화일배경 문제와 컴퓨터 위치선정 문제를 동시에 고려한 분산 정보시스템의 모형화

A Solution Methodology for Distributed Information System Configuration Problem Simultaneously Considering File Allocation and Computer Location Assignment

강석호 (서울대학교 산업공학과 교수)
장윤희 (서울대학교 산업공학과)

We have undertaken to develop an efficient solution methodology to adress distributed information system configuration problems through mathematical programming. We have simultaneously considered file allocation and computer location assignment problems which are two aspects of the design tighly coupled in a distributed computer system.

A model for solving the problem is shown to be a class of nonlinearinteger programming problems and procedures are developed for computing its lower bound. A heuristic algorithm is also developed and some results are obtained. Numerical results yield practical low cost solutions with substantial savings in computer processing time.

1. 서론

지역적으로 분산되어 있는 조직체나 기업은 경영의 효율성을 향상시킬 수 있는 통합적인 정보 시스템을 필요로 한다. 이러한 필요성에 따라 자료베이스를

공유하면서 정보의 교환은 통신 네트워크를 이용하는 지역적으로 분산된 컴퓨터 시스팀이 등장하였다. 즉, 이러한 네트워크는 2개 이상의 정보 프로세서를 포함하고, 이들은 각각의 부속 기억장치에 할당된 보관용 화일을 갖고 있으며, 이 화일들은 각 시스팀에 직접 연결되어 있는 사용자 뿐만 아니라 네트워크상의 다른 시스팀 사용자들도 이용할 수 있게한다.(7)

이러한 시스팀은 컴퓨터 네트워크에 대한 기술의 발달과 꾸준히 감소되고있는 하드웨어 비용의 추세에 힘입어, 집중화되어 있는 시스팀 (centralized computer system)에 비해 그 효율성이 커지고 있다. (2)

한편, 이제까지 진행되어온 분산정보 시스팀(Distrubuted information System)에 대한 연구 논문들은 다음의 두가지로 분류할 수 있다.

1) 고정적으로 주어진 네트워크상에서 자료 화일을 분배시키는 문제에 관한 연구
2) 시스팀 네트워크를 디자인하는 문제에 관한 연구

그런데, 이렇게 두가지 문제를 분리하여 시스팀을 접근하는 방법은 실제의 시스팀에서 볼때 부분 최적해만을 낳는 결과를 가져왔다.

본 연구에서는 위의 두가지 문제가 동시에 고려될 수 있는 포괄적인 디자인 접근법을 택함으로써 이제까지의 부분 최적해보다 더 유효한 시스팀의 구성 방법을 얻어내는 것을 목적으로 한다. 즉 분산정보시스팀을 디자인할 때 컴퓨터 시스팀의 분산배치 문제 (즉, 하드웨어 문제)와 화일의 배정문제 (즉, 소프트웨어의 문제)를 동시에 고려하여 시스팀을 모형화하고, 이모형에 대한 효율적인 해법을 개발하고자 한다.

분산정보시스팀이 보다 효율적인 시스팀으로서의 역할을 하기 위해서는 분산된 정보시스팀에서 고려하여 온 연구문제들 이외에도 다음의 사항들이 해결되어야한다.

첫째, 컴퓨터 시스팀을 지역적으로 분산 설치하는 방법.

둘째, 분할된 자료 화일을 각 분산되어 있는 컴퓨터 시스팀에 할당하는

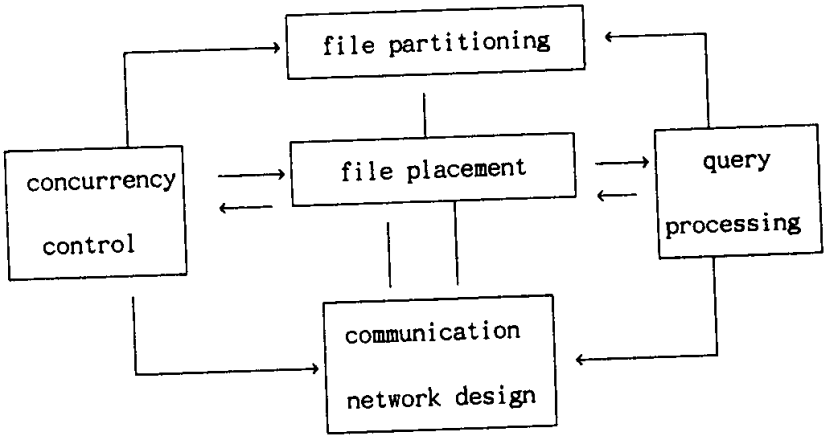
방법,

세째, 지역적으로 분산되어 있는 컴퓨터 시스템을 연결하는 정보통신망을 고안 설치하는 방법,

넷째, 사용자의 요구사항을 효과적이고 효율적으로 수행해 내기 위한 시스템의 운용방안등의 4가지 문제로 크게 나누어 볼 수 있다.

한편, 이제까지의 관련된 연구 논문들을 살펴보면, 위의 네가지 문제들이 서로 연관성을 갖지 못한채 다루어져 왔음을 알 수 있다.

그러나, 아래의 <그림 1>에도 나타나 있듯이 file placement와 communication network design 문제는 시스템 디자인 단계에서 고려해 볼 때도 사실상 분리될 수 있는 성질의 문제라고 할 수 없다.



<그림 1> 화일배치 문제와 연관되어 있는 문제들
(화살표는 정보의 흐름을 나타냄)

이에따라 최근들어 화일배정 문제와 통신 네트워크의 디자인 문제를 동시에 고려하여 시스템 디자인의 해를 얻고자 하는 연구 논문들이 발표되었다. 그러나 이에 관련된 대부분의 논문들은 시스템의 모형화에 역점을 두고 있는 단계이며 해법에 대한 제시가 거의 없는 형편이다. 또한 이러한 문제를 대상으로 해법을

연구한 논문들도 네트워크의 형태를 특정하게(예를들어 tree network 형태) 제한시킨다는 가정하에서 문제를 다루고 있으며 사실상, 다룰 수 있는 문제의 크기도 현실 상황을 고려하여 볼때 부적합하고 효율적이지 못하다.

본 연구는 앞에서 분류하여 기술한 분산정보 시스팀이 해결해야 할 문제 중에서 첫번째와 두번째 문제를 함께 연결시켜 다루어 보려는 의도에서 이루어졌다.

즉, 본 논문은 file placement와 computer location problem을 동시에 고려하여 분산정보 시스팀의 보다 일반화된 모형을 세우는 것과, 이 모형에 대한 효율적인 algorithm을 개발할 필요성에 의해 이루어 진다.

2. 현황분석

정보 통신망이나 컴퓨터 네트워크에서 화일 배정 문제를 다루는데 선구자적 역할을 할 것으로 Chu (11)와 Casey (20)의 논문을 들 수 있다.

Chu는 가지고 있는 각 화일의 갯수가 일정하다고 가정하고, 기억용량에 대한 제약식과 수행작업의 기대 지연시간을 만족시키면서 총 운용 비용을 최소로하는 화일의 배치를 구하였다. 한편, Casey는 각 화일이 갖게되는 동일 화일의 갯수를 결정해야할 변수로 두었는데, 그는 이 논문에서 자료읽기 (query)에 대한 자료수정(up-date)의 비율이 증가함에 따라 설치해야 할 동일화일의 갯수도 증가함을 보였다. 그런데 Casey의 모형은 NP-complet 문제임이 밝혀졌다. (9)

이상의 두 논문은 Simple File Allocation 문제의 대표적인 예로서 분류되는데, 이러한 형태의 문제들은 그후 OR 분야에서 잘알려진 단일상품 창고 위치 선정문제 (single commodity warehouse location problem)와 같은 성격의 문제임이 밝혀졌다. 이로써 이미 개발되어 있는 많은 기법들을 이와 같은 Simple File Allocation 문제에 대하여 프로그램과 특정 하드웨어의 구성을 가정에 포함시키고, 지연시간,기억용량, 화일의 가용성등을 제약식으로 넣었다. 이때의 목적함수는 기억장소 할당에 드는 비용과 통신비용을 고려하고 있다. 이러한 형태의 문제는 General File Allocation Problem 이라고 분류된다. (9)

그런데, 화일 배정을 최적화 (optimal file allocation)하는 문제는 네트워크의 디자인 문제를 동반한다. 즉, 프로그램이나 자료화일을 적정 장소에 배치하는 것 이외에도 통신라인의 용량결정, 컴퓨터 프로세싱 power의 선택, 그리고 connectivity를 크게 하면서도 그 반경 (radius of the network)이 최소가 되게 하는 통신망을 구성하는 등의 문제가 함께 고려되어야 한다. 이러한 필요성에 의하여 네트워크를 디자인하는 문제와 자료화일의 배치문제를 동시에 취급하는 연구가 시작되었는데, 이는 극히 최근의 논문들에서 찾아볼 수 있다. (1,4,5,7)

이러한 최적화 문제의 일반적인 형태는 다음과 같이 나타난다.

목적식: 비용의 최소화 (=기억장소 할당비용 + 통신관련비용)

제약식: 통신 네트워크의 신뢰도
자료화일의 가용도
자료접근 지연시간 등.

이에 관련된 목적함수의 형태에는 여러가지가 있다. 그 중에서도 Mahmoud와 Riordon(5)은 자료화일 저장에 관한 비용과 통신망을 임대하여 사용할 경우의 비용을 목적함수로 하였다. 또, Irani (4) 는 여기에다 통신 네트워크의 termination 비용을 더하였다.이상의 두가지 모형에서는 통신라인의 최적 용량을 구하고 있으며 또한 자료접근 제한시간에 대한 제약식에서는 query와 update에 따르는 차이점도 고려해 넣고있다. 그러나, 이들은 모두 통신네트워크가 특정한 형태로 주어졌다고 (fixed network topology) 가정하고 있다.

한편, Loomis 와 Popek는 더욱 세분화된 요소들로 구성된 목적함수를 사용했다. 즉, 각 query의 필요 용도가 특별하게 주어지는 경우, 각 사용자의 분산정도, 저장 화일에 대한 가용도, 전송지연시간, 동시에 두가지 화일을 접근할 수 있는가에 대한 가능성까지를 고려하고 있다. Chen과 Akoka (1)도 비슷하게 정교한 목적함수를 사용하고 있다. 이들은 자료베이스 소프트웨어, 컴퓨터장비, 통신네트워크장치, 데이타베이스 저장 용량, 사용자-프로그램, 프로그램-자료베이스로 이어지는 query 와 update 작업의 전송 등에 필요한 비용을 고려했다.

이상의 연구등은 프로세싱 power의 설치, 프로그램과 자료베이스의 위치설정, 그리고 시스팀에서 정해진 경로를 따라 전송되는 query 와 update 작업에 알맞는 통신 네트워크의 용량 등의 세가지 문제를 동시에 최적화 시키는 것을 목적으로 하여 쓰여진 것들이다. 그러나 , 이들의 연구에서는 거의 모두 네크워크를 특정하게 주어진 구조로 제한하고 문제를 다루고 있으며, 해법상 다룰 수 있는 문제의 크기도 극히 제한되어있는 상태이다.

3. 연구모형

3.1. 가정

1) 시스팀의 용량에는 제한이 없다고 가정한다. 즉, 시스팀에 연결 가능한 branch의 갯수나 자료화일의 크기에는 제한이 없다.
2) 통신 라인의 용량은 고려하지 않고 있다.
3) 정보처리에 실패하여 같은 transaction을 다시 수행하여야 하는 경우는 없다.
4) 통신 네트워크가 일정한 형태로 고정되어 있다는 가정은 없다.
5) 사용자는 몇 개의 지역으로 분산되어 있으며 각 branch에서 요구하는 transaction의 양은 미리 알려져 있는 것으로 가정한다.

I : 최종 시스팀 사용자가 위치한 branch의 집합
J : 컴퓨터 시스팀 설치 가능 장소의 집합
D : 부분화된 자료화일의 집합
C_{ij} : branch i를 j에 위치한 시스팀에 연결시킬 경우 드는 비용
C_j: 자료화일 d를 j에 위치한 시스팀에 연결시킬 경우 드는 비용
V_j : 위치 j에 시스팀을 설치하는데 드는 비용
Q_i: branch i에서부터 자료화일 d에 대해 요구하는 transaction의 양
r_{ij}':branch i에서 요구하는 transaction이 branch i가 연결되어 있는 시스템 j에서 수행될 경우, 이때 발생하는 통신 비용

r_{ij}'':branch i에서 요구하는 transaction이 branch i가 네트워크 상으로 연결되어 있는 시스팀 j로 옮겨져 수행될 경우, 이때 발생하는 통신비용

χ_{ij}: ┌ 1 : branch i가 j에 위치한 시스팀에 연결되었을 경우
└ 0 : 이 밖의 경우

χ_j : ┌ 1 : 자료 화일 d가 j에 위치한 시스팀에 배정되었을 경우
└ 0 : 이 밖의 경우

y_j : ┌ 1 : 시스팀을 위치 j에 설치할 경우
└ 0 : 이 밖의 경우

3.3 모형의 정립

MODEL NIP (nonlinear integer programming)

$$\text{Min } \{ \sum_j y_j v_j + \sum_i \sum_j C_{ij}\,\chi_{ij} + \sum_i \sum_j C_j^d\,\chi_j^d + \sum_d \sum_i \sum_j r_{ij}' \cdot Q_i^d \cdot \chi_j^d \cdot \chi_{ij} + \sum_d \sum_i \sum_j r_{ij}'' \cdot Q_i^d \cdot \chi_i^d \cdot (1-\chi_{ij}) \}$$

Sub. to

$$\sum_j \chi_{ij} = 1 \qquad \forall i \qquad (1)$$

$$\sum_j \chi_{ij} = 1 \qquad \forall d \qquad (2)$$

$$\chi_{ij} - y_i \le 0 \qquad \forall i,j \qquad (3)$$

$$\chi_j^d - y_i \le 0 \qquad \forall i,d \qquad (4)$$

$$\chi_{ij},\ \chi_j^d,\ y_i \ \ldots\ \{0,1\} \qquad \forall i \in I,\ j \in J,\ d \in D$$

. 목적함수에서 세번째 항까지는 하드웨어 및 소프트웨어의 설치비용이며 네번째 항과 다섯번째 항은 정보전달을 위한 통신비용이다.
. 제약식 (1)은 각 branch 가 정확히 하나의 시스팀에 연결될 것을 보장하여 준다.
. 제약식 (2)는 각 자료 화일이 정확히 하나의 시스팀에 배정될 것을 보장하여 준다.
. 제약식 (3)은 각 branch는 시스팀이 설치되어 있는 곳에만 연결될 수 있음을 나타낸다.
. 제약식 (4)는 자료화일은 시스팀이 설치되어 있는 곳에만 배치될 수 있음을 나타낸다.

4. 연구방법

3.1절에서 제시한 연구 모형은 조합최적화 문제 (combinatorial optimization) 이다. 이 문제는 NP-complete class에 속하는데 이 사실은 모형에서 이차식으로 나타난 항을 제거시켜보면 알 수 있다. 이차식으로 표현된 항을 제거시켜 본래의 문제보다 간단한 형태로 바꾸어 보면 이때의 문제는 uncapacitated plant location problem이 됨을 알 수 있고 이 문제는 이미 NP-complete class problem임이 밝혀져 있다.

NP-complete class 문제에 대해 polynomial time algorithm을 개발한다는 것은 어려운 일이어서, 이때 시스팀 디자이너는 heuristic 방법이나 approximation algorithm 등에 의존하여 해를 구하여야 한다.

본 연구에서는 제시한 모형에 대해 tight한 하한치를 구하여 여기에서부터 feasible solution을 얻어내는 방법과 heuristic한 방법을 이용하여 해를 구해내는 두가지 접근법을 시도해 보았다.

4.1 Lagrangian Relaxation Procedure 를 이용한 방법

Tight한 lower Bound를 얻을 수 있으면 이를 이용하여 최적해에 가까운 feasible solution을 구해낼 수 있으므로 본 연구에서는 Lagrangian Relaxation Procedure를 다음과 같은 방법으로 대상 연구 모형에 적용하여 하한치를 구하였다.

1) 3.1에서 제시한 model NIP에 ϕ_{ij} 라는 변수를 도입함으로써 선형화 시킨다.

[problem LIP]

2) [모형 LIP] 의 제약식중 일부를 Lagrangian Relaxation 시킨후 변수 ϕ_{ij}^{d}와 χ_{ij}의 성질을 이용하여 다시 transformation 시킨다.

[Problem LRIP]

3) Fixed 되어 있는 Lagrangian 승수벡터에 대해 [Problem LRIP]는 OR 분야에서 연구되어온 uncapacitated plant location problem (UPLP)과 convertible한 형태이다. 런데 UPLP에 대해서는 효율적인 algorithm이 이미 개발되어 있으므로, 이러한 algorithm들을 이 모형에 적용, 사용할 수 있다.

4) 3)에서 구해진 solution은 subgradient algorithm 등을 이용하여 보다 tight한 lower bound를 얻어내는 방향으로 개선시킬 수 있다.

위의 과정을 구체적으로 기술하면 다음과 같다.

단계 1 . 3.1에서 제시한 모형 Problem NIP에 var. ϕ_{ij}^{d} 를 도입하여 문제를 선형화 시킨다.

[Problem LIP]

$$\text{Min} \{ \sum_j y_j v_j + \sum_i \sum_j C_{ij} \chi_{ij} + \sum_d \sum_j (C^d_j + \sum_i r'' \cdot Q^d_i) \chi^d_i$$

$$+ \sum_d \sum_i \sum_j (r'_{ij} - r''_{ij}) Q^d_i \cdot x^d_j \cdot \chi_{ij} \}$$

Sub. to (1) - (4)

선형화 시키면,

$$\text{Min} \{ \sum_j y_j v_j + \sum_i \sum_j C_{ij} + \sum_d \sum_j LC^d_j \cdot \chi^d_j$$

$$+ \sum_d \sum_i \sum_j L r_{ij} \cdot Q^d_i \cdot \phi^d_{ij} \}$$

Sub. to (1) - (4)

$$\chi_{ij} + \chi^d_j - 1 \leq \phi^d_{ij} \qquad (5)$$

$$\phi^d_{ij} - \chi_{ij} \leq 0 \qquad (6)$$

$$\phi^d_{ij} - \chi^d_j \leq 0 \qquad (7)$$

$$\chi_{ij},\ \chi^d_j,\ y_i,\ \phi^d_{ij} \cdots\cdots \{ 0,1 \}$$

단계 2. 단계 1의 < Problem LIP > 를 제약식 (5)에 대해서 λ_{ijd} 로 제약식 (7)에 대해서 β_{ijd}로 Lagrangian Relaxation 시킨다.

$$\text{Min } \{ \sum_j y_j v_j + \sum_i \sum_j C_{ij}\chi_{ij} + \sum_d \sum_j LC_j^d \cdot \chi_j^d + \sum_d \sum_i \sum_j Lr_{ij} \cdot Q_i^d \cdot \phi_{ij}^d$$

$$+ \sum_d \sum_i \sum_j \lambda_{ijd} (\phi_{ij}^d - \chi_{ij} - \chi_j^d + 1)$$

$$+ \sum_d \sum_i \sum_j \beta_{ijd} (\chi_i^d - \phi_{ij}^d) \}$$

Sub. to (1) - (4), (6)

목적함수를 재배열하여 정리하면,

$$\text{Min } \{ \sum_j y_j v_j + \sum_i \sum_j (C_{ij} - \sum_d \lambda_{ijd}) \chi_{ij}$$

$$+ \sum_d \sum_j (LC_j^d - \sum_i \lambda_{ijd} + \sum_i \beta_{ijd}) \chi_j^d +$$

$$\sum_d \sum_i \sum_j (\lambda_{ijd} - \beta_{ijd} + Lr_{ij} \cdot Q_i^d) \phi_{ij}^d \} + \sum_d \sum_i \sum_j \lambda_{ijd}$$

새로운 계수를 도입하여 간단한 형태로 정리하면 다음과 같이 된다.

$$\text{Min } \{ \sum_j y_j v_j + \sum_i \sum_j RC_{ij} \cdot \chi_{ij} + \sum_d \sum_j RC_j^d \cdot \chi_j^d$$

$$+ \sum_d \sum_i \sum_j Rr_{ijd} \cdot \phi_{ij}^d \} + \sum_d \sum_i \sum_j \lambda_{ijd}$$

Sub. to (1) - (4), (6)

그런데, 변수 ϕ_{ij}^{d} 와 변수 χ_{ij} 사이에는 다음과 같은 성질이 존재한다.

$$\hat{\phi}_{ij}^{d} = \begin{cases} 0 & \text{if } \hat{\chi}_{ij} = 0 \\ 0 & \text{if } \hat{\chi}_{ij} = 1 \text{ and } Rr_{ijd} \geq 0 \\ 1 & \text{if } \hat{\chi}_{ij} = 1 \text{ and } Rr_{ijd} < 0 \end{cases}$$

단, $\hat{\phi}_{ij}^{d}$ 는 단계2의 [problem LRIP]에서, 고정된 승수백터에 대한 최적해이고, χ_{ij}는 이때 χ_{ij}의 fixed feasible solution.

그러므로 위의 관계를 바탕으로 변수 ϕ_{ij}^{d}를 변수 χ_{ij}에 관해 표현할 수 있게 된다.

$$\hat{\phi}_{ij}^{d} \cdot Rr_{ijd} = \hat{\chi}_{ij} \cdot \min \{ 0, \hat{Rr}_{ijd} \}$$

단계 3. 단계2의 [problem LRIP]를 위의 성질을 이용하여 나타내면 다음과 같다.

[Problem LRIP']

$$\text{Min} \{ \sum_{j} y_j v_j + \sum_{d} \sum_{j} RC_j^{d} \cdot \chi_j^{d} + \sum_{i} \sum_{j} [RC_{ij} + \sum_{d} \min(0, Rr_{ijd})] \chi_{ij} \}$$

Sub. to (1) - (4)

[단계 3] 까지의 procedure를 통해 얻어낸 [problem LRIP']는 그 형태가 simple uncapacitated plant location (SPL) 문제와 convertible한 것을 발견할 수 있다. 즉, J를 candidate plant location의 집합으로, I와 D의 union을 customer location의 집합으로 볼 수 있게 된다.

SPL problem은 NP-complete class의 문제점이나 이를 대상으로 개발된 algorithm들이 많이 있다. 그 중에서도 Erlenkottetr[8] 가 개발한 " a branch and bound dual based procedure for solving plant location problems"를 들 수 있는데 매우 효율적인 algorithm으로 알려져 있다. 그러므로, lagrange 승수 백터가 fixed되어 있다고 가정한다면 Erlenkotter의procedure를 [problem LRIP']에 적용시킬 수 있게 된다. Lagrangian relaxation의 성질에 의하면 주어진 lagrange 승수 백터에 대한 $L(\lambda, \beta)$는 optimal solution의 하한경계치가 된다.

한편, 가능한한 가장 tight한 하한경계치, 즉 최적의 lagrange 승수백터는 다음의 조건을 만족하는 것이 된다.

$$L(\lambda, \beta) = \text{maximize} \{ L(\lambda, \beta) \}$$

그런데, 이와같은 최적의 lagrange 승수 백터를 얻어내는 일은 간단한 일이 아니다.

현재, (λ, β)를 계산해 내는 approximation algorithm들이 많이 소개되어 있는데 대개의 경우 subgradient algorithm을 사용하고 있다.

Subgradient algorithm은 초기 lagrange 승수 백터를 가지고 시작해서 하한치를 개선시키는 방향으로 차례 차례 승수 백터를 구해나가는 방법이다.

4.2 Heuristic 해법의 개발

본 연구에서 개발한 heuristic 해법은 greedy type의 algorithm이다. 이러한 algorithm은 화일 배정문제나 컴퓨터 위치 선정 문제 각각이 모두 simple plant location problem (SPL)으로 취급될 수 있으며, 이 SPL에 대해서는 효율적인 한 algorithm이 이미 개발되어 있어서 이 algorithm을 이용하여 보려는 의도에서 부터 이루어졌다. 만일, 화일의 배치가 이미 결정되어 있다면 우리의 문제는 컴퓨터 위치 선정의단일 문제로 축소된 것을 의미하고, 반대의 경우로 컴퓨터 위치 선정이 결정된 상태라면 화일 배정문제가 되는 것이다.

Heuristic algorithm의 첫 단계는 feasible한 file의 배치를 초기선정하여 주는 것으로부터 시작된다. 주어진 화일배치에 대해 컴퓨터 위치 선정문제를 풀고, 다시 선정된 컴퓨터 위치에 대해 화일을 재배치 시켜서 개선된 해를 얻어낸다. 이러한 방법으로 더 이상 해의 개선이 없을 때까지 같은 과정을 반복한다.

구체적인 단계의 축소된 문제 형태를 보면 다음과 같다.

단계 1. 초기의 feasible file allocation을 정하여 준다.

단계 2. 화일 배치가 이미 결정된 상태 (즉, 변수 χ_j 가 이미 정하여진 상태) 이므로, [problem NIP]는 다음과 같은 문제로 축소된다.

< problem FIP >

$$\text{MIN} \{ \sum_{j \in \hat{J}_d} y_j v_j + \sum_i \sum_j \tilde{C}_{ij} \chi_{ij} \}$$

Sub. to

$$\sum \chi_{ij} = 1 \qquad \forall i$$

$$\chi_{ij} - y_i \le 0 \qquad \forall i, \ j \in \hat{J}_d$$

단, $J_d = J - \{ j \mid x_j = 1 \text{ and } d \in D, \ j \in J \}$

$$\tilde{C}_{ij} = C_{ij} + \sum_d R''_{ij} \cdot Q_i^d + \sum_{\hat{D}_j} (r'_{ij} - r''_{ij}) Q_i^d$$

$$D_j = \{ d \mid x_j^d = \text{and} \ \ d \in D \}$$

단계 3. 단계2에서 구한 각 branch의 컴퓨터 위치 선정에 관한 변수가 고정되어 있다고 가정하고 (즉, 변수 χ_{ij} 가 정해진 상태) 여기에 대해 file을 재배치 시켜 구해지는 해에 개선이 생기는지 살펴본다.

이때 [Problem NIP]는 다음과 같이 축소된다.

< Problem CIP >

$$\text{Min} \{ \sum_{j \in \hat{J}_i} y_j v_j + \sum_d \sum_j \overset{\approx d}{C}_j \overset{d}{\chi}_j \}$$

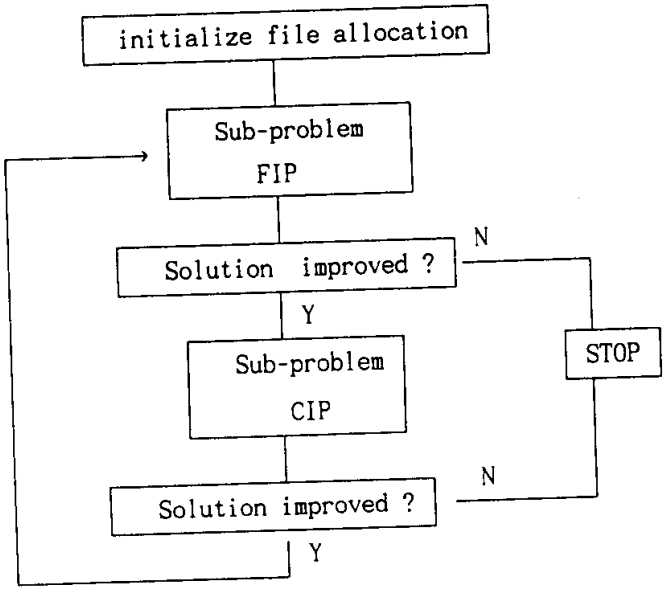
$$\text{Sub. to} \quad \sum_j \overset{d}{\chi}_j = 1 \qquad \forall d$$

$$\overset{d}{\chi}_j - y_j \leq 0 \qquad \forall d, \quad j \in \hat{J}_i$$

단, $\hat{J}_i = J - \{ j \mid \chi_{ij} = 1 \text{ and } i \in I, \ j \in J \}$

$$\overset{\approx d}{C}_j = \overset{d}{C}_j + \sum_i r''_{ij} \cdot \overset{d}{Q}_i + \sum_{\hat{I}_j} (r_{ij}' - r_{ij}'') \overset{d}{Q}_i$$

$\hat{I}_j = \{ i \mid \chi_{ij} = 1 \text{ and } i \in I \}$



<그림 3> heuristic 방법의 iterative process 흐름도

단계 2n 단계 3의 해를 바탕으로, 단계2와 같은 방법으로 해에 개선이 있는지 살펴 본다.

단계 2n+1 단계 2n의 해를 바탕으로, 단계3의 방법을 사용, 해의 개선이 있는지 살펴본다.

더이상 해에 개선이 없을 때까지 위의 과정을 반복한다.

4.2 해법의 적용 예

4.2 절에서 제시한 heuristic algorithm은 IBM - AT computer에 program 되어서, 여러 가지 경우의 예에 대해 실험되었다.

본 절에서는 4가지 경우의 예에 대한 적용결과를 제시하고 있다. 아래의 [표 1]에 그 결과를 요약하여 놓았다. 처음의 두가지 예는 매우 작은 size의 문제이고, 이러한 경우는 전형적인 IP algorithm으로도 적용가능한 시간내에 optimal solution을 얻어 낼 수 있다. 그러나 이러한 경우에도 heuristic algorithm은 최적해에 근접한 해를 더 빠른 시간 내에 찾아낼 수 있었다. 또 다른 두가지 경우는 현실상황에서의 분산 정보 시스팀 문제 size에 부합하는 예이다.

앞서 발표된 논문들의 결과에 따르면 이러한 경우는 IP algorithm으로는 solution을 제공할 수 없는 것이었다. 본 연구에서 제시한 heuristic algorithm에 의하면 이러한 경우에도, 두 예제 모두 2번의 iteration 만에 solution을 제공할 수 있었으며, computaion time은 2분 이내였다. 한편, heuristic한 방법으로 구해지는 해는 feasible solution 이기는 하지만 optimal solution 이라는 보장은 할 수 없다.

최적해를 알 수 없으므로 heuristic한 방법으로 찾아진 solution을 평가할 수 있는 기준이 필요한데, 이러한 필요성에 따라 손쉬운 방법으로 최적해에 대한 하한치를 구하는 방법을 연구, 사용하였다. 최적해에 대한 하한치는 앞의 4.1절에서 제시한 Lagrangian Relaxation Procedure를 통해서도 얻어 낼 수 있으나 이

방법은 계산량이 많아 실제적이지 못하므로, 이 방법으로 보다 tight한 bound는 기대할 수 없으나 실제적인 면에서 손쉽게 하한치를 구할 수 있는 방법을 사용한 것이다.

〈표- 1〉 해법 적용의 결과에 대한 요약

문 제	문 제 size			최적해에 대한 하한치	Heuristic 해법의 적용결과	
	사용자의 수	컴퓨터 위치의수	분할된데이터 베이스의 수		solution	iteration 횟수
1	4	4	4	17,017	17,017	1
2	4	9	13	20,575	21,005	1
3	23	10	10	116,034	124,443	2
4	23	10	10	120,012	124,832	2

5. 결 론

본 연구에서는 computer location assignment problem과 file allocation problem을 동시에고려한 모형을 formulation하여 그 해를 구해냄으로써 distributed information-system 구축시 기존의 연구 해법으로 얻은 결과보다 시스팀 전체적인면에서 볼 때 최적해에 가까운 해를 제공할 수 있었다.

또한, 이제까지의 solution methodology가 네트워크의 제한이 주어진 상태에서 주어졌고, 대상으로 할 수 있는 시스팀의 규모가 실제상황에는 비현실적이었던 것에 비해 수직적인 해법을 이용함으로써 다룰 수 있는 system size (즉, branch의 갯수, computer site의 수)가 커지고, 이전보다 적은 computing-time내에 시스팀 구축에 필요한 최소비용 및 디자인 방법을 얻어낼 수 있었다.

參 考 文 獻

P. Chen and J. Akoka, "Optimal design of distributed information system, IEEE Trans. Comput., Vol. C-29, pp. 1068-1080, Dec. 1980.

B. Gavish and H. Pirkul, "Computer and database location in distributed computer systems," IEEE Trans. Comp., Vol. D-35, pp. 583-590, Jul. 1986.

B. Gavish, "Models for configuring large scale distributed systems," AT & T Technical Jour., Vol. 64, pp. 491-532, Feb. 1985.

K.B. Irani and N.G. Khabbaz, "A methodology for the design of communication network and the distribution of data in distributed supercomputer
systems, "IEEE Trans. Comp Vol. C-31, pp. 419-434, May 1982.

S. Mahmoud and J.S. Riordon, "Optimal allocation of resources indistributed information networks, "ACM Trans. Database Sys., Vol.1, pp. 66-78, Mar. 1976.

A. Mirzaian, "Lagrangian Relaxation for the star-star concentrator location problem: Approximation algorithm and bounds, "Networks, Vol. 15, pp. 1-20, 1985.

S. Ceri and G. Pelagatti, Distributed database: Principles & Systems, McGraw-Hill, New york, 1984.

D. Erlenkotter, "A dual-based procedure for uncapacitated facility location, " Oper. Res. Vol. 26, pp. 992-1009, Nov. 1978.

B. W. Wha, "Fille placement on distributed computer systems," Computer, Vol. 17, pp. 23-32, Jan. 1984.

1.T. Lanning and M.S. Leonard, "File allocation in a distributed communication network," IEEE Trans. Comp., Vol. c-32, pp. 232-244.

W.W. Chu, "Optimal file allocation in a multiple computer system," IEEE Trans. Comp., Vol. C-18, pp. 885-889, Oct. 1969.

B. Gavish, "Augmented lagrangian based algorithms for centralized networks design," IEEE Trans. Comm., Vol. COM-33, pp. 1247-1257, Dec. 1985.

E.G. Coffman, E. Gelenbe and B. Plateau, "Optimization of the number of copies in a distributed database," IEEE Trans. Soft. Eng., Vol. SE-7, pp. 78-84, Jan. 1981.

S. Muro and T. Ibaraki, "Evaluation of the file redundancy in

distributed database systems," IEEE Trans. Soft Eng., Vol. SE-11, pp. 199-204, Feb. 1985.

C.V. Ramamoorthy and B.W. Wah, "The isomorphism of simple file allocation", IEEE Trans. Comp., Vol. C-32, pp.221-231,Mar. 1983.

H.L. Morgan and K.D. Levin, "optimal program and data location in computer networks, "comm. of ACM, Vol.20, pp. 315-322, May 1977.

H.L. Morgan and K.D. Levin, "A dynamic optimization model for distributed databases," Oper. Res., Vol. 26 pp. 824-835, Sep. 1978.

S. Ceri and G. Pelagatti, "A solution method for nonadditive resource allocation problem in distributed system design," Information Proc. Letters, Vol. 15, pp. 174-178, Oct. 1982.

M.L. Fisher, "The lagrangian relaxation method for solving integer programming problems," M.S., Vol.27, pp. 1-18, Jan. 1981.

R.G. Casey, "allocationof copies of a file in an information networks," Proc. 1972 spring joint computer conference, AFIPS, 1972.