

프로세스 분할 병행에 의한 통합 구조설계 운용

Integrated Structural Design Operation by Process Decomposition and Parallelization

황진하†

박종희*

Hwang, Jin-Ha

Park, Jong-Hoi

(논문접수일 : 2007년 12월 21일 ; 심사종료일 : 2008년 1월 9일)

요 지

본 연구는 제품(구조물)최적화와 프로세스 병렬화가 동시에 실행되는 통합구조설계운용 방법을 제시한다. 관행적 설계 방법과 최적화설계방법의 장점들을 선택적으로 수용·보완하는 DB연동 혼성방법을 통해 설계조건을 단계적으로 적용함으로써 대형최적화문제의 계산적 부담 및 이질감을 최소화하였다. PC 네트워크상에서 공학설계를 구성하는 수치연산과 도면 작성이 동시공학적 절차로 구현되고, 여기서 설계의 수치연산은 유한요소기반 최적화 방법에 기초하며, 그래픽작업은 AutoLISP 프로그래밍을 통하여 AutoCAD로 표현된다. 수치연산을 병행하는 서버 및 이산최적화를 지원하는 DB서버와 그래픽스테이션 간의 통신은 메시지패싱 방식으로 이루어진다. 표준최적화모델인 트러스구조에 대한 수치실험은 연구의 타당성과 유용성을 보여준다. 공학설계를 구성하는 주된 작업요소를 분산 병행 처리하는 본 연구는 일반적 방법론 및 산업표준환경 등에 기초한다는 점에서 충분한 적용성과 확장성을 갖는다.

핵심용어 : 다단계 혼성설계, 프로세스 분할 병행, DB연동, 동시공학, 설계최적화, 이산표준단면

Abstract

Distributed operation of overall structural design process, by which product optimization and process parallelization are simultaneously implemented, is presented in this paper. The database-interacted hybrid method, which selectively takes the accustomed procedure of the conventional method in the framework of the optimal design, is utilized here. The staged application of design constraints reduces the computational burden for large complex optimization problems. Two kinds of numeric and graphic processes are simultaneously implemented by concurrent engineering approach in the distributed environment of PC networks. The former is based on finite element optimization method and the latter is represented by AutoCAD using AutoLISP programming language. Numerical computation and database interaction on servers and graphic works on independent clients are communicated through message passing. The numerical experiments for some steel truss models show the validity and usability of the method. This study has sufficient adaptability and expandability, in that it is based on general methodologies and industry standard platforms.

Keywords : *multilevel hybrid design, process decomposition and parallelization, database interaction, concurrent engineering, design optimization, discrete standard sections*

1. 서 론

오늘날의 복합적 시장 환경은 공학도에게 시간, 비용, 품질 등에서 빠르고 유연한 대응을 압박하고 있다. 이러한 요구에 대해 공학설계는 문제형성으로부터 반복 해석과 재설계를 통해 설계도서로 해답한다. 또한 정보처리기술은 공학설계의 수준을 한 차원 높였으며, CAD/CAM/CAE의 이론과

기술은 수학적 모델에 기초하여 정교한 수치해석, 최적화, 시뮬레이션과 그래픽을 구현케 한다. 이러한 다양한 작업들은 각각 상당한 시간과 비용을 요구하고 따라서 이들이 순차적인 과정으로 수행될 경우, 시간과 비용은 물론 의사결정에 많은 낭비가 발생한다. 이러한 문제점은 최근 새로운 엔지니어링 패러다임으로 자리하고 있는 동시병행공학(CE : Concurrent Engineering)방법(Salomone, 1995)으로 상당

† 책임저자, 정희원, 충북대학교 토목공학부 교수
Tel: 043-261-2407 ; Fax: 043-276-6706
E-mail: jhhwang@chungbuk.ac.kr

* 서원대학교 환경건설정보학과 전임강사

• 이 논문에 대한 토론을 2008년 4월 30일까지 본 학회에 보내주시면 2008년 6월호에 그 결과를 게재하겠습니다.

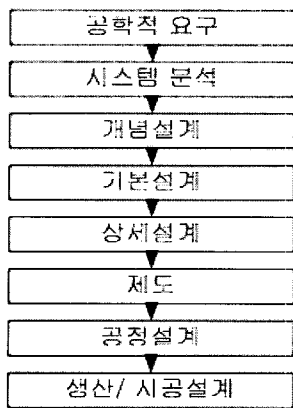


Fig. 1 Sequential Design Cycle

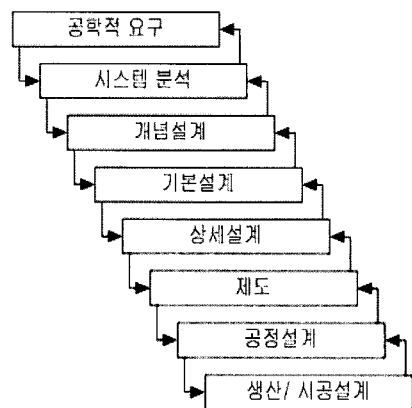


Fig. 2 Concurrent Design Cycle

부분 개선할 수 있다.

동시병행공학에 대한 정의와 그 구현 방법은 다양하나 그것의 기본 개념 및 목표는 제조 및 관련 지원 부분을 포함한 전 공정을 라이프사이클에 걸친 모든 요소를 고려하여 통합 병행토록 설계하는 체계적 접근방법이며 정보화와 자동화를 통하여 개발시간의 단축, 비용 절감 및 품질 향상 등 생산성을 극대화하려는 것이다. 그것에 대한 한 가지 방도인 개별 프로세스의 분산운용은 전체 수명주기관리의 유연성을 높이고 각 단계에서 발생하는 오류를 조기에 검증, 수정할 수 있다. 본 연구에서는 통합 환경에서의 분산운용방식을 통해 각 설계 요소를 분담하여 전체 목표를 병행 수행하는 동시공학 적 접근을 제시하며, 여기서 분산처리란 컴퓨터들을 네트워크로 엮어 하나의 작업을 각 컴퓨터에서 나누어 수행하도록 하는 것을 말한다.

본 연구의 목표는 구조설계프로젝트를 수치적 모델로 정형화하고 동시공학적 개념에 입각하여 그 결과물인 프로덕트 뿐만 아니라 해석/재해석/민감도해석/최적화/도면작성 등 설계과정 전체 프로세스를 분산 최적화하므로써 설계의 유연성과 생산성을 제고하는 데 있다. 즉 프로세스와 프로덕트를 분산운용하여 각 단계에서 발생하는 오류를 조기에 검증, 수정할 수 있도록 한다. 본 연구는 PC 네트워크상에서 분산처리 기법과 데이터베이스에 기초하여 공학설계를 구성하는 설계계산과 도면작성 및 DB연동을 병행함으로써 기존에 순차적으로 이루어지던 설계과정을 동시공학적 절차로 구현하고, 예제를 통해서 타당성 및 효율성을 검증하였다.

2. 프로덕트/프로세스 통합

2.1 프로세스 분할 병행

공학설계는 수치연산, 기호추론 및 그래픽 처리 등을 포함하는 복합적 창조활동이다. 전체 과정은 요구분석 및 문제정

의로부터 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. Fig. 1은 순차적 설계과정을 보이고, Fig. 2는 동시공학 개념에 기초한 설계과정을 단순화하여 나타낸 것이다.

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.

2.2 프로덕트 설계최적화

신뢰성 있고 경제적인 시스템을 설계하는 것은 공학도들이 추구하는 도전적 과제라 할 수 있다. 경제성 관점에서, 최적화설계(Haug, 1979)는 좋은 대안으로 공학의 여러 분야에서 폭 넓게 연구 활용되고 있다. 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.

일반적으로 최적화설계에서는 설계변수들을 연속변수로 가정한다. 그러나 많은 공학설계분야의 최적화문제들에서 이러한 가정은 실용성이 없다. 그 이유는 시공성이나 경제성을 고려할 때, 실무적으로 부재들의 선택이 자유롭지 못하고, 이미 제작되어 있는 단면치수들 중에서 선택해야 하기 때문이다.

이처럼 설계변수들이 실제로 연속변수가 아니라 이산변수

이거나, 또는 연속과 이산변수의 혼합으로 정의되는 경우는 복잡한 불연속 최적화문제로 된다. 이러한 점을 극복하기 위해 구조설계의 최적해를 실무에서 직접 이용할 수 있는 이산 집합으로부터 선택함으로써 그 최적해가 직접적인 가용성을 지니도록 하는 이산최적화에 많은 연구가 이루어지고 있다. 주요한 방법으로는 먼저 분기한계법, 열거법, simulated annealing과 유전자 알고리즘 등(Khan, 1981 ; Gorman, 1984 ; Gupta, 1983)이 시도되고 있으나 실제 문제에 적용하기에는 이론적으로나 절차적으로 일관된 체계가 정립되어 있지 못하고, 특히 계산적 효율성이 떨어지는 어려움이 있다. 구조설계분야에서 이산최적화방법은 연속최적화의 현실적 비실용성을 해결하기 위해 시도되었으나, 기존 기술자들이 오랫동안 사용해 익숙한 관행적설계 나름의 장점을 활용치 못할 뿐만 아니라 최적화설계에 대해 갖는 낯설음이나 비사용성을 해소할 수 있는 근본적 대안이 될 수는 없다.

이상과 같은 관점에서 본 연구는 실제적으로 사용할 수 있는 표준단면들을 찾기 위해 이산최적화를 쓰는 대신, 그보다 효과적인 연속최적화방법의 틀 속에 관행적방법의 직관과 절차를 수용함으로써 두 가지 방법론의 장점이 결합하는 혼성 방법을 제시한다.

2.3 분산환경 구성 및 운용

대형시스템을 최적화하는 문제는 대단히 많은 설계변수와 제한조건을 갖기 때문에 전산처리에 상당한 메모리와 시간 및 많은 비용을 요구하게 된다. 고속화와 관련된 컴퓨터 구조 또는 조직 측면에서 발전은 다양한 형식의 병렬컴퓨터와 분산 처리운용을 들 수 있다(황진하 등, 1999 ; 2000). 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.

분산처리는 대형 프로젝트에 관련된 하나 이상의 프로그램이 여러 개의 프로세스로 나누어져서 전체 목적을 달성하기 위해 각자의 일을 수행한다.

개별메모리형 MIMD 형태의 병렬컴퓨터를 기본 모델로 하여 설계된 PVM(Parallel Virtual Machine)은 서로 다른 종류의 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 구성해 준다(Sunderam, 1990).

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.

본 연구에서는 PVM 프로그래밍 모델 중 주종모델을 사용하며, 마스터 프로세서는 통제프로그램으로서의 역할뿐만 아니라 슬레이브 프로세서와 같이 실제의 계산과정도 수행하였다. 즉, 네트워크로 연결된 3대의 펜티엄급 PC를 사용하여 수치처리와 제어담당서버, 데이터베이스서버 및 그래픽스테이션으로 구성된다.

연산제어서버는 DB서버와 연동하여 구조해석과 최적화 루틴을 반복 수행하면서 그 결과를 수시로 그래픽스테이션에 전달한다. 그래픽스테이션은 모니터를 통해 구조물의 형상과 갱신되는 설계변수 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. 3은 가상 분산환경에서의 시스템 운용도를 보여주며, 여기에서 실제 적용 및 구현한 부분은 설계엔지니어링 부분으로 구조계산과 도면작성을 포함한다. 물론 나머지 부분도

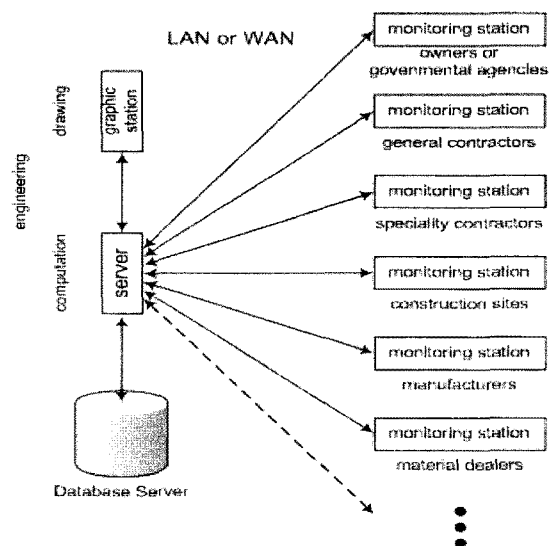


Fig. 3 Conceptual Diagram for Distributed Systems Operation

TCP/IP 프로토콜을 이용하는 인터넷 환경에서 동일하게 운용할 수 있다.

3. 혼성 설계최적화

3.1 설계해석 및 최적화

설계란 해석과 총합의 반복과정으로 이루어지는 종합 작업이며 설계최적화의 실행측면에서의 재해석과 재설계의 반복 결합으로 수행되는 일련의 전산과정이다. 여기서 해석의 보편적 도구는 유한요소법이며, 범용성을 갖는 최적화 방법은 수리계획법(mathematical programming)이다(Vanderplaats, 1984). 유한요소기반 최적화설계는 설계 제한조건을 지키면서 목표함수를 최소화하는 설계변수의 조합을 찾는다. 그러나 최적화문제 또는 프로그램의 제한조건은 설계변수의 함수로 설정되는데 반해 실제적인 구조최적화의 경우 해석모델로부터 산출되는 거동변수의 음함수로 형성된다. 따라서 목표함수와 설계조건에 대한 설계변수들의 영향을 찾는 설계민감도해석과정이 복잡하다.

일반적으로 설계최적화문제는 다음과 같이 형성될 수 있다.

$$\text{Minimize } F(X) \quad (1)$$

$$\text{Subject to } g_j(X) \leq 0 \quad j = 1, 2, \dots, n_g \quad (2)$$

$$h_k(X) = 0 \quad k = 1, 2, \dots, n_h \quad (3)$$

$$X_i^L \leq X_i \leq X_i^U \quad i = 1, 2, \dots, n \quad (4)$$

여기서 식 (1)은 목표함수이고, 식 (2)~(3)은 구조물의 거동을 제한하는 조건식으로 변위, 응력 등에 대한 제한조건을 나타내며, 식 (4)는 설계변수인 단면의 치수에 대한 제한조건이다. $F(X)$ 는 전체 구조물의 총 중량을 나타내고, X_i^L 와 X_i^U 는 각각 설계변수의 하한치와 상한치를 표시한다.

식 (1)~(4)와 같이 수학적으로 형성된 문제를 수치적으로 푸는 최적화 알고리즘은 제한조건을 다루는데 따라 크게 직접법(또는 기본법)과 간접법(또는 변환법)으로 나누어진다. 본 연구는 후자에 속하는 ALM(Augmented Lagrange Multiplier) 방법을 이용한다. ALM방법은 제한조건을 비제한조건으로 변환하여 푸는 변환법의 하나로 라그랑지승수를 개선하여 최적해를 찾는 방법이다. 완화변수를 도입하여 주어진 부등식제한조건들을 등식제한조건들로 변환하고 목표함수를 확장한 확장라그랑지함수는 다음과 같이 나타낼 수 있다.

$$A(X, \lambda, s, r_p) = F(X)$$

$$+ \sum_{j=1}^m [\lambda_j (g_j(X) + s_j^2) + r_p (g_j(X) + s_j^2)^2] \quad (5)$$

여기서, λ 는 라그랑지승수이고, r_p 는 벌칙매개변수, 그리고 s^2 는 완화변수(slack variable)이다. 이는 다음과 같은 유사목표함수로 변환할 수 있다.

$$A(X, \lambda, r_p) = F(X) + \sum_{j=1}^m [\lambda_j \psi_j + r_p \phi_j^2] \quad (6)$$

$$\psi_j = \max \left[g_j(X), -\frac{\lambda_j}{2r_p} \right] \quad (7)$$

최적화루틴과 연결되는 설계해석 및 민감도해석은 유한요소해석에 기초한다. 정적하중을 받는 구조체의 거동은 유한요소 평형방정식

$$K(X)U = P(X) \quad (8)$$

으로부터 산출되고, 여기서 K 는 구조강성도매트릭스, U 는 변위벡터, P 는 하중벡터이다. 유한요소해석에 기초한 구조최적화문제의 설계민감도해석 또한 이 기본식에 근거한다. 설계민감도를 구하는 데는 수치해석에서 널리 사용되는 유한차분법이나 해석적방법과 반해석적방법 등을 사용한다. 유한차분법은 범용성을 가지나 지나치게 비용 소모적이다. 따라서 설계공간이나 상태공간에서 직접 또는 간접 미분을 통한 해석적방법이나 중간 형태의 반해석적방법을 활용할 수 있다. 보다 직관적인 직접미분법은 다음과 같이 구한다. 목표함수나 설계조건 등 설계함수를 음함수

$$\phi = \phi(X, U) \quad (9)$$

로 일반화할 때, 설계변수에 대한 전미분은 연쇄법칙을 이용한 식 (9)의 음함수 미분으로부터

$$\nabla \phi = \frac{d\phi}{dX} \quad (10)$$

또는

$$\frac{d\phi}{dX_i} = \frac{\partial \phi}{\partial X_i} + \frac{\partial \phi}{\partial U} \frac{dU}{dX_i} \quad i = 1, \dots, n \quad (11)$$

여기서

$$\frac{d\phi}{dX_i} = \begin{Bmatrix} \partial \phi_1 / \partial X_i \\ \vdots \\ \partial \phi_r / \partial X_i \end{Bmatrix}, \quad \frac{dU}{dX_i} = \begin{Bmatrix} \partial U_1 / \partial X_i \\ \vdots \\ \partial U_m / \partial X_i \end{Bmatrix}$$

$$\frac{\partial K}{\partial X_i} = \begin{Bmatrix} \frac{\partial K_{11}}{\partial X_i} & \dots & \frac{\partial K_{1m}}{\partial X_i} \\ \vdots & & \vdots \\ \frac{\partial K_{m1}}{\partial X_i} & \dots & \frac{\partial K_{mm}}{\partial X_i} \end{Bmatrix}$$

$\frac{dU}{dX_i}$ 를 얻기 위해 지배방정식 (8)을 미분하면

$$K(X) \frac{dU}{dX_i} + \frac{\partial K}{\partial X_i} U = \frac{dP}{dX_i} \quad i = 1, \dots, n \quad (12)$$

또는

$$K(X) \frac{dU}{dX} = Q(X) \quad i = 1, \dots, n \quad (13)$$

여기서

$$\frac{\partial U}{\partial X} = \left\{ \frac{\partial U}{\partial X_1}, \frac{\partial U}{\partial X_2}, \dots, \frac{\partial U}{\partial X_n} \right\}$$

우변을 유사하중

$$Q = (Q_1, Q_2, \dots, Q_m) \quad (14)$$

$$Q_i(X) = \left(\frac{dP}{dX_i} - \frac{\partial K}{\partial X_i} U \right) \quad (15)$$

라고 하면 식 (13)은 평형조건 식 (8)과 동일한 계수행렬 K 를 갖고 따라서 작용하중 P 를 받는 기본방정식과 동일하게 강성도행렬 K 의 전후대입(forward and backward substitutions)을 통한 분할알고리즘을 이용해 변위에 대한 도함수를 구할 수 있다.

$$K = U^T U \quad (16)$$

여기서 행렬 U 는 상삼각행렬이고 식 (13)으로부터 단일 변위 U_j 에 대한 설계도함수 표현은

$$\frac{\partial U_j^T}{\partial X} = \nabla U_j^T = I_j^T \frac{\partial U}{\partial X} \quad (17)$$

로 쓸 수 있다. 여기서 I_j 는 j 번 성분만 단위성분을 갖고 다른 성분은 모두 0인 벡터이다. 많은 문제에서 하중벡터는 설계변수에 독립적이고 ($\partial P / \partial X = 0$), 따라서 식 (13)은

$$K \frac{dU}{dX_i} = - \frac{\partial K}{\partial X_i} U = Q_i^R \quad (18)$$

가 된다.

기본 미분식 (13)의 양변에 $I_j^T K^{-1}$ 을 전승하고 식 (17)에 대입하면

$$\frac{dU_j}{dX_i} = I_j^T K^{-1} \left(- \frac{\partial K}{\partial X_i} U \right) = I_j^T K^{-1} Q_i^R \quad (19)$$

와 같은 단일변위의 설계변수에 대한 도함수를 얻을 수 있다. 여기서 상태공간법은 K^{-1} 을 구하는 부담을 우회하여 기본식의 형태를 갖는

$$K \mu_i = I_j \quad (20)$$

로 정의되는 수반변수 μ 를 도입하고 식 (19)에 역대입하여 설계도함수를 얻는다.

$$\frac{\partial U_j}{\partial X_i} = \mu_j^T \left(- \frac{\partial K}{\partial X_i} \right) U \quad (21)$$

또는

$$\nabla U_j^T = \frac{\partial U_j^T}{\partial X} = \mu_j^T Q_i^R \quad (22)$$

변위도함수를 건너 바로 설계도함수 $\nabla \phi_i$ 를 구할 경우, 식 (20) 대신에

$$\nu^T K = \frac{\partial \phi^T}{\partial U} \quad (23)$$

또는

$$K(X) \nu = \frac{\partial \phi}{\partial U} \quad (24)$$

을 만족시키는 ν 를 식 (18)에 전승하면

$$\nu^T K(X) \frac{dU}{dX_i} = - \nu^T \frac{dK}{dX_i} U \quad (25)$$

식 (11)로부터 다음과 같이 설계도함수를 구할 수 있다.

$$\begin{aligned} \frac{d\phi}{dX_i} &= \frac{\partial \phi}{\partial X_i} + \nu^T K \frac{dU}{dX_i} \\ &= \frac{\partial \phi}{\partial X_i} - \nu^T \frac{dK}{dX_i} U \end{aligned}$$

$$= \frac{\partial \phi}{\partial X_i} - \nu^T Q_i^R \quad (26)$$

앞 뒤 두 가지 방법은 각각 장단점이 있어 전자는 제한조건의 개수가 설계변수와 하중조건의 개수의 곱보다 큰 경우에 유리하고 그 반대인 경우 후자가 유리하다.

3.2 2차 설계조건

AISC 매뉴얼의 설계 관련 자료를 데이터베이스에 내장하고 SQL 서버는 ODBC를 통해 연속최적화 설계값을 넘는 단면요구에 적합한 단면들을 오름차순으로 제공한다. 다시 후보단면들은 순서대로 아래와 같은 2차 설계조건에 대해 검토되고 반복 시행여부가 결정된다. 부재가 압축력을 받을 경우의 허용응력은 좌굴을 고려한다. AISC에서는

$$C_c = \sqrt{\frac{2\pi^2 E}{F_y}} \quad (27)$$

을 분기점으로 탄성과 비탄성 영역의 좌굴을 반영한다. 즉, 전단면에 대하여 허용응력을

$$F_a = \frac{\left[1 - \frac{(Kl/r)^2}{2C_c^2}\right] F_y}{\frac{5}{3} + \frac{(3Kl/r)}{8C_c} - \frac{(Kl/r)^3}{8C_c^3}} \quad ; \quad Kl/r < C_c \quad (28)$$

$$F_a = \frac{12\pi^2 E}{23(Kl/r)^2} \quad ; \quad Kl/r < C_c \quad (29)$$

로 나누고 설계 매뉴얼에서 세장비에 따른 허용응력을 도표화하여 제공한다. 후보 단면들은 차례로 아래와 같은 2차 설계조건에 대해 검토된다.

$$\frac{L \cdot d}{(b_f \cdot t_f)} \leq \frac{20,000}{F_y} \quad (30)$$

$$L \leq \frac{76b_f}{\sqrt{F_y}} \quad (31)$$

여기서, F_y : 항복응력

L : 압축플랜지의 비지지길이

다음, 선택된 단면의 상세치수는 국부좌굴에 대한 다음 식으로 주어지는 폭/두께비를 만족해야 한다.

$$\frac{b_f}{2t_f} \leq \frac{95}{\sqrt{F_y}} \quad (32)$$

$$\frac{h - 2t_f}{t_w} \leq \frac{1400}{\sqrt{F_y(F_y + 16.5)}} \quad (33)$$

4. DB연동 및 자동 작도

4.1 데이터베이스 연동

구조 설계는 계획, 예비설계, 해석, 검토, 재설계와 같은 과정을 반복하면서 이루어지고 매 단계마다 많은 입출력 설계정보를 처리해야 한다. 따라서 설계과정의 정보관리를 효율적으로 하기 위해서 데이터베이스관리시스템(DBMS)을 이용하는 것이 효과적이다. 대부분의 실제 설계에서, 가능한 부분부터 대량으로 생산되는 표준공장제품으로 규격화 할 때 총비용 면에서 많은 효과가 있으며 국가별로 여러 분야의 표준규격 데이터베이스도 잘 구축되어 있다. 본 연구는 표준단면을 데이터베이스에 내장하고 SQL 서버는 ODBC를 통해 애플리케이션의 연속최적화 설계값을 넘는 단면요구에 적합한 단면들을 오름차순으로 제공한다.

본 연구에서는 MS SQL Server를 DB서버로 하는 C/S(Client-Server) 환경을 기반으로 설계시스템을 구성하고, 임의의 클라이언트에서 원격데이터에 접근한다. 설계시스템은 데이터베이스를 중핵으로 관행적설계와 최적화설계를 결합하는 새로운 다단계 혼성설계방법을 구현한다. C/S 환경에서 후단(Back-End) 서버에는 데이터베이스와 내장 프 로시저 등을 작성하여 프로그래밍 환경을 구성하고 전단(Front-End) 클라이언트에 사용자 인터페이스와 함께 데이터 액세스 및 애플리케이션 로직을 작성한다. 여기서 DB서버는 다수의 클라이언트 사이에서 리소스를 효율적으로 관리하고, 클라이언트 애플리케이션은 사용자와 대화하며 SQL을 통해 서버 DBMS의 데이터 원본에 설계논리를 수행한다. DBMS는 결과만을 데이터소스로 되돌리고 프로파일을 구성한다.

마이크로소프트사가 개발한 표준데이터베이스 인터페이스인 ODBC(Open DataBase Connectivity)는 C/S 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 네 가지의 기본 객체 및 OpenDatabase와 Open Recordset의 두 가지 메소드를 사용한다. 자료 조작은 보통 SQL 명령문을 사용, ODBC드라이버 매니저를 통해 RDBMS에 전달된다.

Fig. 4는 응용프로그램과 연동된 데이터베이스의 데이터 원도우를 보여주며, 시스템은 연동뿐만 아니라 자체적으로 입·출력 수정 및 검색기능을 갖는 전단 기능을 갖는다.

Variable	Shape	Area	Depth	WebT	FlangeW	FlangeT	Fy	Fv3	Ix	Sx	Iy
0.1	W 8-10	2.96	7.89	0.17	3.94	0.215	50.3	54.8	16.4	5.56	2.2
1.77183	W 10-12	3.54	9.87	0.19	3.96	0.21	47.5	24.5	53.8	10.9	3.3
1.77383	W 6-12	3.55	6.03	0.23	4	0.28			22.1	7.31	2
0.1	W 4-13	3.83	4.16	0.28	4.06	0.345			11.3	5.46	1
1.77156	W 8-13	3.84	7.99	0.23	4	0.255		54.7	38.6	8.91	3
Length	W 12-14	4.16	11.91	0.2	3.97	0.225	54.3	18.6	88.6	14.9	4
60	W 10-15	4.41	9.99	0.23	4	0.27		35	68.9	13.8	3
36	W 6-15	4.43	5.99	0.23	5.99	0.26	31.8		29.1	9.72	2
30	W 8-15	4.44	8.11	0.245	4.015	0.315		60.3	48	11.8	3
36	W 5-15	4.68	5.01	0.24	5	0.36			21.3	8.51	2
46.8615	W 12-16	4.71	11.99	0.22	3.99	0.264		22.2	103	17.1	4
Force	W 5-15	4.74	5.28	0.26	4.03	0.405			32.1	10.2	2

Fig. 4 Database Interacted Hybrid Design

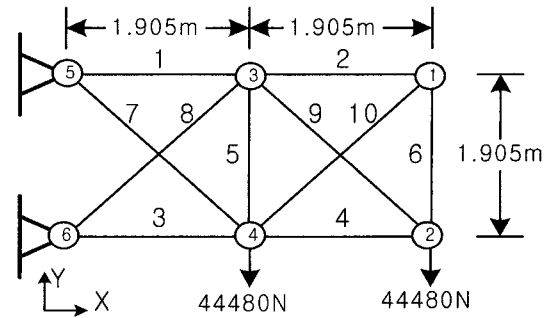


Fig. 5 10 Bar Cantilever Truss

4.2 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 읽어 들여 이를 분석해서 AutoCAD 화면에서 그래픽 작업을 반복한다.

AutoCAD의 도면 데이터베이스는 DXF 코드로 이루어져 있으며 AutoLISP을 통해 도면의 모든 요소에 대한 DXF 코드를 검색 및 수정할 수 있다. 이러한 관점에서 도면 개체의 요소는 도면개체(drawing entity)로 표현되며 각 도면개체에 대한 DXF 코드는 개체의 종류에 따른 몇 개의 코드 번호와 그에 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 속도에 기인해 생산성의 향상과 비용의 절감을 가져다준다. 특히 반복적이고 오류가 발생하기 쉬운 작업에 효과적이다.

Table 1 Design Data for 10 Bar Cantilever Truss

Modulus of elasticity	: 206,840MPa
Allowable stress limit	: 150MPa
Displacement limit	: ± 0.05m
Specific weight	: 76,820N/m ³
Upper limit on cross sectional area	: 0.0645m ²
Loading condition : 1	
(1) -44,480N acting in y-direction at node point 4	
(2) -44,480N acting in y-direction at node point 2	

5. 적용 및 분석

본 연구에서 제시한 동시공학적 분산이용방법의 타당성 및 효율성을 검증하기 위해 상기한 분산환경에서 펜티엄 IV 3.0GHz(RAM 512MB) IBM 호환 PC 3대를 기본 구성으로 하여 10 부재 및 200 부재 트러스구조에 대해 수치실험을 수행하였다.

설계프로세스는 연속최적화 단계(METHOD I), DB연동 이산화단계(METHOD II)와 도면작업을 수행하는 단계(METHOD III)로 구성된다. 각 단계별로 얻어진 설계 정보의 검증을 위해 연속최적화를 통해 얻어진 설계 결과값은 상용패키지(VR&D, 2001)를 수행하여 얻은 결과치와 비교하였고 최종 설계값인 표준화된 이산단면값은 Huang 등(1997)의 문헌에서 제시된 방법과 비교하여 나타내었다.

5.1 Ten Bar Truss

10개의 부재를 갖는 트러스 구조는 Fig. 5와 같으며 설계 변수는 중간변수로 단면적을 취하였다. 트러스 모델에 대한 설계변수와 하중조건 및 탄성계수 등에 관한 설계정보는 Table 1과 같으며 변위제한조건은 5, 6번을 제외한 모든 노드에 주어졌다.

먼저 설계조건에 대한 연속최적화 결과는 Table 2에 나타내었다. 목표함수값이 상용패키지를 수행하여 얻은 값과 유사한 결과를 나타내고 있다.

Table 2 Results of The Continuous Variable Optimization

	Present	Reference
Total Weight(N)	5118.66	5448.18

Table 3 AISC Sections for 10 Bar Truss

No. of Mem.	Present	GADSS	SADSS	BBMDSS
1	W 8×28	W 14×30	W 27×84	W 16×36
2	W 6×9	W 6×15	W 8×28	W 6×20
3	W 8×40	W 10×39	W 21×44	W 10×45
4	W 6×15	W 6×15	W 16×36	W 6×20
5	W 6×9	W 12×26	W 6×15	W 6×20
6	W 6×9	W 6×15	W 10×33	W 6×20
7	W 12×30	W 10×49	W 16×67	W 6×20
8	W 8×24	W 10×49	W 24×68	W 6×20
9	W 10×17	W 10×54	W 16×67	W 4×13
10	W 6×12	W 12×53	W 24×68	W 5×19
Total Weight(N)	6316.47	11778.89	17130.10	7659.83

Table 4 CPU Time

	METHOD I	METHOD II	METHOD III
CASE I	0.032s	0.01s	0.045s
TOTAL CPU TIME		0.087s	
CASE II	0.032s	0.01s	0.045s
TOTAL CPU TIME		0.045s	

Table 3에서는 AISC 매뉴얼의 표준단면에 대한 데이터 베이스를 작성하고 연속최적화와 연동된 데이터베이스 상에서 국부적인 시방사항 검토를 통해 얻은 이산 표준단면값과 문헌에서 제시한 유전자 알고리즘, Simulated Annealing 및 Branch and Bound 방법을 이용한 이산최적화 결과값을 나타내고 있다(Huang, 1997).

Table 2에서 연속최적화에 의한 총중량이 5118.66N인데 비해 표준단면으로 구성된 Table 3에서 총중량은 6316.47 N으로 증가한 것을 볼 수 있다. 이는 연속설계변수가 표준단면들의 이산설계변수로 실제화 되고 압축부재에 대한 허용좌굴응력조건을 부가한데 기인한 것이나 여타의 직접적 이산최적화방법보다 대단히 좋은 결과를 보인다.

또한 같은 설계조건을 갖는 다른 방법과 비교할 때 본 연구의 방법이 1.2배(BBMDSS)에서 2.7배(SADSS)의 효과

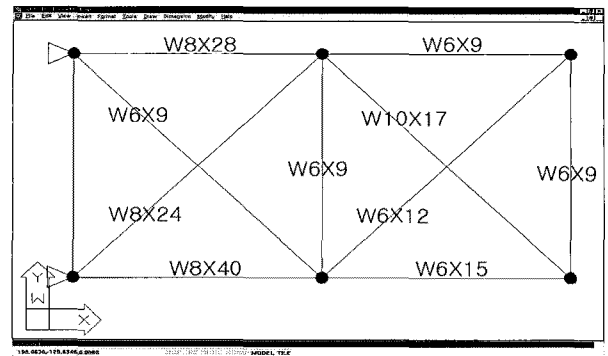


Fig. 6 AutoCAD Presentation by AutoLISP in Distributed Environments

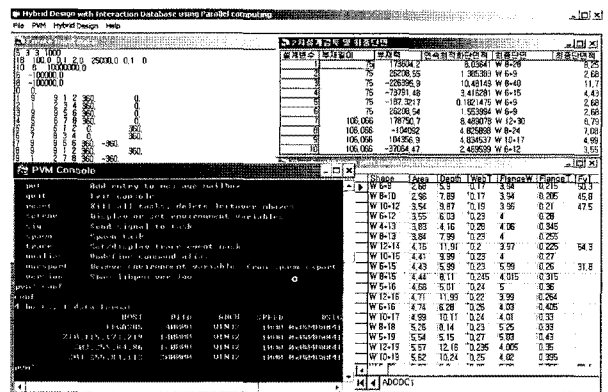


Fig. 7 Distributed Operation of Structural Design Process by Process Parallelization

를 가져오고 있다.

다음은 프로세스의 분할 병행을 통해 얻어지는 효과를 검토한다. Table 4는 한 대의 컴퓨터에서 모든 설계 프로세스를 수행한 결과(CASE I)와 프로세스 분산 병행 결과(CASE II)를 나타낸 것이다.

프로세스 분산운용의 효과는 거의 두 배에 달함을 알 수 있다. 여기서 나타낸 단일 컴퓨터상의 각 단계별 실행시간은 해당 프로그램, 즉 이산최적화를 위한 데이터베이스와 도면 작성을 위한 AutoCAD 로딩시간을 제외한 순수한 해당 작업시간이다.

Fig. 6과 Fig. 7은 각각 그래픽스테이션이 AutoCAD를 실행하여 표현한 설계 모델의 도면작성 과정과 마스터 프로세서에서 설계프로세스를 제어하는 모습을 나타내고 있다.

5.2 Two Hundred Bar Truss

두 번째 시험모델은 150개의 자유도 차수와 200개의 부재를 갖는 트러스 구조로 Fig. 8과 같다.

Table 5는 모델의 하중조건 및 설계정보를 나타내고 있다. 설계조건에서 목표함수는 총중량을, 설계변수는 개별부재

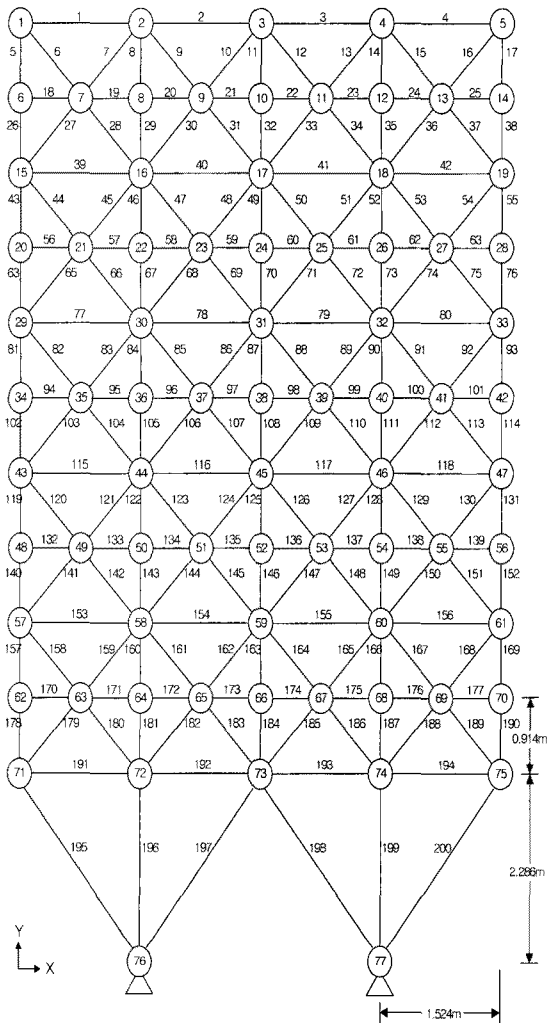


Fig. 8 200 Bar Truss

Table 5 Design Data for 200 Bar Truss

Modulus of elasticity	: 206,840MPa
Allowable stress limit	: ± 150 MPa
Displacement limit	: ± 0.0127 m
Specific weight	: 76.820 N/m ³
Upper limit on cross sectional area	: 0.0645 m ²
Loading condition : 1	
444.8 N acting in positive x-direction at nodal points 1,6,15,20,29,34,43,48,57,62,71	
Loading condition : 2	
4448.0 N acting in negative y-direction at nodal points 1,2,3,4,5,6,8,10,12,14,15,16,17,18,19,20,22,24,...,71,72,73,74,75	

의 단면적으로 정하였으며, 설계제한조건은 부재용력 200개, 절점의 처짐 150개, 그리고 단면적 200개에 가해진다.

Table 6은 200 부재 트러스의 설계변수 연계정보, Table 7은 모델의 하중조건 및 설계정보를 나타내고 있다.

1차 설계기준만 부가된 연속최적화 결과는 Table 8에 보여진다. 목표함수값이 상용패키지의 값과 유사한 결과를 나

Table 6 Grouping Information for 200 Bar Truss

Group	Members numbers(k=0,1 & j=0,1,2,3,4)	Notes
1	1 to 5	Horizontal, top.
2	5+3j	Verticals.
3	19 to 24	Horizontal, interior.
4	18+7k, 56+7k, 94+7k, 132+7k, 170+7k	Horizontal, exterior.
5	26+3j	Verticals.
6	6+k, 9+k, 12+k, 15+k, 27+k, 30+k, 33+k, 36+k	Diagonals.
7	39 to 42	Horizontals.
8	43+3j	Verticals.
9	57 to 62	Horizontal, interior.
10	64+3j	Verticals.
11	44+k, 47+k, 50+k, 53+k, 65+k, 68+k, 71+k, 74+k	Diagonals.
12	77 to 80	Horizontals.
13	81+3j	Verticals.
14	95 to 100	Horizontal, interior.
15	102+3j	Verticals.
16	82+k, 85+k, 88+k, 91+k, 103+k, 106+k, 109+k, 112+k	Diagonals.
17	115 to 118	Horizontals.
18	119+3j	Verticals.
19	133 to 138	Horizontal, interior.
20	140+3j	Verticals.
21	120+k, 123+k, 126+k, 129+k, 141+k, 144+k, 147+k, 150+k	Diagonals.
22	153 to 156	Horizontals.
23	157+3j	Verticals.
24	171 to 176	Horizontal, interior.
25	178+3j	Verticals.
26	158+k, 161+k, 164+k, 167+k, 179+k, 182+k, 185+k, 188+k	Diagonals.
27	191 to 194	Horizontals.
29	195, 197, 198, 200	Inclined, at base.
29	196, 199	Verticals, at base.

Table 7 Design Data for 200 Bar Truss

Modulus of elasticity	:206,840MPa
Yield stress	:250MPa
Displacement limit	:± 0.0127m
Specific weight	:76,820N/m ³
Upper limit on cross sectional area	:0.0645m ²
Lower limit on cross sectional area	:0.000065m ²
Loading condition : 1 444.8 N acting in positive x-direction at nodal points 1,6,15,20,29,34,43,48,57,62,71	
Loading condition : 2 4448.0 N acting in negative y-direction at nodal points 1,2,3,4,5,6,8,10,12,14,15,16,17,18,19,20,22,24,....,71,72,73,74,75	

Table 8 Results of The Continuous Variable Optimization

	Present	Reference
Total Weight(N)	16326.57	16542.49

Table 9 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 6×9	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
Total Weight(N)	35363.35	56686.78	67853.15	60850.32

타내고 있다.

2차 설계규준을 만족하는 이산최적화 상용단면은 문헌에서 제시한 비교목표함수 및 이산최적화 값과 함께 Table 9에 나타내었다.

선정된 상용 표준단면들은 모두 제한조건을 만족시키며 연속최적화값에 비해 약 2배 정도의 절감을 나타내고 있다. 이것은 압축부재의 국부설계조건에 대한 영향과 단면의 최소크기 제한조건을 AISC 상용표준단면의 최소단면적 0.0017m²으로 설정하였기 때문이다.

설계환경 등의 여건으로 실행 시간의 직접비교는 어려우나 일반적으로 이산최적화가 갖는 가장 큰 단점은 많은 시간이 소요된다는 것이다.

이러한 관점에서 본 연구에서 제시한 방법은 이산최적화과정의 연속최적화과정과 연동된 데이터베이스를 통해 수행되므로 연동된 데이터베이스 접근시간만이 소요되므로 설계문제가 커질수록 본 연구의 혼성설계방법이 갖는 효율성은 더욱 크게 나타날 것이다.

프로세스의 분할 병행을 통해 얻어지는 효과는 다음과 같다.

Table 10 CPU Time

	METHOD I	METHOD II	METHOD III
CASE I	76.35s	0.01s	80.45s
TOTAL CPU TIME	156.81s		
CASE II	76.35s	0.01s	80.45s
TOTAL CPU TIME	80.45s		

분산운용환경은 전과 동일하고 Table 10은 본 연구의 혼성설계방법을 순차 처리한 것(CASE I)과 병행 운용한 결과(CASE II)를 비교한 것이다.

분산 운용의 결과는 순차 처리에 비해 거의 2배의 효과를 나타냄을 알 수 있으며 이 결과 역시 단일 컴퓨터상에서 각 설계 프로그램의 로딩 시간을 제외시켰다.

10 부재 트러스와 비교할 때 효과적인 면에서는 거의 동일하나 설계 시간을 직접 비교하게 되면 설계 문제가 커지면 커질수록 분산운용의 효과가 상대적으로 커짐을 알 수 있다. 또한 순차처리에서는 각 단계별 프로세스가 끝난 후에야 설계의 오류검토나 재설계를 위한 Feed back이 이루어진다는 점에서 설계문제가 커지면 커질수록 상당한 시간 및 경비가 소요된다. 이러한 점에서, 본 연구에서 제시한 방법은 전체 설계 과정이 이질적 단위 프로세스로 분할되어 동시에 병행 운용된다는 점에서 설계공간이 클수록 그 효과는 더욱 배가된다.

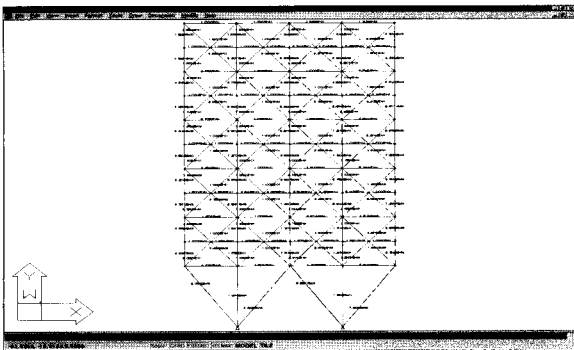


Fig. 9 AutoCAD Presentation by AutoLISP in Distributed Environments

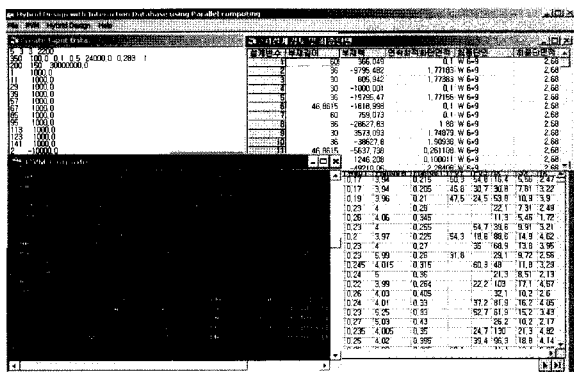


Fig. 10 Distributed Operation of Structural Design Process by Process Parallelization

Fig. 9는 도면작성을 담당하는 슬래브 노드에서 마스터 노드로부터 수치설계결과를 전달받아 각 단계별 진행사항을 도면으로 나타내고 있다. Fig. 10은 마스터 노드에서 전체 설계프로세스를 제어하는 모습을 보여주고 있다.

6. 결 론

본 연구는 PC 네트워크상에서 데이터베이스에 기반을 두고 공학설계를 구성하는 수치설계작업과 도면작성을 동시에 병행함으로써 기존에 순차적으로 이루어지던 설계과정을 동시공학적 절차로 구현하고, 표준최적화모델에 적용하여 연구의 타당성과 유용성을 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 표준상업단면그룹의 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 대형최적화문

제의 계산적 부담을 줄이면서 최적화설계방법에 대한 이질감을 최소화한다.

설계의 수치연산은 유한요소기반 최적화 방법에 기초하고, 그래픽작업은 AutoLISP 프로그래밍을 통하여 AutoCAD로 실행하였다. 수치연산을 병행하는 서버 및 이산최적화를 수행하는 DB서버와 그래픽스테이션 간의 통신은 메시지패싱 방식으로 이루어진다. 공학설계를 구성하는 주된 작업요소를 분산 병행처리한 본 연구는 PC와 Windows를 기본 플랫폼으로, 해석 및 최적화 알고리즘으로서 유한요소법과 수리계획법을 실행하고, AutoCAD와 연동하여 설계도면을 자동적으로 생성한다는 점에서 작업환경 및 방법론상 충분한 적용성과 확장성을 갖는다.

감사의 글

이 논문은 2004년도 한국학술진흥재단의 지원에 의하여 연구되었음(KRF-2004-002-D00403).

참 고 문 헌

- 황진하, 박종희 (1999) PC 트랜스퓨터 시스템을 이용한 구조 최적화의 병렬처리, 한국전산구조공학회 논문집, 12(2), pp.233~241.
- 황진하, 박종희, 정진덕, 전한규 (2000) 분산구조해석을 이용한 구조설계최적화, 한국전산구조공학회 학술발표회, pp. 125~132.
- 황진하, 박종희, 고덕구 (2000) 부구조화에 기초한 매트릭스 구조해석의 병렬/분산처리, 대한토목학회 논문집, 20(1-A), pp.115~124.
- Adeli, H., Kamal, O. (1992) Concurrent Analysis of Large Structures I - Algorithms, *Computers & Structures*, 42, pp.413~424.
- Gorman, M. R. (1984) Optimum Structural Design by Implicit Enumeration, *Proc. 3rd Conf. on Computing in Civ. Eng.* ASCE, San Diego, Calif. April, pp.675~681.
- Gupta, O. K., Ravindran, A. (1983) Nonlinear Integer Programming and Discrete Optimization, *Journal of the Mechanisms, Transmissions, and Automation in Design*, ASME, 105(2), pp.160~164.
- Haug, E. J., Arora, J. S. (1979) *Applied Optimal Design*, Wiley.
- Huang, M. W., Arora, J. S. (1997) Optimal Design of Steel Structures using Standard Sections, *Structural Optimization Journal*, 14.

- Khan, M. R.** (1981) Optimality Criterion Techniques Applied to Frames having Nonlinear Cross-sectional Properties, *Journal of the AIAA*, 81, pp.233~241.
- Kirsch, U.** (1993) *Structural Optimization*, Springer-Verlag.
- Przemieniecki, J. S.** (1968) *Theory of Matrix Structural Analysis*, McGraw-Hill. Inc.
- Salomone, T. A.** (1995) *Concurrent Engineering*, Marcel Wekker Inc.
- Sunderam, V. S.** (1990) PVM : A Framework for Parallel Distributed Computing, *Journal of Concurrency : Practice and Experience*, 2(4), pp.315~339.
- Vanderplaats, G. N.** (1984) *Numerical Optimization Techniques for Engineering Design*, McGraw-Hill. Inc.
- VR&D** (2001) GENESIS User Manual, Vanderplaats Research & Development, Inc.