의된다.[13] 결국 제약 최적설계 문제가 비제약 최적설계 문제로 변환된다.

5. 수치 예제

5.1 선형 계획 문제

문제 설명

GOD 시스템 내에서 심플렉스법을 적용하기 위한 예제로써 다음과 같은 선형 문제를 채택하였다[10]. 최소화하기 위한 목적함수는 다음과 같이 정의한다.

$$f = -x_1 - 2x_2 + 2x_3 \tag{3}$$

문제가 만족해야 할 제약조건들은 다음 식으로 주어진다.

$$3x_1 + 2x_2 - 2x_3 \le 12 \tag{4}$$

$$2x_1 + 3x_2 - 3x_3 \ge 6 \tag{5}$$

제약조건들을 심플렉스법에 적용할 수 있도록 변형하면 다음과 같다.

$$3x_1 + 2x_2 - 2x_3 + x_4 = 12 \tag{6}$$

$$2x_1 + 3x_2 - 3x_3 - x_5 + x_6 = 6 \tag{7}$$

여기서 x_4는 완화변수(slack variable), x_5는 부가변수(surplus variable)이며, x_6는 인위변수(artificial variable)이다. ≥형태의 부등호 제약조건이 있으므로 우선 설계점을 유용영역 이동하고 최적값을 찾는 이단 심플렉스(2-phase simplex)법을 사용하였다. 이단 심플렉스법으로 문제를 풀기 위해 인위 목적함수를 도입하면 다음과 같다.

$$w = 6-2x_1-3x_2+3x_3+x_5 \quad (8)$$

선형 문제 적용

문제 입력과정은 전처리기의 알고리즘 선택부에서 적용할 최적화 알고리즘으로 심플렉스법을 선택하면 사이드 메뉴에서 입력 파일 작성을 요구한다. 입력 파일을 작성한 후 심플렉스법의 목적파일을 컴파일 하여 실행 파일을 만든다. 실행부에서 심플렉스 알고리즘을 이용한 최적화 실행 파일을 실행하여 수치 해석 결과 파일을 만든다. 각 반복 단계에서 목적함수, 설계변수 등의 변화를 수치 해석한 결과를 GOD 시스템을 통해 화면으로 출력하였다. 그림 7은 반복회당 목적함수의 변화를 나타낸다. 목적함수는 4회 반복회 동안 최소값이 −12이었다. 그림 8과 그림 9는 각각 반복회 동안 설계변수 x_1과 x_2가 최적값으로 이동하는 과정을 나타내는 그래프이다. x_3s는 설계변수 x_1과 동일하기에 도시하지는 않았다.

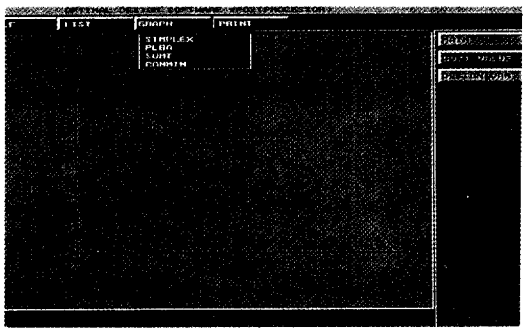

그림 7 반복회당 목적함수 변화

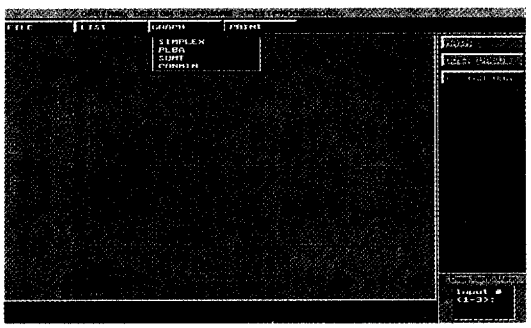

그림 8 반복회당 설계변수 1의 변화

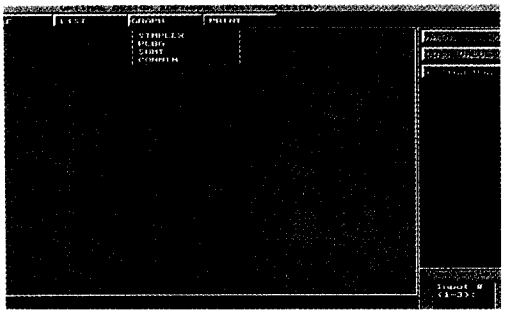

그림 9 반복회당 설계변수 2의 변화

5.2 부재 3개로 구성된 트러스

문제 설명

길이(l)이 1m인 부재 3개로 구성된 트러스가 그림 10에 나타나 있다.[14] 그림 10에서 요소수는 3개, 절점수는 4개이다. 설계 데이터는 탄성계수(E)가 70×10^9 Pa, 비중(ρ)이 2,800kg/m^{3}이다. 하중은 절점 4에 30^0의 각을 이루면서 1.0×10^5 N이 작용한다. 본 문제에 대한 정식화는 다음과 같다.

$$\text{Minimize} \quad F(A_1, A_2) = l(2\sqrt{2}A_1+A_2) \quad (9)$$

$$\text{Subject to } \Psi_1 = \frac{\sqrt{2}lP_u}{A_1E} - u_l \le 0 \quad (10)$$

$$\Psi_2 = \frac{\sqrt{2}lP_v}{E(A_1+\sqrt{2}A_2)} - v_l \le 0 \quad (11)$$

$$\Psi_3 = \frac{1}{\sqrt{2}}\left[\frac{P_u}{A_1}+\frac{P_v}{(A_1+\sqrt{2}A_2}\right] - \sigma_a \le 0 \quad (12)$$

$$\Psi_4 = \frac{\sqrt{2}P_u}{(A_1+\sqrt{2}A_2}) - \sigma_a \le 0 \quad (13)$$

$$\Psi_5 = (2\pi\omega_0)^2 - \frac{3EA_1}{[\rho l^2(4A_1+\sqrt{2}A_2]} \le 0 \quad (14)$$

$$\Psi_6 = -\frac{1}{\sqrt{2}}\left[\frac{P_u}{A_1}+\frac{P_v}{(A_1+\sqrt{2}A_2}\right] - \frac{\pi^2E\beta A_1}{2l^2} \le 0 \quad (15)$$

$$\Psi_7 = -\frac{\sqrt{2}P_u}{(A_1+\sqrt{2}A_2)} - \frac{\pi^2 E\beta A_2}{l^2} \le 0 \qquad (16)$$

$$\Psi_8 = -\frac{1}{\sqrt{2}}\left[\frac{P_v}{(A_1+\sqrt{2}A_2)} - \frac{P_u}{A_1}\right] - \frac{\pi^2 E\beta A_1}{2l^2} \le 0 \qquad (17)$$

and the bounds $A_1, A_2 \ge A_{min}$ (18)

식 (9)에서 F는 최소화되어질 스칼라 양으로 구조물의 부피이다. A_1과 A_2는 설계변수 벡터로 부재의 단면적인데, A_1은 부재 1과 부재 3의 단면적이 같도록 설정하였다. 식 (10)과 (11)은 각각 절점 4의 수평, 수직 방향의 변위 제한조건식이다. 허용 변위인 u_1과 v_1은 각각 0.005m의 값을 설정하였다. 식 (12)와 (13)은 각각 부재 1과 2에 대한 응력제한식인데 부재 3은 θ가 각도 0°와 90° 사이에서 주어지므로 부재 1의 응력이 부재 3의 응력보다 항상 크므로 고려하지 않았다. 식 (14)는 전체 구조물의 고유 진동수 제한식으로 특정 진동수 ω_0보다 커야 하는 식이다. 문제에서 ω_0는 50 Hz로 설정하였다. 식 (15)에서 (17)은 압축력이 작용했을 경우 좌굴에 견디기 위한 제한조건식인데, 여기서 형상계수 $\beta = I/A^2$이고 I는 단면 2차 모멘트이다. 형상계수 β는 1.0으로 선정하였다. 마지막으로 설계변수 A는 설계시 항상 양의 특정 크기 이상이어야 하는 식이다.

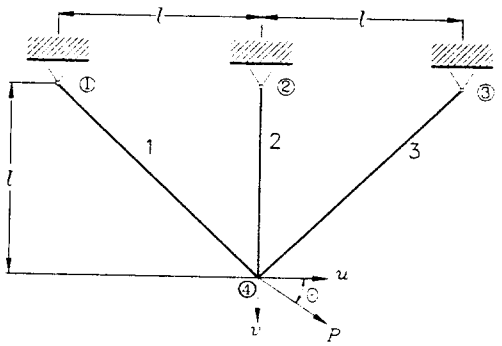

그림 10 부재 3개로 구성된 트러스

부재 3개로 구성된 트러스 문제는 비선형 계획 문제이다. 본 문제는 GOD에서 제공하는 최적화 알고리즘 중 PLBA와 SUMT를 이용하였다. 그 과정은 다음과 같다. 문제 입력을 위하여 GOD 시스템의 초기 화면에서 전처리기로 들어간다. 먼저 알고리즘 선택부에서 PLBA나 SUMT를 선택한다. 알고리즘이 선택되면 사이드 메뉴 박스에 각 알고리즘에 필요한 입력값과 부프로그램을 만들기 위한 메뉴가 나타난다. 이 경우에는 PLBA나 SUMT는 입력 파일과 부프로그램 작성을 요구하는 사이드 메뉴가 나타난다. 입력 파일 작성부에서 선택된 알고리즘에 필요한 정보 즉, 설계변수의 개수, 제약함수의 개수, 설계변수의 범위, 매개변수들의 정보 등을 입력한다. 부프로그램 작성부에서 SUMT의 경우 목적함수와 제약함수를 입력하기 위한 부프로그램을 작성하고 PLBA의 경우에는 목적함수와 제약함수 외에도 설계변수에 대한 목적함수와 제약함수의 미분식을 입력하기 위한 부프로그램을 작성한다. 컴파일부에서 앞 과정에서 작성한 부프로그램을 컴파일하여 GOD 시스템에서 제공하는 PLBA나 SUMT의 목적 파일과 링크하여 최적설계 실행 파일을 만든다. 실행부에서는 컴파일부에서 만들어진 실행 파일을 실행하여 수치 해석 결과 파일을 만든다.

이것으로 전처리 과정은 끝나고 수치 결과를 보기 위해 다시 GOD 시스템의 초기 상태에서 후처리기로 작업환경을 전환한다. 파일 관리부에서는 전처리기의 실행부에서 만들어진 최적설계 결과 파일을 실행을 불러온다. 리스트부에서 결과 파일을 직접 텍스트 화면에서 볼 수 있으며 그래프부에서는 통합환경 내에 3-부재 트러스 문제의 수치 결과를 그래프로 출력한다.

PLBA를 이용한 해석 결과

PLBA 알고리즘에서 사용한 매개변수는 벌칙 매개변수가 1.0, 활성화 제한조건식의 폭은 0.1, 최적해의 정확도는 0.0001, 선탐색의 정확도는 1.0×10^{-5}로 택하였다. 결과는 후처리기에서 표나 그래프로 볼 수 있다. 그림 11은 PLBA로써 최적설계를 수행했을 경우 반복횟수당 목적함수의 변

화를 나타내는 그림이다. 반복횟수 19회에서 목적함수값이 $0.1925\times10^{-2}m^3$로 수렴하였다.

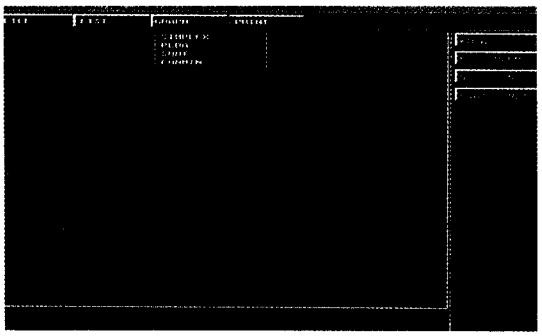

그림 11 반복회당 목적함수 변화

그림 12는 설계변수 A_1이 최적해로 수렴하는 과정을 나타낸 그림이다. 설계변수 A_1는 $0.6099\times10^{-3}m^2$로 수렴하였으며, 설계변수의 초기치를 더 작게 잡는다면 반복횟수가 줄어들 것으로 평가된다.

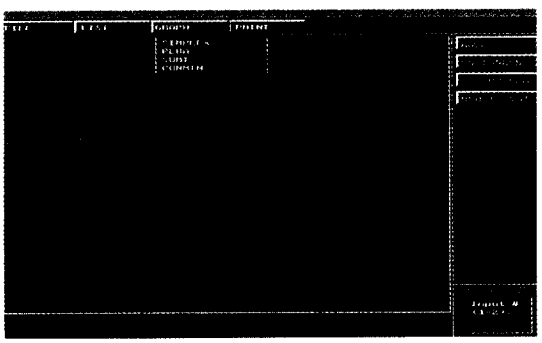

그림 12 반복회당 설계변수 A_1 변화

그림 13은 제약함수 Ψ_4가 반복회당 변화 과정을 나타내는 그래프이다. 수치 해석 전 단계에서 제약함수 4는 수치 해석 사용자에 의해 주어진 ε범위 $(-0.1\sim0.1)$에 있으므로 ε-활성화(ε-active) 제약조건임을 알 수 있다. 그러므로 제약조건 4는 최적설계시 설계 방향에 영향이 있는 것으로 평가할 수 있다.

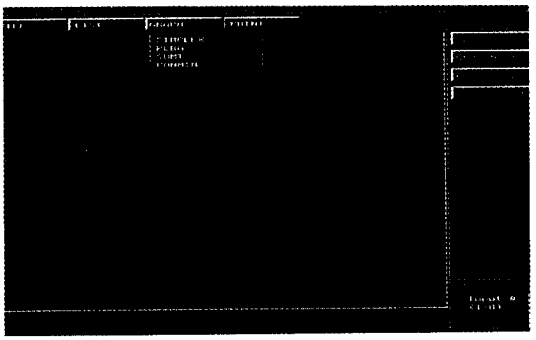

그림 13 반복회당 제약조건 Ψ_4 변화

SUMT를 이용한 해석 결과

SUMT에서 사용한 매개변수는 수렴 정확도를 1.0×10^{-5}, 최적치 정확도를 1.0×10^{-4}, 설계변수의 초기치를 각각 $0.1m^2$, $0.1m^2$로 설정하였다. 해석 결과 46회 반복회 동안 목적함수는 $0.1927\times10^{-2}m^3$에 수렴하였으며, 설계변수 값은 A_1이 $0.6086\times$

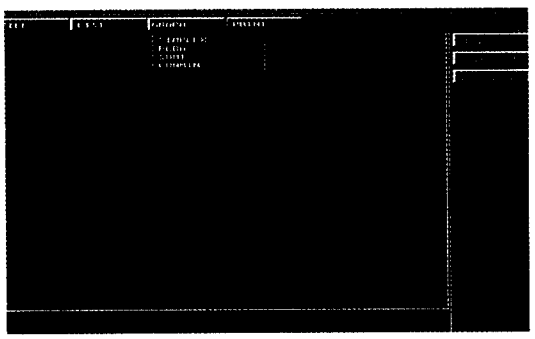

그림 14 반복회당 목적함수 변화

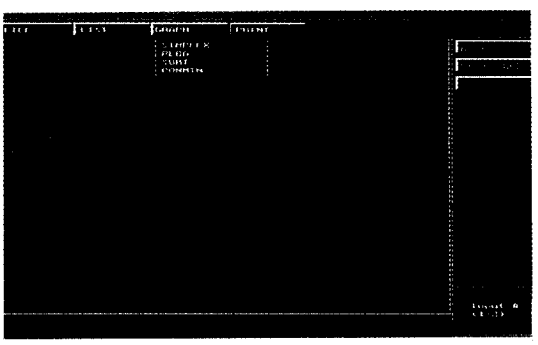

그림 15 반복회당 설계변수 A_1 변화

$10^{-3}m^2$ A_2가 $0.2056\times10^{-3}m^2$이었다. 최적해에서 Ψ_1, Ψ_3, Ψ_4의 제한 조건식이 활성화되었다.

SUMT에 의한 해석 결과를 GOD 시스템의 화면상에서 그래프로 출력하였다. 그림 14는 SUMT에서 반복횟수당 목적함수의 변화를 나타내는 그래프이다. 그림 15는 설계변수 A_1이 반복 과정에서 최적해로 수렴하는 과정을 나타낸 그래프이다.

6. 결 론

최적설계 기법을 공학 설계에 적용할 때 문제의 정식화 및 결과 해석 및 검토의 어려움, 여러 가지 알고리즘의 차이 등으로 자주 어려움을 겪는다. 상기의 문제점을 극복하기 위하여 본 연구에서는 사용자가 편리하게 사용할 수 있도록 사용자 환경 개발하였다. 개발된 최적설계 통합 환경을 IBM호환 PC에서 볼랜드 C언어를 사용하여 간단한 문제에 대해서 적용한 결과를 아래에 정리하였다.

1) 일반적인 설계 최적화 소프트웨어는 알고리즘에 따라서 설계변수의 수, 제약 조건식의 수, 목적함수와 제약함수, 목적함수와 제약함수의 경사도를 입력해야 한다. 또한, 각기 다른 알고리즘에 필요한 매개변수도 각 알고리즘에 맞게 입력할 수 있어야 한다. GOD 시스템은 문제에 적합한 최적화 알고리즘을 선택하면 정식화 및 실행 과정을 도와준다.

2) GOD 시스템에서는 최적화 프로그램이 수행되는 동안 각 반복 과정에서 발생하는 설계변수값, 목적함수값, 제약함수값 등의 수치 결과들을 그래프로 가시화 한다. 사용자는 가시화된 그래프를 통해 문제 정식화시 발생하기 쉬운 오류의 발견 및 문제 특성 파악이 용이하다.

3) 몇 가지 최적화 알고리즘들이 GOD 시스템 내에서 구현되고 있으므로 문제 특성에 따른 각 알고리즘의 효율성(efficiency), 정확성(accuracy), 신뢰성(reliability) 등을 파악할 수 있다.

4) 수치 결과의 그래픽 처리는 각 알고리즘의 매개변수값을 조정하여 수치 해석을 수행하였을 때 매개변수가 최적설계 결과에 미치는 영향을 쉽게 파악할 수 있게 해준다. 사용자는 이를 바탕으로 각 알고리즘마다 타당한 매개변수값을 정할 수 있다.

GOD 시스템은 앞에서 언급한 장점이외에 PC의 기본 기억용량 제한과 처리속도가 문제가 된다. 대규모 최적설계 문제를 PC에서 수행하기 위해서는 컴퓨터 기억장소를 사용하지 않고, 데이터베이스 파일을 사용하는 등의 일련의 연구과정이 동반되어야 한다.

참 고 문 헌

1. J. S. Arora and C. H. Tseng, IDESIGN Program (above Version 3.5), The University of Iowa, Iowa City, Iowa 52242, U.S.A., 1986.
2. G. N. Vanderplaats, ADS-A Fortran Program for Automated Design Synthesis Ver. 2.01, Engineering Design Optimization, Inc. 1275 Camino Rio Verde Santa Barbara, CA 9311, 1987.
3. Santos, J. L. T., Godse, M. M. and Chang, K. H., "An Interactive Post-Processor for Structural Design Sensitivity Analysis and Optimization: Sensitivity Display and What-If Study," Computers & Structures, Vol. 35, No. 1, pp. 1-13, 1990.
4. Saouma, V. E. and Sikiotis, E. S., "Interactive Graphics Nonlinear Constrained Optimization," Computers & Structures, Vol. 21, No. 4, pp. 759-769, 1985.
5. Adeli, A. and Fiedorek, J., "A Microcad System for Design of Steel Connections-I," Computers & Structures, Vol. 24, No. 2, pp. 281-294, 1986.
6. 황희융, C 프로그램 이렇게 짠다, 교학사, 1991.
7. 신동준, 박진감 있는 신동준의 IBM-PC 내부, 기전출판사, 1991.
8. 임인건, 터보C 정복, 가남사, 1992.
9. Hearn, D. and Baker, M. P., Computer Graphics, Prentice-Hall, 1986.
10. Arora, J. S., Introduction to Optimum Design, McGraw-Hill, Inc., 1989.

11. Luenberger, D. G., Linear and Nonlinear Programming, Addison-Wesley, 1984.
12. Lim, O. K. and Arora, J. S., "An Active Set RQP Algorithm for Optimal Design," Computer Methods in Applied Mechanics and Engineering, Vol. 57, No. 10, pp. 51-65, 1986.
13. Kuester, J. L. and Mize, J. H., Optimum Techniques with Fortran, McGraw-Hill. Inc., 1973.
14. Haug, E. J. and Arora, J. S., Applied Optimal Design, Wiley Interscience, 1979.

(접수일자 : 1996. 6. 15)