



# SEZGİSEL YÖNTEMLERE DAYALI ÇOKLU-DİLME BIÇKI MAKİNESİ MİL TASARIMI VE YAZILIMI

Vural AKSAKALLI  
İbrahim ARI

## ÖZET

Mobilya üretim endüstrisinde karşılaşılan tipik bir kereste işleme yöntemi, kerestenin değişik genişliklerde boylamasına kesilmesidir. Bu işlemi gerçekleştirmek için üzerinde birden fazla kesme kanalı bulunan bir bıçkı makinesi geliştirilmiştir. Bu bıçkı, dönen bir mil üzerinde belirli aralıklarla yerleştirilmiş bıçaklardan oluşur. Bu yöntem “çoklu-dilme kereste kesimi” adı verilir. Bu işlemdeki temel problem --ki biz bunu “çoklu-dilme makinesi mil tasarım problemi” olarak adlandırıyoruz-- müşterinin talep ettiği genişliklerdeki dilimlerden, yine müşterinin istediği miktarda üretilirken ortaya çıkan kenar dilim artıklarını ve talaşı en aza indirecek bıçak yerleşim düzenini bulmaktır. Çoklu-dilme mil tasarım problemi zor bir kombinatoriyal eniyileme problemidir. Bu çalışmada, bu problemin sayısal karmaşıklığı incelenecek ve yaklaşık çözümü için yerel iyileştirme esasına dayalı bir sezgisel yöntem sunulacaktır. Daha sonra çoklu-dilme mil tasarım probleminin yaklaşık çözümü için geliştirilmiş, eniyileme motoru olarak bu sezgisel yöntemi kullanan, görsel bir kullanıcı arayüzü olan ve kullanımı kolay bir yazılım tanıtılacaktır.

**Anahtar Kelimeler:** Mil tasarımı, sezgisel yöntem, yerel iyileştirme prosedürü.

## ABSTRACT

A typical lumber processing method within the furniture manufacturing industry is the cutting of lumber along the length into strips of different widths. A saw with multiple cutting channels has been developed to carry out this task. This saw has fixed blades at specific positions on a rotating shaft which rip incoming lumber boards lengthwise into desired finished widths. This method is commonly referred to as the “gang-rip method” of lumber cutting, and the setting of the blades (i.e., the pattern of cutting channels) along the saw shaft is called an “arbor”. The fundamental challenge in this process, called the “gang-rip saw arbor design problem”, is to find the optimal arbor that minimizes edge strips and saw dust while meeting demand for a set of desired finished widths and their corresponding quantities. The gang-rip saw arbor design problem is a difficult combinatorial optimization problem. In this study, we investigate the problem’s computational complexity and propose a local improvement based heuristic method. We also present an easy-to-use software with a graphical user interface that uses this heuristic method as the optimization engine.

**Keywords:** Saw arbor design, heuristic method, local improvement procedure.

## 1. GİRİŞ

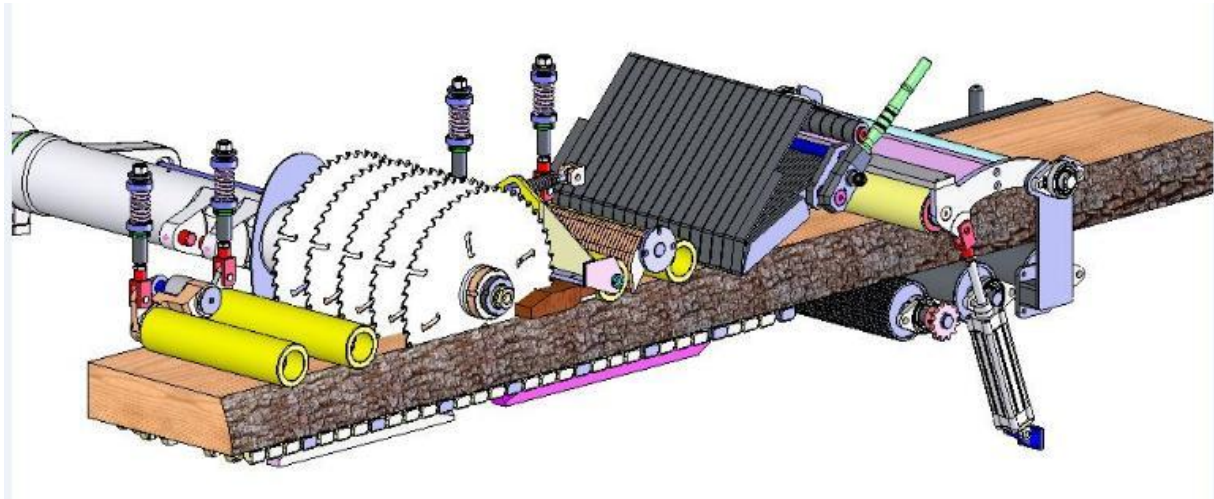
Mobilya üretim endüstrisinde tipik bir kereste işleme yöntemi, kerestenin boyunca (gövde eksenine paralel) farklı genişliklerde dilimler halinde kesilmesidir. Sonra da bu dilimler gerekli uzunluklarda kesilerek üretimde kullanılır.



Bu işlemi gerçekleştirmek için çoklu kesme kanallı ticari bir bıçkı makinesi geliştirilmiştir (bakınız Şekil 1 ve 2). Bu bıçkı makinesi, dönen bir şaftın üzerinde belirli bir konumda sabitlenmiş bıçakları vardır ve gelen kereste tahtalarını istenilen ürün genişliğinde uzunlamasına biçer. Bu yöntem genellikle kerestelerin “çoklu-dilme yöntemi” ile kesilmesi olarak adlandırılır ve bıçkı şaftının boyunca bıçakların ayarı (yani, kesme kanallarının deseni) “mil” olarak adlandırılır.



Şekil 1. Ticari Bir Çoklu Dilme Makinesi (Kaynak: www.fordaq.com)



Şekil 2. Ticari Birçoklu Dilme Makinesinin Genel Bir Çalışma Şeması (Kaynak: www.fordaq.com)

Kesim işlemi süresince, kesilmemiş tahtalar gelen öbekten rastgele bir şekilde çekilip bıçkı makinesi beslenir; genişliğe göre bir ön sıralama gerçekleştirilmez. Gelen tahtaların genişliğinin olasılık dağılımı, örnek bir tablodan türetilmiş ampirik frekans dağılımına dayanarak bulunur.



Bir kesim işi sürecinde üretilen “zayıat” miktarı, problemde anahtar meseledir. Zayıat, (1) talaş kaybı ve (2) gelen tahtaların talep planındaki hiçbir ürün genişliğine uygun olmamasından dolayı kenarlarının dilimler halinde kesilmesinin (kenar dilimi) sonucudur. Buradaki kesim işinde kullanılan bıçkı makinesi ekstra yeteneklere sahiptir. Bu yetenekler, gelen kereste tahtalarının kullanılabilir genişliklerinin kaydedilmesi ve tahtaların mil üzerindeki birbirine komşu kesme kanalları bloğundan, gelen tahta için en az zayıat üretene doğru (şafta paralel) taşınmasıdır.

Zayıatı en aza indirmenin ötesinde, çoklu dilme makinesinin çalışmasıyla ilgili iki mesele vardır. Birinci mesele “fazlalık”, yani talep planındaki ürün genişliklerinden herhangi birinden fazla üretimdir. Her bir ürün genişliği için üretilen miktarın, karşılık gelen talebi önemli ölçüde aşmaması arzulanır (fakat gerekli değildir). İkinci mesele de bir kesim işini yapmak için kullanılacak mil sayısıdır. Doğal olarak, çok sayıda mil kullanmak daha fazla esnekliğe izin verir ve muhtemelen daha iyi bir verim elde edilmesine yol açar. Fakat mil değiştirmek için 15 dakika bıçkıyı durdurmamız gerekmektedir. Bundan dolayı, belirli bir kesme işinde böyle bir üretim kaybından sakınmak için genellikle en az mil sayısını (muhtemelen bir) kullanmak tercih edilir.

Kereste miktarını ifade etmek için çoğunlukla iki ölçüm birimi kullanılır. Bunlardan biri olan lineer foot sadece uzunlukla ilgilidir; on ayak uzunluğundaki bir tahta, genişlik ve kalınlığına bakılmaksızın on lineer foot olarak ölçülür. Diğer taraftan board foot birimi, kerestenin hacminin bir ölçüsü olup 3 boyutu da eş zamanlı olarak içerir. Sunumun kolaylığı için bu makalenin tamamında kullandığımız board foot birimi kalınlığını dikkate almaksızın kerestenin bir lineer foot karesine karşılık gelmektedir. Çünkü kesme işindeki tüm kesim stokunun aynı kalınlıkta olduğu varsayılmıştır. Terminolojideki bu değişiklik bizim sonuçlarımızın geçerliliğini etkilemez.

Yukarıda açıklanan durumu da göz önünde bulundurarak “Çoklu-Dilme Makinesi Mil Tasarımı ve Planlama Problemi” ni şu şekilde tanımlıyoruz:

ÇDMT: Verilenler, (1) istenen ürün genişliklerinin (kesilen-genişlikler) bir kümesi ve bunlara karşılık gelen miktarları (talep planı), (2) gelen tahtaların (hammadde) dağılım frekansı, (3) shaft uzunluğu ve (4) bıçak genişliğidir. İstenen ise, millerin (bir veya daha fazla) bir kümesi (yani, bıçak ayarları) ve buna karşılık gelen her bir milden geçecek kerestenin miktarı, öyle ki zayıatın toplam board foot miktarı en aza indirilerek talep karşılsın.

Bu tanımda, millerin sayısı belirtilmemiştir ve problem çözücü uygun bulunduğu çoklukta mil seçmekte serbesttir. Kesme ve paketleme üzerine olan literatür bağlamında, bu problem bir boyutlu stok kesim problemi (veya [3]’de önerilen gösterime göre SK1) olarak kategorize edilebilir. Bunun uygulamalarının geniş bir alanı olmasından dolayı SK1, kesme ve paketleme üzerine olan açık literatürde en fazla referans gösterilen problemlerden biridir.

Fakat burada üzerinde çalıştığımız problem, mil tasarımının ilişkilendirilmesiyle diğerlerinden ayrılır. Milin üzerindeki birbirine komşu kesme kanalları bloğunda mevcut olan genişlik için, her bir gelen tahta genişliği en iyi kesme desenine göre kesilir. Böylece, bu problemdeki başlıca en zor kısım, bir veya daha fazla milin bir kümesini ve bunların her birine karşılık gelen koşma sürelerini tasarlamak, böylece her bir gelen tahta toplam zayıatı en aza indirecek şekilde ve talebi karşılama bağlamında verimli bir tarzda kesilmesidir. Bu mil tasarım probleminin kombinasyonel doğasıdır (yani, bıçkı shaftının boyu boyunca kesme kanallarının ve karşılık gelen kesilen-genişlik ağırlıklarının düzenlenmesi) ki bununla diğer SK1 problemlerinden ayrılır.

Daha önceki makalelerden [3]’de, ÇDMT’yi çözmek için bir doğrusal programlama (DP) modelini önerdik ve bu modeli çözmek için sütun üretme kavramını kullandık. Bu yaklaşımı kullanarak, makul bir CPU zamanında iyi sonuçlar (az zayıat ve az fazlalık) elde ettik. Fakat, birçok durumda optimal çözümde kullanılan millerin toplam sayısı diğerlerine nazaran çok fazla (çoğunlukla gerekli ürün genişlik miktarlarının sayısı kadar çok mil). Bu, yukarıda bahsedildiği gibi üretimde önemli bir zaman kaybıdır. Açıkça biz her zaman az zayıatı gerçekleştiren ve nispeten az sayıda mil kullanan (ideal olarak bir mil) çözümü tercih ederiz. Fakat DP modeli bağlamında mil sayısını sınırlamak [3], az çok problematiktir. Bu sayı, DP modelindeki bazların büyüklüğü ile ilişkilidir ve bu sayıyı sınırlamak, nispeten çok fazla sayıda ikili değişkenlerle tanışmamızı gerektirir. Bunun sonucu olan karışık



tamsayı doğrusal programlama modelin çözülmesi, önemli ölçüde zordur. Bu makalede biz, millerin sayısının sınırlı olması durumunda bu problemi çözmek için farklı bir yaklaşım öneriyoruz.

Daha belirgin olarak, biz “Bir Milli Çoklu Dilme Bıçkısı Mil Tasarımı ve Planlama Problemi” (ÇDMT/1) tanımladık, ÇDMT’yi 1 mil kullanarak çözdük. ÇDMT/1’i çözmek için birkaç prosedür bulduk ve bunların efektifliğini araştırmak için hesaplamasal deneyler gerçekleştirdik. Bölüm 2’de, problemin kalıbını tanımlamak için gerekli olan çeşitli gösterimler ve terminolojiyi tanımladık. Bölüm 3, ÇDMT/1’in karmaşıklığı üzerinedir. Sonraki üç bölüm, bu problem için geliştirdiğimiz üç sezgisel prosedürün açıklamasına tahsis edilmiştir. Son olarak, Bölüm 7’de, bu prosedürlerle yapılan hesaplamasal deneylerin sonuçlarının bir özetini sunduk ve sonuç olarak bazı çıkarımlar yaptık.

## 2. GÖSTERİM VE TERMİNOLOJİ

Bu bölümde, gösterimlerle tanışacağız ve verilen bir mili değerlendirmek için kullandığımız prosedürleri açıklayacağız.

### 2.1. Problemin Bir Kalıbı

Aşağıdaki gösterimler problemimizin kalıbını gelen (kesilmemiş) kerestenin karakteristiği, talep planı ve bıçkı karakteristikleri cinsinden belirtir.

M	Talep planındaki farklı ürün genişliklerinin (kesilmiş-genişlikler) sayısı
$w_i$	i nci farklı ürün genişliği, $i=1, \dots, M$
$r_i$	$w_i$ ürün genişliği için gerekli board feet’i (yani, lineer feet kare), $i=1, \dots, M$
K	Kesilmemiş stoktaki ayrık gelen tahta genişliklerinin sayısı
$b_k$	k ıncı ayrık gelen tahta genişliği, $k=1, \dots, K$
$l_k$	Gelen kerestenin rastgele bir örnekleminde $b_k$ genişlikli tahtanın lineer feet miktarı, $k=1, \dots, K$
l	Rastgele bir örneklemdaki tüm gelen tahtaların toplam uzunluğu (lineer foot). Bu,

$$l = \sum_{k=1}^K l_k \text{ dir.}$$

$p_k$	Bıçkıya beslenen gelen tahta genişliği (anlık rastgele seçimde) $b_k$ ’nın olasılığı, $k=1, \dots, K$ ( $l_k/l$ ile hesaplanır)
b	Gelen kerestenin bir lineer footundaki kerestenin beklenen board feet miktarı. Bu da,

$$b = \sum_{k=1}^K p_k \frac{b_k}{12} \text{ denklemdir}$$

L	Şaft uzunluğu
h	Bıçak genişliği

### 2.2. Mil Gösterimi ve Karakteristikleri

Mümkün olan tüm millerin kümesini  $\Omega$  gösterebiliriz. Bu kümedeki j milini göstermek için aşağıdaki notasyonu kullanacağız:

$n_j$	j mili üzerindeki kesme kanallarının sayısı
$R_j$	$n_j$ büyüklüğünün vektörüdür. Başarılı girdiler, j mili üzerindeki başarılı kesme kanallarının genişliklerini gösterir



- $R_{ij}$  j milinin üzerindeki soldan i nci kesme kanalının genişliği,  $i=1, \dots, n_j$ . Örneğin,  $R_1=(w_1, w_3, w_2, w_1, w_3)$  soldan sağa başarılı bir şekilde düzenlenmiş  $w_1, w_3, w_2, w_1$  ve  $w_3$  genişliklerine sahip 5 kesme kanallı bir mili gösterir. (yani,  $R_{11}=w_1, R_{21}=w_3, \dots$ )
- $L_j$  j milinin uzunluğu

$$L_j = h + \sum_{i=1}^{n_j} (h + RR_{ij})$$

Bu da, denklemdir ve mili verilen shaft uzunluğuna uydurmak için  $L_j \leq L$  olmalıdır. Herbir  $j \in \Omega$  mili için, aşağıdaki karakteristikleri de tanımladık

- $A_{ijk}$  j mili tarafından gelen tahta genişliği  $b_k$  'dan kesilen  $w_i$  genişlikli dilimlerin sayısı,  $i=1, \dots, M$  ve  $k=1, \dots, K$ .  $A_{ijk}, b_k$  genişlikli gelen tahtaların 1 linear foot'undan kesilen  $w_i$  ürün genişliğinin linear feet'i olarak da yorumlanabilir.
- $T_{jk}$   $b_k$  genişlikli gelen tahtanın j mili ile kesilmesiyle üretilen zayıtın (talaş artı kenar dilim artıkları)

$$T_{jk} = b_k - \sum_{i=1}^M w_i A_{ijk}$$

- genişliği. Denklemi, şeklindedir.
- $a_{ij}$  Gelen kerestenin 1 linear foot'undan j mili ile kesilen  $w_i$  ürün genişliğinin beklenen linear feet'i,

$$a_{ij} = \sum_{k=1}^K p_k A_{ijk}$$

yani, şeklindedir.

- $a'_{ij}$  Gelen kerestenin 1 linear foot'undan j mili ile kesilen  $w_i$  ürün genişliğinin beklenen board feet'i, yani,  $a'_{ij} = a_{ij} \left( \frac{w_i}{12} \right)$  şeklindedir.

- $d_j$  Gelen kerestenin 1 linear foot'undan j mili ile üretilen zayıtın beklenen board feet'i, yani

$$d_j = \sum_{k=1}^K \frac{p_k T_{jk}}{12}$$

şeklindedir.

- $x_j$  Talebi karşılamak için j mili tarafından kesilen gelen kerestenin beklenen linear feet'i. Bu da,  $x_j = \max \left\{ \frac{r_j}{a_{ij}}, \text{ tüm } i = 1 \dots M \text{ için} \right\}$  şeklindedir.

- $T_j$  j miline karşılık gelen toplam zayıt, buda  $d_j x_j$  ye eşittir.

### 2.3. Mil Değerlendirişi İçin Kriterler

Verilen bir j milini ÇDMT/1 bağlamında değerlendirmek birincil kriterimiz olarak kullandığımız ölçümler aşağıdadır:

j miline karşılık beklenen zayıt yüzdesi  $\equiv WP_j = 100 d_j x_j / b$ .

j miline karşılık beklenen ürün yüzdesi  $\equiv YP_j = 100 - WP_j$ .

$$j \text{ mile karşılık beklenen aşırı üretim yüzdesi } \equiv OP_j = \frac{\sum_{i=1}^M (a'_{ij} x_j - r_i)}{\sum_{i=1}^M r_i} \times 100$$

Problemin verilen kalıbı bağlamında bir mile ilişkin aşağıdaki iki terimi kullanacağız:

Uygulanabilir mil: En az bir tane  $w_i$  (bütün  $i=1 \dots M$  için) genişlikli kesme kanalını içeren mildir.

Tam-uzunluklu mil:  $L-L_j < w_1 + h$  eşitsizliğini sağlayan  $L_j$  uzunluklu j milidir, burada  $w_1$  talep planındaki en az genişliği gösterir. Başka bir deyişle, tam-uzunluklu mil için, shaft üzerinde geriye kalan uzunluk, en küçük ürün genişliği artı bıçak genişliğinden daha küçüktür.



## 2.4. Bir Milin Değerlendirilmesi

Bir  $j$  mili, bu mile karşılık beklenen toplam  $T_j$  zayıtının bulunmasıyla değerlendirilir. Değerlendirme prosedürü her bir  $b_k$  genişlikli gelen tahtalara karşılık gelen  $A_{ijk}$  (bütün  $i$  değerleri için) değerini belirlemek için bıçkı işleminin simülasyonu ile başlar (yani, mil üzerindeki tüm kesme kanalları taranır ve  $b_k$  için zayıtı en aza indiren kesme deseni belirlenir).  $d_j$  ve  $x_j$  değerleri yukarıda açıklandığı gibi belirlenir ve karşılık gelen  $T_j = d_j x_j$  hesaplanır. Şuna dikkat edin, bu değerlendirme prosedürü nispeten hesaplamasal olarak yoğundur, çünkü kesilmemiş stoktaki  $b_k$  genişlikli gelen tahtaların her biri için ayrı ayrı bıçkı işleminin simülasyonuna ihtiyaç vardır (tipik olarak çok büyük sayıda ayrık tahta genişliklerine sahibiz).

## 3. ÇDMT/1'İN KARMAŞIKLIĞI ÜZERİNE

Bu bölümde, ÇDMT/1 ile ilişkili karar problemini tanımlayacağız ve bunun NP-tam problemi olduğunu göstereceğiz.

Eğer her bir  $b_k$  genişlikli gelen tahta, talep planına ve milin üzerinde gerçekten uygun desenin olup olmamasına bakılmaksızın, belirli bir  $b_k$  için toplam zayıtı en aza indiren desene göre kesilirse, her bir problem örneği için ürün yüzdesi üzerine teorik üst sınır elde edilebilir. Biz bu üst sınırı, bu örnek için "Ürün Sınırı" olarak adlandırıyoruz ve "Ürün Sınırı Problemi" ni (ÜSP) aşağıdaki gibi tanımlıyoruz:

YLP: ÇDMT/1'deki ürün sınırı, talep planına bakılmaksızın, verilen shaft uzunluğu ve bıçak kalınlığı ile başarılabilir mi?

ÜSP'nin aslında NP-tam problemi olduğunu gösterilmesi yeterlidir ve bu [1]'de gösterilmiştir.

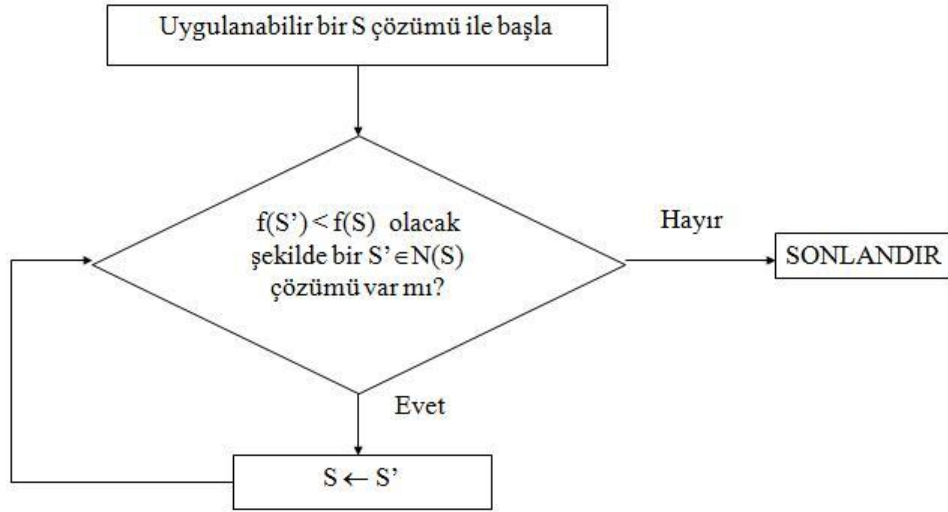
Benzer bir karar problemini, ÜSP'ye eşit veya daha küçük herhangi bir ürün değeri için tanımlayabilir ve bu karar probleminin de NP-tam olduğu aynı argümanlarla gösterebiliriz. Şuna dikkat edin, verilen bir ürün değerini başarmak için mil, her bir knapsack problemi için karşılık gelen çözümlerin bir koleksiyonunda en az bir adet olmalıdır.

Pratikte ÇDMT/1 için optimal çözümü bulmak için, bir toplam numaralandırma prosedürü geliştirdik. Bu yöntemin küçük boyutlu problemler için iyi çalışmasına rağmen, birçok pratik uygulama için zaman gereksinimi aşırı derecede fazladır (çok fazla uygulanabilir-mil sayısından dolayı ve nispeten her bir milin değerlendirilmesi için çok fazla zamanın gerekmesinden). Bu prosedürün detayları ve hesaplamasal gereksinimleri [2]'de bulunabilir.

Bu problemi çözmek için tam olmayan (sezgisel) prosedürler, [2]'de geliştirilmiştir. Yine aynı makalede optimal çözüm için en iyi prosedürün yerel iyileştirme prosedürleri olduğu gösterilmiştir. Bu nedenle, bundan sonraki bölümlerde, yerel iyileştirme prosedürlerini inceleyeceğiz, sonra bu yerel iyileştirme prosedürlerinden en iyi sonuç veren için akış şemasını vereceğiz ve en sonunda bu akış şemasını gerçekleyen yazılımımızı tanıfatacağız.

## 4. YEREL İYİLEŞTİRME PROSEDÜRLERİ

Bu bölümde, ÇDMT/1'i çözmek için birkaç tane yerel iyileştirme prosedürü (YİP) sunacağız ve bu prosedürlerin hesaplamasal deneylerinin, ampirik temeller üzerine karşılaştırmasını tartışacağız. Şekil 3, en aza indirme problemleri için YİP'in genel bir tarifini içerir. Burada,  $f(S)$  verilen bir  $S$  çözümündeki amaç fonksiyonudur ve  $N(S)$   $S$ 'nin tüm komşularının kümesini gösterir. Belirli bir problem kapsamında YİP'i uygulamak için temel elemanları, yani, (1) çözüm değerlendirmesi ve sunumunu, (2) komşuluk yapısını/yapılarını ve (3) arama stratejisini tanımlamamız gerekir. Biz, ÇDMT/1 kapsamında bu elemanlarının her birini kısaca açıkladık ve prosedürlerin değişik varyasyonlarını tarif ettik.



Şekil 3. Genel Bir YIP

Çözüm değerlendirmesi ve sunumu:  $j$  milini göstermek için  $R_j$  vektörünü kullanacağız ve buna karşılık gelen  $T_j$  beklenen toplam zayıtı Bölüm 2'de gösterildiği gibi belirleyeceğiz.

Komşuluk yapıları:  $j$  milinin, verilen bir çözümü için,  $N_s(j)$  kaydırma komşuluğu ve  $N_u(j)$  birim komşuluğu olmak üzere iki farklı komşuluk yapısını göz önüne alacağız. Aşağıda her bir komşuluğun açıklaması vardır.

$N_s(j)$   $j$  üzerindeki bir kesme kanalı seçilerek bulunduğu konumdan çıkarılıp başka bir konuma sokulmasıyla elde edilebilecek tüm  $j'$  millerinin kümesidir.

$N_u(j)$   $j$  üzerindeki bir kesme kanalı seçilerek genişliğinin başka bir ürün genişliğine değiştirilmesi ve gerektiğinde yeni mil uzunluğunda "düzeltme" yapılarak elde edilebilecek tüm  $j'$  millerinin kümesidir.

$N_s(j)$  kaydırma komşuluğu yapısında, her çözüme diğer bütün çözümlerden bu tip bir kaydırma hareketi ile ulaşılamaz ve bu ciddi dezavantajından dolayı uygulanabilir değildir. Diğer yandan  $N_u(j)$  birim komşuluğu yapısının dezavantajı, kesme kanalının genişliğini değiştirdikten sonra elde ettiğimiz  $j'$  milinin uzunluğunun şaft için çok kısa veya çok uzun olabilmesi durumudur. Bu durumda,  $j'$  milini uygulanabilir ve tam-uzunluklu yapmak için "düzeltme işlemi" gerçekleştireceğiz. Bu işlem iki adımdan oluşmaktadır:

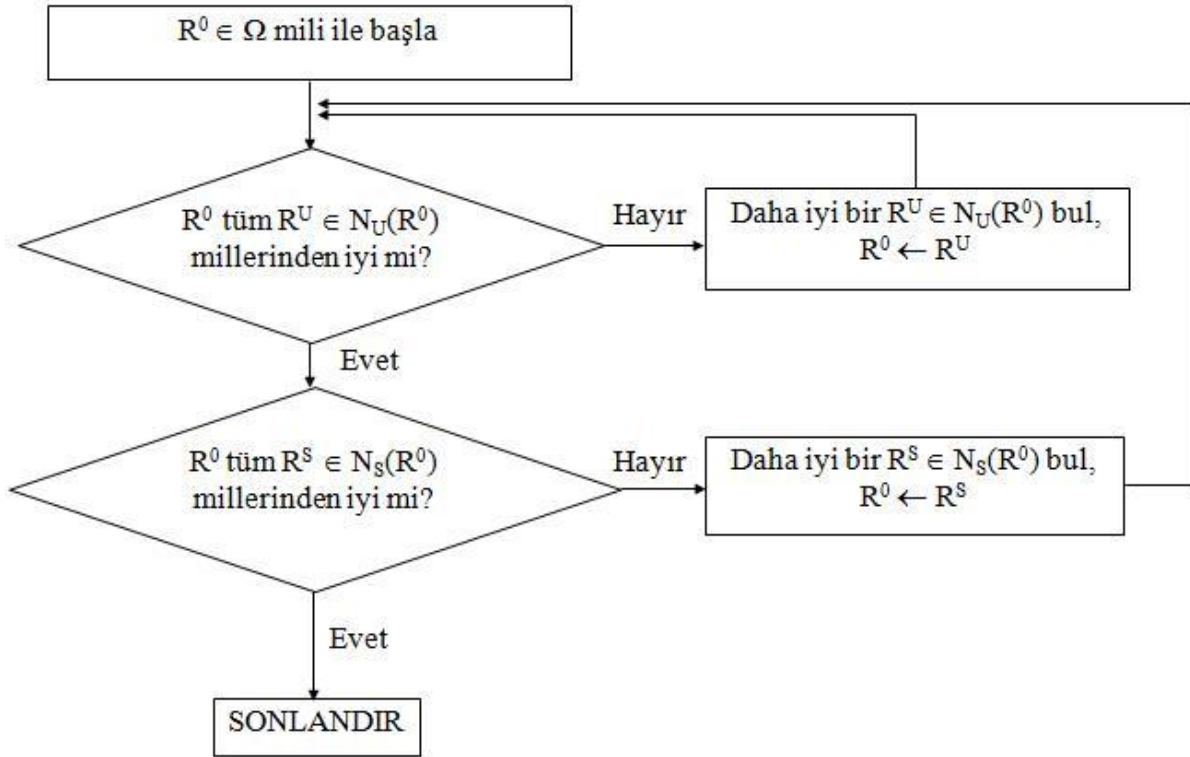
- i. **Çıkarma:** Eğer mil şaft için çok uzunsa, mil boyu şaft boyunu aşmayıncaya kadar milin sağ ucundan kesme kanalları çıkartılır.
- ii. **Ekleme:** Eğer mil çok kısa ise, mil üzerindeki kalan boşluk rasgele üretilen kanalların milin sağına eklenmesi ve bu işlemin milin uzunluğunun şaftınkine eşit oluncaya kadar devam etmesidir.

Arama stratejisi: YIP bağlamında kullanılan genellikle iki adet arama stratejisi vardır ve bunlar birinci-iyileştirme stratejisi ve dik iniş stratejisidir. Birinci-iyileştirme stratejisinde, tatminkâr bir değişimle karşılabilir karşılansız kabul edilir ve bu çözümle daha fazla arama yapılmaz. Diğer taraftan, dik iniş stratejisinde ise tüm komşuluklar aranır ve en az maliyetli çözüm seçilir. Biz YIP'lerin uygulanmasında birinci-iyileştirme stratejisini kullandık, çünkü daha önceki araştırmalarımızda tüm komşuluğu aramanın (dik iniş stratejisinde olduğu gibi) sonuçlarına değmeyecek kadar çok zaman gerektirdiğini gözlemledik. Tüm durumlarda, aramaya başlamak için rastgele üretilmiş uygulanabilir tam-uzunluklu milleri kullandık.

Algoritmaların açıklaması: Yukarıda tarif edilen komşuluk yapılarını kullanarak ÇDMT/1 için dört ayrı YIP geliştirdik. İlk iki prosedür, kaydırma ve birim komşuluklarını yukarıda anlatıldığı gibi kullanarak



Şekil 3'de gösterilen genel YİP temellidir. Bu algoritmaları sırasıyla YİP/S ve YİP/U olarak adlandıracağız. Diğer iki prosedür, bu iki komşuluk yapısının bir karışımına dayanmaktadır. Bu prosedürleri Bileşik Kaydırmalı YİP (BYİP/S) ve Bileşik Birim YİP (BYİP/U) olarak adlandıracağız ve bunlardan yazılımımızda kullandığımız BYİP/S için akış şeması Şekil 4'de verilmiştir. Bu bileşik YİP'lerin her biri iki tane iç içe döngüye sahiptir, her bir döngüde farklı bir komşuluk yapısıyla yerel arama gerçekleştirilir. Buradaki ana fikir, bir mil üzerinde bir komşuluğa göre yerel aramaya başlamadan önce, bu mil üzerine diğer komşuluğa göre yerel arama gerçekleştirilir. Bu YİP'lerin performansını karşılaştırmak için, problemin birkaç örneği için dört prosedürün her biri için çoklu koşmalar gerçekleştirdik. Adil bir karşılaştırma olması için, her bir prosedürün çalışma süresini bir dakikaya sabitledik ve bu zaman diliminde olabildiğince çok rastgele üretilmiş başlangıç noktaları ile prosedürü koştuk. Her bir problem örneği üzerinde her bir prosedürden elde ettiğimiz en iyi sonucu raporladık. Bütün problem örnekleri için BYİP/S'nin diğer prosedürlerden daha iyi olduğunu gözlemledik. Bu nedenle, ÇDMT/1'in çözümünde BYİP/S'i birincil YİP'imiz olarak seçtik.



Şekil 4. BYİP/S İçin Akış Şeması

## 5. YAZILIM

Bu bölüme kadar, ÇDMT/1 problemi ve bu problem için gerekli kavram ve parametreleri tanımladık, sonra bu kavram ve parametreleri kullanarak problemin sayısal karmaşıklığını inceledik, daha sonrada gözlemlerimiz sonucunda bu problemin optimal çözümü için en iyi prosedürün YİP olduğunu belirttik [2]. Bu prosedürün değişik varyasyonlarından (seçilen komşuluk yapılarından ve bunların karışımından doğan) en iyisi olan BYİP/S'in akış şemasını Şekil 4'de verdik.

Bu bölümde ise, BYİP/S akış şemasını gerçekleyen bilgisayar programımızı tanıtacağız. Bu yazılımın amacı ÇDMT/1 probleminin optimal çözümünü kullanıcının istediği bir zaman aralığında bulmasıdır. Bölüm 2'deki gösterimleri daha anlaşılabilir etiketlerle, kullanıcı dostu, sade ve basit bir kullanıcı arayüzü ile sunan çok kullanışlı bir bilgisayar programı yazdık. Bu yazılım ile ek herhangi bir





matematik veya bilgisayar bilgisi gerektirmeden, sadece problemi bilen herhangi biri bu problem için optimal çözüme rahatlıkla ulaşabilir.

Yazılımın kullanıcı arayüzünü tek bir ekran olarak tasarlayarak kullanıcının menüler arasında kaybolmasının önüne geçmeyi ve aynı zamanda da tüm parametreler ile sonuçları aynı ekranda görülmesini sağlayarak problemin girdi ve çıktılarına tam bir hakimiyetin elde edilmesini amaçladık. Şekil 5'de yazılımın kullanıcı arayüzü gösterdik ve sonrada bu arayüzdeki etiket ve alanların açıklamasını yaptık.

Width(mm)	Demand (bf)	Best	2nd Best	3rd Best
A - 30	300			
B - 35	270			
C - 42	250			
D - 49	200			
E - 51	100			
F -				
G -				
H -				
I -				
J -				
K -				
L -				

Şekil 5. Sezgisel Yöntemlere Dayalı Çoklu-Dilme Makinesi Yazılımı

### 5.1. Yazılımın Parametre ve Etiketlerin Açıklaması

**Units:** Seçilecek veya girilecek olan "Saw Kerf", "Arbor Length" ve "Width" parametrelerinin birimi, alabileceği değer mm veya inç'dir.

**Saw Kerf:** Bıçak kalınlığıdır.

**Arbor Length:** Toplam mil uzunluğudur (aynı zamanda şaft uzunluğu).

**Lumber Distribution File:** Kesilmemiş kereste yığınının genişliklerine göre dağılımının girildiği dosyadır. Var olan bir dosya kullanılabilir veya "Change" butonuna basılarak Şekil 6'daki gibi üzerinde değişiklik yapılabilir. Eğer hazır bir dağılım dosyası yoksa "New" butonuna basarak gelen ekran aracılığıyla yeni bir dosya oluşturulup kaydedilir.



Uncut lumber distribution (enter quantities in linear feet)

	0"	1/8"	1/4"	3/8"	1/2"	5/8"	3/4"	7/8"	
3 -	1012	123	145			167	155	157	Clear
4 -		234	256	214	213	145		156	
5 -	145	167	167		156		145	159	Save & Exit
6 -	178	190		189	188	177	167		
7 -	167	178	156	145		145		145	Cancel
8 -	345	312	155	163	144	134	167	199	
9 -	178		98	174	152	177		245	
10 -		177		144	187	164	145	113	
11 -	167		178	143	109	202		209	
12 -	188			178		155		167	
13 -									
14 -									
15 -									
16 -									
17 -									
18 -									

Şekil 6. Kereste Dağılım Fonksiyonu Dosyasının Düzenlenmesi

**Minimization Mode:** Problemimiz için, "Total lumber usage" seçeneği ile en az kereste harcanarak, "Total waste generation" seçeneği ile de toplam zayıt en az olacak şekilde optimal çözüm bulunur.

**Deman Schedule:** Talep planıdır. İstenilen ürün genişlikleri "Width" alanına, istenilen miktarları da "Demand" alanına girilir.

**Overages (%):** Milin en iyi, 2. en iyi ve 3.en iyi dizilimleri için fazla üretim miktarıdır.

**Arbor Found:** Milin en iyi, 2. en iyi ve 3.en iyi dizilimlerini verir.

**Properties:** Çözümün çıktı verilerini içeren gruptur.

- Yield: Ürün verimidir.
- Overage (%): Toplam üretilen fazlalığın yüzdesidir.
- Waste: Toplam zayıt miktarıdır.
- Overage (bf): Toplam üretilen fazlalığın board feet cinsinden miktarıdır.
- Total (bf): Toplam kullanılan kerestenin board feet cinsinden miktarıdır.
- Linear Feet: Üretim/Kerestenin toplam uzunluğunun lineer feet cinsinden miktarıdır.
- Yield Limit: Ürün verimi için teorik üst sınır değeridir.

**Search Time:** Optimal çözümü ne kadar bir süre aranacağı seçilir. Uzun zaman aralığı, daha iyi bir optimal çözüm verme ihtimaline sahiptir.

**Time Remaining:** Yazılım koşturulmaya başladığında seçilen "Search Time" dan geriye doğru saymaya başlar ve bu alanda kalan zamanı görüntüler.

**Mode:** Tüm kesim sürecinde toplam kaç mil kullanılacağı belirtilir.

**Submit Current Settings:** Bu butona basıldığında, girdiğiniz değerlerin uygunluğu kontrol edilir ve eğer uygunsa yazılım koşturulmaya hazır duruma getirilir.

**Run:** Son olarak bu butona basılır ve yazılım koşturulmaya başlanır.



## 5.2. Yazılımın Koşturulması ve Sonuçlar

Bu bölümde, yazılıma örnek bir talep planı girerek, 3 mm bıçak kalınlıklı ve tam-uzunluğu 711 mm olan mil için, bir mülle toplam zayıyatı minimum yapacak şekilde optimal çözümü aramak için yazılımı 5 saniyeliğine koşturduk ve sonuçları Şekil 7’de sunduk.

The screenshot shows the Visual GRADS software interface. It includes a Control Panel with settings for Units (Millimeters), Saw Kerf (3), Arbor Length (711), and Lumber Distribution File (oak2.ldf). The Minimization Mode is set to Total waste generation. The Arbors Found section lists three best arbor solutions: Best Arbor (EAEACABDBCCDABAA), Second Best Arbor (ABAAACDBBCCAABEADC), and Third Best Arbor (EBDACBBAAACEABCAA). The Demand Schedule table shows widths from 30 to 51 mm and demands from 100 to 300 bf. The Overages (%) table compares the Best, 2nd Best, and 3rd Best solutions. The Properties table shows Yield (%), Overage (%), Waste (bf), Overage (bf), Total (bf), and Linear Feet for each solution. The interface also includes a Search time of 5 sec, a Mode of 1-Arbor, and a Time remaining of 01 min.

Width(mm)	Demand (bf)	Best	2nd Best	3rd Best
A - 30	300	0.0	0.1	0.4
B - 35	270	0.1	0.7	0.0
C - 42	250	1.3	1.0	0.6
D - 49	200	10.7	0.0	2.0
E - 51	100	5.4	1.5	11.7
F -				
G -				
H -				
I -				
J -				
K -				
L -				

	Best	2nd Best	3rd Best
Yield (%)	90.1	89.7	89.8
Overage (%)	2.7	0.5	1.7
Waste (bf)	127	129	129
Overage (bf)	30	6	19
Total (bf)	1277	1255	1268
Linear Feet	2504	2460	2485
Yield Limit (%)	91.0		

Şekil 7. Yazılımın Koşturulması ve Sonuçların Görüntülenmesi

Yazılım bize burada, girilen koşullar altında istenen talep planını en az zayıyatla üretmek için gerekli olan en iyi, 2.en iyi ve 3.en iyi mil desenini verdi. Burada yazılımızın bir artısı daha görülmektedir, o da şudur: Ürün veriminin birbirine çok yakın olduğu durumlarda, sonraki üretim planlarını da göz önüne alarak hangi ürün genişliğinden fazlalık üretilmesini istiyorsak (“Overages” grubuna bakarak), 3 mil deseninden buna en uygun olanını seçebiliriz.

## KAYNAKLAR

- [1] Aksakalli, V., Heuristic methods for gang-rip saw arbor design and scheduling, Master’s Thesis, Department of IE, NCSU, Raleigh, NC, 1999.
- [2] Fathi, Y., Aksakalli, V., Heuristic methods for gang-rip saw arbor design, European Journal of Operational Research, 154, 626-640, 2004.



- [3] Fathi, Y., Kegler, S.R., Culbreth, C.T., A column generation procedure for gang-rip arbor design and scheduling, International Journal of Production Research , 2, 34, 313–327, 1996.

## ÖZGEÇMİŞ

### **Vural AKSAKALLI**

Lisans derecesini Orta Doğu Teknik Üniversitesi Matematik Bölümü'nden ve yüksek lisans derecesini North Carolina State University Endüstri Mühendisliği'nden aldı. İkinci yüksek lisans derecesini ve doktora derecesini Johns Hopkins University (Baltimore, Maryland) Uygulamalı Matematik ve İstatistik Bölümü'nden aldı. Şuan İstanbul Şehir Üniversitesi Endüstri Mühendisliği'nde yardımcı doçenttir. İlgili alanları, sezgisel eniyileme, veri madenciliği ve elektronik ticarettir.

### **İbrahim ARI**

Lisans ve yüksek lisans derecelerini Yıldız Teknik Üniversitesi Fizik Bölümü'nden aldı. Özel bir şirkette, iki yıl bilgisayar programcısı olarak çalıştı. Şuan İstanbul Şehir Üniversitesi Endüstri Mühendisliği'nde ikinci yüksek lisansını yapmaktadır.