BİR BİLGİSAYAR HABERLEŞME ŞEBEKESİNDE LAGRANGE YÖNTEMİ İLE OPTİMUM BAĞLANTI VE AKIŞ MALİYETİNİN HESAPLANMASI İÇİN BİR ALGORİTMA

Dr. Mehpare TİMOR

ÖZET:

Bu çalışmada, bir bilgisayar haberleşme şebekesinde terminaller arasında optimum maliyetli bağlantı ve haber akışının sağlanması hedeflenmektedir. Merkezi birime bağlanacak olan terminaller arasında Minimum Maliyetli Ağaç yapısına uygun hatlar kurularak, bağlantı kurulan hatlardan geçecek olan mesaj akış maliyeti minimize edilecektir. Bu amaçla doğrusal programlama ile ifade edilen probleme **Lagrange Yöntemi** uygulanarak etkin bir çözüm algoritması geliştirilmeye çalışılmıştır. Geliştirilen algoritmanın çeşitli örnekler üzerinde test edilmesi ile büyük ölçekli şebekeler için de etkin sonuçlar üretilebildiği gösterilmiştir.

1) GİRİŞ

İşletmecilikte yaygın kullanım alanı bulan Yöneylem Araştırması Tekniklerinden birisi Şebeke Problemleridir. Dağıtım kanalları, boru hattı taşıma sistemleri, cadde ve sokak sistemleri, personel atama, teçhizat yenileme ve çok önemli üretim planı problemleri şebeke problemlerinin uygulama alanlarındandır[(1)]. Geniş bir bilimsel ve endüstriyel uygulama alanı gösteren şebeke problemlerine nakit akışı, havayolu taşımacılığının programlanması, mal ve malzeme nakliyatı, su kaynaklarının optimum kullanımı, makine yükleme, haberleşme ve ulaşım problemleri ör-

(1) M.M. SYSLO, N. DEO, J.S. KOWALİK, **Discrete Optimization Algorithms with Pascal Programs**, (New Jersey: Prentice-Hall, Inc., 1983), S. 54.

nek verilmektedir(2). Şebeke problemlerinin özel bir hali olan haberleşme problemlerine bilgisayar haberleşme şebekeleri, havayolu trafik sistemi, posta dağıtım kanalları örnek verilebilir.

Şebeke problemlerinin özel bir şekli olan bilgisayar haberleşme şebekelerinin tasarımı oldukça karmaşık bir işlemdir(3). Bu nedenle bir bütün halinde çözümü güç olan problem, çözülmesi nispeten daha kolay olan alt gruplara ayrılmaktadır. Herbir alt grupta yer alan problemler belirli iterasyonlar sonucu çözülmek suretiyle problemin bütünü için kabul edilebilecek bir çözüm elde edilmesine çalışılmaktadır(4).

Bilgisayar haberleşme şebekesi tasarımında, merkezi birim ile haberleşecek olan terminallerin doğrudan merkezi birime bağlanması tercih edilebilir. Ancak bu bağlantı şeklinden daha düşük maliyetli ve Minimum Maliyetli Ağaç (MMA) olarak adlandırılan bir şebeke yapısı mevcut olup, herbir düğümün merkezi birime doğrudan bağlanması yerine, yüksek taşıma kapasiteli hatların birden fazla terminal tarafından ortaklaşa kullanımı şeklindeki Minimum Maliyetli Ağaç yapısı, diğer bağlantı şekillerine göre(5), toplam maliyeti minimize etmesi nedeniyle üstünlük göstermektedir(6).

Şebeke problemleri doğrusal programlama ile formüle edilerek çözülebilmektedir. Ancak, şebekedeki düğüm sayısının artması halinde mevcut doğrusal programlama paketleri ile problemin bilgisayar çözümünü elde etmek imkansız hale gelmektedir. Düğüm sayısı arttıkça üstel karmaşıklık gösteren şebeke problemlerinin çözümünde Sezgisel Yöntemle (Heuristic) birlikte çalışan Lagrange Yöntemi (Lagrange Relaxation) kullanılması sonucu optimuma çok yakın (veya optimum) bir çözüm değeri elde edilebilmektedir. Problemin sergilediği karmaşık yapı dikkate alındığında, elde edilen çözümün optimumluktan ne derece uzak olduğunun belirlenmesi ve bu ölçünün kabul edilebilir sınırlar içinde kalması halinde etkin bir algoritma geliştirildiği ifade edilmektedir.

---

(2) F.S. HILLIER, G.J. LIEBERMAN, Introdution to Operatiorts Research, (New York: Mc Graw-Hill Publishing Co., 1990), S. 333.

(3) V. AHUJA, Design and Analysis of Compufer Communication Networks, (New York: McGraw-Hill, 1987), S. 3.

(4) H. PİRKUL, S. NARASİMHAN, P. DE, "Locating Concentrators for Primary and Secondaıy Coverage in a Computer Communications Network", IEEE Transactions, V. 36, No. 4, April 1988, S. 450.

(5) Diğer bağlantı şekilleri hakkında ayrıntılı bilgi için bkz. R.D. DOWSING, F.W.D. WOODHAMS, Compufers form Locic to Architecture. (London: Van Nostrand Reinhold Co. Ltd., 1990), S. 245.

(6) AHUJA, op. cit., S. 123.

Bu çalışmada, bir haberleşme şebekesinde yer alan merkezler arasında optimum maliyetli bağlantı ve mesaj akışını sağlamak üzere, haberleşme problemi doğrusal programlama ile ifade edilmektedir. Şebekede yer alan terminaller arasında kurulacak olan ana bağlantıların Minimum Maliyetli Ağaç yapısına uygun olması, birincil (ana) terminaller ile ikincil terminaller arasındaki bağlantıların da minimum maliyetle gerçekleştirilmesi amaçlanmaktadır. Aynı zamanda bu merkezler arasında, mevcut bağlantılar üzerinden geçecek olan akışın maliyetinin de minimize edilmesi hedeflenmektedir. Doğrusal programlama ile formüle edilen problemin çözümünün elde edilebilmesi için Lagrange Yöntemi kullanılmakta ve böylece problem, çözülmesi "göreceli olarak daha kolay" olan yeni bir probleme indirgenmektedir. Lagrange yöntemi ile orjinal probleme bir sınır değeri (minimizasyon problemi için alt sınır) üretilmektedir. Üretilen bu sınır değeri genellikle orjinal problem için uygun çözüm oluşturmamaktadır. Bu nedenle, probleme bir uygun çözüm üretebilmek için, her problemin yapısına uygun olarak tasarlanan bir sezgisel yaklaşım kullanılmaktadır. Lagrange Yöntemi ile probleme sınır değerleri üretilirken Aşamalı optimizasyon Yönteminden (Subgradient Optimization) yararlanılmaktadır. Algoritma sona erdiğinde sınır ve uygun çözüm değerleri arasında makul sayılabilecek bir fark (gap) kalabilmektedir. Üstel karmaşıklık gösteren bir problem için, uygun bir hesaplama zamanı içinde, optimum çözüme yakın bir sonuç elde edilebildiğinde yeteri kadar iyi bir algoritma geliştirildiği ifade edilebilmektedir.

2) MODEL FORMÜLASYONU

Bu bölümde, bir bilgisayar haberleşme şebekesinde yer alan terminaller arasındaki bağlantıların maliyeti ile terminallerden merkezi birime ulaşan mesaj akış maliyetinin minimize edilebilmesi için problem doğrusal programlama ile formüle edilmiştir. Problemdeki terminal sayısının, bağlantı maliyetinin ve birim mesaj akış maliyetinin bilindiği varsayılmaktadır. İkincil terminaller en yakın birincil (ana) terminale veya doğrudan merkezi birime bağlanabilmekdedir. Tüm terminaller merkezi birimle haberleşeceğinden doğrudan veya dolaylı olarak merkezi birime bağlanmalıdır. Amaç, ikincil ve birincil terminallerin bağlantılarının toplam maliyetini minimum kılmak, aynı zamanda bu bağlantılar üzerinden gerçekleşecek olan toplam akış maliyetini minimize etmektir.

Modelde kullanılan değişkenler ve bu değişkenlere ilişkin açıklamalar aşağıda verilmiştir:

C = Merkezi Birim
J = Birincil Terminaller Seti
I = İkincil Terminaller Seti
c_{ij} = i. terminalin j. terminale bağlanmasının maliyeti

$\hat{c}_{ij}$ = i. terminalle j. terminal arasında MMA kurulmasının maliyeti
$\tilde{c}_{ij}$ = Bir birim akışın i'den j'ye taşınmasının maliyeti

$$x_{ij} = \begin{cases} 1 & \text{i ikincil terminali j birincil terminaline bağlanmışsa,} \\ 0 & \text{bağlanmamışsa} \end{cases}$$

f_i = i ikincil terminalinden kaynaklanan akış miktarı

$$y_{ij} = \begin{cases} 1 & \text{i birincil terminali j birincil terminaline bağlanmışsa,} \\ 0 & \text{bağlanmamışsa} \end{cases}$$

f_{ij} = i merkezi ile j merkezi arasındaki hatta gerçekleşen akış miktarı

F = İkincil terminallerden kaynaklanan akış miktarları toplamı

olmak üzere problem üç alt grup halinde ele alınmaktadır;

a) İkincil Terminaller Birincil terminallere minimum maliyetle bağlanmalıdır,

b) Birincil terminaller merkezi birim ile Minimum Maliyetli Ağaç (Minimum Spanning Tree) oluşturacak şekilde bağlanmalıdır,

c) Tüm merkezler arasındaki akış, minimum maliyetle (Minimum-Cost Flow) gerçekleşmelidir.

Yukarıda tanımlanan değişkenler kullanılarak, problem aşağıdaki şekilde formüle edilebilir:

PROBLEM – P:

Amaç Fonksiyonu:

$$Z_p = \text{Min} \ \left(\sum_{i \in I} \ \sum_{j \in J \cup C} c_{ij} \ x_{ij} + \sum_{i \in J} \ \sum_{j \in J \cup C} \hat{c}_{ij} \ y_{ij} + \sum_{i \in J} \ \sum_{j \in J \cup C} \tilde{c}_{ij} \ f_{ij} \right)$$

Kısıtlar:

(1) $\sum_{j \in JUC} x_{ij} = 1$ $(\forall\, i \in I,\ C: \text{Merkezi Birim})$

(2) $f_{ij} \leq x_{ij}\ f_i$ $(\forall\, i \in I,\ j \in JUC)$ $(x_{ij} = 0,1)$

(3) Minimum maliyetli ağaç bağlantısı,

(4) $f_{ij} \leq y_{ij} * F$ $(\forall\, i \in J,\ \forall\, j \in JUC)$ $(\sum_{i \in I} f_i = F$: Toplam Akış$)$

(5) $\sum_{j \in JUC} f_{ij} = f_i$ $(\forall\, i \in I)$

(6) $\sum_{i \in I} f_{ic} + \sum_{j \in J} f_{jc} = \sum_{i \in I} f_i \ (=F)$

(7) $\sum_{i \in I} f_{ij} + \sum_{l \in J} f_{lj} - \sum_{k \in JUC} f_{jk} = 0$ $(\forall\, j \in J)$

Amaç fonksiyonunun ilk kısmı, ikincil terminallerin bağlantılarını ve bu bağlantıların maliyetlerini, ikinci kısmı birincil terminaller arasında kurulacak bağlantıları, üçüncü kısmı ise akış maliyetlerini temsil etmektedir. Terminaller arasındaki bağlantı batlarının her ne kadar sınırlı kapasiteleri mevcut olacaksa da, gerekli kapasiteler model tarafından hesaplanmaktadır.

Modelde yer alan (1). kısıt ile tüm ikincil terminallerin sadece ve sadece tek bir üniteye bağlanması sağlanmaktadır. (2). kısıt, ikincil terminallerden kaynaklanan akışa bağlantı bulunması halinde izin vermektedir. (3). kısıt birincil terminaller arasındaki bağlantıların Minimum Maliyetli Ağaç yapısına uygun olmasını sağlamaktadır. (4). kısıt sayesinde birincil terminaller arasında bağlantı kurulmuş olması halinde akışa izin verilmektedir. (5). kısıt, bir ikincil terminalden kaynaklanan akışın mutlaka herhangi bir birincil terminal veya merkezi birim tarafından alındığını göstermektedir. (6). kısıt ile ikincil ve birincil terminallerden merkezi birime gönderilen toplam akışın hedef akış miktarına eşit olmasını sağlamaktadır. (7). kısıt, ikincil terminallerden kaynaklanan ve şebeke üzerinde dolaşan akışın tamamının merkezi birim tarafından alındığını, akışta herhangi bir kayıp olamayacağını ifade etmektedir. Bu koşul "Akışın muhafazası" (Flow Conservation) olarak bilinmektedir[7].

(7) SYSLO, DEO, KOWALİK, op. cit., S. 56.

3) PROBLEMİN ÇÖZÜMÜ

Optimizasyon problemlerinin bir kısmı kolaylıkla çözülebiliyorken, geriye kalan büyük bir grup çözümü zor problemleri oluşturmaktadır. Problemin yapısı veya ölçeği nedeniyle çözümü güç olan NPC sınıfı problemler[8] için etkin çözüm algoritmalarına gerek duyulmaktadır. Bu tür problemlerde paket program kullanılması halinde sadece küçük ölçekli problemlerin (özellikle tamsayılı programlama problemlerinde) çözülmesi mümkün olabilmektedir. Problemin ölçeğinin büyümesi halinde paket programların ihtiyaca cevap verememesi nedeniyle etkin çözüm üreten algoritmaların geliştirilmesi gerekmektedir. Bir tamsayılı programlama problemi olan Problem-P'nin çözülebilmesi için, Lagrange Yöntemi uygulanmakta, aynı zamanda Lagrange yöntemi ile birlikte çalışan bir sezgisel yöntem geliştirilmek suretiyle orjinal problem için uygun çözüm değerleri üretilmektedir.

3.1) LAGRANGE YÖNTEMİ İLE PROBLEMİN ÇÖZÜMÜ

Problemi üç alt grup halinde ele almak mümkündür:

Birinci Grubun Amaç Fonksiyonu	**İkinci Grubun Amaç Fonksiyonu**	**Üçüncü Grubun Amaç Fonksiyonu**

$$Z_p = \text{Min} \left(\sum_{i \in I} \sum_{j \in JUC} c_{ij} x_{ij} + \sum_{i \in J} \sum_{j \in JUC} \hat{c}_{ij} y_{ij} + \sum_{i \in J} \sum_{j \in JUC} \tilde{c}_{ij} f_{ij} \right)$$

Birinci Grup Kısıtlar:	İkinci Grup Kısıtlar:	Üçüncü Grup Kısıtlar:

$$\sum_{j \in JUC} x_{ij} = 1 \quad (1)$$

$$f_{ij} \leq x_{ij} \; f_i \quad (2)$$

Minimum Maliyetli Ağaç (MMA) (3)

$$f_{ij} \leq y_{ij} * F \quad (4)$$

$$\sum_{j \in JUC} f_{ij} = f_i \quad (5)$$

$$\sum_{i \in I} f_{ic} + \sum_{j \in J} f_{jc} = \sum_{i \in I} f_i \quad (6)$$

$$\sum_{i \in I} f_{ij} + \sum_{l \in J} f_{lj} = \sum_{k \in JUC} f_{jk} = 0 \quad (7)$$

(8) NPC sınıfı problemler hakkında ayrıntılı bilgi için bkz. M.R. GAREY, D.S. JOHNSON, **Computers and Intractability A Guide to the Theory of NP-Completeness, A.B.D.:** Bell Telephone Lab., Inc., 1979.

P problemine sınır değerleri üretebilmek için probleme Lagrange Yöntemi uygulanmaktadır. Modeldeki (2). ve (4). kısıtlar α ve β ile çarpılarak amaç fonksiyonuna ilave edilmektedir; $f_{ij} - x_{ij} f_i \leq 0$ eşitsizliği $\alpha_{ij} \geq 0$ ile çarpılırsa eşitsizliğin yönü değişmez. $\alpha_{ij} (f_{ij} - x_{ij} f_i) \leq 0$'dır. Aynı şekilde $f_{ij} - y_{ij} * F \leq 0$ eşitsizliği $\beta_{ij} \geq 0$ ile çarpılırsa yine eşitsizlik aynı kalır; $\beta_{ij} (f_{ij} - y_{ij} * F) \leq 0$'dır. Dual değişkenler olarak adlandırılan α ve β ile çarpılan (2). ve (4). kısıtların amaç fonksiyonuna ilave edilmesi ile elde edilen yeni formülasyon aşağıda verilmiştir:

$$Z_L = \text{Min} \ \{\sum_{i \in I} \sum_{j \in JUC} c_{ij} x_{ij} + \sum_{i \in J} \sum_{j \in JUC} \hat{c}_{ij} y_{ij} + \sum_{i \in J} \sum_{j \in JUC} \tilde{c}_{ij} f_{ij})$$

$$\underset{(\forall i \in I,\ j \in JUC)}{\alpha_{ij} (f_{ij} - x_{ij} f_i)} + \underset{(\forall i \in J,\ J \in JUC)}{\beta_{ij} (f_{ij} - y_{ij} * F)}\}$$

(1) $\sum_{j \in JUC} x_{ij} = 1 \qquad (\forall i \in I,$ C: Merkezi Birim)

(3) Minimum maliyetli ağaç bağlantısı,

(5) $\sum_{j \in JUC} f_{ij} = f_i \qquad (\forall i \in I)$

(6) $\sum_{i \in I} f_{ic} + \sum_{j \in J} f_{jc} = \sum_{i \in I} f_i \quad (=F)$

(7) $\sum_{i \in I} f_{ij} + \sum_{l \in J} f_{lj} - \sum_{k \in JUC} f_{jk} = 0 \quad (\forall\ j \in J)$

Problem yeniden düzenlenirse;

$$Z_L : \text{Min} \ \{\sum_{i \in I} \sum_{j \in JUC} (c_{ij} - \alpha_{ij} f_i) x_{ij} + \sum_{i \in J} \sum_{j \in JUC} (\hat{c}_{ij} - \beta_{ij} * F) y_{ij} +$$

$$+ \sum_{i \in I} \sum_{j \in JUC} \alpha_{ij} f_{ij} + \sum_{i \in I} \sum_{j \in JUC} (\tilde{c}_{ij} + \beta_{ij}) f_{ij}\}$$

(1) $\sum_{j \in JUC} x_{ij} = 1 \qquad (\forall i \in I,$ C: Merkezi Birim)

(3) Minimum maliyetli ağaç bağlantısı,

(5) $\sum_{j \in JUC} f_{ij} = f_i \qquad (\forall i \in I)$

(6) $\sum_{i \in I} f_{ic} + \sum_{j \in J} f_{jc} = \sum_{i \in I} f_i \quad (=F)$

(7) $\sum_{i \in I} f_{ij} + \sum_{l \in J} f_{lj} - \sum_{k \in JUC} f_{jk} = 0 \quad (\forall\ j \in J)$

elde edilecektir.

Problem–P'ye Lagrange Yöntemi uygulanması sonucunda Z_L elde edilmektedir. Z_L'den yararlanarak sınır değerleri üretmede kullanılan α ve β değerlerinin belirlenmesi oldukça güçtür. İstisnalar dışında bu işlemin çok zor olduğu gösterilmiştir[9]. Uygulamada optimal olmasa da "iyi" olarak ifade edilblecek çarpanların hesaplanmasında Aşamalı Optimizason Yöntemi (Subgradient Optimization) veya Çarpan Ayarlama Yöntemi (Multiplier Adjustment Method) kullanılmaktadır[10]. Bu çalışmada anlaşılma ve uygulanmasındaki kolaylık açısından Aşamalı Optimizasyon Yöntemi tercih edilmiştir. Aşamalı Optimizasyon Yönteminde, Problem-L'ye sınır değerleri üretilirken kullanılan çarpanların belirlenmesi için, başlangıçta bir α° seçilip, bu değer kullanılarak α değerleri ardışık olarak aşağıdaki formül kullanılarak hesaplanmaktadır[11]:

$$\alpha^{k+1} = \alpha^k + t_k\,(Ax - b)$$

x_k, Z_L probleminin k. aşamadaki çözümü, t_k pozitif büyüklükte bir ölçek olmak üzere,

$$\sum_{i=0}^{\infty} t_k = \infty \qquad \text{olduğu gösterilmiştir.}$$

t_k'nin hesaplanmasında aşağıdaki formül kullanılmaktadır:

$$t_k = \frac{\delta_k(Z_f - Z_L{}^k)}{\|\, Ax - b \,\|^2}$$

Z_f tespit edilen en iyi uygun çözüm değeri, δ_k ise $0 \leq \delta_k \leq 2$ şartını sağlayan bir çarpandır. δ_k algoritmanın başında 2 alınarak, ilerleme kaydedilmeyen 15 iterasyon sonunda ikiye bölünmektedir. Algoritma önceden belirtilen sabit sayıdaki iterasyondan sonra sona erdirilmektedir[12].

(9) B. GAVISH, "On Obtaining the Best Multipliers for a Lagrangian Relaxation for Integer Programming", Comput. Oper. Res., (V. 5, 1978), SS. 55-71.

(10) M.L. FISHER, "The Lagrangian Relaxation Method For Solving Integer Programming Problems", Management Science, (U.S.A.: V. 27, No: 1, January 1981), S. 7-9.

(11) PİRKUL, NARASİMHAN, DE, op. cit., SS. 450-458.

(12) G.L. NEMHAUSER, L.A. WOLSEY, **Integer and Combinatorial Optimization**, New York: John Wiley & Sons, Inc., 1988, S. 46.

(Problemin güçlük derecesine göre işlemler 200 iterasyona[13], veya 500 iterasyona[14] kadar sürdürülmektedir.) Üretilen sınır ve uygun çözüm değerleri arasında kullanıcının belirleyeceği bir fark (gap) kaldığında da algoritma sona erdirilebilmektedir. Optimum çözüm, sınır ve uygun çözüm değerleri arasında yer almaktadır. Bu nedenle, uygun bir hesaplanma süresi içinde, optimum çözüm yerine bu çözüm değerinin çok yakınında bir çözümün elde edilmesi, büyük ölçekli şebeke problemlerine çözüm getirilebilmesini sağlamaktadır.

3.2. EN İYİ ÇÖZÜMÜN ARAŞTIRILMASINDA AŞAMALI OPTİMİZASYON YÖNTEMİNİN (SUBGRADIENT OPTIMIZATION) KULLANIMI

Lagrange yöntemi uygulanmış olan problemin çözümü için kullanılan aşamalı optimizasyon (Subgradient optimization) yönteminin aşamaları aşağıda sıralanmıştır:

1.) Maks. İterasyon = 200
En İyi Çözüm Değeri = ∞
$\alpha_{ij} = 0, \ \beta_{ij} = 0$
İterasyon = 1, $d_k = 2$, Sayaç = 0

2.) Aşağıdaki işlemler iterasyon sayısı > 200 oluncaya kadar uygulanmaktadır;
   a) Birinci alt problem çözülür; (Z_L probleminin ilk alt grubu çözülür.)
   b) İkinci alt problem çözülür, (Z_L probleminin ikinci alt grubu çözülür.)
   c) Üçüncü alt problem çözülür, (Z_L probleminin üçüncü alt grubu çözülür.)
   d) Her üç çözüm birarada dikkate almarak Yeni Sınır değeri hesaplanır,
   e) Eğer Yeni Sınır > En İyi Sınır değeri ise;
   En İyi Sınır = Yeni Sınır
   $\alpha'_{ij} = \alpha_{ij}, \quad \beta'_{ij} = \beta_{ij}$

(13) J. CURRENT, H. PİRKUL, E. ROLLAND, "Efficient Algorithms for Solving The Shortest Covering Path Problems", The Ohio State University Working Paper, WPS 90-56, 1990, S. 15.

(14) B. GAVISH, H. PİRKUL, "Algorithms for The Multi-Resource Generalized Assignment Problem", Management Science, (USA: V. 37, No. 6, June 1991), S. 700.

f) Model üzerinde yerel incelemeler gerçekleştirerek (sezgisel yaklaşım ile) problem için bir uygun çözüm bulunur,
Eğer Yeni Çözüm < En İyi Çözüm değeri ise;
En İyi Çözüm = Yeni Çözüm
$\alpha_{ij} = \alpha'_{ij}, \quad \beta_{ij} = \beta'_{ij}$
Aksi takdirde; Sayaç = Sayaç + 1
Eğer Sayaç > 15 ise Sayaç = 0, $d_k = d_k/2$
$\alpha_{ij} = \alpha'_{ij}, \quad \beta_{ij} = \beta'_{ij}$

g) Eğer (En İyi Uygun Çözüm Değeri-En İyi Sınır Değeri)<1 ise DUR: OPTİMUM ÇÖZÜM BULUNMUŞTUR.
Aksi takdirde;
Eğer Sayaç > 0 ise,

$$t_k = \frac{d_k\,(\text{En İyi Çözüm} - \text{Yeni Sınır Değeri})}{\sqrt{\sum_i \sum_j (f_{ij} - x^k_{ij}\, f_i)^2 + \sum_i \sum_j (f_{ij} - y^k_{ij} * F)^2)^2}}$$

denkleminin eşidi olan;

$$t_k = \frac{d_k\,(\text{En İyi Çözüm} - \text{Yeni Sınır Değeri})}{\sum_i \sum_j (f_{ij} - x^k_{ij}\, f_i)^2 + \sum_i \sum_j (f_{ij} - y^k_{ij} * F)^2}$$

formülasyonu kullanılarak t_k değeri hesaplanmaktadır.

Algoritmanın her aşamasında, α ve β değerleri bir önceki aşamada hesaplanan α ve β değerleri kullanılarak aşağıdaki formüle göre hesaplanmaktadır:

$$\alpha^{k+1}_{ij} = \alpha^k_{ij} + t_k\,(f_{ij} - x^k_{ij}\, f_i)$$
$$\beta^{k+1}_{ij} = \beta^k_{ij} + t_k\,(f_{ij} - y^k_{ij} * F)$$

h) İkinci aşamanın (a) şıkkına dönerek işlemler tekrarlanır.

Problemi çözerken, herhangi bir aşamada ardarda gerçekleştirilen 75 iterasyon sonunda daha iyi bir çözüm değeri elde edilememesi halinde algoritmanın sona erdirilmesi tavsiye edilmektedir[15].

(15) İbid., S. 700.

3.3 PROBLEMİN ÇÖZÜMÜNÜ ELDE ETMEDE SEZGİSEL YAKLAŞIM

Üstel karmaşıklık gösteren problemlerde, problemin optimum çözümü her zaman makul bir sürede elde edilememektedir. Bu nedenle, büyük ölçekli problemleri çözebilmek için uygun çözüm değerleri üreten Sezgisel Süreçlere (Heuristic Procedures) gerek duyulmaktadır. Kullanılan sezgisel yöntemin problemi çözmedeki başarısı, sezgisel yöntemle elde edilen uygun çözüm değeri ile sınır değeri (minimizasyon problemi için alt sınır, maksimizasyon problemi için üst sınır değeri) arasındaki farklı ölçülmektedir. Bu ölçü, kullanılan yöntemin performansını göstermekledir. Optimum çözüm, sezgisel ve lagrange yöntem çözümleri arasında yer alır(16).

Ele alınan problem NPC sınıfına girmektedir. Problemin ölçeği büyüdükçe çözüm zamanı üstel olarak artacaktır. Bu nedenle büyük ölçekli problemler için de uygun çözüm üretebilen çözüm algoritmalarına gerek duyulmaktadır. Probleme Lagrange yöntemi uygulanarak, çözümü güç olan kısıtlar dual değişkenler ile amaç fonksiyonuna eklenmektedir. Yeni problemi çözmek, problemin orjinal şekline göre göreceli olarak daha kolaydır. Bu problemin çözümü ile orjinal probleme bir sınır değeri getirilmektedir. Bu çözüm değeri genellikle orjinal problem için uygun değildir. Probleme bir uygun çözüm değeri bulabilmek için Aşamalı Optimizasyon Yöntemi kullanılmıştır. Aşamalı optimizasyon yönteminin her iterasyonunda, orjinal probleme bir uygun çözüm üretebilmek için aşağıda adımları belirtilen sezgisel yaklaşım uygulanmaktadır:

a) Aşamalı optimizasyon yönteminin başlangıcında elde edilen çözüm (Z_L çözümü), orjinal problem için (Z_p için) uygun değildir. Aşamalı optimizasyon yönteminin her iterasyonunda orjinal problem için bir uygun çözüm oluşturmaya çalışılmaktadır. Algoritma sona erdiğinde elde edilen çözüm, en iyi uygun çözüm değerini verecektir.

b) x_{ij}, y_{ij} ve f_{ij} problem Z_L'nin çözümleridir. eğer, bu çözümler Z_p problemi için de uygun çözüm oluşturuyorsa problemin optimum çözüm tespit edildiğinden algoritma sona erdirilir. Aksi takdirde bir sonraki aşama ile devam edilir,

c) Z_L probleminin sonuçlarından yola çıkılarak, Z_p için de uygun çözüm olabilecek bir çözüm araştırılır. Bunun için f_{ij} akışları, x_{ij} ve y_{ij} bağlantılarının açık olduğu düğümlere yöneltilir. Tüm akışlar açık olan

(16) PİRKUL, NARASİMHAN, DE, op. cit., S. 451.

hatları izlediği sürece elde edilecek çözüm Z_p için bir uygun çözüm oluşturacaktır. Bunun için, öncelikle ikincil terminallerin birincil terminallere gönderdiği akışlar x_{ij}'in açık olduğu hatlara yönlendirilir:

$$\text{Eğer, } \begin{cases} x_{ij} = 0 \text{ ise } f_{ij} = 0 \text{ alınır,} \\ x_{ij} = 1 \text{ ise } f_{ij} = \text{Akış miktarı} \end{cases}$$

olarak değiştirilir. (Akış miktarı, i–nci ikincil terminal ile j–nci birincil terminal arasındaki akış toplamıdır.) Daha sonra, birincil terminaller arasındaki akışlar da hatların açık olup olmadığına bakılarak aşağıdaki koşullara göre yeniden düzenlenmektedir:

$$\text{Eğer, } \begin{cases} y_{ij} = 0 \text{ ise } f_{ij} = 0 \text{ alınır,} \\ y_{ij} = 1 \text{ ise } f_{ij} = \text{Akış miktarı} \end{cases}$$

olarak değiştirilir. Böylelikle f_{ij} akışlarının bağlantısı kurulmuş olan terminaller arasında dolaşması sağlanmaktadır.

Sezgisel yöntemin adımları aşamalı optimizasyon yönteminin her iterasyonunda tekrarlanarak, daha önce Z_p için tespit edilen uygun çözümden daha iyi bir uygun çözüm bulunup bulunmadığı araştırılır. Algoritma sona erdiğinde en iyi uygun çözüm değerine ulaşılmış olacaktır. Sezgisel yöntemin yapısı veya tamsayı olma kuralının bir sonucu olarak, Aşamalı Optimizasyon Yöntemi sona erdiğinde en iyi uygun çözüm değeri (best feasible solution) ile en iyi sınır değeri (best bound) arasında bir fark bulunabilir. Bu fark 1'den küçük ise (en iyi uygun çözüm değeri-en iyi sınır değeri < 1 ise), amaç fonksiyonu katsayılarının tamsayılı değerlerden oluştuğu varsayıldığında bulunan sonucun optimum olduğu söylenebilir(17).

4) UYGULAMA VE BULGULAR:

Bu çalışmada, bir bilgisayar haberleşme şebekesinde terminaller arasındaki bağlantı ve akış maliyetleri toplamının minimize edilmesi amacıyla doğrusal programlama ile ifade edilen problem, Lagrange Yöntemi uygulanarak çözülmeye çalışılmıştır. Bu amaçla 100x100'lük bir alanda yer alan rasgele koordinat noktaları üretilerek, bu noktaların temsil ettiği terminaller arasındaki uzaklıklar Euclidean uzaklıklar olarak hesaplanmaktadır. Birincil ve ikincil terminaller arasındaki bağlantı maliyeti terminaller arasındaki uzaklığın fonksiyonu olarak hesaplanmıştır. Birin-

(17) J. CURRENT, H. PİRKUL, E. ROLLAND, op. cit., SS. 9-10.

cil terminaller arasındaki bağlantı maliyeti de (MMA bağlantı maliyeti) uzaklığın fonksiyonu olarak hesaplanmaktadır. Birim akış maliyeti, bir birim mesajın bir birim mesafe taşınmasından ortaya çıkmakta olup, bağlantı maliyetine göre göreceli olarak çok daha düşüktür. Akış maliyeti, uzaklığın belirli bir katsayıya oranı şeklinde hesaplanmaktadır. Birim mesaj akış maliyeti MMA üzerinde sözkonusudur. İkincil ve birincil terminaller arasındaki akış maliyeti çok düşük bir değer olarak gerçekleşeceğinden ihmal edilerek sıfır alınmaktadır. İkincil terminallerin bulunmadığı bir şebeke problemi için, bağlantı maliyetlerini sıfır alarak problemi birincil terminallerden ibaret bir şebeke halinde çözmek mümkündür.

Geliştirilen algoritmanın etkinliğini test etmek üzere değişik düğüm sayısına sahip şebekeler için, toplam akış miktarı ve maliyetler değiştirilmek suretiyle toplam 192 örnek problem çözülmüştür. Ölçeği büyüdükçe üstel karmaşıklık gösteren bir problemle karşı karşıya olunduğu dikkate alındığında, uygun çözüm değeri ile sınır değeri arasında kalan en yüksek farkın en fazla düğüme sahip şebeke probleminde elde edilmesi beklenecektir. Ancak, problem çözümlerinden elde edilen sonuçlar incelendiğinde, en fazla aralık veren problemin ölçeği en büyük olan problem olmadığı tespit edilmektedir. Aralık değeri, problemin ölçeğinden ziyade probleme ait veriler arasındaki ilişkiden etkilenmektedir. Bağlantı ve haber akış maliyetlerinin oransal olarak birbirine yakın olması halinde çözüm güçleşmektedir.

Aşamalı optimizasyon yöntemi uygulanarak gerçekleştirilen şebeke problemi çözümünde, elde edilen aralığın (gap) % 0.0000 ile % 10.8922 arasında değiştiği tespit edilmiştir. Bütün problem çözümleri dikkate alındığında, tespit edilen en yüksek aralığın düğüm sayısı en fazla olan şebekeye ait olmaması algoritmanın büyük ölçekli problemler için yeterince iyi çalıştığının göstergesidir.

Amaç fonksiyonu katsayıları tamsayılardan oluşan 192 örnek problem çözülerek kullanılan yöntemin etkinliğinin test edilmesi sonucu, çözülen problemlerin % 48.44'ünde (93 roblemde) optimum çözüme ulaşıldığı, geriye kalan problemlerde ise (99 problemde) elde edilen aralığın % 1 ile % 10 arasında değiştiği tespit edilmiştir. Problemlerin % 20.31'lik kısmında (39 problemde) % 1'den az aralık bulunduğu, % 6.25'lik kısmında (12 problemde) % 1 ile % 2 arasında aralık kaldığı, geriye kalan 48 problemde % 2'den büyük aralık olduğu tespit edilmiştir. 48 problemden 32'sinde mevcut aralığın % 2 ile % 5 arasında olduğu, 15 problem için % 5 ile % 9 arasında aralık kaldığı ve sadece tek bir problem için % 10'uk bir aralığın mevcut olduğu tespit edilmiştir.

Problemlerin tamamı gözönüne alındığında, çözülen problemlerin çoğunda (144 problemde) % 2'den az aralık kalmaktadır. Ayrıca, geriye kalan problemlerinde çoğunda (32 problemde) % 2 ile % 5 arasında değişen, çok yüksek sayılmayacak aralık değerleri elde edilmektedir. 15 problem % 5 ile % 9 arasında aralık vermekte, yalnızca tek bir problemde % 10 aralık kalmaktadır. Problem çözümlerine ait sonuçların tablolanmış şekli aşağıda verilmiştir:

Aralık Değerlerine Göre Elde Edilen Sonuçların Gruplanmış Tablosu

Tespit Edilen Aralık	İlgili Gruba Giren Problem Sayısı
% 0 (Optimum Çözüm)	93
% 0.0001 - 1'den az	39
% 1 - 2'den az	12
% 2 - 5'ten az	32
% 5 - 9'dan az	15
% 10	1

Yukarıdaki tablo, kümülatif değerler dikkate alınarak incelendiğinde, problemlerin % 68.75'lik kısmında % 1'den az aralık olduğu, % 75'lik kısmında % 2'den az aralık tespit edildiği, % 91.67'lik kısmında ise % 5'ten az aralık kaldığı gözlenmektedir. % 5'ten büyük aralık veren problemler ise tüm sonuçlar içinde düşük bir oranı (% 8.33) temsil etmektedir. Bu sonuçlar dikkate alındığında, geliştirilen algoritmanın optimum çözüme ulaşmada yeterince iyi çalıştığı yargısına varılabilmektedir. Ayrıca, optimum çözümün en iyi uygun çözüm değeri ile en iyi sınır değeri arasındaki (mevcut aralık içinde) bir noktada bulunduğu dikkate alınarak, optimum çözüm civarında, makul bur sürede elde edilen bu sonuçların kabul edilebilir olduğu yargısına varılmaktadır.

5) SONUÇ:

Bu çalışmada bir bilgisayar haberleşme şebekesindeki bağlantı ve akış maliyetlerinin minimize edilmesi problemi ele alınmıştır. Problem "çözümü zor" problemler sınıfına girmektedir. Bu nedenle Lagrange yöntemi uygulanarak probleme sınır değerleri üretilmektedir. Geliştirilen Sezgisel yaklaşım ile, daha önce üretilen sınır değerleri kullanılarak ilgili probleme uygun çözüm değerleri üretilmeye çalışılmaktadır. Geliştirilen algoritmanın test edilmesi amacıyla 192 örnek problemin çözülerek, uygun çözüm ve sınır değerleri arasında ortalama % 0 - 5 aralık (gap) kaldığı ve bu değerin şebeke büyüklüğünden bağımsız olduğu tespit edilmiştir. Problemin optimum çözümü, sınır ve uygun çözüm değerleri arasında yer almaktadır. Bu nedenle, algoritmanın sınırlı sayıda iterasyon sonunda sona erdirilmesi halinde optimum çözüm civarında (veya optimum) bir değere ulaşılması garanti altına alınmaktadır. Özellikle büyük ölçekli şebeke problemlerinde çözümün son derece karmaşık hale geldiği bilinmektedir. Bu nedenle, uygun bir hesaplama süresi içinde optimum çözüm civarında bir çözüm değerinin elde edilebilmesi, geliştirilen algoritmanın büyük ölçekli problemler için de etkin bir şekilde çalıştığını göstermektedir.

KAYNAKLAR

V. AHUJA, **Design and Analysis of Computer Communication Networks, Networks,** New York: McGraw-Hill, 1987.

J. CURRENT, H. PİRKUL, E. ROLLAND, "Efficient Algorithms for Solving The Shorest Covering Path Problems", **The Ohio State University Working Paper,** WPS 90-56, 1990.

R.D. DOWSING, F.W.D. WOODHAMS, **Computers form Locic to Architecture,** London: Van Nostrand Reinhold Co. Ltd., 1990.

M.L. FISHER, "The Lagrangian Relaxation Method For Solving Integer Programming Problems", **Management Science,** (U.S.A.: V. 27, No: 1, January 1981, SS. 1-18.

M. R. GAREY, D.S. JOHNSON, **Computers and Intractability A Guide to the Theory of NP-Completeness,** ABD: Bell Telephone Lab., Inc., 1979.

B. GAVISH, "On obtaining the Best Multipliers for a Lagrangian Relaxation for Integer Programming", **Comput. Oper. Res.,** V. 5, 1978, SS. 55-71.

B. GAVISH, H. PİRKUL, "Algorithms for The Multi-Resource Generalized Assignment Problem", **Management Science,** USA: V. 37, No. 6, June 1991, SS. 695-713.

F.S. HILLIER, G.J. LIEBERMAN, **Introduction to Operations Research,** New York: McGraw-Hill Publishing Co., 1990.

G.L. NEMHAUSER, L.A. WOLSEY, **Integer and Combiuatrial Optimization,** New York: John Wiley & Sons, Inc., 1988.

H. PİRKUL, S. NARASİMHAN, P. DE, "Locating Concentrators for Primary and Secondary Coverage in a Computer Communications Network", **IEEE Transactions,e** V. 36, No. 4, April 1988, SS. 450-458.

M.M. SYSLO, N. DEO, J.S. KOWALIK, **Discrete Optimization Algorithms with Pascal Programs,** New Jersey: Prentice-Hall, Inc., 1983.