

구조 설계 프로세스의 분산운동

Distributed Operation of Structural Design Process

황진하* 박종회** 김경일***
Hwang, Jin-Ha Park, Jong-Hoi Kim, Kyung-III

ABSTRACT

Distributed operation of overall structural design process, by which product and process optimization are implemented, is presented in this paper. The database-interconnected multilevel hybrid method, in which the conventional design method and the optimal design approach are combined, is utilized there. The method selectively takes the accustomed procedure of the conventional method in the conceptual framework of the optimal design. Design conditions are divided into primary and secondary criteria. This staged application of design conditions reduces the computational burden for large complex optimization problems. Two kinds of numeric and graphic processes, are simultaneously implemented on the basis of concurrent engineering concepts in the distributed environment of PC networks. Numerical computation on server and graphic works on independent client are communicated through message passing. Numerical design is based on the optimization methodology and the drawing process is carried out by AutoCAD using the AutoLISP programming language. The prototype design experimentation for some steel trusses shows the validity and usability of the method. This study has sufficient adaptability and expandability in methodology, in that it is based on general theory and industry standard systems.

1. 서론

오늘날의 복합적 시장환경은 공학도에게 빠르고 유연한 대응을 압박하고 있다. 즉, 개발 및 생산기간의 단축과 코스트 절감은 물론 고객 감동의 품질을 동시에 요구하고 있다. 이러한 요구에 대한 응답 활동인 공학설계는 다양한 영역과 문제해결 방법들 및 지식 표현형태를 다루며 마지막 설계의 결과는 많은 부분이 설계도로 표현된다. 오늘날의 정보처리기술은 공학설계의 수준을 한 차원 높였으며 CAD/ CAM/ CAE의 이론과 기술은 수학적 모델에 기초하여 정교한 수치해석, 최적화, 시뮬레이션과 그래픽을 구현케 한다. 그러나 절차면에서 일련의 순차적인 과정으로 이루어지고 하나의 단계가 끝난 후 다음 단계로 이행되어 사람과 시간 등 특히 의사결정 피드백에 많은 낭비가 발생하였다. 이러한 문제점은 최근 엔지니어링 업무와 관련 시스템을 이끄는 새로운 패러다임으로 부상하고 있는 동시공학적 접근방법으로 상당 부분 개선할 수 있다. CE (concurrent engineering)에 대한 정의는 다양하나 간단히 정의하면, 제조 및 관련 지원 부분을 포함한 전 공정을 라이프사이클에 걸친 모든 요소를 고려하여 통합 병행토록 설계하는 체계적 접근방법이며 정보화와 자

* 정회원 · 충북대학교 구조시스템공학과 교수

** 정회원 · 충북대학교 구조시스템공학과 강사

*** 정회원 · 충북대학교 토목공학과 박사수료

동화를 통하여 개발시간의 단축, 비용 절감 및 품질 향상 등 생산성을 극대화하려는 것이다. 이러한 것은 분산운용을 통하여 전체 설계의 유연성을 높이고 각 단계에서 발생하는 오류를 조기에 검증, 수정할 수 있다. 고속화와 관련된 컴퓨터 구조 또는 조직 측면에서 발전은 다양한 형식의 병렬컴퓨터와 분산 처리운용을 들 수 있다. 이러한 방법은 기존의 단일 프로세서로 구현할 수 있는 연산속도의 기술적 한계를 극복하기 위해 다수의 프로세서 또는 컴퓨터를 묶어 성능향상을 도출해내는 방법이다. 이 중 분산처리(distributed processing)는 컴퓨터들을 하나의 네트워크로 엮어 커다란 작업을 각 컴퓨터에서 나누어 수행하는 것으로 병렬컴퓨터처럼 고가가 아닌 기존의 장비를 활용하여 추가비용 없이 대형의 문제를 최소의 비용으로 풀 수 있다는 장점이 있다.

본 연구의 목표는 구조설계프로젝트를 수치적모델로 정형화하고 동시공학적 개념에 입각하여 그 결과물인 프로토타입 뿐만 아니라 해석/ 재해석/ 민감도해석/ 최적화/ 도면작성 등 설계과정 전체 프로세스를 분산 최적화함으로써 설계의 유연성과 생산성을 제고하는 데 있다. 즉 프로세스와 프로토타입을 분산운용하여 각 단계에서 발생하는 오류를 조기에 검증, 수정할 수 있도록 하고, 각 과정의 수치계산 또한 다수의 컴퓨터에 분산처리하므로써 설계시간의 단축 및 비용 절감 등에 있어 획기적 효과를 이루어 내도록 한다. 본 연구는 PC 네트워크 상에서 분산처리 기법과 데이터베이스에 기초하여 공학설계를 구성하는 설계계산과 도면작성 및 DB 연동을 동시에 병행함으로써 기존에 순차적으로 이루어지던 설계과정을 동시공학적 절차로 구현하고, 예제를 통해서 타당성 및 효율성을 검증하였다.

2. 설계 프로세스 및 최적화

2.1 설계프로세스 분산

공학설계는 수치연산, 기호추론 및 그래픽 처리 등을 포함하는 복합적 창조활동이다. 전체 과정은 요구분석 및 문제정의로부터 상세치수와 도면 및 시방 등을 포함하는 실시설계에 이르기까지 반복되는 재해석-평가-재설계 과정을 갖는다. 이들 단위과정들은 반복적 순환을 통해 검증 및 수정되며 전통적으로 연속적으로 수행되었으나, 최근 정보인프라를 바탕으로 병렬화를 지향하고 있다. 실행 측면에서 전체설계프로세스는 다른 단계에 놓여있는 다양한 서브테스크 또는 서브시스템들의 실행을 요구한다. 따라서 설계에 대한 시스템적 해결전략은 전체 설계작업과 요소들을 어떻게 계획하고 분할할 것인지 또, 어떻게 서브테스크 사이의 관계를 표현, 조율할 것인지를 대상으로 한다. 설계문제는 분석이나 진단과 다르게 단일입력/ 다중출력 형식의 의사결정문제로서 방대한 의사결정공간에서 하나의 최적해를 찾는다. 시스템설계문제의 설계의사 결정공간에서 최적의 시스템을 정의하는 최적해를 찾기 위한 방법의 하나가 최적화이다. 설계최적화는 요구된 설계조건을 만족시키면서 최대의 경제성을 갖는 구조계를 도출하는 과정으로 설계문제를 하나의 수학적 문제로 형성하고 체계적으로 접근한다. 대형시스템을 최적화하는 문제는 대단히 많은 설계변수와 제한조건을 갖기 때문에 전산처리에 상당한 메모리와 시간 및 많은 비용을 요구하게 된다. 이러한 어려움은 분산처리를 통해 나누어 처리할 수 있으며 분산처리란 컴퓨터들을 네트워크로 엮어 하나의 작업을 각 컴퓨터에서 나누어 수행하도록 하는 것을 말한다. 분산처리는 특히, 대형 프로젝트에 관련된 하나 이상의 프로그램이 여러 개의 프로세스로 나누어져서 전체 목적을 달성하기 위해 각자의 일을 수행한다.

2.2 분산운용체계 및 구성

개별메모리형 MIMD 형태의 병렬컴퓨터를 기본 모델로 하여 설계된 PVM(Parallel Virtual Machine)은 서로 다른 기종의 컴퓨터들을 엮어 하나의 가상적인 병렬형 컴퓨터처럼 동작하게 해주는 일종의 가상 운용체계이며 이기종(heterogeneous)간의 자료전달 및 동기화 등의 기능을 제공하여 분산처리를 가능하게 해주는 정보교환 환경을 구성해준다. 컴퓨터간의 정보교환은 3단계의 절차를 거쳐 수행되는데, 메시지를 송신할 경우에는 먼저 버퍼를 초기화하고 데이터를 버퍼에 묶은 다음 버퍼의 데이터를 특정위치로 보낸다. 메시지를 수신하는 컴퓨터에서는 먼저 버퍼를 할당하고 데이터를 받아서 버퍼에 저장한 다음 버퍼로부터 데이터를 푸는 절차를 갖는다. 분산처리는 호스트 유무에 따라 두 개의 주된 프로그래밍 모델을 갖는데, 본 연구에서는

주종모델(master-slave)에 기초하며 마스터프로세서는 통제프로그램으로서의 역할뿐만 아니라 슬레이브프로세서와 같이 실제의 계산과정도 수행한다. 즉, 네트워크로 연결된 PC 컴퓨터를 사용하여 수치치리를 병행하는 서버와 그래픽처리를 담당하는 그래픽스테이션으로 구성된다. 서버는 구조해석과 최적화 루틴을 반복 수행하면서 그 결과를 수치로 그래픽스테이션에 전달한다. 변경된 설계변수값을 받은 그래픽스테이션은 설계변수값을 받아 모니터를 통해 구조물의 형상과 함께 각 부재에 대한 설계변수값 즉, 부재 단면적이 변화하는 값을 실시간으로 제공한다. 각 프로세서는 각각의 명령어에 대해서 자신의 독립된 제어흐름에 따라 연산을 수행한다. 따라서 프로세서간의 통신 및 데이터의 동기화는 프로그래머가 담당하게 된다. 또한 각 프로세서가 자신만이 접근할 수 있는 로컬 메모리를 통해 연산을 수행하고 프로세서간의 동기화는 메시지를 주고받음으로써 이루어진다.

Fig. 1은 가상 분산환경에서의 시스템운용도를 보여주며, 여기에서 실제 적용 및 구현한 부분은 실선박스 안의 설계엔지니어링 부분으로 구조계산과 도면작성을 포함한다. 물론 나머지 부분도 TCP/IP 프로토콜을 이용하는 인터넷 환경에서 동일하게 운용할 수 있다.

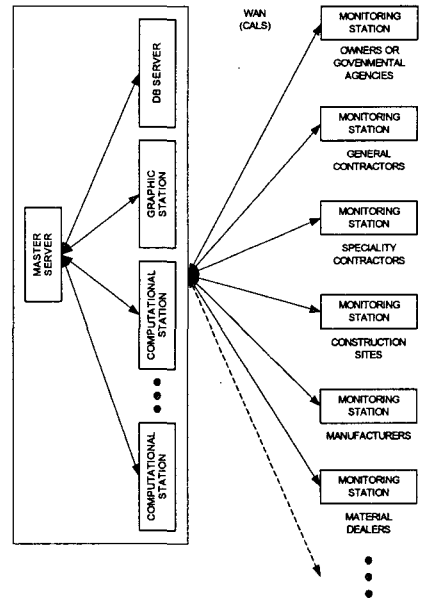


Fig. 1 Conceptual diagram for distributed systems operation

3. 다단계 혼성설계

3.1 설계 방법 및 절차

구조 설계는 계획, 예비설계, 해석, 검토, 재설계와 같은 과정을 반복하면서 이루어지고 매 단계마다 많은 입출력 설계정보를 처리해야 한다. 대부분의 실제 설계에서, 가능한 부분부터 대량으로 생산되는 표준공장제품으로 규격화 할 때 총비용 면에서 많은 효과가 있으며, 따라서 설계과정의 정보관리를 효율적으로 하기 위해서 DBMS(데이터베이스관리시스템)을 이용하는 것이 효과적이다. DB를 중핵으로 관행적설계와 최적화설계를 결합하는 새로운 다단계 혼성설계방법을 구현하는 다단계 혼성방법의 주요 설계절차를 요약하면 다음과 같다. 먼저, 문제에서 설정된 주 규준을 제한조건으로 연속최적화를 수행한다. 이때의 설계변수는 단면적이거나 단면2차모멘트 등의 중간변수를 취한다. 다음, 연동된 데이터베이스로부터 계산된 단면값이 속하는 표준상업단면그룹을 검색, 가용 상위단면들을 나열한다. 마지막으로 부차적 규준에 따라 후보단면들을 순서대로 검토 선택한다. 설계를 지배하는 제한조건을 다루는데 있어 관행적 설계방식과 같이 주 조건과 부 조건으로 나누어 단계적으로 접근함으로써 복합 조건들을 갖는 대형최적화문제의 계산적 부담을 줄인다. 일반적으로, 전체 구조물의 중량을 목표함수로 하는 구조최적화 문제는 다음과 같이 형성될 수 있다.

$$\text{Minimize : } F(X) = \sum_{i=1}^I \rho_i L_i X_i \quad (1)$$

$$\text{Subject to : } G_j(X) \leq 0, \quad j=1,2,\dots,J \quad (2)$$

$$X_i^L \leq X_i \leq X_i^U, \quad i=1,2,\dots,I \quad (3)$$

여기에서 식(1)은 목표함수이고, 식(2)는 구조물의 거동을 제한하는 조건식으로 변위, 응력 등에 대한 제한 조건을 나타내며, 식(3)은 설계변수인 단면에 대한 제한조건이다. $F(X)$ 는 전체 구조물의 총 중량, ρ_i 는 각 부재의 단위중량을 나타내며, L_i 와 X_i 는 각각 부재의 길이와 단면적을 나타낸다. 또한 G_j 는 설계제한조건을 의미하고, X_i^L 과 X_i^U 는 각각 설계변수인 단면적의 하한치와 상한치를 표시한다. 최적화 알고리즘은 ALM(Augmented Lagrange Multiplier)을 이용하였다.

본 연구에서 최적화알고리즘으로 사용한 ALM방법은 제한조건의 문제를 비제한조건의 문제로 변환하여 푸는 변환법의 하나로 라그랑지 승수를 개선하여 최적해에 도달하는 방법이다. 우선, 주어진 모든 부등식제한조건들을 등식제한조건들로 바꾸기 위해 완화변수를 도입하여서 제한조건식들을 변환하면, 확장 라그랑지 함수는 다음과 같이 나타낼 수 있으며

$$A(X, \lambda, Z, r_p) = F(X) + \sum_{j=1}^m [\lambda_j (G_j(X) + Z_j^2) + r_p (G_j(X) + Z_j^2)^2] \quad (4)$$

여기서, λ 는 라그랑지 승수이고, r_p 는 벌칙매개변수이다. 이는 다음과 같은 유사목표함수로 변환할 수 있다.

$$A(X, \lambda, r_p) = F(X) + \sum_{j=1}^m [\lambda_j \psi_j + r_p \psi_j^2] \quad (5)$$

$$\psi_j = \max \left[G_j(X), \frac{-\lambda_j}{2r_p} \right] \quad (6)$$

따라서, 각 부분이나 과정을 연결하는 수학적인 공유기반을 부구조화에 두고, 유사목표함수 $A(X, \lambda, r_p)$ 을 최소화하는 비제한최적화문제로 형성된다.

3.2 DB 연동 및 2차조건 검토

본 연구에서는 MS SQL Server를 DB서버로 하는 C/S(Client-Server) 환경을 기반으로 설계시스템을 구성하고, 임의의 클라이언트에서 원격데이터에 접근하는 설계 애플리케이션을 개발한다. C/S환경에서 후단(Back-End) 서버에는 데이터베이스와 내장 프로시저 등을 작성하여 프로그래밍 환경을 구성하고 전단(Front-End) 클라이언트에 사용자 인터페이스와 함께 데이터 액세스 및 애플리케이션 로직을 작성해야한다. 여기서 데이터베이스 서버는 다수의 클라이언트 사이에서 리소스를 효율적으로 관리하고, 클라이언트 애플리케이션은 사용자와 대화하며 SQL을 통해 서버 DBMS의 데이터 원본에 논리를 수행한다. 이때, DBMS는 결과만을 데이터소스로 되돌리고 프로파일을 구성한다. 마이크로소프트사가 개발한 표준데이터베이스 인터페이스인 ODBC(Open Database Connectivity)는 클라이언트/서버 환경에서 여러 종류의 서로 다른 원격 데이터베이스 원본에 동일한 방법으로 접근할 수 있도록 한다. ODBC는 일종의 표준화된 API로 DBMS에 독립적이어서 일관된 DB 프로그래밍을 가능케 한다. 이를 통해 네트워크 DB서버의 데이터소스에 연결하려면 네가지의 기본 객체와 OpenDatabase와 OpenRecordset의 두 가지 메소드를 사용한다. 자료 조작은 보통 SQL 명령문을 사용, ODBC드라이버 매니저를 통해 RDBMS에 전달된다. 응용프로그램과 연동된 데이터베이스의 데이터원도우를 보여주며 시스템은 연동뿐 아니라 자체적으로 입·출력 수정 및 검색기능을 갖는 전단 기능을 갖는다. 부제가 압축력을 받을 경우의 허용응력은 좌굴을 고려한다. 본 연구는 AISC매뉴얼에서 제공하는 도표화된 세장비에 따른 허용응력을 DB에 내장하고 SQL 서버는 ODBC를 통해 애플리케이션의 연속최적화 설계값을 넘는 단면요구에 적합한 단면들을 오름차순으로 제공한다. 다시 애플리케이션은 아래와 같은 2차 설계조건을 검토하고 반복 시행여부를 결정한다. 또한, 선택된 단면의 상세치수는 국부좌굴에 대해 폭/두께 비를 만족해야 한다. 따라서 DB를 통해 선택된 후보 이산 표준단면에 대해 관련 조건들을 검토한다.

3.3 수치/그래픽 변환 및 제도

본 연구에서 그래픽스테이션은 동시에 두 개의 프로그램을 실행하는데 이것은 데이터를 재구성하기 위해 Fortran으로 작성한 자료변환 프로그램과, AutoCAD를 실행하도록 AutoLISP으로 작성한 그래픽처리 프로그램이다. 그래픽처리 프로그램은 변환기에서 생성된 데이터파일을 이용해 AutoCAD 화면으로 그래픽을 구현한다. 변환 프로그램의 경우, 초기의 설계변수값을 토대로 데이터파일을 생성하고, 이후에 최적화 루틴의 결과로 발생한 새로운 설계변수값을 받을 때마다 기존에 생성한 데이터 파일을 수정하는 작업을 한다. 동시에 실행되는 그래픽처리 프로그램은 변환기와는 독립적으로 출력파일을 읽어들이 이를 분석해서 AutoCAD 화면에 그래픽 작업을 반복한다.

AutoCAD의 도면 데이터베이스는 DXF 코드로 이루어져있으며 AutoLISP을 통해 도면의 모든 요소에 대한 DXF 코드를 검색 및 수정할 수 있다. 이러한 관점에서 도면 개개의 요소는 도면개체(drawing entity)로 표현되며 각 도면개체에 대한 DXF 코드는 개체의 종류에 따른 몇 개의 코드 번호와 코드 번호에 따른 고유한 코드값으로 이루어진다. 이 코드값은 alphanumeric 레코드로 이루어져서 이 레코드의 인위적인 변경에 의해 AutoCAD를 통한 그래픽 이미지가 바뀌어 나타내어질 수 있다. 즉, AutoLISP은 AutoCAD의 데이터베이스 레코드를 취하여 연관리스트(association list)로 재구성하고 그래픽 이미지를 화면에 표현한다. 기호 데이터를 처리하기 위한 간단하고 강력한 프로그래밍 언어인 LISP은 인공지능분야에서 널리 사용되고 있다. 많은 그래픽 처리함수를 내장하고 AutoCAD의 기능을 확장하여 특정 응용프로그램에 대한 제도용 패키지를 만들 수 있도록 Autodesk사가 개발한 AutoLISP 루틴은 그 기능과 속도로 해서 생산성의 향상과 비용의 절감을 가져다준다. 특히 반복적이고 오류가 발생하기 쉬운 작업에 효과적이다.

4. 설계지향 해석

4.1 설계 민감도해석

설계민감도해석은 설계 변화량에 대한 거동량들의 변화량을 산정하는 것으로 설계도함수를 구하는 방법에는 크게 미분법과 변분법이 있으나 이산형 구조에는 주로 음함수 미분법이 적용되며, 이것은 다시 수치적 방법과 해석적 방법으로 나누어진다. 이들은 효율성 면에서 후자가 선호되나, 설계함수들이 설계변수의 비선형 음함수로 표현되는 구조최적화에서 때로 어렵거나 불가능한 반면, 전자는 시간이 많이 걸리기는 하나 어느 문제에나 쉽게 다루어질 수 있는 양면이 있다. 부구조화에 기초한 구조해석은 전체 구조시스템을 여러 개의 작은 부구조로 분할하여 시스템방정식을 경계와 내부에 관계된 양들로 나누어 구성한다.

구조시스템의 시스템방정식을

$$KU = P \quad (7)$$

이라 할 때, s번째 부구조에 대한 시스템방정식은 다음과 같다.

$$\begin{bmatrix} K_{bb}^{(s)} & K_{bi}^{(s)} \\ K_{ib}^{(s)} & K_{ii}^{(s)} \end{bmatrix} \begin{bmatrix} U_b^{(s)} \\ U_i^{(s)} \end{bmatrix} = \begin{bmatrix} P_b^{(s)} \\ P_i^{(s)} \end{bmatrix} \quad (8)$$

여기서 아래첨자 b와 i는 각각 경계와 내부자유도에 관계된 양들을 의미하고, s는 부구조 번호이다. 이 식의 응축과정을 거쳐 경계에 대한 평형방정식이 구해진다. 부구조화에 기초한 설계도함수를 얻는 과정은 다음과 같다. 먼저 목표함수 및 설계조건을

$$\Phi(X, U_b, U_i) \quad (9)$$

로 나타내고 변수 X , U_b 와 U_i 들의 변화량에 대한 해당함수의 전미분을 취하여

$$d\Phi = \frac{\partial \Phi}{\partial X} dX + \frac{\partial \Phi}{\partial U_b} dU_b + \frac{\partial \Phi}{\partial U_i} dU_i \quad (10)$$

을 얻을 수 있다. 이 식의 우변을 설계변수의 변화량 dX 항으로 바꾸기 위하여 먼저 부구조의 평형방정식을 전개하여 다음 선형 근사식을 얻는다.

$$K_{bb}dU_b + K_{bi}dU_i = C_1 dX \quad (11)$$

$$K_{ib}dU_b + K_{ii}dU_i = C_2 dX \quad (12)$$

여기서,

$$C_1 = \frac{\partial R_b}{\partial X} - \frac{\partial}{\partial X}(K_{bb}U_b) - \frac{\partial}{\partial X}(K_{bi}U_i) \quad (13)$$

$$C_2 = \frac{\partial R_i}{\partial X} - \frac{\partial}{\partial X}(K_{ib}U_b) - \frac{\partial}{\partial X}(K_{ii}U_i)$$

식 (11), (12)에서 dU_i 를 소거하면

$$K_b dU_b = C dX, \quad C = C_1 + q^T C_2 \quad (14)$$

를 얻고 마지막으로 식 (12)의 dU_i 와 식 (14)의 dU_b 를 식 (10)에 대입하면 설계변수의 변화량에 대한 목표함수 및 거동제한함수의 변화량을 얻을 수 있다.

$$d\phi = \Lambda^T dX \quad (15)$$

$$\Lambda = \frac{\partial \phi^T}{\partial X} + C^T \lambda_i + C^T \lambda_b \quad (16)$$

이 식의 Λ 는 설계민감도벡터를 나타내며 λ_i 와 λ_b 는 강성도매트릭스 K_{ii} , K_b 의 역을 효율적으로 계산하기 위하여 도입된 수반변수로서 내부변위와 관계된 응축량을 이용하면 다음 관계식의 해로 정의된다.

$$K_b \lambda_b = \frac{\partial \phi^T}{\partial U_b} + q^T \frac{\partial \phi^T}{\partial U_i} \quad (17)$$

$$K_{ii} \lambda_i = \frac{\partial \phi^T}{\partial U_i} \quad (18)$$

4.2 근사 구조재해석

구조 재해석은 구조최적화과정에서 설계변수가 수정될 때, 새로운 정해석 대신에 초기의 해석정보값을 이용하여 수정된 구조의 거동량을 산정하는 것으로, 설계변화량을 고려한 식(7)의 시스템방정식은 다음과 같이 다시 쓸 수 있다.

$$(K^* + \Delta K)(U^* + \Delta U) = P \quad (19)$$

여기서, ΔK 는 구조 강성도매트릭스의 변화량, ΔU 은 변위벡터의 변화량 그리고 *은 전단계의 해석값을 나타낸다. 즉, 시스템방정식 (7)을 사용하여 전체구조의 거동량을 산정하는 대신 전단계 해석정보값인 $\{U^*\}$ 를 이용하여 새로운 거동량을 구한다. 근사재해석 방법으로 보존근사화와 차원축소법을 결합한 합성근사화를 사용하며, 보존근사화(conservative approximation)는 다음 식으로 표현된다.

$$g_c = g^* + \sum_{j=1}^{N^*} G_j (X_j - X_j^*) \left(\frac{\partial g^*}{\partial X_j} \right) \quad (20)$$

$$G_j = \begin{cases} 1 & \text{if } X_j^* (\partial g^* / \partial X_j) \geq 0 \\ \frac{X_j^*}{X_j} & \text{otherwise} \end{cases} \quad (21)$$

차원축소법은 기저벡터들로 이루어진 축소된 부공간에서 거동량을 산출하는 방법으로, 초기 해석정보값들 중 독립된 n개의 기저벡터를 선형결합하여 근사변위 U를 구한다. 제시된 합성 근사화 접근방법은 보존근사화의 각 항들을 차원축소법의 기저벡터로 선정하는 것으로, 각 부구조의 내부변위 $\{U_i\}$ 와 경계변위 $\{U_b\}$ 는

$$\begin{Bmatrix} U_i \\ \dots \\ U_b \end{Bmatrix} \cong \begin{Bmatrix} C_i^* \\ \dots \\ C_b^* \end{Bmatrix} \{u\} \quad (22)$$

에 의해 구해지며, 미정계수벡터 $\{u\}$ 는 다음의 축소된 시스템방정식을 풀어 결정된다.

$$[K_c] \{U_c\} = \{P_c\} \quad (23)$$

$$[K_c] = \sum_{i=1}^S [K_{c_i}], \quad \{P_c\} = \sum_{i=1}^S \{P_{c_i}\} \quad (24)$$

각 부구조에 대한 $[K_{c_i}] \{P_{c_i}\}$ 는 다음 식으로 정의된다.

$$[K_{c_i}] = [C_i^*]^T [K_{ii}] [C_i^*] + [C_b^*]^T [K_{ib}] [C_i^*] + [C_i^*]^T [K_{ib}] [C_b^*] + [C_b^*]^T [K_{bb}] [C_b^*] \quad (25)$$

$$\{P_{c_i}\} = [C_i^*]^T \{P_i\} + [C_b^*]^T \{P_b\} \quad (26)$$

식(13)의 기저벡터 C_i^* 와 C_b^* 는 $X_j^* (\partial U / \partial X_j) \geq 0$ 인 경우,

$$[C_b^*] = \left[\{U_b^*\}, \sum_{j=1}^{N^*} \left(\frac{X_j^*}{X_j} \right) \left(\frac{\partial U_b^*}{\partial X_j} \right) (X_j - X_j^*) \right] \quad (27)$$

$$[C_j^*] = \left[\{U^*\}, \sum_{i=1}^{N_d} \left(\frac{X_i^*}{X_j} \right) \left(\frac{\partial U_i^*}{\partial X_j} \right) (X_j - X_i^*) \right] \quad (28)$$

$X_j^*(\partial U/\partial X_j) < 0$ 인 경우,

$$[C_j^*] = \left[\{U^*\}, \sum_{i=1}^{N_d} \left(\frac{\partial U_i^*}{\partial X_j} \right) (X_j - X_i^*) \right] \quad (29)$$

$$[C_j^*] = \left[\{U^*\}, \sum_{i=1}^{N_d} \left(\frac{\partial U_i^*}{\partial X_j} \right) (X_j - X_i^*) \right], \quad j = 1, 2, 3, \dots, N_d \quad (30)$$

으로 구성된다. 위 식에서 N_d 는 설계변수의 수를 나타낸다.

5. 적용 및 분석

5.1 200부재 트러스 모델

150개의 자유도와 200개의 부재를 갖는 표준 200부재 트러스(Arora, 1997)에 대한 연속최적화 결과는 Fig. 2 및 Fig. 3과 같다(모델, 설계조건은 참고문헌과 동일). 또한 2차 설계규준을 만족하는 최종 상용단면은 Table 1에 문헌의 3가지 방법(GADSS, SADSS 및 BBMDSS)의 결과값과 비교하여 나타내었다.

Table 1 AISC Sections for 200 Bar Truss

Group No.	Present	GADSS	SADSS	BBMDSS
1	W 6×9	W 6×15	W 6×15	W 6×15
2	W 6×9	W 8×10	W 12×14	W 6×15
3	W 6×6	W 6×9	W 10×12	W 6×15
4	W 6×9	W 12×16	W 10×12	W 6×15
5	W 6×9	W 12×14	W 6×9	W 6×15
6	W 6×9	W 6×12	W 16×26	W 6×15
7	W 6×9	W 6×15	W 12×26	W 6×15
8	W 6×9	W 8×10	W 14×22	W 6×15
9	W 6×9	W 8×10	W 8×10	W 6×15
10	W 6×9	W 4×13	W 8×10	W 6×15
11	W 6×9	W 6×9	W 14×22	W 6×15
12	W 6×9	W 8×18	W 14×22	W 6×15
13	W 6×9	W 4×13	W 12×14	W 6×15
14	W 6×9	W 6×12	W 14×22	W 6×15
15	W 10×12	W 8×15	W 12×16	W 6×15
16	W 6×9	W 8×13	W 4×13	W 6×15
17	W 6×9	W 6×25	W 16×26	W 6×15
18	W 4×13	W 8×21	W 5×19	W 8×28
19	W 6×9	W 5×19	W 10×12	W 6×15
20	W 6×15	W 10×26	W 10×22	W 8×28
21	W 6×9	W 8×10	W 8×15	W 6×15
22	W 6×9	W 6×25	W 12×26	W 6×15
23	W 5×19	W 18×35	W 12×35	W 8×40
24	W 6×9	W 6×12	W 6×16	W 8×24
25	W 8×21	W 21×50	W 18×35	W 8×40
26	W 6×9	W 8×10	W 12×16	W 6×15
27	W 10×12	W 6×25	W 8×18	W 6×15
28	W 6×20	W 10×54	W 12×53	W 6×25
29	W 12×30	W 8×48	W 16×57	W 12×65
Cost	7.950	12.7437	15.2540	13.6797

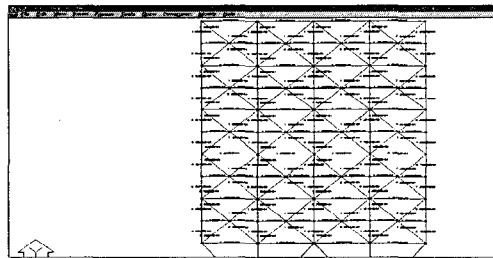


Fig. 2 AutoCAD presentation by AutoLISP in distributed environment(200-bar truss)

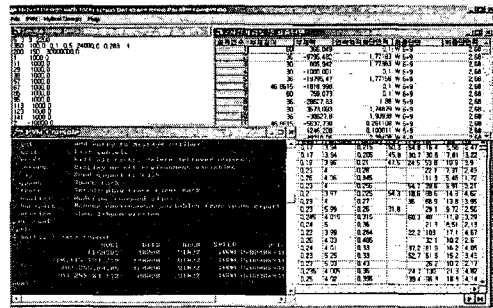


Fig. 3 Distributed Operation of Structural Design Process

5.2 1208부재 트러스 모델

1208개의 부재를 갖는 표준 1208부재 트러스(Schmit, 1994)에 대한 모델은 Fig. 4와 같고 연속최적화 결과는 Table 2 및 Table 3과 같다(모델, 설계조건은 참고문헌과 동일).

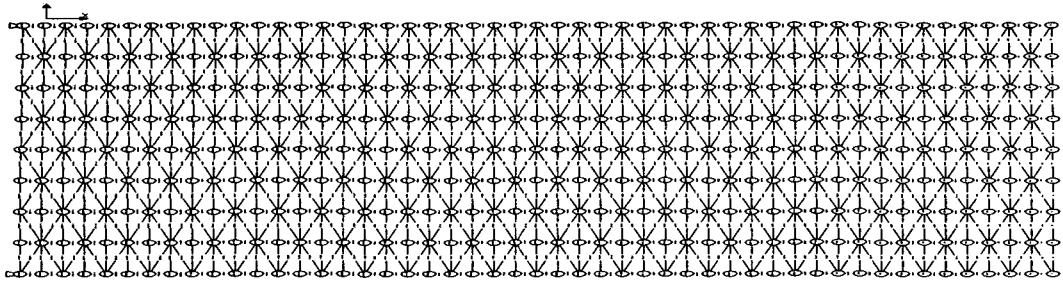


Fig. 4 1208-bar plane truss

Table 2. 알고리즘별 목표함수

Data Algorithm	Objective function (lb)
Method I	1.107E03
Method II	1.070E03
Present	1.070E03
Reference	1.038E03

Table 3. 최적화 수행결과

Design Variable	Value (in ²)
1	3.02616E-01
2	1.95494E-01
3	1.89311E-01
4	1.69755E-01
5	2.99567E-01
6	3.11661E-01
7	3.61560E-01
8	4.61320E-01
9	4.60955E-01
10	4.39068E-01
11	4.88925E-01
12	5.36085E-01
13	5.82208E-01
14	6.28374E-01
15	6.74385E-01
16	7.20808E-01
17	7.66510E-01
18	8.12701E-01
19	8.59146E-01
20	9.06595E-01
21	9.58341E-01
22	1.02896E+00
23	1.18713E+00
24	1.73350E+00

6. 결론

정보화와 자동화의 흐름 속에 개발시간의 단축, 비용절감 및 품질향상 등을 통한 생산성 극대화가 요구되는 시장 환경에서 이에 해당하는 공학적 패러다임으로 동시병행공학이 유용하게 활용되고 있다. 본 연구는 PC 네트워크 상에서 분산운용 및 처리 기법을 이용하여 공학설계를 구성하는 수치설계작업과 도면 작성을 동시에 병행함으로써 기존에 순차적으로 이루어지던 설계과정을 동시공학적 절차로 구현하고, 구조설계모델을 통해서 연구의 타당성과 시스템 개발의 가능성을 검증하였다. 설계방법으로는 최적화설계의 개념 및 틀에 관행적설계의 장점을 선택적으로 수용 보완한 다단계혼성설계방법을 사용하였다. 여기서 단면적 또는 단면계수를 중간설계변수로 하여 연속최적화를 통해 시스템단계설계를 취하고, 계산된 단면값이 속하는 표준상업 단면그룹의 상위단면들을 데이터베이스로부터 후보군으로 검색 나열하고, 순서대로 부 규준에 따라 검토 선택한다. 설계를 지배하는 제한조건을 다루는데 있어 관행적 설계방식과 같이 주 조건과 부 조건으로 나누어 단계적으로 접근함으로써 복합 조건들을 갖는 대형최적화문제의 계산적 부담을 줄이면서 최적화 설계방법에 대한 이질감을 최소화한다. 설계의 수치연산은 유한 요소 기반 최적화 방법에 기초하고, 그래픽작업은 AutoLISP 프로그래밍을 통하여 AutoCAD로 실행하였다. 수치연산을 병행하는 서버와 그래픽스테이션 간의 통신은 메시지패싱 방식으로 이루어진다. 공학설계를 구성하는 주된 작업요소를 분산 병행처리한 본 연구는 PC와 Windows를 플랫폼으로, PVM을 분산운용체제로 하여, 해석 및 최적화 알고리즘으로서 유한요소법과 수리계획법을 실행하고, AutoCAD와 연동하여 자동적으로 설계도면을 생성하였다는 점에서 작업환경 및 방법론상 충분한 적용성과 확장성을 갖는다.

감사의 글

이 논문은 2004년도 한국학술진흥재단의 지원에 의한 연구의 일부임((KRF-2004-002-D00403)

참고문헌

1. Salomone, T. A., *Concurrent Engineering*, Marcel Wexler Inc, 1995
2. Kirsch, U., "*Structural Optimization*", Springer - Verlag, 1993
3. Adeli, H. and Kamal, O., "Concurrent Analysis of Large Structures I - Algorithms", *Computers & Structures*, Vol. 42, 1992, pp. 413~424
4. Sunderam, V. S., "PVM : A Framework for Parallel Distributed Computing", *Journal of Concurrency : Practice and Experience*, Vol. 2, No. 4, 1990, pp. 315~339
5. Przemieniecki, J. S., *Theory of Matrix Structural Analysis*, McGraw-Hill. Inc, 1968
6. Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design*, McGraw-Hill. Inc, 1984
7. Khan, M. R., "Optimality Criterion Techniques Applied to Frames having Nonlinear Cross-sectional Properties", *Journal of the AIAA*, Vol. 81, 1981, pp.233~241
9. Gorman, M. R., "Optimum Structural Design by Implicit Enumeration", *Proc. 3rd Conf. on Computing in Civ. Eng. ASCE*, San Diego, Calif, April, 1984, pp.675~681.
10. Gupta, O. K. and Ravindran, A., "Nonlinear Integer Programming and Discrete Optimization", *Journal of the Mechanisms, Transmissions, and Automation in Design, ASME*, Vol.105, No. 2, 1983, pp.160~164.
11. Haug, E. J., and Arora, J. S., *Applied Optimal Design*, Wiley, 1979
13. Huang, M. W. and Arora, J. S., "Optimal Design with Discrete Variable: Some Numerical Experiment", *Int. J. Num. Meth. Eng.* Vol.40, 1997, pp.165~188
14. Huang, M. W. and Arora, J. S., "Optimal Design of Steel Structures using Standard Sections", *Structural Optimization Journal*, Vol.14, 1997, pp.24~35.
16. 황진하, 이학술, 박종희, "계층적 모델링을 통한 사용자 지향 구조시스템통합 프로그래밍", 대한토목학회 논문집, 제19권 제I-3호, 1999, pp.297~305
17. 황진하, 박종희, "분산해석에 기초한 대형구조계의 설계최적화", 대한토목학회논문집, 제22권 제3-A호, 2002, pp.545~552