

KALİTELİ İŞ PAYLAŞIMI PROBLEMİ İÇİN BULANIK MANTIK YAKLAŞIMI

Efendi N.NASİBOV*, **A. Övgü KINAY****

ÖZET

Bu çalışmada, işlerin işçiler arasında daha kaliteli paylaşılması problemi ele alınmıştır. Her bir işçinin her işi yapabilme yeteneği bulanık ilişki matrisi şeklinde verilmektedir. Optimallik kriterleri olarak, tüm atamaların bir araya getirilmiş (aggregation) yeterlilik derecesinin ve esas işçilerin toplam meşgulliyet derecesinin maksimleştirilmesi kriterleri dikkate alınmıştır. Problemin çözümü için klasik konteynerlere yerleştirme probleminin çözümünde kullanılan Best Fit Decreasing (BFD) algoritmasına benzer heuristik algoritmalar önerilmiştir.

Anahtar Kelimeler: Atama Problemi, Bulanık İlişki, Çok Kriterli Optimizasyon.

FUZZY LOGIC APPROACH TO THE PROBLEM OF TASK SHARING IN HIGH QUALITY

ABSTRACT

In this study, task sharing between workers in high quality is examined. Each worker's working ability of each tasks is taken as a fuzzy relationship matrix. Maximizing the competence value of aggregated assignments and total task loadings of the staff criteria are taken into consideration as optimizing criteria. Heuristic algorithms that are similar to the Best Fit Decreasing (BFD) algorithm which is used for the solution of the classical bin packing problem are suggested for the solution of the assignment problem.

Keywords: Assignment Problem, Fuzzy Relation, Multi-Criteria Optimization.

*Dokuz Eylül Üniversitesi Fen-Edebiyat Fakültesi İstatistik Bölümü,35160, Buca-İZMİR,
efendi_nasibov@yahoo.com

**Dokuz Eylül Üniversitesi Fen-Edebiyat Fakültesi İstatistik Bölümü,35160, Buca-İZMİR,
ovgu.tekin@deu.edu.tr

I. GİRİŞ

Varsayalım ki, n sayıda yapılacak p_1, p_2, \dots, p_n işleri ve bu işleri yapabilecek m sayıda s_1, s_2, \dots, s_m esas işçileri olsun. Ayrıca ek olarak bir tane s_{m+1} olarak işaretlenmiş kiralanan işçiler grubu olsun. Kiralanan işçiler bir grup halinde olmuş daha fazla sayıda işçi de olabilir. Her bir p_j işini yapmak için $a_j > 0$ zamanı gerekmektedir (bu zaman birimi günler, saatler vb. olabilir). İşler ve esas işçiler arasında ise $C = \|c_{ij}\|, i = \overline{1, m}; j = \overline{1, n}$ şeklinde gösterebileceğimiz bulanık ilişkiler mevcuttur. Bu matrisin $c_{ij} \in [0, 1]$ elemanı i . işçinin j . işi yapabilme yeteneğini gösterir. $c_{ij} = 0$ olduğunda i . işçinin j . işi yapma yeteneği yoktur, $c_{ij} = 1$ olduğunda ise işçi, bu işi yapma konusunda uzmandır. $0 < c_{ij} < 1$ olması durumu ise aradaki durumları belirtmektedir. Belirli koşulları sağlayarak, işleri, işçilere optimum şekilde atamak gerekmektedir. Her bir işçinin yapabileceği toplam iş miktarı vb. lineer koşullar olabilir. Aynı zamanda atanmış işleri, işçilerin uzmanlıkla yapabilmeleri gerekmektedir. Yani, atamaların kalitelerinin bir araya getirilmesiyle elde edilen sonuç atanma kalitesinin maksimalleştirilmesi gerekmektedir. Ayrıca ek maliyet gerektirdiği için kiralanan işçilere atanmış toplam iş miktarının minimmalleştirilmesi gerekmektedir.

İşlerin işçilere atanmasını $X = \|x_{ij}\|, i = \overline{1, m+1}; j = \overline{1, n}$ matrisiyle gösterelim. Başka deyişle,

$$x_{ij} = \begin{cases} 1, & \text{eger } j. \text{ is } i. \text{ isciye atanmissa} \\ 0, & \text{aksi halde} \end{cases} \quad (1.1)$$

Atama sırasında sağlanması gereken kesin kısıtlar aşağıdaki gibidir:

a. Her bir esas işçinin toplam iş yüklemesi işçinin kapasitesini aşmamalıdır.

$$\sum_{j=1}^n a_j x_{ij} \leq b_i, \quad i = \overline{1, m} \quad (1.2)$$

Burada $b_i > 0$ her bir i . esas işçinin yapabileceği toplam iş miktarını gösterir;

b. Her bir iş, sadece bir işçiye atanabilir.

$$\sum_{i=1}^{m+1} x_{ij} = 1, \quad j = \overline{1, n} \quad (1.3)$$

Optimalleştirilmesi gereken kriterler olarak, tüm atamaların bir araya getirilmiş sonuç geçerlilik değeri $\gamma(x)$ 'in ve esas işçilere atanmış işlerin toplam miktarı derecesi $\xi(x)$ 'in maksimalleştirilmeleri istenmektedir, başka deyişle

$$\gamma(x) = \min_{x_{ij}=1} c_{ij} x_{ij} \rightarrow \max \quad (1.4)$$

$$\xi(x) = \frac{\sum_{i=1}^m \sum_{j=1}^n a_j x_{ij}}{\sum_{j=1}^n a_j} = 1 - \frac{\sum_{j=1}^n a_j x_{m+1j}}{\sum_{j=1}^n a_j} \rightarrow \max \quad (1.5)$$

Böylece, (1.1) - (1.3) koşullarını sağlayarak (1.4) ve (1.5) kriterlerini maksimalleştirmek gerekmektedir. Ayrıca $\gamma(x)$ değeri olarak farklı bir araya getirme yöntemleri de kullanılabilir (Calvo ve Mesiar, 2001; Nasibov ve Nasibova, 2003; Yager, 2003).

Bu problem, çok kriterli optimizasyon problemidir. Problemin çözümü için farklı yöntemler kullanılabilir. Bu çalışmada ayrı ayrı kriterler üzerine iki aşamalı optimizasyon tekniği uygulanmıştır; başka deyişle aşağıdaki çözüm şeması uygulanmaktadır.

$$\text{I. Aşama:} \quad \xi(x) \rightarrow \max \quad (1.6)$$

$$(1.1)-(1.3) \text{ koşulları sağlanarak} \quad (1.7)$$

$$\gamma(x) \geq \gamma_0 \quad (1.8)$$

$$\text{II. Aşama:} \quad \gamma(x) \rightarrow \max \quad (1.9)$$

$$\xi(x) = \xi_{\max} \quad (1.10)$$

$$(1.1)-(1.3) \text{ koşulları sağlanarak} \quad (1.11)$$

Burada γ_0 değeri, minimum kalite derecesine getirilmiş alt sınır ve ξ_{\max} değeri, I. aşamanın sonunda $\xi(x)$ fonksiyonunun aldığı maksimal değerdir. Açık ki, II. aşama sonucunda bulunan çözüm, (1.1) - (1.5) probleminin pareto optimal çözümü olacaktır.

2. ÇÖZÜM ALGORİTMALARI

Nasibov vd. (2004/b) çalışmasında (1.6) - (1.8) problemi $\gamma_0 = 0$ halinde ele alınmış ve işçiler maksimum kapasitelerine göre önceden azalan sırada sıralanarak çözüm algoritması kullanılmıştır. Bu algoritma aşağıdaki gibidir (Algoritma 1.a).

Algoritma 1a (I. Aşama)

Adım 0. Başlangıç değerler:

$x_{ij} = 0$, $i = 1, m$; $j = 1, n$ esas işçilerin yeterlilik dereceleri c_{ij} , $i = 1, m$; $j = 1, n$ belirlenir; işlerin yapılma süreleri a_j , $j = 1, n$ belirlenir; işçilerin maksimum toplam yüklenme miktarları b_i , $i = 1, m$ belirlenir; esas işçilerin boş kalan iş miktarları $s_i := b_i$, $i = 1, m$;

DegMin:=1, tüm atamanın sonuç geçerlilik derecesine başlangıç değeri verilir;

Adım 1. İşler, iş miktarlarının büyüklüğüne göre azalan sırada sıralanır.

Adım 2. Adım 2.Esas işçiler maksimum yüklenme kapasitelerine göre azalan sırada sıralanırlar.

Adım 3. for $j:=1$ to n do begin /* her bir iş sırayla ele alınır*/

Adım 4. for $i:=1$ to m do begin /* her bir işçi sırayla ele alınır*/

Adım 5. if $a_j \leq s_i$ and $c_{ij} \geq \gamma_0$ then begin

$x_{ij} := 1$; $s_i := s_i - a_j$; $DegMin := \min\{DegMin, c_{ij}\}$;

Bir sonraki iş için Adım 3 tekrarlanır.

end /* if */

end /* for i */

end /* for j */

Adım 6. $x_{ij}, i = 1, m+1; j = 1, n$, I. Aşama'nın sonunda işlerin atanma haritasını ve DegMin tüm atamanın sonuç geçerlilik derecesini yansıtmış olur.

Stop.

Adım 2.

Şimdiki çalışmada önerilen algoritmanın Nasibov vd.'nin (2004/b) çalışmasındaki algoritmadan farkı Adım 2 ve Adım 3'den kaynaklanmaktadır. Başka deyişle esas işçiler maksimum kapasitelerine göre önceden sıralanmamışlardır. Bunun yerine her bir iş ele alındığında onu yapabilecek işçiler yeteneklerine göre dinamik sıralanırlar (Algoritma 1.b).

Algoritma 1b (I. Aşama)

Adım 0. Başlangıç değerleri:

$x_{ij} = 0, i = 1, m; j = 1, n$ esas işçilerin yeterlilik dereceleri $c_{ij}, i = 1, m; j = 1, n$ belirlenir;

işlerin yapılma süreleri $a_j, j = 1, n$ belirlenir; işçilerin maksimum toplam yüklenme miktarları $b_i, i = 1, m$ belirlenir; esas işçilerin boş kalan iş miktarları $s_i := b_i, i = 1, m$;

DegMin:=1, tüm atamanın sonuç geçerlilik derecesine başlangıç değeri verilir;

Adım 1. İşler, iş miktarlarının büyüklüğüne göre azalan sırada sıralanır.

Adım 2. for $j:=1$ to n do begin /* her bir iş sırayla ele alınır*/

Adım 3. İşçiler j . işi yapabilme yeteneklerine göre azalan sırada sıralanırlar.

Adım 4. for $i:=1$ to m do begin /* her bir işçi sırayla ele alınır*/

Adım 5. if $a_j \leq s_i$ and $c_{ij} \geq \gamma_0$ then begin

$x_{ij} := 1$; $s_i := s_i - a_j$; $DegMin := \min\{DegMin, c_{ij}\}$;

Bir sonraki iş için Adım 3 tekrarlanır.

end /* if */
 end /* for i */
 end /* for j */

Adım 6. $x_{ij}, i = \overline{1, m+1}; j = \overline{1, n}$, I. Aşama'nın sonunda işlerin atanma haritasını ve DegMin tüm atamanın sonuç geçerlilik derecesini yansıtmış olur.
Stop.

Algoritma 2 (II. Aşama)

I. Aşama sonuçlarının daha fazla iyileştirilmesi için bu çalışmada da Nasibov vd.'nin (2004/b) işine benzer olarak II. aşama kullanılır. Bu süreçte, işler “az yetenekli” işçilerden alınarak “daha yetenekli” işçilere atanır (Tablo 1).

Tablo 1. II. Aşama'da İşlerin Yerlerinin Değiştirilmesi Tablosu

İş İşçi	1	2	...	j	...	j_0	...	n	Boş iş miktarı	Maksimum iş miktarı
1	c_{11}	c_{12}	...	c_{1j}	...	c_{1j_0}	...	c_{1n}	s_1	b_1
2	c_{21}	c_{22}	...	c_{2j}	...	c_{2j_0}	...	c_{2n}	s_2	b_2
⋮	⋮	⋮		⋮		⋮		⋮	⋮	⋮
i_0	c_{i_01}	c_{i_02}	...	c_{i_0j}	←	$c_{i_0j_0}$...	c_{i_0n}	s_{i_0}	b_{i_0}
⋮	⋮	⋮		⋮		⋮		⋮	⋮	⋮
i	c_{i1}	c_{i2}	...	c_{ij}	→	c_{ij_0}	...	c_{in}	s_i	b_i
⋮	⋮	⋮		⋮		⋮		⋮	⋮	⋮
$m-1$	c_{m-11}	c_{m-12}	...	c_{m-1j}	...	c_{m-1j_0}	...	c_{m-1n}	s_{m-1}	b_{m-1}
m	c_{m1}	c_{m2}	...	c_{mj}	...	c_{mj_0}	...	c_{mn}	s_m	b_m
İş miktarı	a_1	a_2	...	a_j	...	a_{j_0}	...	a_n	DegMin	

Adım 0. Giriş değerler:

I. Aşamamın sonunda elde edilmiş atanma haritası;
 esas işçilerin yeterlilik dereceleri $c_{ij}, i = \overline{1, m}; j = \overline{1, n}$; işlerin miktarları $a_j, j = \overline{1, n}$;
 işçilerin maksimal yüklenme miktarları $b_i, i = \overline{1, m}$; esas işçilerin önceki algoritma sonucu boş kalan iş miktarları $s_i, i = \overline{1, m}$.

Adım 1. while (REPEAT) do begin

Adım 2. Esas işçilere verilmiş işler içerisinde en “az yetenekli” işçi bulunur; İşaretlenir:

DegMin:= $c_{i_0j_0} \equiv \min\{c_{ij} \mid \forall i = \overline{1,m}; \forall j = \overline{1,n} : x_{ij} = 1\}$;

REPEAT:=**false**;

Adım 3. for $i:=1$ **to** m **do begin**

Adım 4. if $i = i_0$ **then** i için bir sonraki iterasyona geç;

Adım 5. for $j:=1$ **to** n **do begin**

Adım 6. if $x_{ij} \neq 1$ **then** j için bir sonraki iterasyona geç;

Adım 7. if $a_j + s_i \geq a_{j_0}$ and $a_{j_0} + s_{i_0} \geq a_j$ and $\min\{c_{i_0j}, c_{ij_0}\} > c_{i_0j_0}$ **then begin**

j_0 işi i . işçiye, j işi i_0 işçisine verilir, yani

$x_{ij_0} := 1, x_{i_0j} := 1, x_{i_0j_0} := 0, x_{ij} := 0$;

REPEAT:=**true**;

Adım 1 baştan tekrarlanır;

end /* if */

end /* for j */

end /* for i */

end /* while */

Adım 8. $x_{ij}, i = \overline{1,m+1}; j = \overline{1,n}$, işlerin sonuç atanma haritasını ve DegMin tüm atamanın sonuç geçerlilik derecesini yansıtmış olur.

Stop.

3. HESAPLAMA DENEYİMLERİ

Algoritma 1b ve Algoritma 2, C++ algoritmik dilinde programlanarak, INTEL PENTIUM IV, 1.70 GHz, 256 MB RAM işlemcisi olan bir bilgisayarda denemeler yapılmıştır. Denemelerde, $\|c_{ij}\|$ matrisinin elemanları $[0,1]$ aralığında, $[0.0, 0.1, 0.2, \dots, 1.0]$ değerlerini alabilen tesadüfi sayı üreticisiyle oluşturulmuşlardır. İşlerin hacimleri ise $[1,100]$ aralığında tam değerler alabilen tesadüfi sayı üreticisiyle oluşturulmuşlardır.

İşçilerin maksimum yüklenme miktarları iki şekilde belirlenmektedir: benzer maksimum miktarlar ve benzer olmayan maksimum miktarlar. Benzer maksimum miktarlarda $m-1$ sayıda b_i değeri

$$b_i \in \left[\frac{4}{10} \sum_{j=1}^n \frac{a_j}{m}, \frac{6}{10} \sum_{j=1}^n \frac{a_j}{m} \right], i = \overline{1, m-1} \quad (3.1)$$

aralığında tesadüfi olarak belirlenmiştir. Benzer olmayan durum için maksimum yüklenme miktarları ise

$$b_i \in \left[0, \frac{1}{2} \sum_{j=1}^n a_j - \sum_{k=1}^{i-1} b_k \right], i = \overline{1, m-1} \quad (3.2)$$

aralığında tesadüfi olarak belirlenmiştir. Sonuncu b_m maksimum yüklenme miktarı ise her iki durumda da

$$b_m = \frac{1}{2} \sum_{j=1}^n a_j - \sum_{i=1}^{m-1} b_i \quad (3.3)$$

olarak belirlenmiştir.

Deneyler sonucu Tablo 2 ve Tablo 3'de verilmektedir. Deneylerin istatistiksel açıdan daha anlamlı olması için tabloların her satırında aynı verileri içeren 200 deney için iş ve işçi sayıları, işçi/iş oranları ve Algoritma 1.a ile Algoritma 1.b'ye bağlı olarak II. Aşama sonucu iyileştirme gözlenmiş deneylerin sayısı verilmiştir.

Tablo 2. II. Aşama Sonucu Benzer Kapasiteler İçin Elde Edilen Sonuçlar.

İşlerin sayısı	İşçilerin sayısı	İşçi/İş yüzdesi	II. Aşama sonucu iyileşmelerin sayısı	
			Algoritma 1a	Algoritma 1b
50	5	%10	32	19
50	10	%20	53	28
50	25	%50	90	24
50	30	%60	92	12
50	40	%80	91	8
200	20	%10	8	9
200	40	%20	19	15
200	100	%50	81	32
200	120	%60	113	10
200	160	%80	100	1
400	40	%10	4	8
400	100	%25	44	67
600	60	%10	12	12
600	120	%20	22	18

Tablo 3. II. Aşama Sonucu Farklı Kapasiteler İçin Elde Edilen Sonuçlar.

İşlerin sayısı	İşçilerin sayısı	İşçi/İş yüzdesi	II. Aşama sonucu iyileşmelerin sayısı	
			Algoritma 1a	Algoritma 1b
10	2	%20	15	9
10	4	%40	20	11
10	6	%60	19	1
20	2	%10	12	7
20	8	%40	44	22
30	6	%20	26	14
30	12	%40	31	20
50	5	%10	20	7
50	10	%20	21	11
100	10	%10	6	8
100	15	%15	8	11

Benzer kapasiteler için Tablo 2 incelendiğinde II. Aşama'nın her zaman belirli bir miktarda iyileştirme yaptığı görülmektedir. İş büyüklüğü 200, 400 ve 600 için işçi/iş yüzdesi %10 olduğu durumda, hangi Algoritma uygulanırsa uygulansın II. Aşama'nın yaklaşık aynı sayıda iyileştirme yaptığını, başka deyişle Algoritma 1b'nin beklenen faydayı sağlamadığı görülmektedir. Ancak işçi/iş yüzdesi arttıkça başka deyişle işçi başına düşen iş sayısı azaldıkça Algoritma 1b'nin daha etkin bir şekilde çalıştığını II. Aşama iyileştirmesinin giderek daha az sayıda olmasından görüyoruz. Ayrıca Algoritma 1b'nin Algoritma 1a'ya göre daha az iyileştirmeye gereksinim duyduğunu istatistiksel olarak inceledik. Sonuç olarak her ne kadar normallik koşulları sağlansa bile veri sayısının yetersiz olması sebebiyle Wilcoxon işaretli sıra sayıları testi uygulandığında 0.0071 olasılıkla bu sonucun gerçekten anlamlı olduğunu görüyoruz.

Benzer olmayan kapasiteler için benzer kapasiteler denemeleri ile aynı büyüklüklerde bir çalışma yapılamamıştır. Bunun sebebi ise (3.2) ifadesidir. Bu çalışmada tesadüfi sayı üreticisi olarak (3.2) ifadesinin kullanılmış olması benzer olmayan kapasiteler sınamasını kısıtlamıştır. İş büyüklükleri arttıkça veri türetilmesi zorlaşmış ve bir noktadan sonra mümkün olamaz hale gelmiştir. Bu yüzden daha küçük iş sayısı ve işçi/iş yüzdeleriyle ilgili deney yapılabilmektedir. Tablo 3'den de görüldüğü gibi benzer olmayan kapasiteler için de Tablo 2'nin sonuçlarına yakın sonuçlar elde edilmiştir. Bu tabloda da iş sayısı 100 olduğunda ve işçi/iş yüzdesi %10 ile %15 için Algoritma 1a'nın Algoritma 1b'ye göre daha etkin olduğunu görüyoruz. Başka deyişle iş sayısının fazla olduğu durumlarda işçi/iş yüzdesi küçük bir değer ise Algoritma 1a daha etkindir. Ancak diğer denemelerde Algoritma 1b'nin kullanılması atama işleminin II. Aşama sonucunda daha az iyileştirme sağladığını gösteriyor. Benzer olmayan kapasiteler için de aynı şekilde parametrik olmayan

istatistiksel test yöntemlerinden Wilcoxon işaretli sıra sayıları testi uygulandığında 0.0038 olasılıkla Algoritma 1b'nin anlamlı düzeyde az iyileştirme yaptığını görüyoruz.

Sonuç olarak tablolardan ve Wilcoxon test istatistiklerinden de gözüktüğü gibi II. Aşama sonucu belirli bir oranda iyileşme sağlamıştır. Bu durum özellikle Algoritma 1a sonrası çalışmalarda daha fazladır. Bu çalışmada sunulan Algoritma 1b sonrası ise II. Aşama'da daha az iyileşmeler gözlenmektedir. Bunun da sebebi Algoritma 1b'nin Algoritma 1a'ya göre daha etkin çalışmasıdır.

4. SONUÇ

Bu çalışmada bulanık yetenek dereceleri göz önünde bulundurularak işlerin işçilere daha kaliteli paylaşılması problemi ele alındı. Önceki çalışmalardan farklı olarak bu çalışmada önerilen Algoritma 1b, her bir işin paylaşımında işçilerin yeteneklerini dinamik olarak dikkate almaktadır. Bunun sonucu iş paylaşımı kalitesi belirli derecede artış göstermektedir. Yapılan deneylerde de II. Aşama'da Algoritma 2'nin genellikle sonuçları iyileştirebilmesinin yanı sıra Algoritma 1b sonrası daha az iyileştirmeler gözlenmiştir. Bu da Algoritma 1b'nin çoğu zaman iyileştirilemeyecek daha iyi sonuçlar verdiğini göstermektedir.

Önerilen algoritmanın zaman ve bellek karmaşıklığı bilinen FFD ve BFD algoritmalarına benzerlik göstermektedir (Kim vd, 2001; Martello vd, 1990; Nasibov, 2004/a). Fakat bu konunun daha ayrıntılı araştırılması iyi olurdu. Daha sonraki çalışmalarda, bu tip araştırmalar da yapılabilir.

5. KAYNAKÇA

Calvo T., Mesiar R., (2001), "Generalized Medians", Fuzzy Sets and Systems, 124, 59-64.

Kim J.-K., Lee-Kwang H., Yoo S.W., (2001), "Fuzzy Bin Packing Problem", Fuzzy Sets and Systems, 120, 429-434.

Martello S., Toth P., (1990), "Knapsack Problems: Algorithms and Computer Implementations", J.Wiley.

Nasibov E.N., (2004/A), "An Algorithm for Constructing an Admissible Solution to the Bin Packing Problem with Fuzzy Constraints", Journ. of Comp. and Syst. Sci. Int., 43, 2, 205-212.

Nasibov E.N., Nasibova R.A., (2003), "OWA and MIN Aggregation Methods in Fuzzy Bin-Packing Problem", Transac. of the National Academy of Sciences of Azerbaijan, Phus.-Tech. and Math. Series, 2, 45-50.

Nasibov E.N., Senol S., Nasibova R.A., (2004/B), "An Optimal Task-Assignment Problem with A Fuzzy Competence Matrix", *Automatic Control and Computer Science*, 37, 6, 28-40.

Yager R.R., (2003), "Induced Aggregation Operators", *Fuzzy Sets and Systems*, 137, 59-69.