김일성종합대학학보
정보과학

주체109(2020)년
제66권 제2호

0－1옹근수계획법문제를 병렬처리하기 위한 주콤퓨터의 알고리듬

전철용, 김유성

0－1옹근수계획법문제는 한가지 특수한 선형계획법문제로서 배낭문제, 공장부지설정문제, 가공문제 등 많은 문제들이 여기에 속한다.

현재 0－1옹근수계획법문제를 풀기 위한 풀이법들이 많이 나왔으나 변수의 개수나 제한식의 개수가 많아지면 계산시간이 매우 길어지거나 심지어 계산못하는 경우까지도 있게 된다.

선행연구[1]의 간접렬거법은 비교적 계산속도가 빠르지만 직렬계산법으로서 역시 규모가 큰 문제들에 대하여서는 계산을 못하는 경우가 있게 된다.

론문에서는 규모가 큰 0－1옹근수계획법문제를 처리하기 위한 한가지 병렬계산법을 제안하였다.

0－1옹근수선형계획법문제는 다음과 같이 정식화된다.

$$\min\{\boldsymbol{c}^{\mathrm{T}}x \mid Ax \le \boldsymbol{b}, x=(\xi_1, \xi_2, \cdots, \xi_n),\ \xi_i \in \{0, 1\}(i=1, 2, \cdots, n)\}$$

여기서 $A=(a_{ij}) \in Z^{m\times n}$, $\boldsymbol{Z}$는 어떤 옹근수모임, $\boldsymbol{b}$는 m차원, $\boldsymbol{c}$는 n차원옹근수벡토르이다.

벡토르 $\boldsymbol{c}=(c_1, c_2, \cdots, c_n)^{\mathrm{T}}$에 대하여 일반성을 잃지 않으면서 $0\le c_1 \le c_2 \le \cdots \le c_n$이라고 하자.

이때 0－1옹근수계획법문제에는 2^n개의 벡토르가 존재한다.[2]

그러면 임의의 벡토르는 다음의 $n+1$개 모임중의 하나에 속한다.

$$X^k = \left\{ x \left| \sum_{j=1}^{n} \xi_j = k \right. \right\} (k=0, 1, \cdots, n)$$

웃식은 k가 옹근수값 $0, 1, \cdots, n$을 취하는 보조변수방정식이다.

정리 1 제한조건

$$\sum_{j=1}^{n} a'_{ij}\,\xi_j \le b'_i,\ \ i=1, 2, \cdots, m$$

은 주어진 0－1계획법의 제한조건과 동등하며 결수들은 모두 부아닌 수들이다. 여기서

$$a_i^L = \min_{1\le j\le n}\{a_{ij}\},\ \ a'_{ij}=a_{ij}-a_i^L,\ \ b'_i=b_i-ka_i^L\ \ (i=1, 2, \cdots, m;\ j=1, 2, \cdots, n)$$

이다.

임의의 X^k에서 최량풀이를 구한다고 하면 주어진 0－1옹근수계획법문제의 제한조건은 정리 1에 의하여 동등하게 변환되며 서로 다른 X^k에 대하여 제한조건도 다르다는 것을 알수 있다. 즉 k가 주어지면 그것에 대한 제한조건이 성립됨으로써 새로운 0－1옹근수계획법문제가 생겨난다고 볼수 있으며 이것은 주어진 문제를 병렬처리하기 위한 기

본조건으로 된다.

정리 2 만일

$$k^L = \max_{1\le i\le m+1}\left\{\frac{b_i}{a_i^L}\,|\,a_i^L<0\right\},\ k^U = \min_{1\le i\le m+1}\left\{\frac{b_i+\sum_{j=1}^{n}a_{ij}''}{a_i^U}\,|\,a_i^U>0\right\}$$

이라고 하면 0－1옹근수계획법문제에서 보조변수 k의 값은

$$k\ge k^L,\ k\le k^U$$

이다. 여기서

$$a_i^U=\max_{1\le j\le n}\{a_{ij}\},\ a_{ij}''=a_i^U-a_{ij},\ b_i''=ka_i^U-b_i\,(i=1,\,2,\cdots,\,m;\ j=1,\,2,\cdots,\,n)$$

이다.

정리 3 X^k의 모든 벡토르들에 대응하는 목적함수값 $f^{(k)}$는

$$\sum_{j=1}^{k}c_j\le f^{(k)}\le\sum_{j=n-k+1}^{n}c_j$$

를 만족시킨다.

병렬계산에서 주콤퓨터의 알고리듬을 정리 1, 2, 3을 리용하여 다음과 같이 작성할수 있다.

걸음 1 주어진 0－1옹근수계획법문제를 입력하고 X^0의 유일한 벡토르

$$x=(0,\,0,\cdots,\,0)$$

에 대한 계산을 진행한다. 만일 이 벡토르가 허용벡토르 즉 $b_i\ge 0(i=1,\,2,\cdots,\,m)$이면

$$f^*=0,\ x^*=(0,\,0,\cdots,\,0)$$

이라고 하고 걸음 4로 이행한다. 그렇지 않으면 다음걸음으로 이행한다.

걸음 2 $x^*=(0,\,0,\cdots,\,0)\in R^n,\ f^*=\sum_{j=1}^{n}c_j+1$이라고 하고 정리 2를 리용하여 보조변수 k의 상계 k^U와 하계 k^L을 계산한다. 만일 $k^U<k^L$이라고 하면 걸음 4로 이행하고 그렇지 않으면 다음걸음으로 이행한다.

걸음 3 $k^L\le k\le k^U$을 만족시키는 k에 대하여

$$\sum_{i=1}^{k}c_i\ge f^*$$

이면 걸음 4로 이행하고 아니면 매 마디콤퓨터들에 정리 1을 리용하여 서로 다른 k에 따르는 제한식과 최량벡토르계산지령을 준다.

걸음 4 어느 한 마디콤퓨터에서 k에 대한 최량벡토르를 구하고 이 최량벡토르를 x^*로, 이때의 목적함수값을 f^*로 하고 걸음 3으로 이행한다.

걸음 5 $f^*=\sum_{j=1}^{n}c_j+1$이면 풀이가 존재하지 않는다고 출력하고 아니면 x^*을 최량벡

토르, f^*을 최량값으로 출력하고 계산을 끝낸다.

마디콤퓨터들의 계산법은 선행연구[1]에서 제출한 직렬계산법과 같으며 하나의 0－1옹근수계획법문제를 우의 알고리듬을 리용하여 여러대의 콤퓨터로 계산하면 그 계산시간은 몇배, 몇십배로 당겨진다.

맺 는 말

0－1옹근수계획법문제를 병렬처리하기 위하여 2^n개의 벡토르를 $n+1$개 모임으로 분류한데 기초하여 매 모임에서 최량풀이를 구하기 위한 주콤퓨터의 알고리듬을 제안하였다.

참 고 문 헌

[1] 김유성 등; 조선민주주의인민공화국 과학원통보, 6, 16, 주체104(2015).

[2] B. Soylu; European Journal of Operational Research, 245, 3, 690, 2015.

주체109(2020)년 2월 5일 원고접수

Algorithm of Master Computer for Parallel Processing of 0−1 Integer Programming Algorithm

Jon Chol Yong, *Kim Yu Song*

In this paper we studied an algorithm of basic computer that got the optimal solution in each set, being based on classifing 2^n vectors in the $n+1$ sets, about for parallel processing of 0−1 integer programming algorithm.

Keywords: 0−1 integer programming, parallel algorithm