

(19) 대한민국특허청(KR)

(12) 특허공보(B1)

(51) Int. Cl.⁵

H04M 7/00

H04Q 3/66

G06F 15/20

(45) 공고일자 1994년 10월 17일

(11) 공고번호 94-009780

(21) 출원번호

특 1986-0700887

특 1987-7000253

(22) 출원일자

1986년 12월 12일

1987년 05월 30일

(86) 국제출원번호

PCT/US 86/000569

(87) 국제공개번호 WO 86/06569

(86) 국제출원일자

1986년 03월 28일

(87) 국제공개일자 1986년 11월 06일

(30) 우선권주장

725, 342 1985년 04월 19일 미국(US)

(71) 출원인

아메리칸 텔리폰 앤드 텔레그라프 캠퍼니 엘리 와이스

미합중국, 뉴욕 10022, 뉴욕, 매디슨 애비뉴 550

(72) 발명자

나렌드라 크리쉬나 카르 마르 카르

미합중국, 뉴저지 07060, 노스 플레인필드, 웨스터벨트 애비뉴 114

(74) 대리인

이병호

심사관 : 김성배 (책자공보 제3777호)(54) 자원 할당방법 및 장치**요약**

내용 없음.

대표도**도1****영세서**

[발명의 명칭]

자원 할당방법 및 장치

[도면의 간단한 설명]

제 1 도는 선형 프로그래밍 문제에서 최적 자원 할당을 결정하는 본 발명의 방법을 그래프적으로 표시한 도면.

제 2 도는 선형 프로그래밍 모델에서 최적 자원 할당을 결정하는 종래의 심플렉스법을 그래프적으로 표시한 도면.

제 3 도는 선형 프로그래밍 문제에서 최적 자원 할당을 결정하는 종래의 타원체법을 그래프적으로 표시한 도면.

제 4 도는 본 발명에 따른 선형 프로그래밍 방법의 개괄적인 순서도.

제 5 도는 최적 자원 할당을 결정하기 위해 사영 변환(projective transformations)을 이용하는 본 발명의 여러 방법에 대한 더욱 상세한 순서도.

제 6 도는 자원 할당을 제어하기 위해 제 4 도 및 제 5 도의 방법을 이용하는 자원 할당 시스템의 블록도.

제 7 도는 본 발명을 이용할 수 있는 간략한 통신 루팅 할당 문제를 그래프적으로 표시한 도면.

제 8 도는 본 발명에 따라 구성되어 제 4 도 및 제 5 도의 방법을 이용하는 전화 루틴 장치의 일반적인 블록도.

[발명의 상세한 설명]

[기술적 분야]

본 발명은 다수의 자원 사용자 사이에 자원을 할당하는 시스템에 관한 것이다. 특히, 그러한 할당에 대한 이익을 최대로 하거나 또는 비용을 최소로 하기 위해 기술적 및 산업적 자원의 할당을 효과적으로 최적화하는 장치 및 방법에 관한 것이다.

[발명의 배경]

자원 할당 결정에 대한 필요는 전화 전송 시스템의 전송 설비 배당, 공장의 생산물 혼합 관리, 산업 장비의 배치, 재고품 관리, 및 기타 등등과 같은 광범위한 기술적 및 산업적 영역에서 발생한다. 이와 관련하여, 자원 할당은 일반적으로 특별한 기술적 또는 산업적 결과의 생산물에 대한 특정한 기술 또는 산업적 자원의 배치를 의미한다.

자원 할당 결정은 전형적으로 그러한 할당상의 제한을 받는다. 자원은 총 이용도(availability)에서 항상 제한을 받으며, 더 나아가, 다소 특별한 적용에서 특별한 자원의 유용성도 또한 제한을 받을 수 있다. 예를 들면, 전기 통신 시스템에 제공되는 전체 트래픽은 제한되며, 통신 시스템에서 개별 링크 각각의 트래픽 운반 용량도 또한 제한된다. 각각의 특별한 자원 할당은 "페이오프(payoff)" 즉, 그 할당의 비용 또는 할당이익(즉, 수익)에 연관될 수 있다. 그때의 문제는 모든 제한을 만족시킴과 동시에 페이오프를 최대로 하도록 즉 비용을 최소로 하거나 또는 이익을 최대로 하도록 모든 자원을 할당하는 것이다.

대표적인 할당 결정 문제의 한 방법은 선형 프로그래밍 모델이라 불리운다. 이 모델은 있을 수 있는 다양한 할당과 그에 따른 제한들, 및 비용 또는 그에 따른 이익들 사이의 정량적 관계를 표시하는 다수의 선형 표현으로 이루어져 있다. 만일 모든 관계들이, 정수와 같거나 또는 그 이상이거나 또는 그 이하인 미지 할당값의 계수배들의 합이라면 이 일련의 관계들은 선형적이라 말한다. 물론, 많은 자원 할당 문제들이 이러한 선형 관계로 표시되어지는 것은 아니지만 미지수 또는 다른 비선형의 고차 제곱들을 관계식에 관련시키므로 선형 프로그래밍 방법에 영향을 미치지는 않는다.

전술한 자원 할당 문제들은 실제 물리적 시스템에서 일어나는 실제의 문제라는 사실에 주의해야 한다. 물리적 문제의 중요한 양적 양상은 선형 프로그래밍 모델로 표시될 수 있는 반면, 이 모델의 목적은 물리적 시스템을 구성 또는 작동시키기 위해 물리계에서 사용되는 최적 값을 제공하는 것이다. 물리적 시스템의 특성을 기술하는 이러한 수학적 모델을 사용하는 전형적인 종래 기술의 예는 라디오 안테나를 구성하거나 고무-주형 동작을 제어하는 방정식의 사용이다.

한때, 전술된 많은 자원 할당 문제들은 인간의 직관 또는 경험을 이용하여 해결되었다. 최근에는 통계, 모델링, 그래프 및 선형 프로그래밍과 같은 정량적 방편들이 의사 결정 행위에서 인간을 보조하기 위해 이용되고 있다. 예를 들면, 제조 설비들은 선형 프로그래밍 모델을 사용하여 판매 수요를 만족시킬 재고품 레벨 및 생산품 예정표를 제어하며 동시에 생산품 및 재고품 비용을 최소화한다. 유사하게, 통신 시스템은 전체 트래픽 요구를 만족시키고 전송 링크의 과부하를 방지하며 전송 비용을 최소화하기 위해 선형 프로그래밍 모델을 이용하여 전화 트래픽을 전송 설비망을 통해 송신한다.

선형 프로그래밍 모델로서, 제기된 할당 문제를 해결하는 가장 공지된 종래의 기술은, 1947년 조지 비. 댄지그씨에 의해 창안되어, 1963년, 뉴저지, 프린스턴, 프린스턴 대학신문에서

Linear Programming and Extension에 기재된 심플렉스(simplex)법이다. 심플렉스 법에 따르면, 제1단계는 초기의 실행가능한(feasible) 할당을 시작점으로 선택하는 것인데, 이는 원래 모델의 변형인 다른 선형 프로그래밍 모델을 사용할 수도 있다. 이 경우, 실행가능한 할당이란 모든 제한들을 만족은 시키지만 최적이라고 알려진 것은 아니다. 그후 최적화되어질 함수(목적 함수라 칭함)를 개선하는 새로운 할당들이 연속해서 확인된다. 이 과정은 반복적으로 되풀이되어 항상 최적 할당에 더 근접하는 새로운 일시적 할당들이 선택된다. 이 반복 과정은 현재의 일시적 할당이 더 이상 개선되어지지 않을 때 정지한다.

이 심플렉스법은 제 2 도에 도시된 선형 프로그래밍 모델의 간략화된 그래프를 참조하면 더 잘 이해될 수 있다. 제 2 도에는 다수의 각면(facet)을 갖는 볼록한 폴리토프(polytope)(10)의 3차원적 표시가 도시되어 있다. 폴리토프(10)의 각 각면들은 정상 선형 프로그래밍 모델에서 제한 관계들의 일부분에 대한 그래프적 표시이다. 즉, 각각의 선형 제한은 폴리토프(10)의 공간에서 평면으로 정의되며, 그 평면의 부분은 폴리토프(10)의 각면을 형성한다. 폴리토프(10)는 폴리토프(10)의 표면상의 임의 두 점을 잇는 선이 폴리토프(10) 내부에 있다는 점에서 볼록면체이다.

폴리토프(10)가 단지 예시적인 목적으로 3-차원적 다각형으로 도시된 것에 유의하라. 사실상, 선형 프로그래밍 모델의 폴리토프 표현은 미지의 할당값의 수와 동일한 (제 2 도에서와 같이 도시될 때) 또는, 더 자세히 말하자면, 부등(inequality) 제한 관계의 수에서 대등(equality) 제한 관계의 수를 뺀 수와 동일한 다수 차원을 갖는 4차원 이상의 공간에 속한다. 실제로, 폴리토프는 4차원 이상의 공간을 폴리토프(10)의 실행가능 영역 및 폴리토프(10) 외부의 실행 불가능 영역으로 분할한다.

선형 프로그래밍 모델에서 최적 자원 할당이 폴리토프(10)의 정점에 있다는 것은 널리 공지된 사실이다. 몇몇 모델에서, 폴리토프(10)의 모든 모서리 또는 모든 각면들은 최적 할당을 표시하며, 그리하여 정점이 하나 이상일 때 최적화될 수 있다. 심플렉스법의 전략은 폴리토프(10)의 인접 정점들을 연속적으로 확인하는 것이며, 여기에서 각각의 새로운 정점(새로운 실행가능한 일련의 할당들을 각기 표시하는 점)은 목적 함수로 측정된 바와 같이 이전의 할당보다 최적점(21)에 더 근접한다. 제 2 도에서, 심플렉스법은 먼저 정점(12)을 확인할 것이며, 그 다음에 최적점(21)에 도달할 때까지 경로(13)를 따라 정점에서 정점으로 ((14)에서 (20)까지) 이동한다. 그리하여 심플렉스법은 폴리토프(10)의 표면상에서 이동하도록 특히, 폴리토프(10)의 한 정점(12)에서 인접 정점(즉, (14)으로 이동하도록 제한된다. 수천, 수십만 또는 수백만의 변수와 관련된 많은 선형 프로그래밍 문제에 있어서, 폴리토프상의 정점의 수는 상용하게 때로는 지수적으로 증가한다. 경로(13)의 길이는 평균적으로 변수의 갯수에 정비례하여 증가한다. 더욱이, "최악의 상황"이라 불리우는 문제가 있는데, 그것은 최적 정점에 도달하기 위해 대부분의 정점들을 거쳐야만 하는 폴리토프의 토플로지(topology)이다.

이런 저런 요인의 결과로서, 심플렉스법에 의해 선형 프로그래밍 모델을 풀이하는데 필요한 평균 계산 시간은 적어도 이 모델에서의 제한들의 제곱에 비례한다. 적당한 규모의 할당 문제에서 조차도 이 시간은 종종 너무 커서, 최적 할당이 계산되기 전에 제한들이 변하는 경우는 이 모델을 사용하는 것

이 실용적이지 못하거나 또는 모델을 사용하여 할당을 최적화시키는데 필요한 계산 시간(아마 컴퓨터상에서)이 너무 커서 적당한 비용으로 단순히 그 목적용만으로는 이용할 수 없다. 최적 할당은 일반적으로 진행중인 처리, 시스템 또는 장치에 다소간의 연속 제어를 제공하기에 충분히 빠른 "실시간(real time)" 내에 만들어지지 못한다.

선형 프로그래밍 모델을 착수하는 두번째 방법은, 1970년 소련의 엔. 제트. 쇼어씨가 발명한, 타원체법이라 불리우는 방법으로, 1979년도판 **Doklady Akademii Nauk** SSSR 244 : S, **Soviet Mathematics Doklady** 1, 191면 내지 104면에도 기재되어 있음) 엘.지.카치얀씨의 논문, "선형 프로그래밍의 다항식 알고리즘"에 기술되어 있다. 타원체법에서 제 3 도의 폴리토프(30)는 중심점(32)을 갖는 타원체(31)에 둘러사인다. 타원체(31)의 중심점(32)은 이것이 폴리토프(30)의 내부에 있는지 또는 외부에 있는지를 알기 위해 검사된다. 만일 점(32)이, 제 3 도에 도시된 바와 같이, 폴리토프(30)의 외부에 있다면, 평면(33)은 폴리토프(30)의 어떤 각면과 평행하게 중심점(32)을 통과하여 점(32)이 그 각면을 포함하는 제한의 반대쪽(외부)에 있게 된다. 그때 타원체(31)의 어느쪽 반분이 폴리토프(30)를 포함하는지가 결정된다. 제 3 도에서는 타원체(31)의 상반분이다.

그 다음, 중심점(35)을 갖는 두번째의, 더 작은 타원체(34)가 타원체(31)의 상반분 주위에 구성된다. 다시, 중심점(35)이 폴리토프(30)이 내부 또는 외부에 있는지 검사된다. 만일 이것이 외부에 있다면(제 3 도), 상기 과정은 중심점이 폴리토프(30)내에 있을 때까지 반복된다. 둘러싼 타원체의 중심점이 폴리토프(30)내에 있을 때, 중심점을 통과하는 평면은 폴리토프(30)를 두 구획으로 나누도록 목적 함수의 방향에 대해 수직으로 그려진다. 그때 폴리토프(30)의 어느쪽 구획이 최적점(36)을 포함하는지 결정된다. 또 다른 타원체는 최적점(36)을 포함하는 폴리토프(30)의 반분 주위에 구성되며, 또 다른 평면은 그 중심점을 통과하고, 타원체의 어느쪽 반분이 최적점(36)을 포함하는지 확인하기 위해 시험이 수행된다. 이 과정은 중심점이 최적점(36)과 어느 정도 일치할 때까지 계속된다. 중심점의 좌표는, 제1사례에서 모델을 표현하기 위해 사용된 넘버링 시스템(numbering system)의 정확도내에서 점(36)으로 표시된 최적 할당의 정확한 값으로 "포위(round)"될 수 있다.

Operations Research.

제29권, 제6

호(1981), 1039-1091페이지의 "타원체 방법 : 관찰"에서 알.지.브랜드씨등에 의해 논의된 이유때문에 대부분의 선형 프로그래밍 모델에 대해 심플렉스법보다 더 느리게 실행된다.

선형 프로그래밍 모델을 착수하기 위한 새롭고 더 효과적인 방법 또는 절차가 요구된다. 예를들어, 국내 전화 네트워크를 통한 장거리 전화 트래픽의 최적 루팅(routing)은 방대한 분량의 가능한 연결, 그와 연관된 비용 및 제한들에 관련된다. 이러한 특별한 문제의 해결은 합당한 시간, 즉, 최적값을 이용하기 위해 트래픽 루팅을 조정함으로써 그 해결법을 실제 사용할 수 있을 정도로 충분히 짧은 시간내에 이용할 수 있어야 한다. 이것은 본 발명이 지향하는 이러한 유형의 문제를 해결하는 방법이다.

[발명의 요약]

본 발명의 예시적인 실시예에 따라, 최적 자원 할당은 종래 기술의 자원 할당 절차에 의해 가능했던 것보다 더 빠르게 성취된다. 특히, 본 발명의 원리를 이용함으로써, 다수의 선형 프로그래밍 모델이 "실시간"에서, 즉, 시스템 또는 장치의 연속 제어를 다소 허용할 수 있을 정도로 충분히 빠르게 해결되어질 수 있다. 다른 할당 문제들은 종래 기술의 선형 프로그래밍 해결책이 경제적으로 실행 불가능한 경우 경제적으로 관심을 끄는 선형 프로그래밍 해결책으로서 충분히 빠르게 해결되어질 수 있다. 마지막으로, 너무 방대해서 선형 프로그래밍이 가능한 해결책으로서 간주되지 못하는 다수의 할당 문제들이 본 발명에 따른 선형 프로그래밍 해결책을 사용함으로써 이제는 해결되어질 수 있다.

이하에서 엄밀히 정의될, 이러한 현저히 개선된 이익을 얻기 위한 절차는 제 1 도의 폴리토프(50)를 고려함으로써 이해될 수 있다. 제 2 도 및 제 3 도의 폴리토프(10 및 30) 같이 폴리토프(50)는 제한 평면에 대응하는 다수의 각면들과 평면의 교차점에 대응하는 다수의 가장자리들(edges), 및 가장자리의 교차점에 대응하는 다수의 정점들을 갖는다. 폴리토프(50) 표면상의 각 점 및 폴리토프(50) 내부의 각 점은 실행가능한 자원의 할당, 즉 모든 제한을 충족시키는 할당을 표시한다. 각 점의 좌표는 할당값의 크기이다.

본 발명에 따라, 폴리토프(50) 내부의 점(51)이 시작 할당점으로서 선택된다. 그 다음 연속 단계(52, 55 및 56)들이 최적 할당점(53)을 향해 궤도를 따라 폴리토프(50) 내부에서 취해진다. 연속 단계들의 크기는 인접한 정점 간격(심플렉스법에서 처럼)에 의해 한정되지 않기 때문에, 더 넓은 단계가 확보될 수 있으며 더 적은 수의 단계가 필요해지며, 최적 할당을 확인하기 위해 요구되는 시간은 짧아진다.

특히, 폴리토프(50) 내부에서 임의적으로 또는 체계적으로 선정된 할당점(51)은 시작점으로서 이용된다. 선형성 및 볼록면체를 유지하는 할당 변수들의 선형 변화를 이용하여, 선형 프로그래밍 모델에서의 변수들은 시작점이 변환 폴리토프의 사실상 중심에 있으며 모든 각면들이 중심에서 어느 정도 등거리에 있도록 변환된다. 이러한 등거리 배정(equidistancing) 절차는 정규화(normalization), 센터링, 등거리 배정 정규화, 정규화 변환, 또는 센터링 변환으로 불리울 수 있다. 다음 할당점은 목적 함수의 음의 기울기 방향(가장 가파른 경사 방향)으로 이동시킴으로써 그리고 폴리토프의 경계에 의해 제한되는 거리에 의해 선택된다(폴리토프 내부를 벗어나지 못하게). 마지막으로, 새로운 할당점을 원래 변수, 즉, 원래 폴리토프의 공간으로 복귀시키기 위해 역변환(이 새로운 할당점상에서 실행된다. 변형된 새로운 점을 새로운 시작점으로서 이용하여, 전체 과정이 반복된다.

각각의 단계는 폴리토프 표면상의 원주 형태이기보다는 폴리토프내에서 방사상이기 때문에, 많은 더 적은 단계들이 최적점에 수렴하기 위해 필요하다. 일단 선택된 내부 점은 최적점에 충분히 근접하여

원래 제안된 문제가 갖는 정확도의 한계내에 존재하므로, 최적점은 원래 문제의 정확도로 값을 근접시키거나, 최적한 해결을 포함하는 제한들(각면들)을 확인하거나, 또는 종래 기술에서 유용한 다른 정지 기준중 어느것에 의해 확인되어질 수 있다.

본 발명의 주요한 장점은 자원 할당 변수들의 값이 얻어지는 속도이다. 이 속도는 심플렉스법 및 타원체법으로 현재 제공된 것보다 더 능률적인 최적 자원 할당 계산을 만들뿐만 아니라, "실시간"에서 자원을 할당하는 실용 가능성을 최초로 허용한다. 또한 본 발명은 시스템에서 이전에는 너무 커서 종래 기술방법에 의해 알맞은 시간내 실행되어지지 않았던 자원 할당을 최초로 허용한다.

진행중인 처리를 제어하기 위해 동적으로 사용되어질 정도로 충분히 빠른 "실시간"에서 자원을 할당하는 가능성은, 환경(제한)이 변화하는 것처럼 연속적으로 자원을 최적하게 할당 및 재할당함으로써 제어되어질 제조 또는 조립 처리, 항해 처리 및 전화 루팅 처리를 허용한다. 게다가, 어떠한 실용적인 시간 분량으로 제어될 수 없었던 매우 큰 자원 할당용 시스템이 이제 본 발명의 기술을 이용하여 신속히 제어되어진다.

[발명의 상세한 설명]

선형 프로그래밍 모델로 최적 자원 할당을 만드는 새롭게 구성된 방법이 먼저 논의되고 그 다음에 기술적 및 산업적 자원 할당 시스템, 장치 및 절차에서의 이 방법의 사용이 논의될 것이다.

선형 프로그래밍 모델의 형식은 최대화되어지거나 또는 최소화되어질 목적 함수와, 수용가능한 할당상의 물리적 제한을 표현한 다수의 제한 관계들의 형태를 취한다. 이러한 제한들은 물리적 시스템에 존재하는 실제 물리적 제한들을 가능하면 정확하게 표시하며, 이에 대응한다. 표준 벡터 표시법에서, 전형적인 선형 프로그래밍 모델은 다음과 같이 표현된다.

$$\text{최소화} : \mathbf{C}^T \mathbf{X}$$

$$\text{조건} : \mathbf{A}\mathbf{x} = \mathbf{b} \text{ 및 } (1)$$

$$\mathbf{L} \leq \mathbf{x} \leq \mathbf{U}$$

가 되도록 길이 n 의 벡터 \mathbf{x} 를 찾아라. 여기에서 $\mathbf{C} = (C_1, C_2, \dots, C_n)$ 은 비용계수의 벡터이고, 위첨자 T 는 행렬 전치(transposition) 연산을 표시하고, $\mathbf{x} = (x_1, x_2, \dots, x_n)$ 은 할당값의 벡터이고, n 은 그러한 할당값의 갯수이며, $\mathbf{A} = (a_{11}, a_{12}, \dots, a_{mn})$ 은 $m \times n$ 제한 계수 행렬이며, $\mathbf{b} = (b_1, b_2, \dots, b_m)$ 은 m 정수 벡터이며 $\mathbf{L} = (l_1, l_2, \dots, l_n)$ 및 $\mathbf{U} = (U_1, U_2, \dots, U_n)$ 은 각각 \mathbf{x} 값상의 하한 및 상한 경계이다.

전형적으로, \mathbf{x} (할당값)의 성분 값은 음수가 아닌 값으로 한정되지만, 다른 제한들은 가능하다. 모든 목적 함수 및 모든 제한 관계는 간단한 대수학 조작에 의해 이 형태로 축소될 수 있다. 예를 들어, "크거나 같은" 제한들은 제한 행렬에 인위적인 "잉여(surplus)" 변수들을 더함으로써 "대등" 제한으로 변화될 수 있다. 유사하게, "작거나 같은" 제한들은 인위적인 "부가(slack)" 변수들을 더함으로써 "대등" 제한으로 변화될 수 있다. 이러한 기법은 종래 기술에서 널리 공지된 사실이다.

본 발명에 따라, 심플렉스법 및 타원체법 양쪽의 단점은 선형 프로그래밍 모델로 자원을 할당하는 전적으로 다른 방법을 사용함으로써 극복된다. 심플렉스법은 \mathbf{x} 의 다양한 성분들 \mathbf{x}_i 중 어느것이 최적 \mathbf{x} 의 경계 $(\mathbf{x}_i = 0)$ 에 있을 것인가 추측하고, 할당 성분 \mathbf{x} 의 최적 집합(set)이 얻어질 때까지, 알고리즘 진행에 따라, 한번에 \mathbf{x} 의 한 성분씩 이 추측을 개정한다. 본 발명의 방법에 따라, $\mathbf{Ax} = \mathbf{b}$ 및 $\mathbf{L} \leq \mathbf{x} \leq \mathbf{U}$ 가 되도록 (폴리토프 내부에서) 강하게 실행가능한 \mathbf{x} 의 성분들이 선택된다. "강하게 실행가능한 (strictly feasible)"이란 모든 제한들을 만족은 시키지만 경계값은 아닌 값을 의미한다. 변수의 선형 변화가 \mathbf{x} 성분에 만들어지고, 이는 \mathbf{x} 성분에 대응하는 변화된 변수 성분의 단위 변화가 원래 \mathbf{x} 성분에서의 더 작은 변화로 다시 환언되어 (translate) 경계에서 멀리 떨어진 \mathbf{x} 성분에 대응하는 변화된 성분이 될 것이다. 이 처리는 정규화, 센터링, 등거리 정규화, 정규화 변환 또는 센터링 변환으로 불리운다. 그 다음 가장 가파른 경사의 방향이 새로운 변수에서 결정되고 원래 변수에서의 단계 방향으로 다시 환언된다. 강하게 실행 가능한, 즉, $l_i \leq x_i^{\text{new}} \leq U_i$ 인 새로운 \mathbf{x} 성분을 유지하게 하는 방향 및 크기로 단계가 취해진다.

상기 절차는 제 4 도에서 요약된다. 제 4 도에 도시된 바와 같이, 박스(160)에서 선형 프로그래밍 모델을 공식화하는 것이 첫째로 필요하다. 그 다음, 강하게 실행가능한 시작점 $\mathbf{x}^{\text{start}}$ 인 박스(161)에서 선택되며, 현재 반복(current iterate) \mathbf{x}^{curr} 이 박스(162)에서 시작점 $\mathbf{x}^{\text{start}}$ 으로 설정된다. 강하게 실행가능한 시작점을 선택하는 기술은 이후에 기술될 것이다. 일정 쇄선으로 된 박스(163)에 포함된 제 4 도의 나머지 부분은 본 발명에 따른 절차의 반복부이다.

제 4 도의 반복 절차(163)는 다음의 단계들을 포함한다. \mathbf{x} 성분의 강하게 실행가능한 반복이 가정된다 :

- 1) 박스(164)에서, 경계에 대하여 현재 반복을 정규화할 변수의 변화를 선정한다 :
- 2) 박스(165)에서, 새로운 변수에서 가장 가파른 경사 방향을 계산하고 그 방향을 원래 변수로 다

시 환언한다 :

3) 박스(166)에서, 또한 강하게 실행가능한 $\underline{\mathbf{x}}$ 성분의 새로운 반복을 유지하는 크기만큼 환언된 방향으로 진행한다 : 및

4) 판정 박스(167)에서, 어느 중대한 개선이 목적 함수에서 관찰되지 않을 때 절차를 종료한다. 그 렇지 않으면, 박스(169)에서 현재 반복 $\underline{\mathbf{x}}^{\text{curr}}$ 과 동일한 새로운 $\underline{\mathbf{x}}^{\text{next}}$ 를 설정하고 단계(1) 내지 (4)를 반복하도록 박스(164)로 복귀한다.

반복 절차를 정지시키는 한가지 방법은 "프라이멀(primal)" 선형 프로그래밍(LP) 모델 및 "듀얼(dual)" LP모델 양쪽을 동시에 해결하는 것이다. 만일 프라이멀 모델이 다음과 같이 표현된다면,

$$\text{최소화} : \mathbf{c}^T \mathbf{x}$$

$$\text{조 건} : \mathbf{Ax} = \mathbf{b} \text{ 및}$$

$$\mathbf{x} \geq \mathbf{0}$$

그때 듀얼 모델은 다음과 같이 표현될 수 있다.

$$\text{최대화} : \mathbf{U}^T \mathbf{b}$$

$$\text{조 건} : \mathbf{A}^T \mathbf{U} \leq \mathbf{c}$$

이러한 두 모델은 동일한 최적 목적 함수를 갖게 되지만, 반복 절차는 반대 방향으로부터 이러한 최적 값에 근접한다. 최적값은 현재 프라이멀 목적 함수 값과 현재 듀얼 목적 함수값 사이의 충분히 작은 차를 선택함으로써 간단히 가깝게 근접될 수 있다. 다른 정지 절차는 종래 기술로도 이용 가능하며 또한 사용될 수도 있다.

본 발명의 방법은 풀리토프 표면상의 이동에 관련되지 않을뿐 아니라 인접 정점의 간격에 인해 단계-크기가 제한되지도 않는다는 것에 유의하라. 그 결과, 본 발명의 방법은 본질적으로 최적점을 향하여 더 직접적으로, 그리고 더 적은 단계로 이동할 수 있다. 본 발명은 실질적으로 모든 LP모델용의 심플렉스법 및 타원체법에 대해 속도의 장점을 제공할 뿐만 아니라, 이 장점은 모델의 크기(변수의 수)에 따라 증가한다. 그러므로 해들이 더 이상 유효하지 못하여 쓸모없어질 만큼 문제가 많이 변하기 전인 실시간에서 활용될 수 있을 정도로 빠르게 선형 프로그래밍 모델을 해결하는 것이 가능해진다. 게다가, 심플렉스법 또는 타원체법으로는 합당한 비용으로 해결될 수 없는 큰 선형 프로그래밍 모델(많은 변수들과 관련있는)을 해결하는 것이 가능해진다.

상기 절차의 중요한 양상들 중 하나는 단계(1)에서 변수 변화의 설정이다. 이러한 변수들의 변화는 대각선 스케일 행렬 \mathbf{D} 로 표시될 수 있다. 현재 반복을 모든 경계로부터 어느 정도 동일한 거리상에 배치시키는 정규화 함수를 실행하기 위하여, \mathbf{D} 의 i 번째 대각선 기재사항(entry)의 값은 $\underline{\mathbf{x}}_i$ 가 L_i 또는 U_i 의 근처에 있을 때 작아져야 한다. \mathbf{D} 의 i 번째 대각선 기재사항을 위한 명백한 설정은

$$D_{ii} = \min\{1, \underline{\mathbf{x}}_i^{\text{curr}} - L_i, U_i - \underline{\mathbf{x}}_i^{\text{curr}}\} \quad (2)$$

이다. 여기에서 $\underline{\mathbf{x}}^{\text{curr}}$ 은 $\underline{\mathbf{x}}$ 의 현재 반복이다. 만일 경계가 매우 큰 값이거나, 또는 $\underline{\mathbf{x}}$ 가 양 또는 음방향으로 경계되지 않았다면, 합당한 경계가 그럼에도 불구하고 \mathbf{D} 상에 배치되어져야 한다. 즉,

$$D_{ii} = \min\{1, \underline{\mathbf{x}}_i^{\text{curr}} - L_i, U_i - \underline{\mathbf{x}}_i^{\text{curr}}\} \quad (3)$$

특히 성분이 많이 변하지 않는다면, 또는 대응하는 $\underline{\mathbf{x}}$ 성분이 경계에서 멀리 떨어진다면, 한번 이상의 반복을 위해 고정된 약간의 \mathbf{D} 성분들을 유지시키는 것이 가능하다.

다음 반복에 대한 탐색 방향은

$$\mathbf{P} = \mathbf{D} \{ \mathbf{I} - (\mathbf{A} \mathbf{D}^2 \mathbf{A}^T)^{-1} \mathbf{A} \mathbf{D} \} \mathbf{D} \mathbf{C} \quad (4)$$

로 주어진다. 여기에서 \mathbf{I} 는 항등 행렬(주 대각선의 모두 1임)이며 위첨자 T 는 행렬의 전치(transpose)(행렬과 열을 교환)를 표시한다. 계산적 견지에서 가장 어려운 연산은 행렬곱을 반전하는 것이며 근사 기법 또는 충분 변화 기법은 이러한 목적을 위해 충당되어질 수 있다.

새로운 반복의 값은

$$\underline{\mathbf{x}}^{\text{new}} = \underline{\mathbf{x}}^{\text{curr}} + \alpha \mathbf{P} \quad (5)$$

로 표현될 수 있으며, 여기에서 α 는 $\frac{\mathbf{P}}{\|\mathbf{P}\|}$ 로 기술된 방향에서의 단계의 크기이다. 강하게 실행가능한 새로운 반복 $\underline{\mathbf{x}}^{\text{next}}$ 를 유지하기 위하여, α 는 가장 가까운 경계까지의 거리보다 작아야 한다. 한가지 간단한 설계는 가장 가까운 경계에 대해 β 비율로 진로를 이동시키는 것이다.

$$\alpha = \beta \min\{\min\{(L_i - X_i)/P_i \mid P_i \neq 0\}, \min\{U_i - X_i \mid P_i \neq 0\}\}$$

(6)

β 의 값은 1보다 작아야 한다.

전술한 방법은 강하게 실행 가능한 시작점 즉, 폴리토프 내부내의 점을 요구한다는 것에 유의하라. 그러한 점은 어떤 상황에서는 쉽사리 확인될 수도 있는 반면, 일반적인 경우에서는 실행 가능한 영역이 있는지 없는지 조차도 알 수 없다. 전술한 절차를 이용하는 예비 단계는 선형 프로그래밍 모델에 대한 어떠한 해결이 조금이라도 있는가, 그리고 만일 그럴다면, 강하게 실행 가능한 시작 점의 값은 얼마인가를 결정하는 것이다. 종래 기술에서 이것은 실행 가능성 문제라고 불리우며, 그 해결은 보통 자원 배당의 최적 값을 찾는 선형 프로그래밍 모델의 해결에 선행한다.

본 발명에 따라서, 전술된 절차의 변형이 선형 프로그래밍 모델에 대한 실행 가능성 문제를 해결하기 위해 사용될 수 있다. 심플렉스법에서, 이것은 인위적인 부가 변수 또는 임의 변수를 제한 관계에 더하고 이러한 인위적인 변수의 합이 할당 값의 몇몇 집합에 대해 0으로 감소되어지는가를 알기 위해 심플렉스법을 사용함으로써 행해진다. 만일 그럴지 않다면, 문제는 실행 불가능하며 그러므로 해결 불가능하다. 이 합이 0으로 감소될 수 있다면, 그 목표를 달성하기 위해 필요한 할당 값은 시작점으로 이용될 수 있다. 사실상, 새로운 목적 함수가 인위적 변수의 합을 최소화하는, 제한 관계와 함께 사용된다.

비슷한 방법이 본 발명에서 실행 가능성 문제를 해결하기 위해 사용된다. 강하게 실행 가능한 시작점이 개시점으로 요구되기 때문에, 새로운 목적 함수가 그 목표를 달성하도록 설계된다. 특히, 선형 프로그래밍 모델이, 각 단계에서, 실행 불가능한 할당 값의 경계로부터의 거리를 최소화하기 위해 해결된다면, 이 실행 가능성 문제의 해결에서 기인하는 할당값은 강하게 실행 가능할 수도 있으며 주 절차의 시작점으로 이용될 수 있다. 그리하여 실행 가능성 문제는 다음과 같이 서술된다.

$$\text{최소화 : } \sum_{i=1}^n \max\{0, (L_i - X_i), (X_i - U_i)\}$$

조건 : $Ax=b$ 및

$L \leq X \leq U$

이 절차를 위한 시작점은 제한 $Ax=b$ 를 만족하는 X 의 어떤 값도 될 수 있다. 또 다른 시작 절차는 1984년 4월 30일자, 계산 이론상의 미국 컴퓨터 학회(ACM) 심포지움의 회보에 "선형 프로그램용 새로운 다항식-곱 알고리즘"이란 제목의 출원인의 논문에 기술되어 있다.

D. 대각선 스케일 행렬 D 값상의 많은 변화는 정규화 특성이 유지되는 동안은 가능하다. 유사하게, α 값상의 많은 변화도 다음 반복이 폴리토프의 내부에 포함되는 즉, 강하게 실행 가능한 동안은 마찬가지로 가능하다. 그러한 한가지 선택적인 정규화 방법이 상기 지적된 본 출원인의 출판물에 기술되어 있다.

출원인의 출판물에서 개요된 절차는 변수를 변화(사영 변환)시켜 이 새로운 변수에 대하여 "포텐셜 함수"(아래에 서술됨)의 가장 가파른 경사 방향을 계산하고, 새로운 반복이 여전히 실행 가능하도록 이 방향으로 약간의 거리를 이동하며, 그리고 결과점을 원래 변수로 다시 환연시킨다. 출원인의 출판물에서 기술된 변수의 변화는 현재 반복 X 가 단위(unit) 심플렉스의 도심으로 환연하도록 선정되며 그리하여 어떤 점에서는, 모든 부등 제한으로부터 등거리가 된다.

이 문제는 다음과 같이 먼저 재기술된다.

$$\text{최소화 : } C^T X$$

조건 : $Ax=b$ 및

$X \geq 0$

$C^T X$ 를 직접 최소화하도록 시도하기보다, 공표된 절차는

$$f(X) = \sum_{i=1}^n \log \frac{C^T x}{X_i} \quad (9)$$

로 정의된 "포텐셜 함수"를 감소시키는 단계들을 취하는데, 여기에서 C 는 방정식(8)의 어떤 최적 해결 X^{opt} 이 C 를 C 로 대체시켜 방정식(8)을 풀이하여 $CX^{opt} = 0$ 이 되도록 선정된 C 의 변형된 형태이며, 최적 목적 함수 값상의 하한 및 상한 경계를 유지하기 위해 "슬라이딩 목적(sliding objective)" 방법을 사용한다.

참조된 출판물에서, 대수학의 어려움은 제한들을 다음의 특별한 형태로 함으로써 감소한다.

$$\text{최소화 : } C^T X$$

조건 : $Ax=b$

(10)

$e^T X = 1$ 및

$X \geq 0$

여기에서 e 는 1들의 벡터이다. 이것은 하나 이상의 성분을 \underline{X} 에 더하고, 대응하는 0들의 열 (column)을 \underline{A} 에 더하고 \underline{A} 를 적당히 스케일링하여, 그 결과에서 \underline{be}^T 를 빼므로써 달성된다. 특별한 형태의 방정식(10)으로 재기술된 문제에 따라, 절차는 제 5 도에 도시된 바와 같이 요약될 수 있다. 이러한 형태로 서술된 모델에 따라, 적당한 스케일링 행렬은 현재 반복 값, 즉, $D_i = X_i$ 이다. 더욱이, 이 문제는 계산의 복잡성을 감소시키기 위해 단위 심플렉스상의 문제로 변환된다.

특히 제 5 도를 참조하면, 박스(160)에서, 선형 프로그래밍 문제가 표준형으로 공식화된다. 박스(161)에서 폴리토프 내부의 시작점 $\underline{X}^{\text{start}}$ 이 상기 제안된 바와 같은 실행 가능성 결정과 결부되어 선택된다. 초기의 현재 반복 $\underline{X}^{\text{curr}}$ 으로서 $\underline{X}^{\text{start}}$ 를 사용함으로써, 다음 반복 $\underline{X}^{\text{next}}$ 를 발생하는 절차가 제 5 도의 순서도의 나머지 부분(68)에서 개요된다. 그 단계는 다음과 같다.

1. i 번째 대각선 기입내용이 ($d_i = X_i^{\text{curr}}$)인 대각선 스케일 행렬 \underline{D} 를 선정한다. 이러한 선택은 다음 관계에 의해 변수 \underline{X}' 로 사영 변환을 결정한다.

$$X'_1, \dots, n = \frac{X^{\text{curr}}/d_i}{1 + e^T \underline{D}^{-1} \underline{X}^{\text{curr}}}$$

$$\text{및 } X'_{n+1} = 1 - e^T X'_1, \dots, n \quad (11)$$

여기에서 $\underline{X}'_{1, \dots, n}$ 은 $(n+1)$ 벡터 \underline{X}' 의 제1의 n 성분을 표시한다. 이 사영 변환은 단위 심플렉스로의 직교 변환으로서 생각될 수 있으며, 이에 의해 정규화 또는 센터링 특성을 달성한다. 박스(62)는 인접 단위 심플렉스의 공백(null) 공간으로의 변수 정규화 변환을 요약한다.

2. 지금, 단위 심플렉스의 도심으로 사영된, 현재 반복으로부터 변환된 목적 함수의 제한된 가파른 경사 방향 \underline{P} 를 계산한다. 이 방향은,

$$P = -[I - B^T (BB^T)^{-1} B] D C \quad (12)$$

로 주어지며, 여기에서

$$B = \frac{\underline{A} \underline{D}, -b}{e^T}$$

(13)

이 계산은 제 5 도의 박스(63)에 도시된다.

3. $(\underline{X}'^{\text{next}} := \underline{X}' + \alpha \underline{P})$ 가 강하게 실행 가능하도록, 즉 $(X'^{\text{next}} \geq 0)$ 이도록, 그리고 포텐셜 함수 $g(X')$ 가 감소되도록 (양호하게는 거의 최소화되도록) α ($\alpha \geq 0$)의 값을 선정한다. 포텐셜 함수는 $g(X) = f(T(X))$ 이고, 여기에서

$$T(X) = D X'_1, \dots, n / X'_{n+1}$$

이 단계는 제 5 도의 박스(66)에 도시된다.

4. $\{X'^{\text{next}} = T(X'^{\text{next}})\}$ 를 계산한다. 여기에서 T 는 방정식(14)로 주어진다. 이 단계는 제 5 도의 박스(67)에서 도시된다.

제 5 도의 일정쇄선으로 표시된 박스(68)의 반복 절차가 완결된 다음, 전술된 것을 포함하는 어떠한 공지된 정지 기준은 판정 박스(69)에 제공된다. 만일 정지 기준이 박스(69)에서 만족되면, 절차는 완결되어 종료 박스(70)에서 정지된다. 만일 정지 기준이 박스(69)에서 만족되지 않으면, 계산된 다음 반복 $\underline{X}^{\text{next}}$ 은 박스(71)에서 현재 반복 $\underline{X}^{\text{curr}}$ 로 대체되며, 다음 반복을 위해 박스(62)로 다시 들어간다.

제 6 도에는 처리(80)를 제어하는 처리 제어 시스템이 도시된다. 처리기(80)는 전화 통신 시스템 제조 처리, 항해 처리, 또는 최적화될 어떤 다른 산업적 또는 기술적 처리일 수도 있다. 비용 레지스터(81)는, 제어된 처리(80)에서 다양하고 가능한 자원 할당의 단위(unit)당 비용을 표시하는 비용 데이터를 리드(82)에서 수신한다. 비용 데이터는 컴퓨터 단말기나 또는 이를 비용을 동적으로 결정하는 독립된 처리로부터 레지스터(81)로 입력될 수도 있다. 보통 이 비용 데이터가 비교적 천천히 변하는 반면, 그럼에도 불구하고 필요할 때마다 입력 리드(82)를 통해 이 데이터를 갱신하는 능력이 있다. 만일 해결 값상에 0이 아닌 한계(limit)(방정식(1)에서 L 및 U)가 존재한다면, 비용 데이터와 같은 이러한 한계들은 레지스터(81)와 같은 데이터 입력 레지스터에 의해 LP제어기(85)에 공급되어야 한다.

유사하게, 한계 레지스터(83)는 각각의 특정 자원 할당상의 모든 물리적 한계들의 표시를 저장하기

위해 제공된다. 마찬가지로 이 한계들은 비교적 정적이며 컴퓨터 단말기나 또는 독립적인 한계-결정 처리로부터 리드(84)를 통해 레지스터(83)로 들어갈 수 있다. 레지스터(81 및 83)의 출력은, 제 4 도 또는 제 5 도의 순서도에서 요약된 처리를 실행하는 선형 프로그래밍(LP) 제어기(85)에 인가된다. 양호한 실시예에서, LP제어기(85)는 제 4 도 또는 제 5 도의 순서도를 구현하는 프로그램이 저장된 프로그램 디지털 컴퓨터이다. 제어기(85)는 또한 제 4 도 또는 제 5 도의 절차를 실행하기 위해 설계된 하드웨어 회로의 복합체, 절차의 병렬 실행을 위한 가능성을 이용하는 다수의 병렬 처리기, 또는 이 목적을 위해 프로그램된 다수의 프로그램 선형 배열들을 구비할 수도 있다.

다수의 제한 센서들(86), (87), …, (88)은 제한 관계의 제한 계수를 동적으로 감지하기 위해 제공된다. 제한 센서는 (86 내지 88)은 제어된 처리(80)의 환경 변화에 연속적으로 응답하고, 그 변화들은 제한 관계에 영향을 주며 그러므로, 처리(80)를 제어하도록 주적되어야 한다. 각각의 제한 센서(86 내지 88)는 대응 변화(밀타) 검출기(89), (90), …, (91)를 갖는데, 이 검출기들은 센서(86 내지 88) 각각의 출력 변화를 감지한다. 각각의 검출기(89 내지 91)로부터의 변화-지시 신호는 변화 버스(92)에 인가되어 그리하여 AND게이트(93)에 인가된다. 또한 절차 실행의 종료를 지시하는 신호가 LP제어기(85)로부터 리드(94)를 통해 AND게이트(93)에 인가된다. 센서(86 내지 88)로부터의 출력은 각각의 검출기(89 내지 91)를 통해 제어기(85)에 인가된다.

작동중에, 센서(86 내지 88)로부터의 출력은 방정식(1)의 제한 행렬 A 의 계수로서 제어기(85)에 의해 사용된다. 레지스터(81)의 비용 데이터는 방정식(1)에서 비용 벡터(C)로서 사용되며 레지스터(83)의 한계 데이터는 방정식(1)의 한계 벡터(b)로서 사용된다. 이러한 입력이 주어지면, LP제어기(85)는 제 4 도 또는 제 5 도의 절차를 실행할 수 있으며 그리고 디지털 해결값(X')를 제어 레지스터(95), (96), …, (97)에 공급한다. 레지스터(95 내지 97)에서의 값은 처리(80)를 제어하도록 사용된다.

제 6 도의 LP제어기는 제 4 도 또는 제 5 도의 극히 신속한 절차를 이용하기 때문에, 제어 값은 매우 짧은 시간에 레지스터(95 내지 97)용으로 유용하다. 게다가, 제한 변화처럼, 이런 변화들은 센서(86 내지 88)에 의해 감지되고, 검출기(89 내지 91)에 의해 검출되며, AND게이트(93)를 부분적으로 인에이블하기 위해 사용된다. 제 4 도 또는 제 5 도의 절차가 완결될 때, LP제어기(85)는 제어 신호를 발생하며 그리고 그 신호를 레지스터(95 내지 97)로 전송하며, 동시에 AND게이트(93)로 가는 리드(94)상에 인에이블 신호를 발생시켜, AND게이트(93)의 인에이블먼트를 완결한다. 그 뒤, 전체 처리가 반복된다.

문제의 복잡성(센서(86 내지 88)에 의해 감지된 제한의 수) 및 처리(80)의 안정성에 따라, 이 방법에 의해 처리(80)를 다소 연속적으로 제어하는 것이 가능하다. 사실, 센서(86 내지 88)에 의해 감지된 환경 요소의 변화율이 LP제어기(85)의 작동 속도와 같거나 작다면, 이 처리(80)는 연속적으로 제어될 것이다. 더 빨라진 환경 변화 속도는 입상(granularity)을 제어과정에 도입하지만, 그래도 평균적으로, 처리(80)의 최적 동작에 가깝도록 허용한다. 사실, 주변 변화에 대한 약간의 변천(history)가 주어진다면, 센서(86 내지 88)의 출력에서의 장자의 변화에 대한 방향 및 크기를 예언하도록 약간의 예언 메카니즘의 검출기(89 내지 91)에 설치될 수 있다.

본 발명이 적용될 수 있는 통신 분야의 전형적인 문제형태는 1981년 10월, 제60권, 제8호의 벨 시스템 기술지에서 두개의 논문으로 기술되어 있다. 지. 알. 애쉬씨등에 의한 "역학루팅을 갖는 네트워크의 설계 및 최적화"라는 제목의 첫번째 논문(페이지 1787)은 일반용 전화 트래픽 루팅 문제를 서술한 반면, 역시 지. 알. 애쉬씨등에 의한 "역학 루팅을 갖는 네트워크의 서비스 및 실-시간 제어"라는 제목의 두번째 논문(페이지 1821)은 트래픽 부하의 잘못된 예언에서 기인한 유휴용량을 최소화하는 보조적인 문제를 서술한다. 이 논문들 모두는 여기에서 참조로 반영된다.

제 7 도의 간략화된 형태로 도시된 바와 같이, 국내전화 네트워크는 대다수의 전화 스위칭 점을 상호 접속하는 대다수의 전송 설비로 구성된다. 네트워크의 한 부분에서 발생하는 전화 호출은 전송 설비를 통하여 네트워크의 또 다른 부분의 특정 전화국으로 루팅되어야 한다. 전송 설비에서 각각의 연결은 최대 용량 제한 뿐만 아니라 그것과 연관된 비용을 갖는다. 각각의 스위칭 노드에서 야기되는 트래픽 양은 아직도 또 다른 변수이다. 전화 네트워크는 모든 호출을 최소 경비 노선으로 적당한 목적지로 루팅하는 반면, 동시에, 용량제한을 침해하지 않을 것이 요구된다. 전화 네트워크 제어 시스템에서 목적 함수는 모든 다양한 전송 링크를 통한 트래픽 루팅 비용의 합산으로, C 는 비용 계

X_i 는 링크 (a_i) 는 (초과될 수 없는) 전송선의 용량 및 (서브되어져야 하는) 트래픽 부하를 표시한다. 제 6 도의 일반적인 시스템에서와 같이, 양의 링크 부하 값만이 허용 가능($X_i \geq 0$)하다.

더욱 특히, 전화루팅 시스템은 애쉬씨의 참조문헌에 도시된 바와 같이 선형 프로그램 모델로서 표시될 수 있다.

$$\text{최소화 : } \sum_{i=1}^L M_i a_i$$

$$\text{조건 : } \sum_{k=1}^k \sum_{j=1}^{J_k} P_{ij}^k r_{jk}^k \leq a_i$$

$i=1,2, \dots, L$; $h=1,2, \dots, K$

$$\sum_{j=1}^L \frac{r_{jk}^h}{1-g_{jk}^h} = P_k^h$$

$h=1,2, \dots, H$; $K=1,2, \dots, K$

$$r_{jk}^h \geq 0, a_i \geq 0$$

여기서 L =네트워크에서 총 링크의 수, K =수요 쌍(제공된 부하)의 수, H =설계 시간의 수, J_k^h =시간 h 에서 수요 쌍 k 의 노선의 수. P_{jk}^{ih} =시간 h 에서 링크 i 상의 점 대 점 수요량 k 에 대한 노선 j 상에서 운반된 부하의 비율, M_i =링크 i 에 대해 운반된 트래픽의 얼랭(erlang)당 달려 비용의 항으로된 충분 링크 비용 행렬. R_k^h =시간 h 에서 수요 쌍 k 로 제공된 부하, r_{jk}^h =시간 h 에서 수용 쌍 k 의 노선 j 상에 운반된 부하, r_{jk}^j =시간 h 에서 수용 쌍 k 의 노선 j 상에 운반된 부하. A_k^h =시간 h 에서 링크 i 로 제공된 부하, a_i =전시간에 걸쳐 링크 i 상에서 최대로 운반된 부하, g_{jk}^h =시간 h 에서 수용 쌍 k 의 노선 j 상의 노선 블로킹, 및 b_j^h =시간 h 에서 링크 i 의 블로킹.

이러한 형태의 LP모델을 해결하기 위한 시스템은 제 8 도에 도시된다.

제 8 도는 전화 네트워크(100)용 노선 공식을 위한 반복 루프를 도시한다. 제 7 도의 장치는 최단 경로(가장 경제적인 경로), 즉 네트워크(100)에서 점(104),(105)간의 점(101),(102),(103)을 찾는다, 알맞는 블로킹 레벨이 가정되며(또는 실질 블로킹이 박스(106)에서 측정된다). 라우터(router)(107)가 경로(101),(102),(103)를 후보 노선(경로의 순열)으로 형성한다. 또한 라우터(107)는 제공된 부하 각각의 단위(unit)에 대한 노선에서 각 경로에 제공된 트래픽 흐름 비율을 결정한다. 여기에서 이 트래픽 부하의 박스(109)에 의해 연속적으로 공급된다. 이때 선형 프로그램 제어기(108)는 전체 네트워크 루팅 비용을 최소화하도록 트래픽 흐름을 후보 노선에 할당한다. 선형 프로그래밍 제어기로부터의 출력은 각 링크상의 트래픽 흐름을 제어하기 위한 루팅 테이블(110)에 의해 사용될 수 있는 최적 루팅 계획안이다.

제 8 도의 전화 루팅 장치는 전화 네트워크를 연속적으로 또는 규칙적 간격으로 제어하기 위해 사용될 수 있다. 그리하여, 제 4 도 및 제 5 도의 더욱 빠른 절차와 더불어, 변화 수요 및 변화 링크가 이용 가능할 때 전화 네트워크를 동적으로 제어하기 위해 제 8 도의 장치를 사용하는 것이 가능하다.

전화 루팅 문제의 해결은 각각의 전송 링크에 배치될 최적 트래픽 부하를 제공하여 모든 전화 호출에 대한 최적 루팅을 제공함을 알 수 있다. 더욱이, 국내 전화 네트워크는 대다수의 그러한 링크를 포함하기 때문에, 문제를 해결하기 위해 요구된 시간은 해결의 실제 유용성에 대해 상당히 중요한 것이다. 트래픽 부하 변화, 연계 중지(linkage outages) 및 링크 비용 변화 모두는 최적 할당에 영향을 미친다. 그러므로 루팅 제언은 문제 자체가 크게 변하기 이전에 제공되어야만 한다. 시행착오에 의해 문제를 해결하는 방법은 이 점에 관해서는 보조하는 정도인 반면, 빠른 선형 프로그래밍 방법은 특히 예기치 않은(예견할 수 없는) 부하를 조작할 때, 극단적으로 유용한 것이다.

여기에서 서술된 새로운 절차로부터 이익을 얻을 수 있는 다른 문제들은 산업용 처리 제어, 고객 서비스를 제공하는 직원 배치, 무역 상품을 만드는 재료 배합, 정유물 혼합, 다수 사용자에 대한 컴퓨터 자원 할당, 및 그외의 것을 포함한다. 각각의 경우에 있어서, 비용(또는 이익) 계수가 측정되거나 또는 결정되어야 하며, 제한 한계도 설정되어야 하며 그리고 이런 제한에 대한 모든 판정 변수의 분배도 역시 측정 또는 결정되어야 한다. 각각의 경우에 있어서, 절차를 실행한 결과는, 실제 상황에 적용될 때 최적 처리 또는 장치를 생성하는 일련의 제어 파라미터의 내역이다.

가장 실용적인 프로그래밍 모델에 연관된 행렬은 스파스(sparse) 행렬이며, 이 스파스 행렬 기술은 또한 제 4 도 및 제 5 도에서 탐색 방향 P 를 평가할 때 사용될 수 있음을 유의하라.

본 발명자는 선형 프로그래밍 문제를 해결하는 새로운 방법을 구성했지만, 본 발명의 특허청구범위는, 시스템의 특성을 나타내는 변수 및 제한들에 대해 선형적으로 표현하는 것이 가능한 실제의 기술적 및 산업적 시스템에서 최적 자원 할당을 결정하는 장치, 즉 처리의 실행, 기계, 물질의 제조 또는 합성을 최적화하기 위해 자원이 어떻게 인가될 것인가를 결정하는 물리적 장치에 대한 이 신규 방법의 응용에만 관련되었다는 것을 이해해야 한다. 계산 탐색, 알고리즘 탐색, 또는 선형 대수 탐색 활동과 같이 모든 새로운 방법의 다른 이용은 본 발명의 어떠한 요소도 형성하지 않는다. 유사하게, 비-기술적 또는 비-산업적 시스템에서 새로운 방법을 사용하는 것도 본 발명의 요소를 만들지 않는다.

(57) 청구의 범위

청구항 1

전기 통신 전송 설비를 자공하는 총 비용이 최소화되도록 특정 시간에 서비스를 요구하는 전화 가입자들 사이에 유용한 전기 통신 전송 설비를 할당하는 방법에 있어서, 상기 총 비용이 최소화되게끔 상기 이용가능한 전기 통신 전송 설비를 상기 가입자들에게 일시적 및 반복적으로 재배당하는 단계와, 각각의 상기 재배당이 상기 할당상의 제한에 대해 이전 배당을 정규화함으로써 결정되는 단계와, 상기 비용이 최소화될 때 상기 반복적 재배당 단계를 종료하는 단계, 및 최소 비용 배당에 따라서 상기 전송 설비를 할당하는 단계를 포함하는 것을 특징으로 하는 전기 통신 전송 설비 할당 방법.

청구항 2

다수의 제2전기 통신 스위칭 노드들과 상호 접속하는 다수의 제1링크들 및, 각각의 상기 노드에서 발생한 트래픽을 트래픽 운반 비용을 최소화하게끔 상기 링크에 배당하는 수단을 포함하고, 상기 배당 수단이 상기 최소 비용 배당들의 견적(estimates)들을 반복적으로 선택하는 수단을 구비하여 각각의 상기 반복 선택이 상기 배당상의 물리적 제한을 나타낸 다차원 볼록면체 해결 공간의 내부에서 배당 값들을 완전히 표시하도록 하는 것을 특징으로 하는 전기 통신 전송 시스템.

청구항 3

유용한 사용자 설비를 제공하는 비용이 최소화되도록 다수의 사용자 사이에 상기 사용자 설비를 할당하는 방법에 있어서, 상기 비용이 최소화되게끔 상기 사용자 사이에 상기 설비들을 일시적 및 반복적으로 재배당하는 단계와, 각각의 상기 반복적 재배당이 상기 할당상의 제한에 대해 이전 할당을 집중화함으로써 결정되는 단계와, 상기 비용이 최소일 때 상기 반복적 재배당 단계를 종료하는 단계, 및 최종 반복적 재배당에 따라서 상기 사용자 사이에 상기 유용한 사용자 설비를 할당하는 단계를 포함하는 것을 특징으로 하는 사용자 설비 할당 방법.

청구항 4

제 3 항에 있어서, 상기 설비는 전기 통신 전송 설비를 포함하며 상기 사용자는 전화 가입자를 포함하는 것을 특징으로 하는 사용자 설비 할당 방법.

청구항 5

제 3 항에 있어서, 상기 설비는 정보 조작 설비를 포함하는 것을 특징으로 하는 사용자 설비 할당 방법.

청구항 6

제 3 항에 있어서, 상기 설비는 데이터 처리 설비를 포함하는 것을 특징으로 하는 사용자 설비 할당 방법.

청구항 7

제 3 항에 있어서, 상기 설비는 제조 설비를 포함하는 것을 특징으로 하는 사용자 설비 할당 방법.

청구항 8

사용하기에 유용한 다수의 제1물리적 자원과, 상기 물리적 자원을 사용하는 다수의 제2자원 사용자 및, 상기 자원을 공급하는 비용이 최소화되게끔 상기 자원 사용자에게 상기 물리적 자원을 배당하는 수단을 포함하고, 상기 배당수단이 상기 배당들중 실행가능한 배당을 일시적 및 반복적으로 선택하는 수단을 포함하며, 상기 실행가능한 배당의 각각이 정규화된 다차원 볼록면체 실행가능 해결 공간의 내부에 집중되도록 하고, 상기 일시적인 배당의 최종 배당에 따라서 상기 물리적 자원을 할당하는 수단을 포함하는 것을 특징으로 하는 최적 자원 할당 시스템.

청구항 9

제 8 항에 있어서, 상기 물리적 자원은 전기 통신 설비를 포함하며 상기 사용자는 전화 가입자를 포함하는 것을 특징으로 하는 최적 자원 할당 시스템.

청구항 10

제 8 항에 있어서, 상기 물리적 자원이 정보 조작 설비를 포함하는 것을 특징으로 하는 최적 자원 할당 시스템.

청구항 11

제 8 항에 있어서, 상기 물리적 자원이 데이터 처리 설비를 포함하는 것을 특징으로 하는 최적 자원 할당 시스템.

청구항 12

제 8 항에 있어서, 상기 물리적 설비는 제조 설비를 포함하는 것을 특징으로 하는 자원 할당 시스템.

청구항 13

최적화 기준에 따라서 제어된 처리의 실행을 최적화하는 시스템에 있어서, 제어 신호 집합(set)에 응답하여 상기 처리를 제어하는 처리 제어 장치와, 상기 처리의 동작에 영향을 주는 다양한 조건들을 감지하는 다수의 감지기와, 상기 처리의 동작에 영향을 주는 조건들을 규정하는 다수의 데이터 입력 장치, 및 상기 감지기 및 상기 입력 장치에 응답하여 최적 제어 신호 집합을 상기 처리 제어

장치에 제공하는 선형 프로그래밍 제어기를 포함하고, 상기 제어기가 연속적이고 일시적인 강하게 실행가능한 제어 신호 집합을 반복적으로 확인하고 상기 최적화 기준의 정규화된 형태의 가장 가파른 기울기 방향으로 각각의 다음 일시적 제어 신호 집합을 선택하는 수단을 구비하는 것을 특징으로 하는 최적화 시스템.

청구항 14

제어된 시스템의 동작을 최적화하는 제어기에 있어서, 상기 시스템 동작상의 물리적 제한 및 제한한계들을 결정하는 수단과, 상기 시스템의 동작에 대해 실행 측정 기준을 규정하는 수단과, 상기 제한 및 상기 제한 한계를 강하게 만족시키는 일시적인 동작 제어 값 집합을 연속적으로 확인하는 수단과, 상기 제한 한계로부터 상기 제어 값들이 등거리가 되게끔 상기 일시적인 제어 값 집합을 정규화하는 수단, 및 상기 규정된 실행 측정 기준에 따라서 각각의 다음 연속하는 정규화된 제어 값 집합을 선택하는 수단을 포함하는 것을 특징으로 하는 제어기.

청구항 15

선형 프로그래밍 모델을 이용하는 자원 할당 방법에 있어서, 상기 자원의 실행가능한 할당을 적절히 기술하는 목적 함수 및 다수의 제한들로 선형 프로그래밍 모델을 규정하는 단계와, 강하게 실행가능한 일시적 자원 할당을 확인하는 단계와, 상기 제한들에 대해 상기 일시적 자원 할당을 정규화하고 상기 목적 함수에 의해 지정된 방향으로 상기 일시적 자원 할당을 변경함으로써 상기 일시적 자원 할당에 따라서 상기 자원을 할당하는 단계를 포함하는 것을 특징으로 하는 자원 할당 방법.

청구항 16

다수의 사용자 사이에 자원을 최적으로 할당하는 개선된 선형 프로그래밍 방법에 있어서, 단지 강하게 실행가능한 할당상에서만 반복하는 단계, 및 할당을 정규화하는 단계를 포함하는 것을 특징으로 하는 개선된 자원 할당 선형 프로그래밍 방법.

청구항 17

자원 할당 메카니즘에 의해 부과된 자원 할당 요소들을 구비하여, 다수의 사용자 실체 사이에 기술적 자원을 할당하되, 각각의 상기 할당은 할당상에 부과된 제한들과 할당에 관련된 정량화 가능한 비용 또는 이의를 갖도록 한 기술적 자원 할당 시스템에 있어서, 상기 자원 할당 메카니즘이, 상기 제한들을 표시하는 각면(facets) 및 양호한 자원 할당을 표시하는 표면(surface)을 갖는 다차원적 볼록면체 폴리토프로서 상기 제한들을 표시하는 수단과, 시작점으로서 상기 폴리토프 내부 점에 대응하는 상기 자원의 할당을 일시적으로 선택하는 수단과, 상기 시작점으로부터, 각기 선택점보다 더 최적의 할당을 표시하는 상기 폴리토프 내부의 연속 후행점들로 진행하는 수단, 및 상기 표면상의 점에 의해 지정된 양호한 자원 할당을 수용시키기 위해 상기 시스템의 요소를 배치하는 단계를 포함하는 것을 특징으로 하는 기술적 자원 할당 시스템.

청구항 18

할당상의 제한들에 복종하여 할당의 총비용이 최소화되는 방식으로 다수의 자원 사용자 사이에 기술적 자원을 할당하는 기술적 자원 할당 시스템에 있어서, 상기 시스템의 상기 할당 결정 방법이

- 1) 다차원 공간에서 폴리토프로서 상기 제한들을 표시하는 단계와,
- 2) 상기 다차원 공간에서 벡터로서 상기 할당 비용을 표시하는 단계와,
- 3) 상기 폴리토프의 내부에 위치된 초기 할당 점을 선택하는 단계와,
- 4) 상기 초기 할당 점이 실제로 상기 폴리토프의 중심에 있도록 상기 폴리토프를 정규화하는 단계와,
- 5) 상기 정규화된 폴리토프내의 상기 제한들의 공백(null) 공간으로 사용된 상기 비용 벡터의 방향을 결정하는 단계와,
- 6) 상기 사영된 비용 벡터의 반대 방향으로 상기 정규화된 폴리토프내의 새로운 할당 점을 선택하는 단계, 및
- 7) 상기 새로운 할당 점에 대해 단계(3) 내지 (6)을 반복하는 단계를 포함하는 것을 특징으로 하는 기술적 자원 할당 시스템.

청구항 19

다수의 제2종료 결과를 달성하는 다수의 제1자원을 포함하고, 상기 종료 결과를 달성하기 위해 현재 자원 할당 배열 \mathbf{X}^i 가 방정식 $\mathbf{X}^{i+1} = \phi(\mathbf{X}^i)$ 에 의해 \mathbf{X}^i 로부터 웃도된 개선된 자원 할당 배열 \mathbf{X}^{i+1} 로 대체되고 이 개선된 자원 할당 배열 \mathbf{X}^{i+1} 이 다음 반복의 현재 자원 할당 배열이 되는 반복방법(recursive)을 이용하여 상기 자원을 할당하는 시스템이 있어서, 상기 함수 ϕ 는

- 1) 다른 부분은 영인 대각선 행렬 \mathbf{D} 의 대각선을 따라 상기 현재 자원 할당 배열의 성분들을 배열하고,
- 2) 행렬 곱 \mathbf{AD} 를 전개하고 이 전개된 곱을 n 개의 1로 이루어진 최종 행으로 증가시킴으로써 행렬 \mathbf{B} 를 형성하고,
- 3) \mathbf{I} 가 항등 행렬인 $[\mathbf{I} - \mathbf{B}^T(\mathbf{B}\mathbf{B}^T)\mathbf{B}]^{-1}$ 의 값을 획득함으로써 벡터 \mathbf{Cp} 를 전개하고,

4) 정규화된 포인터 \mathbf{Cp} 를 전개하게 위해 그 크기로 \mathbf{Cp} 를 나누고,

5) $\underline{\mathbf{X}}^i$ 의 현재 변환 견적 $\underline{\mathbf{X}}^i$ 로부터 $\alpha \mathbf{Cp}$ 를 뺏으로써 $\underline{\mathbf{X}}^i$ 의 변환된 새로운 견적 $\underline{\mathbf{X}}^{i+1}$ 을 만들고(여기에서 α 는 1보다 작다),

6) $\mathbf{Dx}^{i+1}/\mathbf{e}^T \mathbf{Dx}^{i+1}$ 을 평가함으로써 $\underline{\mathbf{X}}^i$ 의 변형되지 않은 새로운 견적을 만들고(여기에서 $\mathbf{e}=\{(1, 1, \dots, 1)\}$),

7) $\underline{\mathbf{X}}^i$ 의 상기 새로운 견적 $\underline{\mathbf{X}}^{i+1}$ 을 상기 시스템에 인가하는 것을 포함하는 것을 특징으로 하는 시스템.

청구항 20

다수의 자원 사용자 사이에 산업적 자원을 할당하는 방법으로서, 각각의 상기 할당이 자원 사용상의 물리적 제한들과 각각의 상기 할당에 연관된 정량화 가능한 비용 또는 이익을 갖는 상기 산업적 자원 할당 방법에 있어서,

1) 상기 물리적 제한의 각각을 나타내는 각면을 갖는 다차원 폴리토프를 정의하는 것과 함께 선형 관계의 시스템으로 상기 물리적 제한을 표시하는 단계와,

2) 시작 할당점으로서 상기 폴리토프의 내부점으로 표시된 상기 자원의 일시적인 할당 집합을 선택하는 단계와,

3) 상기 시작 할당점이 실질적으로 상기 변환된 폴리토프의 기하학적 중심에 위치하고 모든 상기 각면들이 상기 중심으로부터 실질적으로 등거리에 위치하도록 상기 폴리토프를 변화하는 단계와,

4) 상기 시작 할당점으로부터 상기 리스케일(rescaled)된 폴리토프의 표면에 더 근접하는 상기 리스케일된 폴리토프 내부의 다른 할당점으로 이동하는 단계와,

5) 상기 다른 할당점을 상기 폴리토프의 원래 스케일로 다시 변환하는 단계와,

6) 상기 표면과 실질적으로 일치하는 상기 폴리토프 내부 점이 선택될 때까지 단계(3) 내지 (5)를 반복하는 단계와,

7) 상기 표면과 실제로 일치하는 상기 점과 관련된 할당 값을 확인하는 단계, 및

8) 확인된 자원 할당 값에 따라 상기 자원을 할당하는 단계를 포함하는 것을 특징으로 하는 산업적 자원 할당 방법.

청구항 21

각각의 자원 할당이 물리적으로 제한되고 상기 할당과 연관된 정량화 가능한 비용을 갖는, 다수의 자원 소비자 사이의 산업적 자원 할당 시스템에 있어서, 상기 시스템의 상기 산업적 자원 할당 방법이,

1) 상기 물리적 제한을 폐쇄된 볼록 다차원 입체(closed convex multidimension solid)를 정의하는 선형 관계 시스템으로 표시하는 단계와,

2) 상기 다차원 입체의 내부 점에 대응하는 자원 할당을 시작점으로 선택하는 단계와,

3) 상기 선택된 시작점이 실질적으로 변환된 입체의 기하학적 중심에 위치하고 상기 입체의 표면이 상기 중심으로부터 실질적으로 등거리에 위치하도록 상기 폐쇄 입체를 변환하는 단계와,

4) 상기 시작점보다 상기 표면에 더 근접한 상기 리스케일된 폐쇄 입체 내부의 또 다른 점에 대응하는 또 다른 상기 자원의 할당을 선택하는 단계와,

5) 선택된 할당이 실질적으로 입체의 표면상의 점에 대응할 때까지 새로운 할당에 대해 단계(3) 및 (4)를 반복하는 단계, 및 상기 입체의 상기 표면상의 점과 연관된 최종 자원 할당에 따라서 상기 자원을 할당하도록 상기 시스템을 배열하는 단계를 포함하는 것을 특징으로 하는 산업적 자원 할당 시스템.

청구항 22

범용 디지털 컴퓨터로 사용하기 위한 선형 프로그래밍 제어기에 있어서, 상기 디지털 컴퓨터에 의한 실행을 위해 컴퓨터 프로그램이 내장된 컴퓨터 프로그램 저장 매체를 포함하고, 상기 컴퓨터 프로그램이 상기 다수의 선형 관계에 대해 실행 가능한 해결 집합을 표시하는 다차원 볼록 폴리토프를 정의한 다수의 선형 관계를 처리하기 위한 수단, 및 최적화될 함수를 포함하며, 상기 폴리토프의 내부에 완전히 포함되는 강하게 실행 가능한 해결 경로를 따라 연속 단계로 진행함으로써 상기 다수의 선형 관계에 대한 최적 해결을 표시하는 상기 폴리토프의 경계상의 점을 확인하기 위한 수단을 포함하는 것을 특징으로 하는 선형 프로그래밍 제어기.

청구항 23

할당상의 제한에 복종하여 관련된 할당 비용을 최소화하는 방식으로 다수의 자원 사용자 사이에 물리적 자원을 할당하는 방법에 있어서,

1) 상기 제한을 다차원 공간의 폴리토프로서 표시하는 단계와,

2) 상기 할당 비용을 상기 다차원 공간의 벡터로서 표시하는 단계와,

- 3) 상기 폴리토프 내부에 위치한 초기 할당점을 선택하는 단계와,
- 4) 상기 폴리토프를 실질적으로 그 중심상에 상기 초기 할당 점을 갖는 등가의 공간으로 변환하는 단계와,
- 5) 상기 등가의 공간에서 상기 비용 벡터의 방향을 결정하는 단계와,
- 6) 상기 등가의 공간에서 상기 비용 벡터의 반대방향으로 새로운 할당점을 선택하는 단계와,
- 7) 상기 새로운 할당 점을 상기 폴리토프의 원래 공간으로 다시 변환하는 단계, 및
- 8) 상기 새로운 할당 점에 대해 단계(4) 내지 (7)를 반복하는 단계를 포함하는 것을 특징으로 하는 물리적 자원 할당 방법.

청구항 24

제한 $\mathbf{A}_i \mathbf{x}_i \leq \mathbf{b}_i$ 및 $\mathbf{x}_i \geq \mathbf{0}$ ($i=1, n : j=1, m$)에 복종하여 비용 함수 $\mathbf{C}_i^T \mathbf{x}_i$ 를 최적화하는 방식으로 다수의 자원 사용자 사이에 산업적 또는 기술적 자원 \mathbf{x}_i ($i=1, n$)을 할당하는 방법에 있어서,

- 1) 상기 제한을 충족시키는 초기 할당 $\mathbf{x}^{\text{start}} = (x_1^{\text{start}}, x_2^{\text{start}}, \dots, x_n^{\text{start}})$ 를 선택하는 단계와,
- 2) 상기 제한을 실질적으로 그 중심상에 $\mathbf{x}^{\text{start}}$ 를 갖는 아핀 공간(affine space)으로 변환하기 위해

D= 대각선 행렬 $\{x_1^{\text{start}}, x_2^{\text{start}}, \dots, x_n^{\text{start}}\}$ 및

$\mathbf{e} = (1, 1, 1, \dots, 1)$ 인 사영 변환

$\mathbf{x}' = \frac{\mathbf{D}^{-1} \mathbf{x}}{\mathbf{e}^T \mathbf{D}^{-1} \mathbf{x}}$ 를 사용하는 단계와,

- 3) 상기 아핀 공간에서,

I는 항등 행렬

C는 비용 벡터

D는 상기와 같으며

$\mathbf{B} = \frac{\mathbf{A}\mathbf{D}}{\mathbf{e}^T}$ 인 관계

$\mathbf{C}_p = \{I - \mathbf{B}^T (\mathbf{B}\mathbf{B}^T)^{-1} \mathbf{B}\} \mathbf{D} \mathbf{C}$ 에 의해 비용 벡터 \mathbf{C}_p 를 결정하는 단계와,

- 4) 관계

$\mathbf{C}_p = \frac{\mathbf{C}_p}{\|\mathbf{C}_p\|}$ 에 의해 상기 비용 벡터를 정규화하는 단계와,

- 5) 상기 아핀 공간에서

r은 상기 아핀 공간에서 가장 큰 내접구 (inscribed sphere)의 반경으로

$r = \frac{1}{n(n-1)}$ 이고

$\alpha = 1$ 보다 작은

$b' = x^{\text{start}} - \alpha r \mathbf{C}_p$ 에 의해 새로운 초기 할당 b' 를 선택하는 단계와,

- 6) 상기 새로운 초기 할당점을 변환

$\mathbf{b} = \frac{\mathbf{D} \mathbf{b}'}{\mathbf{e}^T \mathbf{D} \mathbf{b}'}$

를 이용하여 원래 공간으로 다시 변환하는 단계, 및

- 7) 새로운 초기 할당으로서 \mathbf{a}_o 대신 \mathbf{b} 를 사용하여 단계(2) 내지 (6)을 반복하는 단계를 포함하는 것을 특징으로 하는 산업적 또는 기술적 자원 할당 방법.

청구항 25

벡터 \mathbf{b} 는 다수의 종결 결과를 표시하고, 벡터 \mathbf{x} 는 다수의 자원에 요구되는 기여 (contribution) 집합을 표시하고, 벡터 \mathbf{c} 는 비용 계수 집합을 표시하며, 행렬 \mathbf{A} 는 시스템 제한을 표시할 때, 다수

의 기술적 종료결과 b 를 성취하기 위해 일제히 동작하는 다수의 n 개 산업적 자원으로서, 상기 산업적 자원의 각각은 시스템의 특징을 기술하는 제한들에 복종하여 부수 비용 계수와 함께 상기 종료 결과의 각각에 기여하는 상기 산업적 자원을 갖는 시스템에서 총 비용을 개선하는 방법에 있어서, X 의 현재 견적으로서 상기 시스템 제한을 만족하는 기여 집합 X^{curr} 를 선택하고, 다른 부분은 영인 대각선 행렬 D 의 대각선을 따라 상기 현재견적 X^{curr} 를 배열하고, 행렬 곱 AD 를 전개하고 이 전개된 곱을 n 개의 1로 이루어진 최종행으로 증가시킴으로써 행렬 B 를 향상하고, $[I - B^T(BB^T)^{-1}B]DC$ 를 평가함으로써 포인터 벡터 Cp 를 전개하고 (I 는 항등 행렬임), 정규화된 포인터 Cp 를 전개하기 위하여 Cp 를 자신의 크기로 나누고, e/n 에서 αCp 를 뺀으로써 X 의 변환된 새로운 견적 X' 를 만들어내고 (α 는 1보다 작다), $e=\{1, 1, \dots, 1\}$ 인 Dx'/e Dx' 를 평가함으로써 X 의 변환되지 않은 새로운 견적 X^{next} 를 만들어내며 ($e=\{1, 1, \dots, 1\}$ 임), X 의 상기 새로운 견적 X^{next} 를 상기 시스템에 인가하는 것을 특징으로 하는 시스템의 총 비용 개선 방법.

청구항 26

산업적 또는 기술적 자원 할당 방법에 있어서, 상기 자원과 관련된 변수에 대한 값의 결정과 상기 변수의 목적 함수의 값을 최적화하기 위해 볼록면체 집합인 상기 변수 및 실행될 상기 결정의 실행 가능한 결합 집합을 포함하고, 상기 결정은 각 단계에서 상기 변수에 대해 일시적인 값이 대체되는 일련의 단계로 이루어지고, 상기 대체 작업은 상기 볼록면체 세트를 집중화함으로써 얻어진 집합내의 방향을 선택하는 것에 근거하는 것을 특징으로 하는 산업적 또는 기술적 자원 할당 방법.

청구항 27

선형 목적 함수의 각 요소가 개개의 시스템 실체에 분배가능한 특정 자원 할당을 나타내고, 변수 및 공지된 변수 계수를 포함하며, 하나 또는 그 이상의 제한 선형 형태가 목적 함수의 하나 또는 그 이상의 변수의 항으로 표현되는 선형 목적 함수에 의해 특징이 기술되어지는 시스템에서 자원 할당을 최적화하는 방법에 있어서,

- 1) 초기 값에 의해 정의된 n 차원 공간의 벡터가 상기 제한 선형 형태에 의해 정의된 폴리토프내에 존재하도록 목적 함수의 각 변수의 초기 값을 결정하는 단계와,
- 2) 초기 벡터 및 제한 선형 형태를 포함하는 폴리토프를 실질적으로 초기 벡터의 원점이 그 중심상에 위치되는 심플렉스 $S=\{X \mid X \geq 0, \sum X_i=1\}$ 로 변환되는 단계와,
- 3) 변환된 초기 벡터를 직각으로 심플렉스에 사영하는 단계와,
- 4) 상기 심플렉스에서 변환된 초기 벡터의 사영 방향을 결정하는 단계와,
- 5) 심플렉스 S 의 중심 e/n 으로부터, 상기 심플렉스에 내접하고 변환된 초기 벡터의 원점에 중심이 있는 가장 큰 구의 반경의 배수와 동일한 상기 심플렉스내의 거리만큼 상기 결정방향의 반대 방향으로 이동시킴으로써 새로운 초기 벡터에 대한 새로운 시작 점을 결정하는 단계와,
- 6) 상기 새로운 시작 점을 폴리토프 공간으로 다시 변환하는 단계와,
- 7) 목적 함수가 만족스럽게 최소화될때까지, 목적 함수 변수의 초기 값을 상기 변환된 새로운 시작 점에 의해 정의된 값으로 대체시켜 단체(2) 내지 (6)을 반복하는 단계와, 상기 목적 함수 요소의 최종 값에 따라 시스템 자원을 개개의 시스템 실체(entities)에 할당하는 단계를 포함하는 것을 특징으로 하는 자원 할당 최적화 방법.

청구항 28

제 27 항에 있어서, 단계(2)가 목적 함수 변수의 초기 값들로 구성된 대각선 행렬에 제한 선형 형태의 계수 행렬을 곱함으로써 행렬 B 를 발생하고, 행의 각 행렬 위치에 1의 값을 갖는 부가적인 최하위 행을 행렬 B 에 더하는 단계를 더 포함하는 것을 특징으로 하는 자원 할당 최적화 방법.

청구항 29

제 27 항에 있어서, 단계(3)가 초기 벡터 곱하기 초기 변수 값의 대각 행렬 곱하기 행렬식 $[I - B^T(BB^T)^{-1}B]$ 으로부터 변환된 초기 벡터의 직각 사영을 계산하는 단계(여기에서 I 는 항등 행렬, B^T 는 상기 행렬 B 의 전치 행렬)와, 상기 직각 사영을 정규화하는 단계를 더 포함하는 것을 특징으로 하는 자원 할당 최적화 방법.

청구항 30

제 27 항에 있어서, 단계(5)가 상기 변형된 비용 벡터 곱하기($X^{start} - \alpha r$)의 값으로부터 새로운 변형된 초기 벡터를 계산하는 단계(여기에서 $X^{start} = e/n$, r 은 상기 내접구의 반경, α 는 미리 선택된 상수)를 더 포함하는 것을 특징으로 하는 자원 할당 최적화 방법.

청구항 31

제 30 항에 있어서, 공식 $1/\sqrt{n(n-1)}$ 로부터 상기 반경을 계산하는 단계를 더 포함하는 것을 특징으로 하는 자원 할당 최적화 방법.

청구항 32

목적 함수의 각 요소가 개개의 시스템 실체에 분배 가능한 특정 자원 할당을 나타내고 변수와 공지된 변수를 포함하며, 하나 또는 그 이상의 제한 관계가 하나 또는 그 이상의 목적 함수 변수의 항으로 표현되는 n 차원 목적 함수에 의해 특징이 기술되어지는 시스템에서 자원 할당을 최적화하는 방법에 있어서,

- 1) 초기 값에 의해 정의된 초기 벡터가 상기 제한 관계에 의해 정의된 폴리토프내에 존재하도록 목적 함수의 각 변수의 초기 값을 결정하는 단계와,
- 2) 초기 벡터와 제한 관계를 포함하는 폴리토프를 실질적으로 변형된 초기 벡터가 그 중심상에 위치되는 심플렉스 $S=\{X | X \geq 0, \sum X_i=1\}$ 로 변환되는 단계와,
- 3) 상기 목적 함수의 변환된 벡터를 상기 변환된 제한 관계의 공백 공간상에 직각으로 사영하는 단계와,
- 4) 상기 변형된 목적 함수의 상기 사영 방향을 결정하는 단계와,
- 5) 상기 심플렉스의 중심 e/n 으로부터, 상기 심플렉스내에 포함되고 상기 변형된 초기 벡터의 원점에 중심이 있는 가장 큰 구의 반경의 사전 정의된 배수와 동일한 거리만큼 이동시킴으로써 상기 목적 함수의 각 변수에 대해 새로운 값을 결정하는 단계와,
- 6) 상기 새로운 값을 원래 변수로 다시 변환하는 단계와,
- 7) 목적 함수가 만족스럽게 최소화될 때까지, 상기 목적 함수 변수의 초기 값을 상기 새로운 값으로 대체시켜 단계(2) 내지 (6)를 반복하는 단계와,
- 8) 상기 목적 함수 요소의 최종 값에 따라 개개의 시스템 실체에 시스템 자원을 할당하는 단계를 포함하는 것을 특징으로 하는 자원 할당 최적화 방법.

청구항 33

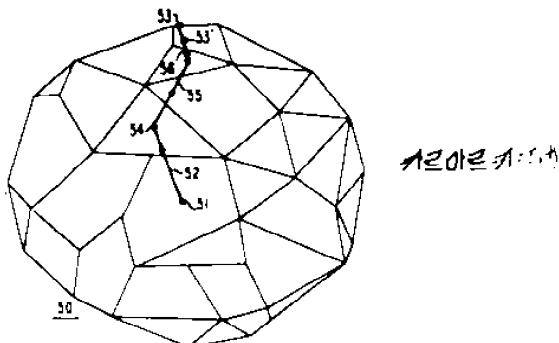
제 32 항에 있어서, 단계(2)가, 상기 제한 관계의 계수 행렬로써 상기 목적 함수 변수의 초기 값을 들로 구성된 대각선 행렬에 상기 제한 관계의 계수 행렬을 곱함으로써 행렬 B 를 발생하는 단계와, 상기 행의 각 행렬 위치에 1의 값을 갖는 부가적인 최하위 행을 행렬 B 에 더하는 단계를 더 포함하는 것을 특징으로 하는 자원 할당 최적화 방법.

청구항 34

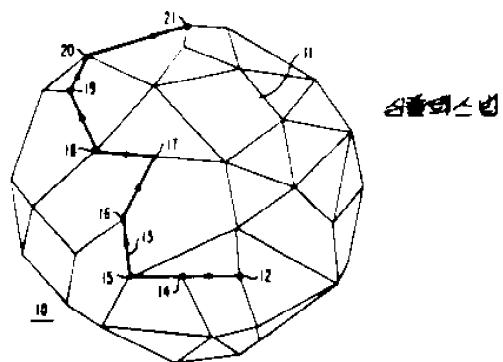
제 33 항에 있어서, 단계(3)가 행렬식 $[I - B^T (BB^T)^{-1} B]$ 곱하기 초기 벡터로부터 변환된 초기 벡터의 각 사영을 계산하는 단계(여기에서 I 는 항등 행렬, B^T 는 행렬 B 의 전치 행렬)와, 상기 각 사영을 사전 정의된 방식으로 정규화하는 단계를 더 포함하는 것을 특징으로 하는 자원 할당 최적화 방법.

도면

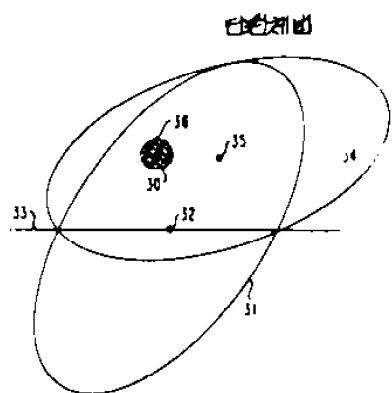
도면1



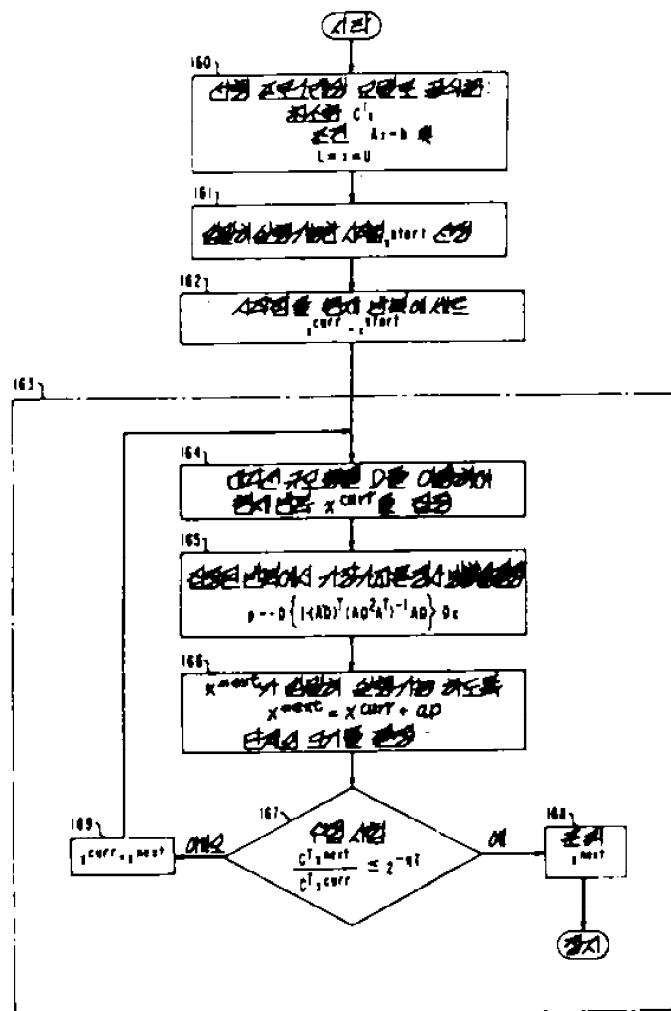
도면2



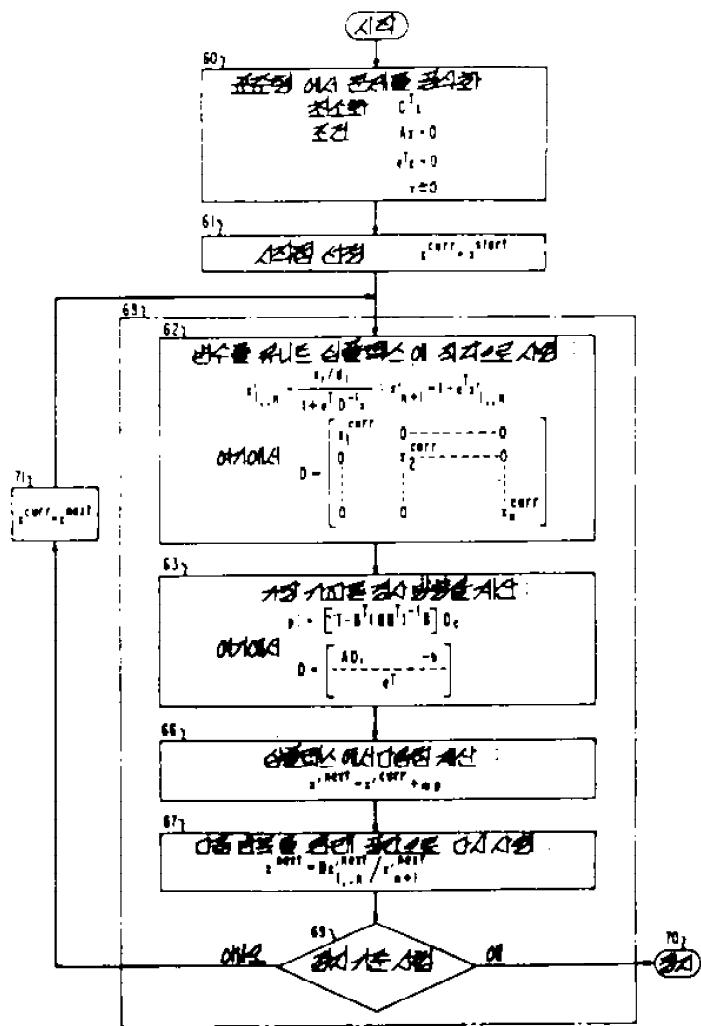
도면3



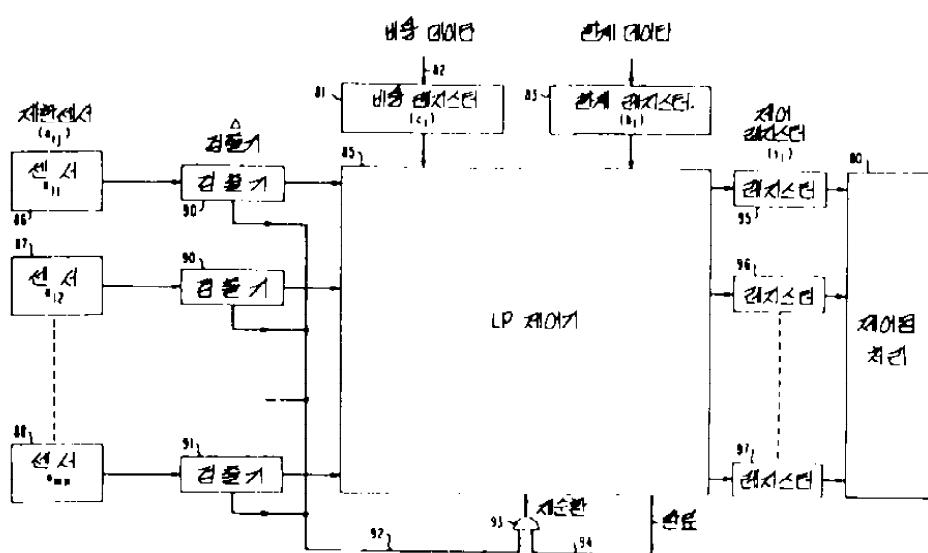
도면4



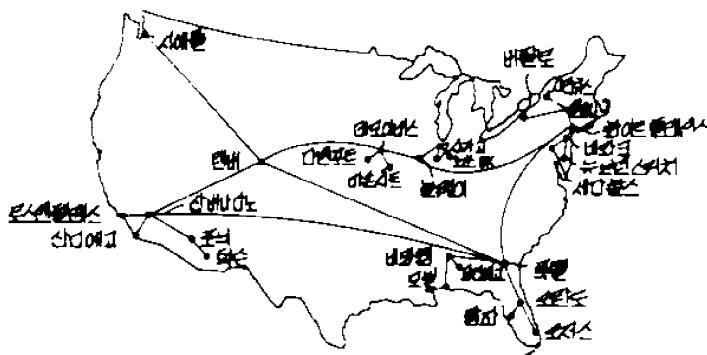
도면5



도면6



도면7



도면8

