

TALAŞLI İMALAT İŞLEM PLANLARININ  
HAZIRLANMASINDA  
UZMAN SİSTEM YAKLAŞIMI

Mehmet TAŞPINAR

Yüksek Lisans Tezi  
Endüstri Mühendisliği Anabilim Dalı  
Yöneylem Araştırması Bilim Dalında

EYLÜL - 1991

Anadolu Üniversitesi  
Merkez Kütüphane

Mehmet TAŞPINAR'ın YUKSEK LİSANS tezi olarak hazırladığı "Talaşlı İmalat İşlem Planlarının Hazırlanmasında Uzman Sistem Yaklaşımı" başlıklı bu çalışma, jürimizce lisansüstü tez yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

10.9.1991

Danışman : Prof.Dr. Musa ŞENEL

Uye : Doç.Dr. Ali GUNEŞ

Uye : Yrd.Doç.Dr. Doğan EROL

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 10.9.1991  
gün ve ...286-2 sayılı kararıyla onaylanmıştır.

Prof. Dr. Rüstem KAYA

Enstitu Muduru

3. TALAŞLI İMALATTA İŞLEM PLANLARI .....	22
3.1. İşlem Planlarının Yapısı .....	23
3.2. Kesme Değişkenlerinin Tanıtımı .....	24
3.2.1. Paso .....	25
3.2.2. İlerleme hızı .....	26
3.2.3. Kesme hızı .....	26
3.3. Kesme Değişkenlerinin İncelenmesi .....	27
3.3.1. Paso ve ilerleme .....	27
3.3.2. Kesme hızı .....	28
3.4. Maliyet Analizleri .....	30
3.4.1. Değişken maliyetler .....	30
3.4.1.1. Fiili işleme maliyeti .....	30
3.4.1.2. Kesici takıma bağlı maliyet .....	31
3.4.1.3. Ek faaliyetlerin maliyeti .....	32
3.4.1.4. Stoklara ilişkin maliyetler .....	32
3.4.2. Sabit maliyet .....	34
3.5. Kesme Hızının Etkilediği Faktörler .....	35
3.5.1. Kesme hızı ile üretim süresi arasındaki ilişki .....	36
3.5.2. Kesme hızı ve toplam birim maliyet arasındaki ilişki .....	37
3.5.2.1. Kesme hızı birim sabit maliyet arası ilişki .....	37
3.5.2.2. Kesme hızı birim değişken maliyet arası ilişki .....	38
3.5.2.3. Kesme hızı ile kâr arasındaki ilişki .....	40
3.6. Sayısal Bir Uygulama .....	42
3.6.1. Uygulamaya ilişkin bilgiler .....	42
3.6.2. Sayısal uygulamadan elde edilen sonuçlar .....	46
3.6.2.1. Üretim süresi ve kesme hızı ilişkisi .....	46
3.6.2.2. Toplam birim maliyet ve kesme hızı ilişkisi .....	47
3.6.2.3. Kâr ile kesme hızı arasındaki ilişki .....	50

4. GELİŞTİRİLEN YÖNTEM .....	52
4.1. Yöntem Hakkında Genel Bilgi .....	52
4.2. Yapılan Varsayımlar ve Gözönüne Alınan Kısıtlar .....	53
4.3. Geliştirilen Yöntemin Genel Yapısı .....	55
4.4. Geliştirilen Programın İçeriği .....	58
4.4.1. Programın içerdiği modüller ve işlevleri	58
4.4.2. Programın içerdiği veri-bilgi tabanları	65
5. SONUÇ ve ÖNERİLER .....	67
6.KAYNAKLAR DİZİNİ .....	69

#### EKLER

- EK-1 Yapay Zeka ve Uzman Sistemlerde yapılan çalışmalar
- EK-2 Talaşlı imalat hakkında yapılan çalışmalar
- EK-3 Geliştirilen program
- EK-4 Geliştirilen programın çıktıları

**UZET**

Talaşlı imalat işletmelerinde işlem planları hazırlanırken, planın ekonomik yönü genelde ihmal edilir. Teknik açıdan eniyi plan için her zaman, ekonomik olarak da eniyi plandır denilemez. Bunun için, talaşlı imalat işlem planları hazırlanırken, işlemlerin hem teknik hemde ekonomik yönü dikkate alınmalıdır.

Çalışma sırasında, işlemlerin ekonomik yönünün önemini vurgulayabilmek amacıyla bir örnek problem üzerinde teknik kısıtları sağlayan birçok plan türetilmiş ve en ekonomik planın seçimi yapılmıştır.

Örnek çalışma üzerinde verilen yöntemle benzer bir yöntem kullanılarak bilgi tabanlı bir uzman sistem geliştirilmiştir. Söz konusu sistem geliştirilirken Turbo Prolog programlama dilinden yararlanılmıştır.

Geliştirilen Uzman Sistem kullanılarak eniyi kesme değişkenleri değerleri, gerekli olan kesici takımlar ve karşılaşılabilecek maliyetler belirlenir.

Eniyi işlem planında, toplam birim maliyet en küçük değerini alır.

**Anahtar kelimeler**

Uzman Sistemler

Talaşlı imalat işlem planları

Talaşlı imalat maliyetleri

## S U M M A R Y

Economical aspect of machining operation plans are neglected in machine shops, in general. The best technical plan is not necessarily the most economical plan.

So, both technical and economical aspects of the processes should be considered to prepare the machining operation plans.

In this study, a lot of plans are generated on a case study to emphasize the importance of the processes and the most economical plan is choosed.

A knowledge base expert system is developed by using a similar method on the case study by making use of Turbo Prolog programming language.

The best cutting variables values and necessary cutting tools would be determined by using the proposed knowledge base expert system.

The unit total cost is the minimum in the best machining operating plan.

### Key Words

Experts Systems

Machining Operating Planing

Machining Operating Costs

## ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
2.1. Uzman Sistemin genel yapısı .....	9
2.2. Derinlemesine öncelikli araştırma metodu ....	19
2.3. Genişlemesine öncelikli araştırma metodu ....	20
3.1. Talaşlı imalat işlem planları genel yapısı ..	24
3.2. Bir delik işleme işlemi .....	28
3.3. Kesme hızı ve üretim süresi arası ilişki ....	37
3.4. Kesme hızı ve üretim maliyetleri arası ilişki	40
3.5. İşlemlerden elde edilecek karın oluşumu .....	41
3.6. Örnek için kesme hızı ve üretim süresi arası ilişki.....	47
3.7. Örnek için kesme hızı ve toplam birim maliyet ilişkisi.....	48
3.8. Örnek için kesme hızı ve üretim maliyetleri arası ilişki.....	50
4.1. Talaşlı imalat için Uzman Sistem yapısı .....	56

**ÇİZELGELER DİZİNİ**

<u>Çizelge</u>	<u>Sayfa</u>
2.1. Çeşitli kesme hızlarına karşı gelen üretim suresi bileşenlerinin alacağı değerler .....	44
2.2. Çeşitli kesme hızlarına karşı gelen üretim maliyeti bileşenlerinin değerleri .....	45



## 1. GİRİŞ

Bilgisayar teknolojisinde son yıllarda önemli gelişmeler olmaktadır. Bu gelişmelerden biri de **Yapay Zeka (Artificial Intelligence)** adı altında yapılan çalışmalardır. Yapay zeka çalışmaları son yıllarda uygulama alanında görülmeye başlamıştır. Yapay zeka çalışmaları çalışma alanına bağlı olarak **Uzman sistemler (Expert systems)**, **Robotlar (Robotics)** vb. değişik adlarla anılırlar. Ticari uygulamalar, en fazla uzman sistemler adı altında gerçekleşmektedir.

Talaşlı imalatta kesme değişkenlerinin alacağı değerlerin, gerçekleştirilen işlemlerin ekonomik yönü üzerinde oldukça fazla etkisi vardır. İşlemlerin ekonomisini etkileyen maliyet bileşenleri, kesici takım maliyeti, kesme işlemi maliyeti, tezgah sabit maliyeti ve ek süre maliyeti olarak verilebilir. Kesme planı, talaşlı imalat işletmelerinde genellikle bir atölye mühendisi tarafından hazırlanır. Atölye mühendisleri, sadece işlemlerin gerçekleştirilebileceği teknik şartları göz önüne alırlar. Atölye mühendisinin yerine getirdiği bu işler, sadece teknik yönü dikkate almasına rağmen yine de oldukça sıkıcı ve zaman alıcı işlerdir.

Kurulan tüm işletmelerin kuruluş amacı kar elde etmek olduğuna göre, işlem planlarının teknik yönüne verilen önem kadar, ekonomik yönüne de önem verilmelidir.

Bu çalışma sırasında, talaşlı imalat işlem planlarının ekonomik yönünü dikkate alacak ve gerekli bilgileri oluşturacak bir uzman sistem yapısı geliştirilmeye çalışıldı. Amaç, fazladan harcanan zaman ve sıkıcı

işlemlerden kurtulmak olduğu kadar, talaşlı imalat işlem planlarının ekonomik yönünün de dikkate alınmasının gerekliliğini vurgulamaktır.

Çalışmanın yapılabilmesine olanak sağlayacak bir temel oluşturması açısından, öncelikle Yapay Zeka ve Uzman Sistemler hakkında çalışıldı ve bu konuya ilişkin bilgiler İkinci Bölümde verilmektedir.

Talaşlı imalat işlem planlarının yapısının anlaşılabilmesi ve bu planların ekonomik yönünün önemini vurgulayabilmek için Üçüncü Bölüm hazırlandı.

İkinci ve Üçüncü bölümlerde oluşan bilgi birikimi yardımıyla bilgi tabanlı bir uzman sistem görevini yerine getirebilecek program, Dördüncü Bölümde geliştirilmeye çalışıldı. Programın oluşturulabilmesi için yapay zeka programlama dilleri incelendi ve **Turbo Prolog 2.0** programlama dili seçildi. Söz konusu program ve yerine getirdiği işlevler Dördüncü Bölümde detaylı olarak açıklanmaktadır.

Yapılan çalışmanın sonucunda, elde edilen sonuç bilgileri ve önerilebilecek çalışmalar Beşinci Bölümde açıklanmaktadır.

## 2. YAPAY ZEKA ve UZMAN SİSTEMLER

Bu bölümde, "Yapay Zeka nedir?", "Nasıl gelişmiştir?", "Uygulama alanları nelerdir?" gibi sorulara cevap olabilecek bilgiler verilmiştir. Bu bölümün amacı, daha sonraki bölümlerde yapılacak çalışmalara temel oluşturmaktır.

### 2.1. Yapay Zeka

Eski çağlardan beri, insanı diğer bütün varlıklardan ayıran özelliğin "düşünmek" olduğu söylenir. Fakat, "düşünce" denen bu ayırıcı özelliğin nasıl bir şey olduğu, diğer yaratıklara özgü hareketlilik biçimlerinden nasıl, hangi biçimlerde farklılık gösterdiğinin araştırılması ancak 17. yüzyıl'dan sonra önem kazanmıştır. Daha önceleri, özellikle filozoflar mantık başlığı altında "düşünmek" ten söz ediyorlardı. Fakat, o zamanlar mantık, daha çok bir araç olarak görüldüğü için, mantık başlığı altında, insanların düşünürken ne yaptıklarına ilişkin değil, istenen bazı şeyleri gerçekleştirmek için doğru olduğu bilinen önermelerden hareketle, doğru bir sonuç elde edebilmek için ne yapmaları gerektiğine ilişkin kurallar incelenirdi (Bilgisayar Ansiklopedisi, 1987).

Burada sözü edilen kural kelimesi YAPAY ZEKA (Artificial Intelligence) araştırmalarının temelinde yatan bir terimdir. Kural teriminin anlamı, gündelik dilde kullanılan anlama benzemekle beraber, Yapay Zeka (YZ)

içerisinde hem sınırları hem de anlamı açısından özel bir yapıyı içerir. Bir şeyin nasıl yapılmakta olduğuna ilişkin kurallar ile, nasıl yapılması gerektiğine ilişkin kurallar arasında farklılık vardır. Bir şeyin nasıl yapılmakta olduğuna ilişkin kurallara **"oluşturucu"** (constitutive) kurallar, nasıl yapılması gerektiğine ilişkin kurallara da **"düzenleyici"** (regulative) kurallar denir (Bilgisayar Ansiklopedisi, 1987).

Oluşturucu kurallar, kuralı oldukları faaliyeti, gerçekleştiren kurallardır. Kendilerinden bağımsız olarak var olan bir faaliyeti düzenlemezler. Örneğin, satranç oyununda, ilk hamle olarak şahın önündeki piyonu açmak belli bir üstünlük sağlar, dolayısıyla bu iyi bir kural olabilir. Ancak ilk hamle olarak başka bir taşı örneğin atı oynamak da mümkündür. Başka bir deyişle, insan bu kurala uymadan da satranç oynayabilir. Dolayısıyla, "oyuna şahın önündeki piyonu açarak başlanır" kuralı "düzenleyici" bir kuraldır. Birde, "filler çapraz ilerler" kuralını ele alalım. Eğer bu kurala uyulmuyorsa oynanan oyun satranç değildir. Dolayısıyla bu kural "oluşturucu" bir kuraldır.

Aslında Yapay zekanın gerçekleştirebileceği düşüncesinin ileri sürülmesi, düşünce gerektiren tüm faaliyetlerin, oluşturucu kurallar bilgisine sahip olmak ve bu kuralları uygulamaktan ibaret olduğunun kabulünden kaynaklanır. (Bilgisayar Ansiklopedisi, 1987)

### 2.1.1. Yapay zekanın tanımı

"Yapay zeka (artificial intelligence) nedir?" sorusuna cevap arandığında hemen hemen konu ile ilgilenen tüm araştırmacıların birer tanım yaptığını görürüz. Bu tanımlamalardan bazıları aşağıda verilmektedir.

-YZ, bilgisayarı düşünen bir makina haline getirilebilme yollarının araştırılmasıdır.

-Bilgisayarın insan gibi davranmasını sağlama yönlü bir girişimdir.

-YZ, insanların bilgisayarlardan olmayacak birşey (bilgisayar zekası) beklemesine sebep olan şaşırtıcı bir terimdir.

-YZ, insan yapısı sistemlerin beklenmedik gerçekler karşısında hayatta kalma yeteneğidir.

-İnsanların yaptıkları işleri bilgisayarlara eniyi nasıl yaptırabiliriz, sorusuna cevap arama konusunda yapılan bir çalışmadır.

-YZ, bilgisayarların karşılaşılan belirsiz süreçleri çözümleyebilme yeteneğidir.

-YZ, insan zekasının bilgisayar ortamında modellenebilmesi çalışmalarıdır.

-Bu terim, insan düşünce sürecini tekrarlayabilen programlar için kullanılmalıdır.

-Kendi varlığını bilmesi YZ'nin anahtar kavramıdır.

-Eğer fiziksel olarak çalışan birşeyler varsa bu YZ değildir.

-Akıllı diye adlandırılan sistemlerin nasıl çalıştığı biliniyorsa bu sistemler akıllı sistemler değil, olsa olsa bilgisayar programları olurlar.

-İnsan bir çok durumda problemleri bir fonksiyon olarak tanımlama gereği duymadan düşünme yeteneği, sezgi ve

deneyimlerini kullanarak çözer. YZ insanın bu yeteneklerini anlama ve taklit etme amacına yönelik çalışmalarla ilgilenen bilim dalıdır (Williamson, 1986).

-YZ'yı, bilgisayarları insanlarda var olan ve akıllı olarak vasıflandırdığımız davranış şekillerine benzer davranışlar gösteren makinalar durumuna getirme konusunda çalışmalar yapan bir bilgisayar bilim dalı olarak tanımlayabiliriz (Timuroglu, 1988).

Yapay zeka ve Uzman Sistemler konusunda yapılan çalışmalar tarih sırasına göre EK-1'de verilmektedir.

## 2.2. Yapay Zekanın Uygulama Alanları

Bir işin yapılması ya da bir problemin çözülmesi için bilgisayar kullanmaya karar verildiğinde, eğer problem matematiksel fonksiyonlar olarak tariflenemiyor ya da çalışma alanının kısıtları tanımlanamıyorsa YZ çalışma alanı içerisindedir, demektir.

Bu durumda, bilgisayarı uzman insanların probleme yaklaşımlarını taklit edecek şekilde programlamamız gerekir. Eger insanın bilgi işleme düzeni tam olarak bilinmiyorsa, bu düzeni yaklaşık taklit etme yoluna gidilir.

Günümüzde bir çok alanda YZ çalışmalarını görmek mümkündür. Bu alanlardan en önemlileri aşağıdaki gibi verilebilir.

- .Araştırma, (Search)
- .Uzman Sistemler, (Expert Systems)
- .Doğal dil işleme, (Natural Language Processing)
- .Desen gösterimler, (Pattern Matching)
- .Robotlar, (Robotics)

- .Öğrenme, (Learning)
- .Mantık, (Logic)
- .Belirsizlik ve Bulanık Kumelerle Çalışma, (Uncertainty and ,fuzzy logic).

Bu guruplar dahilinde yapılan çalışmaların neler olduğu aşağıda açıklanmaktadır.

### 2.2.1. Araştırma

Araştırma terimi YZ uygulamalarında, bir probleme çözümler bulma olarak kullanılır. Buradan çıkarılacak anlam bir veri tabanından özel bir bilgi parçacığının bulunması değildir. İki şehir arasındaki en kısa yolun ya da matematiksel bir teoremin doğruluğunu ispatlama gibi problemlerin çözümü bu çalışma alanında yapılan çalışmalara örnek verilebilir. Bu alanda yapılan çalışmalar sonucu bir çok araştırma yöntemi gelişmiştir (Schildt,1987).

### 2.2.2. Uzman sistemler

Uzman sistemler YZ'nin ilk ticari amaçlı uygulamalarıdır . Bir uzman sistem öncelikle iki önemli işlevi yerine getirir. Birincisi, bilgisayara bir nesne (olay) hakkında bilgi girişini sağlamasıdır. Bu genelde bilgi tabanı (Knowledge Base) olarak isimlendirilir. İkincisi, bu bilgi tabanı üzerindeki, bilgi parçacıklarını kullanarak uzman bir insanın uzmanlık alanında karşılaştığı bir problemi çözmek için yürüteceği mantıksal yapıya benzer bir çıkartım mekanizması kullanmasıdır. Bu tür özellikler gösteren programlar "UZMAN SİSTEM" olarak adlandırılır (Harmon and King,1985). Dağlının tanımlamasına göre ise Uzman Sistem, uzman insanın davranışlarını taklit ederek belirli bir alanda uzmanlık gerektiren karmaşık problemlerin çözümü için bir veya birden fazla yaklaşım

öneren akıllı bilgisayar programlarıdır (Daglı,1986). Yapılan çalışma ile doğrudan ilgili olması sebebi ile bu konuda detaylı bilgi verilmektedir.

#### 2.2.2.1. Uzman sistemin yapısı

Genellikle Uzman Sistemler üç ana bölümden oluşmaktadır.Bunlar,

- i-Bilgi tabanı
- ii-Çıkartım mekanizması
- iii-Kullanıcı bağlantısı

Bilgi tabanı ilgilenilen sistemle ilgili hemen hemen tüm bilgileri içerir. Bu bilgiler konunun uzmanlarından bilgi mühendisleri tarafından toplanmış bilgilerden oluşur. Söz konusu bu bilgiler "Eğer-Dolayısıyla" (IF-THEN) kuralları şeklinde ifade edilmişlerdir. İleriki bölümlerde geliştirilen sistem çerçevesinde benzer kurallar açıklanmaktadır.

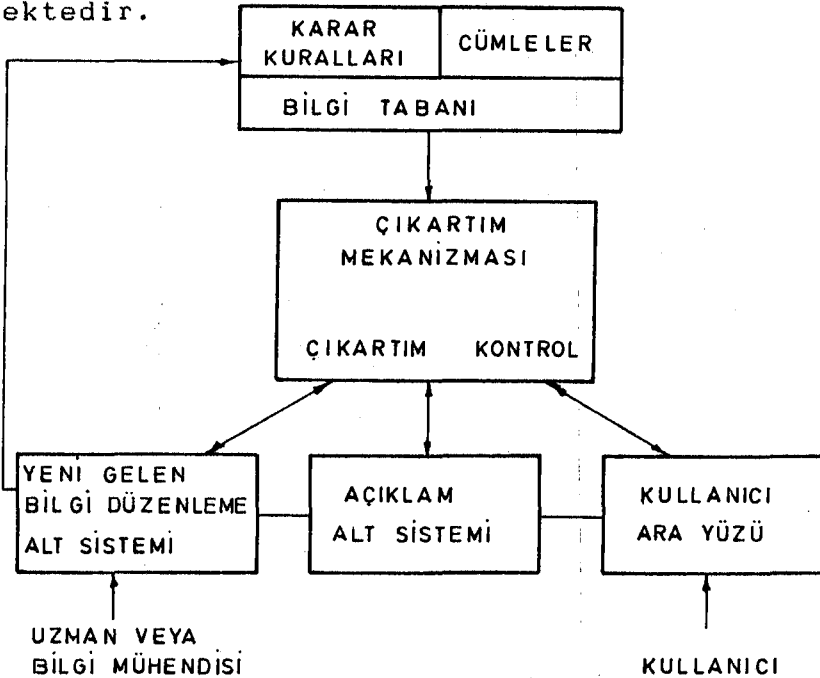
Çıkartım mekanizması ise geliştirilen sistemin tüm işlemlerinin kontrol edildiği yerdir. Burada bir dizi sezgisel yöntem, sembolik mantık ve karşılaştırma yaklaşımları kullanılarak problemi çözmek ve sonuca ulaşmak için bilgi tabanından nasıl yararlanılacağı konusunda kararlar verilir (Daglı,1986).

Talaşlı imalat ortamında da hangi bilgilerden nasıl, ne zaman yararlanılacağına ilişkin kararların verilmesi gerekmektedir. Ürneğin en iyi kesme planı enküçükleme ölçütüne göre seçilecek ise türetilecek birçok işlem planı içerisinde enküçük maliyetli olanın seçilmesi çıkartım mekanizması tarafından yerine getirilir.



Kullanıcı bağlantısı ise Uzman sistem ile kullanıcının iletişimini sağlar. Sistemin bu parçası kullanıcıdan bazı bilgilerin alınması ve kullanıcıya bilgi aktarılması işlerini yerine getirir. Talaşlı imalat işlem planlarının oluşturulmasında da kullanıcıdan yeni tezgah bilgileri gibi bilgiler istenmekte, buna karşılık türetilen eniyi plan kullanıcıya aktarılmaktadır.

Şekil-2.1 bir Uzman Sistemin genel yapısını vermektedir.



Şekil 2.1- Bir Uzman Sistemin Genel Yapısı.

#### 2.2.2.2. Uzman sistemlerin yararlı ve sakıncalı yönleri

Bir Uzman Sistemin, bir uzman kişiye göre üstün ve sakıncalı yönleri maddeler halinde şöyle verilebilir (Ulengin, 1987).

- Uzman kişi bir insan olduğu için fizyolojik ya da ruhsal bir hastalık sonucu faydalı olmaktan çıkabilir, bir kaza sonucu kaybedilebilir. Uzman Sistemlerde sistem çok kolay çoğaltılabileceği için böyle bir durum söz konusu değildir.

- Uzman Sistemler manyetik ortamlar sayesinde çok kolay taşınabilirler, çoğaltılabilirler. Uzman kişinin yetişmesi yıllar alabilir.
- Uzman Sistemlerin işleyiş mekanizması ve kullandığı mantıksal yapı kolay anlaşılır. Uzmanlar ise kendi kullandıkları yapıları bile bilemezler veya bilselerde kolay bir şekilde başkalarına aktaramazlar.
- Uzman Sistemler kararlılık gösterirler. Uzman kimseler ise aynı durumla farklı iki zamanda karşılaştıklarında farklı davranış gösterebilirler. Buldukları psikolojik durum kararlarını etkiler.
- Uzman Sistemler ucuzdur. Uzman kişi ise bulunduğu alanda çok az kişiden biri olduğu için yararlanılması çok pahalıdır.
- Uzman Sistemler, çözümünde duygu gerektiren problemlerin çözümünde kullanılamazlar.
- Uzman sistemler yaratıcı değildir. İnsanlar ise en gelişmiş uzman sistemlerden çok daha fazla yaratıcıdır. Uzman biri bilgiyi yeniden sentezleyerek yeni bilgilere ulaşabilir.
- Uzman kişi dünya bilgisini kullanabilir. Uzman sistemler genel amaçlı olmadıkları için böyle bir şey yapamazlar. fakat uzman sistemler uzmanlık alanında elde ettikleri bilgileri bilgi tabanına eklenmesi ve bilgilerin son şekliyle yararlanması mümkün olabilmektedir. Dördüncü bölümde anlatılan sistemede öğrenme işlevinin nasıl sağlandığı anlatılmaktadır.

- Uzman kişiler girdileri çok daha geniş bir açıyla çok daha kolay elde ederler. Uzman kişi duyu organları sayesinde problemleri ve olayları daha kolay kavrar. Uzman sistemlerde azda olsa bilgi kaybı söz konusudur.

### 2.2.2.3. Uzman Sistemlerin klasik programlardan farklılıkları

Uzman sistemler bilgiyi işler, klasik programlar ise veriyi işler. Klasik programların büyük çoğunluğu algoritmiktir. uzman sistemlerde sergisel yöntemler önemli yer tutar. Klasik programlar herkesin yapabileceği bir takım işleri hızlı ve doğru bir şekilde yapar. Uzman sistemler bir uzmanın işlerini yapar. Uzman sistemler bilgi mühendisi ve uzman kişi tarafından oluşturulur. Klasik programlar ise bilgisayar program alan tarafından yazılır. Uzman sistemlerin kullandığı mantıksal yapı klasik programların aksine programın doğrudan okunması ile anlaşılabilir (Ulengin, 1986). Ayrıca uzman sistem bulduğu sonuçlardan yeni genellemelere gidebilir. Yani öğrenme temelinde oluşturulabilir. Dördüncü bölümde anlatılan uzman sistem yapısı içerisinde anlatıldığı gibi bir uzman sistem bulduğu sonuçları değerlendirebilir. Bu değerlendirme şekli dördüncü bölümü anlatmaktadır.

### 2.2.2.4. Uzman Sistem geliştirme aşamaları

Bir uzman sistem geliştirilirken bazı aşamalardan geçer. Bu aşamalar aşağıdaki gibi verilebilir (Watts, 1987).

1 - Tanımlama

2 - Kavramsallaştırma

- 3 - Formülleştirme
- 4 - Yürütme
- 5 - Test etme ve değerlendirme
- 6 - Bakım planlama

Tanımlama aşamasında problemin ne olduğu ve uzman sistem için uygunluğu belirlenir. Bu aşamada ayrıca problem doğal dil kullanılarak yazılı hale getirilir.

Kavramsallaştırma aşamasında bilgiler düzenlenir. Bilgilerin birbirleri ile ilişkileri tariflenir. Ve uzman sistemin bir prototipi geliştirilir. Ve kullanılacak aletler (diller) belirlenir. Formüle etme aşamasında, bilgiler kullanılacak olan aletlere uygun şekle getirilir. Burada ayrıca uzman kişiden ek bilgiler alınır.

Yürütme aşamasında elde edilen bilgiler kullanılan dilin yapısına uygun şekilde kodlanır. Bu aşamada bilgi mühendisi ve uzmanın beraber çalışması gerekir. Böylece sistemin prototipi ortaya çıkmış olur.

Test etme aşamasında örnek problemler kullanılarak sistem uzman ve bilgiler mühendisi tarafından denenir. Bu aşamada bazı adımların geliştirilmesi yapılır. Sorun çıktıkça geri dönülür düzeltmeler yapılır.

Son adım olan bakım planlamasında ise kullanılan bilgi tanbanın güncelliğini koruyacak düzenlemelerin zamanı ve güncelleştirme aşamasında yapılacak işler tariflenir.

Bu çalışma bir uzman sistem geliştirme çalışması değildir. Çalışmadan amaç, oldukça fazla bilgi içeren ve bu bilgiler arasında ilişkiler tarifienebilen talaşlı imalat

işlem planları hazırlama konusunda bir uzman sistem geliştirileceğini göstermek ve böyle bir sistemin yararlarını tartışmaktır.

### 2.2.3. Doğal dil işleme

YZ'nın en çok araştırma yapılan alanıdır. Bu konu üzerinde yapılan çalışmaların oldukça fazla olmasının nedeni, doğal dil işleme çalışmalarının sonuçlarının, insanların bilgisayarlarla doğrudan iletişim yeteneğini bilgisayara kazandırmak olmasıdır. Bu amaca ulaşmadaki zorluk, insanların kullandıkları dilin çok karmaşık ve çok boyutlu olmasındandır. Bu problemin zorluk gösteren diğer bir yönü de, bilgisayarlarla insanların kuracağı iletişimin çok benzer yapılar içermesidir (Aynı cümlenin farklı durumlarda farklı anlamlar taşıması gibi) (Schildt,1987).

### 2.2.4. Desen gösterimler

Desen işleme ve şekillerle çalışma, robotlar ve şekilsel süreçlerin analizlerini içeren uygulamalar için önemlidir (Turban,1990). Örneğin, bir televizyon ekranı üzerindeki nesnelere başlangıç ve bitiş yerleri veya hangi nesnelere hangi nesnelere üzerine konulabileceğini bilgisayara nasıl tanıttırabiliriz sorusu gündeme geldiğinde bu tür çalışmalara olan ihtiyaç kendini gösterir. Gölgelemlerin nesnelere ayırte edilmesi, bu çalışma alanının en önemli problemidir.

### 2.2.5. Robotlar

Robot uygulamaları, YZ'da hareketlerin nasıl kontrol edileceği hususunda yapılan çalışmalara verilen addır (Turban,1990). Montaj hatlarında kullanılan endüstriyel robotlar bu çalışmaların ürünüdür. Bu alanda yapılan

çalışmalar en önemli uğraşını, kesikli hareketlerden doğal hareketler kümesi üretebilme yönünde gösterirler. Genel amaçlı robotlar üretme çalışmaları içerisinde bulunan araştırmacıların en önemli sorununu, doğal dilin robotlar tarafından anlaşılır hale getirilmesi, değişen çevre koşulları ve beklenmedik olaylar karşısında robotların davranışlarının tarif edilememesi oluşturur (Nilson,1980).

#### 2.2.6. Öğrenme

YZ'nin ilginç çalışma alanlarından biri de öğrenme yeteneğini bilgisayara öğretme yönünde yapılan çalışmalardır. Bu alanda yapılan çalışmalar, yapılması istenen şeylerden, gözlemlerden ve oluşan hatalardan yeni şeyler öğrenen bilgisayar yazılımları oluşturma çabaları olarak tanımlenebilir. Öğrenmenin diğer bir anlamıda tecrübelerden yararlanabilir bilgisayarlar geliştirmektir (Schildt,1987).

#### 2.2.7. Mantık

Diğer bir YZ çalışma alanı, mantığın standart kurallarını kullanarak bir önermenin mantıksal olarak doğruluğunu gösteren programlar üretmektir. (Turban,1990) Bu alanda yapılan çalışmalar, matematiksel teoremlerin ispatı, şekilsel mantık vb. alanlarda yoğunlaşır.

#### 2.2.8. Belirsizlik ve bulanık kümeler

İnsanoğlu bazen vereceği kararları, elinde net bilgiler olmadan da verebilir. Bu tür bilgileri açık olmayan ve/veya olasılık içeren bilgiler oluşturur. Söz konusu bu bilgileri kullanarak, istenen sonuçlara ulaşabilen makineler ve programlar geliştirme çalışmaları bu alan içerisinde tartışılabilir (Schildt,1987).

### 2.3. Yapay Zeka Programlama Dilleri

Genel olarak her programlama dili ile her türlü program yazılabilir. Fakat her programlama dili kullanıldığı alanların gereksinimlerine göre gelişip güçlenir. YZ çalışmaları sayılardan çok kavramlarla ilgilendiği için, kavramsal niteliklere sahip dillerin ortaya çıkmasına neden olmuştur. Bu amaçla geliştirilen iki önemli programlama dili vardır. Bunlar, **PROLOG** (Programing Logic) ve **LISP** (List Processing) diye bilinirler.

LISP 1958 yılında John McCARTY tarafından geliştirilmiştir. (Hekmatpaur, 1989) FORTRAN programlama dilinden sonra en eski yüksek dil olmasına karşın, büyük bellek gereksinimi nedeniyle bilimsel araştırmaların dışında yaygınlaşması büyük belleğin kuçulmesine bağımlı kalmıştır. 50 KB'lık bir belleğin lüks olduğu dönemlerde mikrobilgisayarlarda LISP'in kullanımı imkansızdı. Zamanla, LISP'in daha az bellek gereksinimi gösteren yeni tipleri gelişti ve bu arada bellek sorununa yeni çareler bulundu. Böylece LISP piyasada kullanılır hale geldi. LISP'in kullandığı veri yapısı liste türündedir. Son zamanlarda LISP'in yeni lehçeleri geliştirilmiştir (Timuroglu, 1988).

YZ uygulamalarında kullanılan diğer bir programlama dili olan PROLOG yaklaşık 18 yıl önce öncülüğünü A.Colmeraurer'in yaptığı YZ gurubu (GIA:Grouge d'intelligence Artificielle) tarafından MARSILYA'da geliştirilmiştir. Prologun kullandığı mantıksal çıkarımlarla programlama şöyle izah edilebilir.

- Bilgisayara bilgileri mantık kullanarak vermek,
- Bilgisayara problemleri mantık kullanarak aktarmak,
- Mantıksal çıkarım kuralları kullanarak, problemleri bilgisayarla çözdürmek.

Bu yaklaşımı izleyen GIA, 1973 yılında FORTRAN programlama dilini kullanarak ilk PROLOG yorumlayıcısını gerçekleştirmiştir. Bunu dünyanın çeşitli yerlerinde yazılan PROLOG yorumlayıcıları ve derleyicileri izlemiştir. Ancak 1980 yılına kadar bu çalışmalar akademik çevre içerisinde kısıtlı kalmıştır. PROLOG'un Japonların 1982'de başlattıkları "Beşinci Kuşak Bilgisayar Sistemleri" projesi için çekirdek dil olarak seçilmesinden sonra, Prolog endüstride de kullanım alanı bulmuştur (Baray,1986).

Prolog ile program yazmak için,

- Nesneler ve nesnelere arası ilişkileri anlatan gerçeklerin (Fact) tanımlanması
- Nesneler ile ilgili ve nesnelere arası ilişkileri düzenleyen kuralların tanımlanması gerekmektedir (Goldenthal,1987).

Prolog'un son zamanlarda çeşitli lehçeleri geliştirilmiştir. Bunlara örnek verecek olursak, aşağıdaki gibi bir sıralama yapabiliriz.

- CProlog                      UNIX altında çalışır.
- Quintus Prolog            UNIX ve VMS altında çalışır
- Silogic                      UNIX altında çalışır.  
Knowledge  
Workbench
- Prolog-2                    IBM PC uyumlu makinalarda DOS  
altında çalışır.
- Arity Prolog                IBM PC uyumlu makinalarda DOS  
altında çalışır.



- UNSW Prolog Bir versyonu olan **PROLOG86** MSDOS ve CP/M işletim sistemi altında IBM PC uyumlu makinalarda diğer bir versyonu olan **Chalcedony Prolog** MSDOS ve APPLE MACHINTOSH işletim sistemleri altında çalışır.
- Turbo Prolog Borland ürünü olan Turbo PROLOG MSDOS işletim sistemi altında IBM PC uyumlu makinalarda çalışır. Turbo PROLOG interpreter değil, competibildir (**Rowe, 1988**).

Çalışma sırasında Turbo Prolog 2.0 programlama dili kullanılmıştır.

YZ'nin çalışma alanlarından, uzman sistemler geliştirme çalışmalarında "kabuk" (Expert system shell) adı verilen aletler de kullanılmaktadır. Bu kabuklara örnek olarak VP-EXPERT verilebilir. Burada amaç, uzman sistemi, programlama dillerinde oluşturulacak yapılar yerine, bu yapıları kullanan kabukları kullanarak daha kolay geliştirmektir.

#### 2.4. Temel Problem Çözme Yöntemleri

Problemlerin anlaşılır olmalarını sağlamak için genelde **durum uzayı** (State Space) adı verilen gösterimler kullanılmaktadır. Durum uzayı problemin tipine bağlı olarak uygun düğümlerden oluşturulan bir şema gösterimidir (**Bratko, 1987**). Problemlerin çözümleri sırasında, seçeneklerin araştırılması çabası karmaşık yolların araştırılmasını gerektirir. Bu durumda problemin grafik olarak ifade edilmesinin çözüm kolaylığı sağlayacağı açıktır. Seçeneklerin araştırılmasında en çok kullanılan

iki yöntem vardır. Bunlar, "Derinlemesine Öncelikli Arama" (Depth-First) ve "Genişlemesine Öncelikli Arama" (Breadth-First) olarak adlandırılırlar. Bu arama yöntemleri dışında bir çok yordamsal arama yöntemi vardır. Geliştirilen program içerisinde yeri geldikçe bu arama yöntemlerinden yararlanılmaktadır. Bunlardan önemli olan bazıları aşağıda kısaca açıklanmaktadır.

Bir araştırma tekniğinin performansını değerlendirmek oldukça zor bir iştir. Buna rağmen, değerlendirmede kullanılabilen iki temel ölçüt vardır. Bunlar;

- A- Her hangi bir çözüme ulaşmadaki hızlılık,
- B- En iyi çözüme ulaşmadaki hızlılıktır (Schildt,1987).

Öyle problemler vardır ki, en az çaba ile herhangi bir çözüme ulaşmayı gerektirir. Böyle problemlerde birinci ölçütü değerlendirme ölçütü olarak ele alabiliriz. Bunun dışında öyle problemler vardır ki, eniyi çözüme ulaşılmayı gerektirir. Böyle problemlerde de ikinci ölçütü değerlendirme ölçütü olarak almak mümkündür.

Çözüm yolunun uzunluğu ve karşılaşılan durum sayısına bağlı olarak çözüm zamanı artar ya da azalır. Çözüm zamanı bu iki faktöre bağlı değişkenlik gösterir.

En iyi çözüm ile her hangi bir çözüm arasındaki farkın anlaşılması çok önemlidir. En iyi çözüm, bir problemin çözüm kümesi üzerinde amacı en iyileyecek tek bir değerdir. İyi bir çözüm ise, çözüm alanı üzerinde problemin kısıtlarını sağlayan her hangi bir nokta olabilir. Bu noktalardan her hangi biri için en iyi noktadır denilemez.

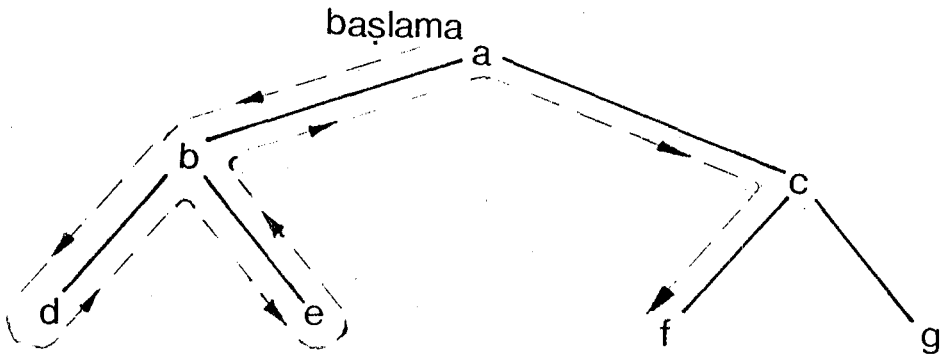
Yukarıda sözü edilen ve/veya başka çözüm tekniklerinden herhangi birisi için, her zaman diğerlerine göre üstündür denemez. Problemin yapısı ve eniyi durumun bulunacağı yola

bağlı olarak yukarıda açıklanan değerlendirme ölçütlerine göre araştırma teknikleri birbirlerine üstünlük sağlayabilirler. Bunun yanısıra bazı tekniklerin çözüme ulaşması diğerlerine göre daha olasıdır.

#### 2.4.1. Derinlemesine öncelikli arama yöntemi

Derinlemesine öncelikli arama yöntemi, çözüm uzayı üzerindeki bir kaç düğümden birini seçmesi gerektiği durumda en dipteki düğümü seçme yoluna gider. En dip düğüm başlangıç düğümüne en uzak olan düğümdür. Aşağıdaki şekil bu yöntemin anlaşılabilirliğini kolaylaştırmak için verilebilir (Bratko,1987).

Bu arama tekniğinin programlanması çok kolaydır. Kesinlik kazanmış durumların araştırılmasında çok iyi sonuçlar verir. Yöntemi kullanırken dikkat edilmesi gereken en önemli konu, düğümler arasında bir döngü oluşturmamaktır.



Şekil-2.2. Derinlemesine öncelikli araştırma

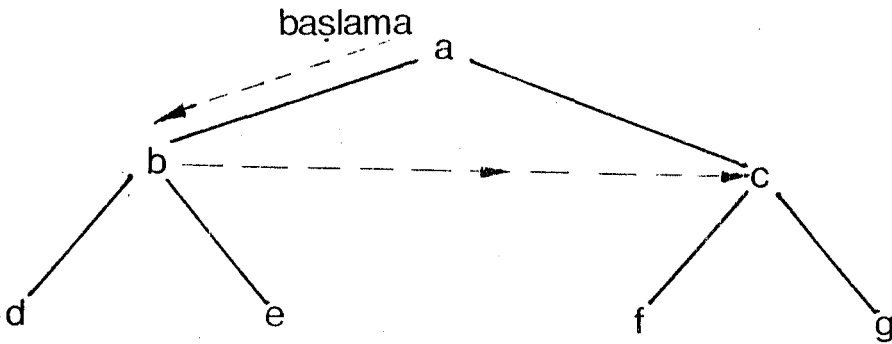
Yukarıdaki grafikte derinlemesine öncelikli arama yöntemi ABDBEBACF yolunu izleyerek çözüme ulaşacaktır. Çözüme bir başlangıç düğümden başlanmış, sonra soldan aşağı inilmiş sol yollar bittikçe aşağıdan sağa kayılmıştır. Bu

döngü, çözüm uzayında çözüme ya da son düğüme ulaşınca ya kadar devam eder. Eğer çözüm yukarıdaki şekilde D düğümü ise bu arama yönteminin çözüme ulaşma zamanı artar.

#### 2.4.2. Genişlemesine öncelikli arama yöntemi

Bu arama yöntemi, derinlemesine öncelikli arama yönteminin tam tersidir. Yöntem, başlangıç düğümü sonra başlangıç düğümüne bağlı tüm düğümleri değerlendirir. Daha sonra bir alt kademe düğümlere geçer. Bu döngü son düğüme ya da çözüme ulaşılınca ya kadar devam eder. Şekil 2.1 yöntemin anlaşılması için verilmiştir (Bratko,1987).

Şekil 2.1'den de anlaşılacağı üzere, bu yöntem derinlemesine öncelikli arama yönteminde olduğu gibi eğer amaç düğümü varsa, bu düğüme kesinlikle ulaşacaktır. Bu yöntem kolay programlanamaz. Bunun nedeni herhangi bir düğümden gidilebilecek tüm yolların araştırma sırasında bir şekilde korunmuş olmasının gerekliliğidir.



Şekil-2.2. Genişlemesine öncelikli araştırma

#### 2.4.3. Yukarı tırmanarak çözüm arama yöntemi

Bazen bir araştırma yönteminden beklentimiz en kısa yol bulma yerine en az düğüm dolaşarak çözüme ulaşmak olabilir.

Böyle bir amacı arařtırmak için kullanılan yordamsal arařtırma yöntemine yukarı tırmanma (Hill Climbing) arama yöntemi adı verilir (Schildt,1987).

Aslında, yukarı tırmanma yönteminin durum uzayı üzerindeki tüm düğümleri tarayacağı açıktır. Bu yöntem bu adın verilmesi, dağa tırmanan bir dağcının tırmanma sırasında kaybolması ile benzerliğindedir. Burada kaybolan dağcının amacı, karanlık basmadan kamp yerine ulaşmaktır. Kamp yerinin yukarılarda olduğu bilinmektedir. Bu yöntem de böyle bir düşünce taklit ederek çalışır.

#### 2.4.4. Enaz maliyetli çözümleri bulma yöntemi

Çözüm bulma işlemi sırasında bazen en kolay aşılabilir yolu bulma ile ilgileniriz. Bu amacı sağlayacak uygun çözümleri bulma tekniğine, enaz maliyet (Least-Cost) tekniği adı verilir. Maliyetin ölçütü uzunluk, para, kolaylık vb. olabilir. Yöntem düğümleri arası geçiş maliyetlerine bağlı olarak gidiş yolunu seçer. Yönteme uygulamalarda çok rastlanmaktadır (Schildt,1987).

### 3. TALAŞLI İMALATTA İŞLEM PLANLARI

Talaşlı imalat atölyelerinde, takım tezgahlarındaki işlem planlarının hazırlanması sırasında, elde edilecek planın en ekonomik kesme şartlarını sağlayacak şekilde oluşturulmasına dikkat edilmelidir. Geleneksel bir üretim işletmesinde işlem planları genellikle bir atölye mühendisi tarafından hazırlanır. Böyle bir karar verme sürecinde atölye mühendisi, hem teknik hem de ekonomik durum uzayını gözönüne alarak, uygun kesici takım, uygun tezgah ve kesme parametrelerini belirler. Ancak bu şekilde elde edilen planlar uygun planlardır. Eğer ekonomik yönden bazı ihmaller yapılırsa, yapılan işlemlerin işletmeye ne kadar kar ya da zarar getireceği bilinemez. Teknik yönden yapılan ihmaller ise ürünün yüzey kalitesi ve ölçü hassasiyeti ile doğrudan ilişkili olduğundan, ürünün kalitesinde istenmeyen değişkenlikler meydana getirebilir (Guangming, 1990). İstenenin üstünde elde edilen kalite ek işlem, dolayısıyla ek maliyet demektir. İstenenin altında gerçekleşen kalite ise, bozuk ürün maliyetini beraberinde getirir. Bunun için elde edilecek plan hem ekonomik hem de teknik yönden dengelenmiş olmalıdır. Önemli nedeniyle, bu bölümde işlem planlarının yapısı ve ekonomik yönü üzerinde durulmaktadır. Bu konu hakkında daha önce yapılan çalışmalar EK-2'de özet olarak verilmektedir.

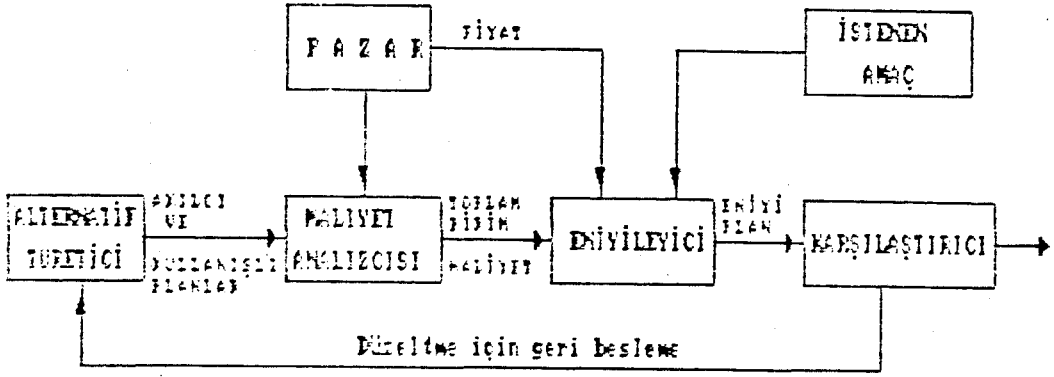
### 3.1. İşlem Planlarının Yapısı

Talaşlı imalat işlem planlarının oluşturulması sırasında, "kesme değişkenlerinin eniyi değerleri nelerdir?" sorusunu cevaplamak oldukça basit bir sorun olarak görülebilir. Ancak, elde edilecek plan, teknik yönünün yanı sıra, üretim sırasında imalat ortamının kaynaklarını en ekonomik şekilde nasıl kullanabiliriz sorusunu da cevaplamayı gerektirdiğinden, karmaşık bir yapı içerir.

Planın hazırlanmasında karar verici genellikle işe, karar değişkenlerinin uygun çözüm aralığını belirlemekle başlar. Öncelikle, ürün kalitesini sağlayacak kesme değişkenlerinin uygun aralıklarını bulur. Bu değişkenler kesme hızı, paso ve ilerlemedir. Sonra belirlenmiş uygun aralıkları sağlayan birçok seçenek plan oluşturulur. Bu planlar kullanışlı, pratik ve uygulanabilir planlardır. Verilen işlem için türetilen planlar, önceden belirlenmiş teknik amacı sağlayan planlardır.

Teknik özellikleri karşılayan bu planların içinden seçilen planın iyi bir plan olup olmadığı, yapılan işlem ile ilişkili sabit ve değişken maliyetler analiz edilerek belirlenebilir.

Böyle bir karar verme süreci Şekil-3.1 ile gösterilebilir.



**Şekil-3.1. Talaşlı imalat işlem planı oluşturma süreci**

Şekil-3.1'deki birinci eleman, imalat sisteminin teknik sınırları dahilinde planlar türetir. İkinci eleman, türetilen planların sabit ve değişken maliyet bileşenlerinin belirlenmesi ve değerlendirilmesi işlemini yapar. Üçüncü eleman, yönetim tarafından belirlenen amaçları sağlayan en iyi planın seçimini yapar. Dördüncü eleman, ekonomik yönden planın eniyiliğini test eder. Karşılaştırmacıdan seçenek üreticiye gelen geri besleme (feed-back), iyileştirme için uç kesme değişkeninde ki değişimin büyüklüğünü ve yönünü belirler. Böylece eniyileme döngüsü kapatılır. Yapıyı oluşturan tüm elemanlar parçanın talebi ve fiyatını etkileyen pazarla iç içedir (Guangming, 1990).

### 3.2. Kesme Değişkenlerinin Tanıtımı

Bir malzeme üzerinden istenmeyen bölgelerin boşaltılması için takım tezgahı ve kesicinin yapması zorunlu uç tur hareket söz konusudur:



- i- Kesici takımın kaldırılacak talaş derinliği için derinlemesine ilerletilmesi (mm).<sup>1</sup>
- ii- Kesici takımın iş parçası üzerinde talaş kaldırılacak yüzey boyunca ilerletilmesi (devir/mm).
- iii- Kesme işlemini sağlayacak dönme hareketi (devir/dk).

Bu çalışmada, tornalama işlemi ele alınıp incelenmektedir. Yukarıda açıklanan bu üç tür hareket, genellikle paso, kesme hızı ve ilerleme hızı diye adlandırılırlar.

### 3.2.1. Paso

Kesici takımın kesme kenarının, talaş kaldırılacak malzeme üzerinde gidip gelişleri sırasında, (malzemenin eksenine dik olarak) malzemeye batırılacak değere paso denir (Erol,1985). Paso bazı kaynaklarda kesme derinliği olarak da adlandırılır. Paso değeri kesicinin gidip gelişleri sırasında (tornalamada) parça üzerinde işlenmiş yüzey ile işlenmemiş yüzey arasında oluşan çap değerlerinin birbirinden çıkarılması sonucu elde edilen değer ikiye bölümü ile bulunabilir. Paso değeri;

$\alpha$  : Paso [mm]

$D_0$  : İşlem öncesi çap [mm]

$D_1$  : İşlem sonrası çap [mm]

olmak üzere,

$$\alpha = \frac{(D_0 - D_1)}{2} \dots\dots\dots(3.1)$$

---

<sup>1</sup> Matkapla delme işleminde böyle bir hareket söz konusu değildir. Yanlız büyük çaptaki deliklerin kademeli olarak delinmesi böyle bir hareket olarak düşünülebilir.

ile formüle edilebilir.

### 3.2.2. İlerleme hızı

Kesici takımın, işleme yüzeyi boyunca işlenmemiş yüzey yönünde hareketine ilerleme hızı denir (Erol,1985). İlerleme hızının değeri, tornalama ve matkapla delme işlemlerinde mm/devir, frezeleme ve planyalama da mm/dakika şeklinde ifade edilir. İlerleme hızı genelde  $f$  harfi ile gösterilir. Tornalama işlemi için ilerleme hızı;

- $n$  : Tezgah devir hızı [dev/dk]  
 $f$  : İlerleme hızı [mm/devir]  
 $f_m$  : İlerleme hızı [mm/dk]

olmak üzere,

$$f = \frac{f_m}{n} \dots\dots\dots(3.2)$$

olarak ifade edilebilir.

### 3.2.3. Kesme hızı

İş parçasının işlem göreceği yüzeyinin, kesici takımın kesme kenarına göre zaman biriminde katettiği uzunluğun m/dakika olarak ifade edilmesine kesme hızı denir. Kesme hızı aşağıdaki gibi hesaplanabilir.

- $D$ : Parçanın işlem çapı (mm)  
 $n$ : Tezgah devir hızı (dev/dk)  
 $V$ : Kesme hızı (m/dk)

$$V = \frac{\pi \cdot D \cdot n}{1000} \dots\dots\dots(3.3)$$

Boyuna tornalama işleminde kesme hızı, bir paso değeri için sabit iken alın tornalamada kesme hızı dış çapta enbüyük, merkezde ise hemen hemen sıfır değerini alır. Hesaplamalarda ise en dış çap dikkate alınır (Erol,1985).

### 3.3. Kesme Değişkenlerinin İncelenmesi

İmalat sistemlerinde talaşlı imalat da işlem planı, "tasarım özelliklerine uygun olarak bir iş parçasının işlenebilmesi için kesme koşullarının belirlenmesi" şeklinde tarif edilebilir (Guanming,1990). Bu faaliyetin temelini ise, kesme değişkenlerinin değerlerinin ayrı ayrı belirlenmesi işlemi oluşturur.

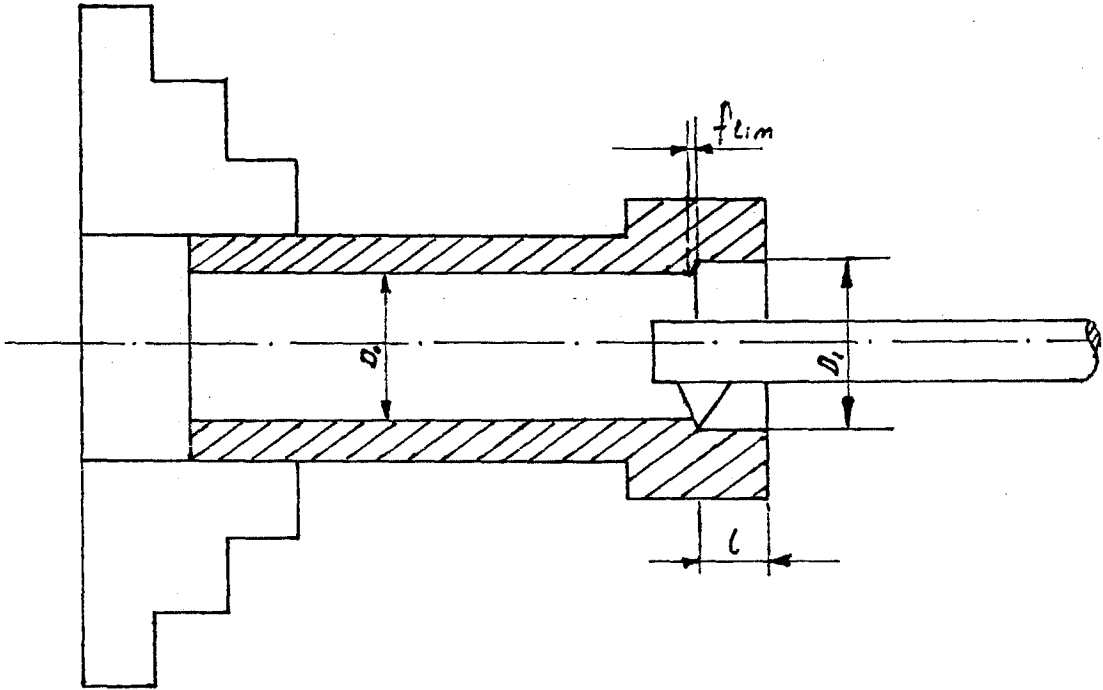
#### 3.3.1. Paso ve ilerleme

Fazla paso ve/veya ilerlemenin daha fazla güç harcanmasına, iş parçasının yüzeyinin bozulmasına, titreşimlere ve boyutsal hatalara neden olacağı açıktır. Bu durumların ortaya çıkması kesici takımın kırılmasına ve/veya çabuk eskimesine neden olabilir. Bunun yanısıra, düşük ilerleme ve paso ise, parça üzerinde daha fazla gidip gelme ve dönme demektir. Bunun sonucu olarak fazladan harcanan zaman, verimde bir azalmaya neden olur.

Parçanın yüzey kalitesi ilerleme hızı ile doğrudan ilişkilidir (Guanming,1990). Yüksek ilerleme, işlenmesi gereken yüzeyde kesilmemiş kısımlar, ve dolayısıyla yüzey kalitesinde bozulmalara neden olur. Bunun için, bu iki kesme değişkeni ile istenen yüzey kalitesi arasında ilişkiyi tanımlayabilecek bir fonksiyona ihtiyacımız vardır. Bu amaçla talaşlı imalat işlemleri için el kitaplarında bulunan mekanistik modeller kullanılabilir. Mekanistik bir model ile verilen bir ilerleme ve paso için

işlem sonunda oluşabilecek yüzey kalitesini, boyutsal hataları ve paçanın titreşimlerini belirleyebiliriz (Guanming,1990).

Şekil-3.2, bir delik işlemini göstermektedir. Kalemin zayıflığı ve titreşimin fazla olmasından dolayı paso oldukça küçük seçilmelidir. Paso en fazla 1 mm olarak verilmiştir. İlerleme hızının üst limiti ise, istenen yüzey kalitesi nedeni ile 0.2 mm/dev olarak alınmıştır. Yüzey kalitesi ise, AA-2.5 $\mu$  olarak verilmiştir.



Şekil-3.2 Bir delik büyültme işlemi

### 3.3.2. Kesme hızı

Yayınlanmış tüm kaynaklarda kesme hızına diğer kesme değişkenlerine göre daha fazla önem verildiği gözlenmektedir. Yüksek kesme hızı verim artışının yanısıra

ürünün yüzey düzgünlüğünün sağlanması açısından da önem taşır. Kesme hızı işlem zamanları, dolayısıyla verimlilik üzerinde önemli bir role sahiptir. İlerleme ve paso değerleri bilinen bir işlem için işlem zamanı aşağıda verilen formül ile hesaplanabilir (Guanming,1990).

$D_{\text{basl}}$  ve  $D_{\text{son}}$  işlemden önce ve işlemden sonraki çap değerlerini,  $l$  işlem göreceğ boyu,  $f$  devir başına ilerleme değerini,  $V$  seçilen kesme hızını ve  $d_{\text{ilm}}$  paso değerini göstermektedir. Buna göre işlem süresi,

$$IZ = \frac{\pi(D_{\text{basl}} + D_{\text{son}})}{2} \frac{1}{1000 V} \frac{(D_{\text{basl}} - D_{\text{son}})/2}{d_{\text{ilm}}} \frac{l}{f} \dots\dots\dots(3.4)$$

formülü ile bulunabilir.

Son çapı elde etmek için gerekli olan gidiş geliş sayısını  $\frac{(D_{\text{basl}} - D_{\text{son}})/2}{d_{\text{ilm}}}$  formülü ile hesaplanabilir.

Yukarıdaki formül dikkatli şekilde incelenirse, düşük işlem zamanının yüksek kesme hızları gerektirdiği görülür. Ancak kesici takımın ömrü kesme hızı ile ters ilişkili olduğundan dolayı, kesme hızı arttıkça kesici takımın ömrü azalır. Kesici takımın ömrü aşağıdaki formül yardımıyla hesaplanabilir (Guanming,1990).

$t_r$  verilen referans hızdaki takım ömrü,  $V_r$  referans alınan kesme hızı ve  $n$  takım ve işlenecek parçanın malzemesine bağlı sabit bir değer olmak üzere, kesici takım ömrü,

$$TAKOM = \left( \frac{V_r}{V_s} \right)^{1/n} t_r \dots\dots\dots(3.5)$$

formülü ile bulunabilir.

Örneğin  $n$ , kesici takım sinter karbür ve kesilen parça çelik olduğunda 0.20-0.30 arasında bir değer, kesici takım elmas, kesilen malzeme çelik olduğunda ise 0.40-0.60 arasında değer alır. (ODTU Ders Notları)

Kesici takımın değiştirilmesi için gerekli zaman, kesme hızının arttırılması ile tasarruf edilen zamana denk veya daha büyük olabilir. Daha da önemlisi kesici maliyeti, üretim maliyetinin büyük bir kısmını oluşturabilir. Dolayısıyla, kesici ömrü ile kesme hızı arasındaki ilişkinin, işlemlerin ekonomisi üzerinde etkisi olduğu rahatlıkla söylenebilir.

### 3.4. Maliyet Analizleri

Talaşlı imalatta işlem planlarını hazırlamanın, işlenmiş parçaların kalitesini istenen düzeyde tutmaktan başka, en büyük amacı; düşük maliyetler içeren tatmin edici bir verimlilik düzeyi sağlamaktır. İşlem toplam maliyeti sabit ve değişken olmak üzere iki tip maliyetten oluşur. Bu maliyet bileşenlerinin her ikisi de işlem planlarının ekonomik olarak değerlendirilmelerinde önemlidirler.

#### 3.4.1. Değişken maliyetler

Genelde bir üretim faaliyetinin değişken maliyeti miktara bağlı olarak doğru orantılı artma gösterir. Talaşlı imalatta değişken maliyet, aşağıdaki bileşenlerden oluşur.

##### 3.4.1.1. Fiili işleme maliyeti

Fiili işlem maliyeti, işçilik, tezgah kullanım maliyeti ve genel imalat giderlerinin işlem başına düşen payından oluşur. Aşağıdaki formül fiili işlem maliyetin hesaplanmasında kullanılabilir.

- FIM** : Fiili işlem maliyeti [TL]  
**SIU** : Saatlik işçi ücreti [TL/Saat]  
**SGIG** : Saatlik genel imalat gideri [TL/Saat]  
**IS** : İşlem zamanı [Dakika]

olmak üzere, fiili işlem maliyeti,

$$FIM = \left( \frac{(SIU + SGIG)}{60} \right) * IS \dots\dots\dots(3.6)$$

olarak verilebilir.

#### 3.4.1.2. Kesici takıma bağlı maliyet

Genellikle bir kesici takım birden fazla parçanın işlenmesinde kullanılabilir. Kesici takım bozulduğunda değiştirilmesi belli bir zaman alır. Bu zaman aralığında işleme ara verilir. Bu nedenle işlenen bir parça için kesici takım maliyeti, iki bileşenden oluşur. Birincisi kesici takım satın alma maliyetinin bir kısmı, ikincisi ise kesici takım değiştirme maliyetidir. Aşağıdaki formül kesici takım maliyetinin hesaplanmasını olanaklı kılar.

- KM** : Kesici takım toplam maliyeti [TL]  
**IS** : İşlem süresi [Dakika]  
**KO** : Kesici takım ömrü [Dakika]  
**KSAM** : Kesici takım satın alma mal. [TL/Adet]  
**KDZ** : Kesici değiştirme zamanı [Dakika]  
**SIG** : Saatlik işçilik gideri [TL/Saat]  
**SGIG** : Saatlik genel imalat gideri [TL/Saat]

olmak üzere, kesici takım maliyeti,

$$KM = \frac{IS}{KO} * KSAM + \frac{IS}{KO} * KDZ * (SIG + SGIG) \dots\dots\dots(3.7)$$

formülü yardımıyla hesaplanabilir.

### 3.4.1.3. Ek faaliyetlerin maliyeti

Bu maliyet bileşenini; işlenecek parçaların tezgaha yuklenmesi, boşaltılması ve işlem sırasında kesici takımın başlangıç pozisyonuna dönmesi sırasında gerekli olan sürelerden oluşan maliyetler oluşturur. Üretimin fiilen yapılamadığı bu zamanlara ilişkin maliyet;

- EFM* : Ek faaliyet maliyeti [TL]  
*SIG* : Saatlik işçilik gideri [TL/Saat]  
*SGIG* : Saatlik genel imalat gideri [TL/Saat]  
*EST* : Ek süreler toplamı [Dakika]

olmak üzere,

$$EFM = \left( \frac{SIG + SGIG}{60} \right) * EST \dots\dots\dots(3.8)$$

formülü ile hesaplanabilir.

### 3.4.1.4. Stoklara ilişkin maliyetler

İmalat sistemlerinde stok seviyesini belli bir düzeyde tutmak önemlidir. Stok dışı kalma ve depolama maliyetleri, talaşlı imalat işlemleri ile ilişkilendirilmiş değişken maliyetin önemli bileşenlerindedir. Stok maliyetini oluşturan en önemli faktörler;

i-Bir parçanın üretimi için gerekli olan süre. Bu süre fiili işlem süresi, eksüreler ve işlem başına düşen kesici takım değiştirme süresinden oluşur. Bu ifadeyi,



*US* : Üretim süresi [Dakika]  
*IS* : İşlem süresi [Dakika]  
*KO* : Kesici takım ömrü [Dakika]  
*KDS* : Takım değiştirme süre payı [Dakika/işlem]  
*EFS* : Ek faaliyetlere harcanan süre [Dakika]

olmak üzere,

$$US = IS + \left( \frac{IS}{KO} \right) * KDS + EFS \dots\dots\dots(3.9)$$

formülü ile gösterebiliriz.

ii-Talep miktarı

iii-işlem için ayrılacak tezgah kapasitesi

Aylık temelde stok tutma maliyet değerini;

*ETM* : Aylık elde tutma maliyeti [TL/Parça]  
*TK* : Toplam tezgah kapasitesi [Saat/Ay]  
*US* : Üretim süresi [Dakika]  
*AT* : Aylık talep [Adet]

olmak üzere,

$$SM = ETM * \left( \frac{TK * 60}{US} - AT \right) \dots\dots\dots(3.10)$$

formülü ile gösterebiliriz.

Formüldeki bölümlü kesim, aylık üretilebilir parça sayısını göstermektedir. Köşeli parantezin içindeki işlemlerin yapılması sonucu talep fazlası üretim miktarı

hesaplanabilir. Bu deęerin elde bulundurma maliyeti ile çarpılması sonucu toplam elde bulundurma gideri hesaplanır. Birim parça üzerine etki eden stok maliyeti;

*BSM* : Birim stok maliyeti [TL/Adet]

*SM* : Stok maliyeti [TL]

*TK* : Toplam kapasite [Saat]

*US* : Üretim süresi [Dakika]

olmak üzere,

$$BSM = \frac{SM}{(TK*60)/US} \dots\dots\dots(3.11)$$

formulu ile gösterilebilir.

Toplam deęişken maliyet ya da toplam birim deęişken maliyet yukarıdaki maliyet bileşenlerinin toplamları ile bulunabilir. Üretim düzeyini talep miktarına yakın bir deęerde tutabilmek için stok maliyetleri, olması gerektiğinden daha büyük deęerler olarak düşünülebilir.

### 3.4.2. Sabit maliyet

Gelecekte düşünölen üretim için yapılan hazırlıklardan ortaya çıkan sabit maliyetler başlıca, amortismanlar, idari harcamalar ve bakım harcamalarından oluşur. Sabit maliyet toplam maliyetin bir kısmıdır ve üretim miktarından bağımsızdır. Eğer, karar verme sırasında, pazarın talebi ile üretim miktarı arasında bir denge kurulmak istenirse, sabit maliyetin önemi ortaya çıkar. Kesme hızının alacağı deęişik deęerlerde, farklı sayıda tezgah ihtiyacı ortaya çıkar. Eğer sabit maliyet tezgah bazında belirlenmiş ise

tezgah sayısı arttıkça parça başı sabit maliyet de artma gösterir. Tezgah sayısı kesme hızına bağlı değiştirilebilir.

Sabit maliyet, hızlı bir değişim göstermez. Birim parça başı sabit maliyet, üretilcek olan ürünlerin toplam miktarının bilinmesi ile değişik kesme hızları için hesaplanabilir. Buna göre, birim sabit maliyet,

*ASM* : Aylık birim sabit maliyet [TL/Adet]

*ATSM* : Aylık tezgah sabit maliyeti [TL/Tezgah]

*TTS* : Toplam tezgah sayısı [Adet]

*BTK* : Aylık birim tezgah kapasitesi [Saat/Ay]

*US* : Üretim zamanı [Dakika]

olmak üzere,

$$ASM = \frac{ATSM * TTS}{(BTK * TTS) / US} \dots\dots\dots(3.12)$$

formülü ile hesaplanabilir.

Sonuç olarak, bir ürünün toplam birim maliyeti, talaşlı işlemlerle ilişkilendirilmiş birim değişken ve birim sabit maliyetlerin toplamı şeklinde ifade edilebileceğini söyleyebiliriz.

### 3.5. Kesme Hızından Etkilenen Faktörler

Daha önceden de söylendiği gibi kesme hızını değiştirmek, hem işleme maliyetleri hem de üretilmiş parçaların kalitesi üzerinde önemli değişikliklere neden olur.

### 3.5.1. Kesme hızının üretim süresi üzerindeki etkisi

Üretim süresi, her ikisinde kesme hızının bir fonksiyonu olan kesici takım değiştirme süresi ile işlem süresinin toplamıdır. Kesme hızının artırılması kesici takım ömrünün azalmasına, dolayısıyla daha sık kesici takım değiştirme işlemine neden olur. Her kesici takım değişiminin belli bir süre alması sonucu, toplam üretim zamanında bir artma olur. Toplam üretim süresindeki bu artışın birim üretim süresine etkisi ile, birim üretim süresi kesme hızının artmasına rağmen artış gösterir.

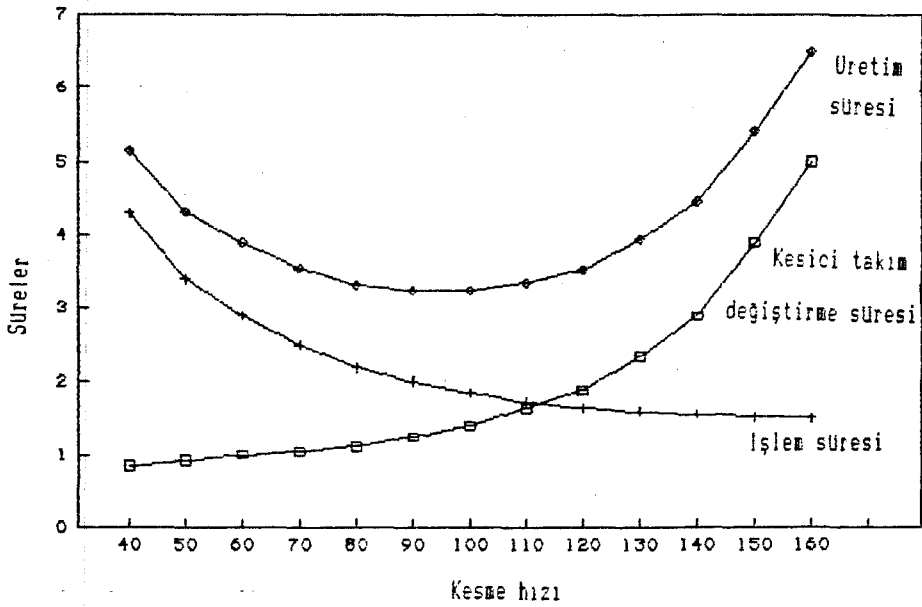
Bunun yanısıra, kesme hızının artması, birim zamanda parça üzerinden kaldırılan talaş miktarının artması anlamına geleceği için, işlem süresinde bir azalma söz konusu olacaktır.

Aslında, zaman açısından, eniyi birim üretim süresi, her ikisi de kesme hızının bir fonksiyonu olan işleme süresi ile kesici takım değiştirme süresi payı arasında bir denge noktasındadır.

Yukarıdaki açıklamalara göre, yüksek kesme hızlarında karbürden yapılmış kesici takımlar ekonomik olmadığı için tavsiye edilmezler. Burada dikkat edilecek önemli bir nokta da, parça bağlama ve sökme sırasında boşa geçen sürelerin azaltılabilmesi için işlemin tipine bağlı olarak özel hazırlanmış kolaylıkların kullanılmasının getireceği yararlarıdır. Fakat her zaman bu süreler işlem süresinin belli bir yüzdesini oluşturur. Bu çalışma sırasında belirtilen bu değer %30 olarak alınacaktır.

Şekil-3.3 kesme hızı ile üretim süresi arasındaki ilişkiyi göstermektedir.

Şekil-3.3'den görülmektedir ki, fiili işlem süresi ve kesici takım değiştirme süresi kesme hızı değişimine bağlı olarak ters orantılı değişim gösterirler.



Şekil-3.3. Kesme hızı ve üretim süresi arası ilişki

### 3.5.2. Kesme hızının toplam birim maliyet üzerindeki etkisi

Toplam birim maliyeti, sabit maliyet ile değişken maliyetlerin oluşturduğunu daha önce söylemiştik. Şimdi kesme hızı ile birim sabit ve değişken maliyetler arası ilişkileri inceleyelim.

#### 3.5.2.1. Kesme hızının birim sabit maliyet üzerindeki etkisi

Sabit maliyet temel olarak, işlemlerin yapılabilmesi için gerekli olan yatırım masraflarından oluşur. Eğer sabit maliyetin birimini TL.tezgaah/dk olarak

belirleyebilirsek, kesme hızındaki deęişimin işlem maliyetine etkisini açıkça görebiliriz. Düşük kesme hızlarında, talebi karşılayabilmek için tezgah sayısında bir artış olacaktır. Örneğin 50 m/dk kesme hızında iki adet tezgahla talebi karşılarken 45 m/dk kesme hızında tezgah sayısı uç olabilir. Veya 60 m/dk'lık kesme hızında tezgah sayısı bir olabilir. Tezgah sayısındaki bu deęişim doğrudan toplam sabit maliyeti etkileyecektir. Sabit maliyet tezgah temelinde ele alındığı için; toplam sabit maliyet tezgah sayısı ile dönemlik sabit tezgah maliyetinin çarpılması ile bulunur. Dönemlik (Aylık) tezgah sabit maliyetini 100.000 TL olarak düşünürsek, bir tezgah kullanılması durumunda 100.000 TL, iki tezgah kullanılması durumunda 200.000 TL'lik toplam sabit maliyetle karşılaşırız. Bu durumda, birim sabit maliyet, toplam sabit maliyetin üretilen parça sayısına bölümü ile elde edilebilir. Birim sabit maliyetin toplam sabit maliyetle aynı yönlü deęişim göstermesinden dolayı, düşük kesme hızlarında yüksek birim sabit maliyetle, yüksek kesme hızlarında düşük birim sabit maliyetle karşılaşılır.

### **3.5.2.2 Kesme hızının birim deęişken maliyet üzerindeki etkisi**

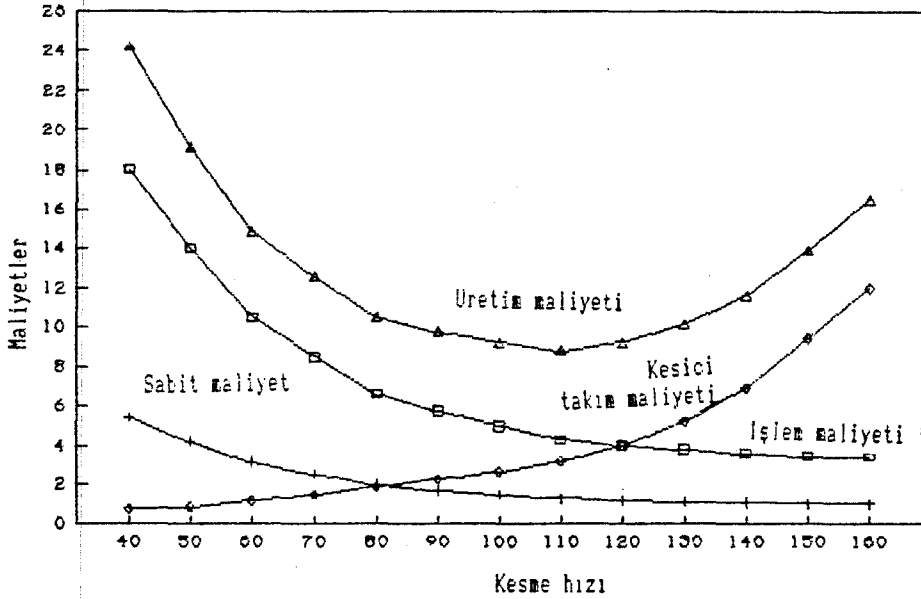
İşlemlerin yerine getirilmesi sırasında ortaya çıkan birim deęişken maliyet boşa geçen zaman maliyeti, kesici takım maliyeti ve tezgah işletim maliyeti bileşenlerinden oluşur. Kesme hızının artması işleme süresinde bir düşüş böylece, işleme maliyetinde bir azalmaya neden olur. Bunun nedeni, işlem maliyetinin işlem süresine doğrudan bağlı olmasıdır.

Fakat bunun tam tersi olarak, yüksek kesme hızlarında kesici takım deęiştirme sıklığı artacak ve dolayısıyla

toplam kesici takım deęiřtirme süresinde bir artış sözkonusu olacaktır. Fazladan harcanan bu süre, kesici takım deęiřtirme ve satınalma maliyetlerini arttırır.

Ayrıca stok maliyetlerinin de hesaba katılması gerekir. Stoklama maliyetinin toplam birim maliyetin hesabında göz önüne alınması, bulunacak çözümlerde üretilecek ürün sayısını talep miktarına yaklařmaya zorlar. Bunun nedeni ise fazladan üretilmiř her bir parçanın işletmeye dönemlik stoklama maliyeti kadar ek bir maliyet getirmesidir.

řekil-3.4, talařlı işleme iliřkin işleme maliyetlerin deęiřimini göstermektedir. řekilden de anlaşılacađı gibi, toplam birim maliyetin enkuçuklendiđi kesme hızında, toplam birim maliyetin önemli bir kısmını birim sabit maliyet oluřturmaktadır. Bu da birim sabit maliyetin işlem planları hazırlanırken dikkate alınmasının önemini gösterir. Birim sabit maliyetin ihmal edilmesi sonucu yapılan çözümlerlerin dođruluđu azalır. Diđer önemli bir nokta, sabit maliyetin yüksek kesme hızlarında hemen hemen belli bir deđerde kalmasıdır. Kesme hızının artırılmasına devam edilse bile birim sabit maliyette bir azalma olmaz. Fakat kesici takım maliyeti hızlı bir řekilde artış göstermeye başlar.

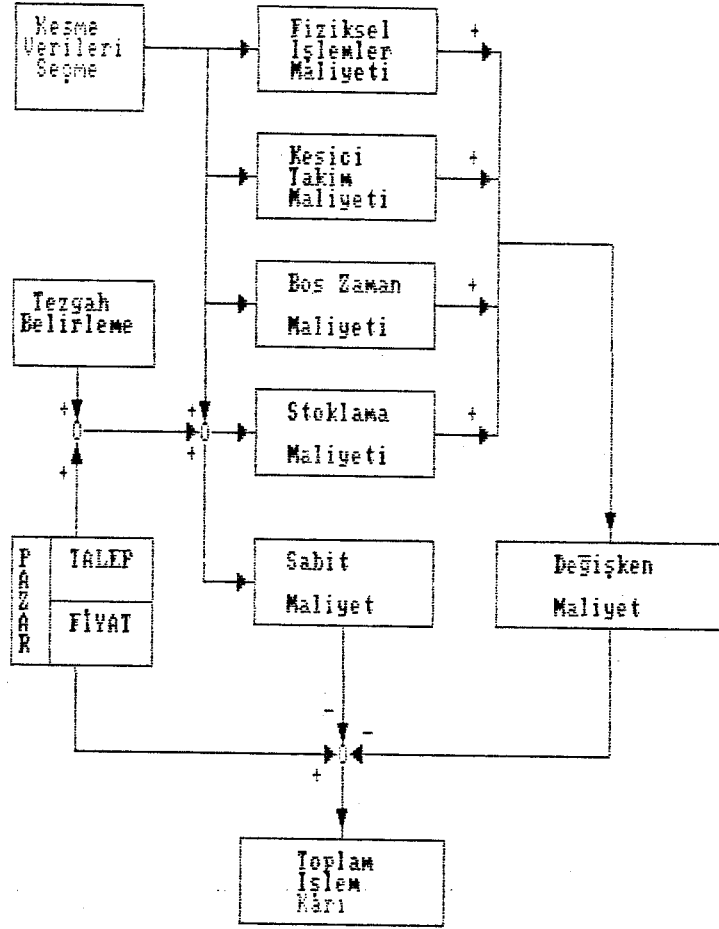


Şekil-3.4. Kesme hızı ve imalat maliyetleri arası ilişki

### 3.5.2.3. Kesme hızı ile kar arasındaki ilişki

İşletmeler için, işlemleri gerçekleştirmenin tek amacı gelecekte yapılacak üretimler için yeni sermaye oluşturmak ya da kar elde etmektir. Kar dolaylı ve doğrudan maliyetlerin toplamının satış gelirinden çıkarılması sonucu elde edilecek artı değerdir. Bunun sonucu olarak, kesme hızının işlemlerden elde edilecek kara etkisinin olduğunu söyleyebiliriz. Şekil-3.5 işlemlerden elde edilecek karın nasıl oluştuğunu göstermektedir.





Sekil-3.5 İşlemlerden elde edilecek kârın oluşumu

İşlemlerden elde edilecek kârın belirlenebilmesi, parçanın fiyat ile talebi arasındaki ilişkinin karmaşıklığından dolayı oldukça güç olabilir. Kâr fonksiyonu ürün fiyatı ve üretilen parça sayısına bağlı olarak,

- BRSF** : Birim satış fiyatı [TL/Adet]  
**TBM** : Toplam birim maliyet [TL/Adet]  
**BTK** : Birim tezgah kapasitesi [Saat/AY]  
**TTS** : Toplam tezgah sayısı [Adet]  
**US** : Birim üretim süresi [Dakika]

olmak üzere,

$$\hat{K\ddot{A}R} = \frac{(BRSF-TBM)*BTK*TTS}{US} \dots\dots\dots(3.13)$$

şeklinde verilebilir.

Genelde eniyi işlem kârı, bulunan en küçük üretim süresine karşı gelen kesme hızı değeri ile en küçük toplam birim maliyetin sağlandığı kesme hızı değerleri arasındaki bir kesme hızı değerinde elde edilebilir.

Aslında işlem kârının değerlendirilmesi çok daha karmaşık bir yapı içerir. Şekil-3.5'de gösterildiği gibi işlem kârı sadece maliyet analizleri ile ilgili değildir, yanısıra fiyat-talep ilişkisi gibi pazar analizlerini de gerektirir.

### 3.6. Sayısal Bir Uygulama

Kesme hızında yapılacak değişikliklerin, üretim süresi ve maliyet bileşenleri üzerinde önemli değişmelere neden olacağını daha önce belirtilmiş idi. Bu değişimler hakkında daha açık bilgiler elde etmek amacıyla bir örnek problem üzerinde durulmaktadır.

#### 3.6.1. Sayısal uygulamaya ilişkin bilgiler

##### İŞ

##### PARÇASI:

Malzeme : 1035 Çelik  
Başlangıç çap: 64 mm  
Bitiş çap : 60 mm  
Uzunluk : 500 mm

**KESICI****TAKIM:**

Malzeme : Karbur (TNMM-71 SANDVIK  
CROMAT)  
(T<sub>r</sub>, V<sub>r</sub>) : (50,200)  
Uç maliyeti : 300 TL/köşe  
Köşe sayısı : 6  
Sökme bağlama  
süresi : 10 Dakika

**TAKIM****TEZGAHI:**

Kapasite : 360 Saat/Ay  
Sabit maliyet: 600000 TL/Ay

**KESME****BİLGİLERİ:**

Enb. paso : 1 mm  
İlerleme : 0.1 mm/devir  
Ek süreler : İşlem süresinin % 30'u

**MALI****BİLGİLER:**

İşçilik gideri : 6000 TL/Saat  
Genel imalat gideri : 2000 TL/Saat  
Birim stok tutma maliyeti: 500 TL/Ay  
Satış fiyatı : 5500 TL/Ay  
Aylık talep : 1700 Parça

Yukarıda verilen bilgilere göre, çeşitli kesme hızı değerlerine karşı gelen süre ve maliyet bileşenleri Tablo-3.1 de özetlenmektedir.

Tablo-3.1. Kesme hızı ve üretim süresi ilişkisi

KESME HIZI (m/dk)	İŞLEM SÜRESİ (dk/par)	UÇ DEĞ. SÜRE PAYI (dk/par)	EK İŞLER SÜRESİ (dk/par)	ÜRETİM SÜRESİ (dk/par)	GEREKLİ MAKİNA SAYISI (Adet)	AYLIK ÜRETİM MİKTARI (Adet)
60	32.46	0.208	9.73	42.40	4	2038
80	24.34	0.355	7.30	31.99	3	2025
100	19.47	0.537	5.48	25.85	3	2507
120	16.23	0.750	4.87	21.85	2	1977
140	13.91	1.000	4.17	19.08	2	2264
160	12.17	1.280	3.65	17.10	2	2526
180	10.82	1.600	3.25	15.67	2	2726
200	9.74	1.948	2.92	14.60	2	2958
220	8.85	2.320	2.65	13.82	2	3125
240	8.12	2.730	2.43	13.27	2	3255
260	7.44	3.170	2.24	12.90	2	3348
280	6.95	3.630	2.08	12.66	1	1706
300	6.49	4.130	1.95	12.57	1	1718
320	6.08	4.650	1.82	12.55	1	1721
340	5.72	5.210	1.71	12.64	1	1708
360	5.41	5.800	1.63	12.84	2	3364
380	5.13	6.410	1.54	13.07	2	3305

Tablo-3.2. Kesme hızı ve toplam birim maliyet arası ilişki

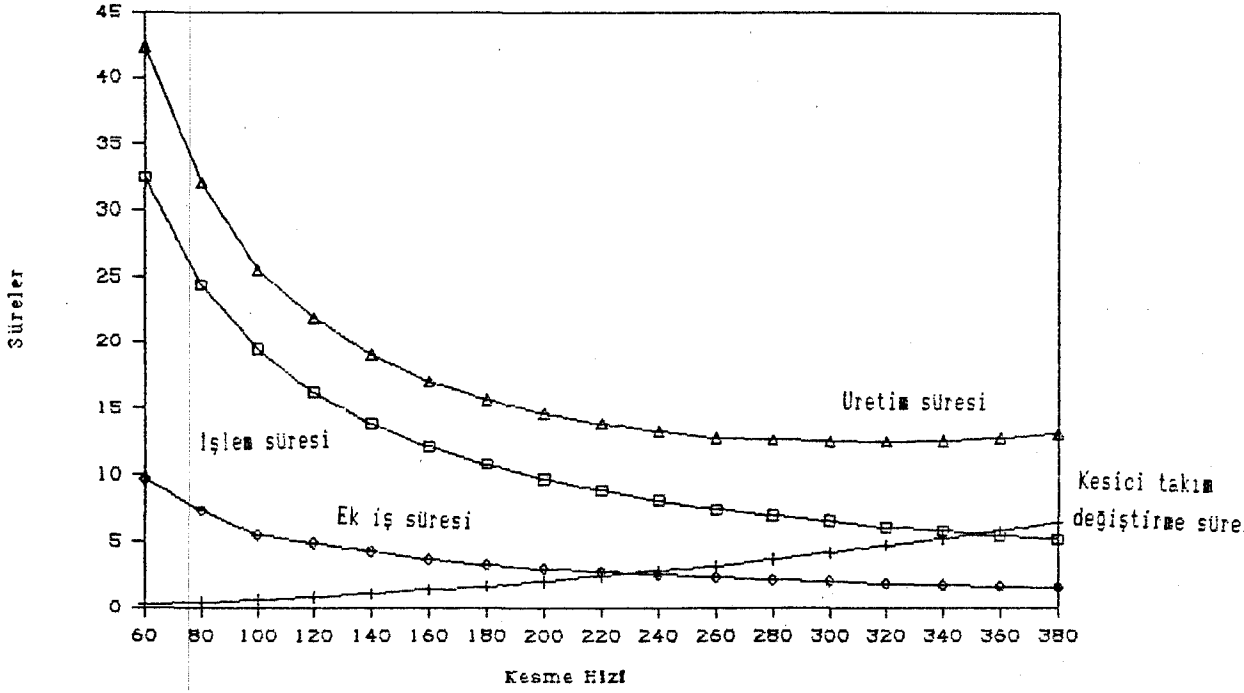
KESME HIZI (m/dk)	İŞLEME MALİ. TL/Par	EK SURE MALİ. TL/Par	KESİCİ MALİ. TL/par	STOK MALİ. TL/Par	BİRİM DEĞİŞ. MALİ. TL/Par	BİRİM SABİT MALİ. TL/Par	TOPLAM BİRİM MALİ. TL/Par
60	4328	1297	90.13	82.87	5798	1177.8	6976
80	3245	973	153.87	80.37	4452	888.6	5341
100	2596	778	232.7	160.91	3768	718.0	4486
120	2164	649	325.00	213.4	3352	607	3959
140	1854	556	433.3	249.7	3093	530	3623
160	1622	486	554.67	276	2939	475	3414
180	1442	433	613.34	294.5	2783	435	3218
200	1298	389	844.13	309	2840	406	3246
220	1180	353	1005.4	318	2857	384	3240
240	1080	324	1183	326	2915	369	3284
260	992	298	1373.6	331	2995	358	3353
280	926	277	1573	1.8	2777	352	3129
300	865	260	1790	5.63	2920	350	3270
320	810	242	2015	6.13	3073	349	3422
340	762	228	2258	2.59	3250	351	3601
360	721	217	2513	248	3699	357	4056
380	683	205	2777	243	3908	363	4271

### 3.6.2. Sayısal uygulamadan elde edilen sonuçlar

Kesme hızındaki değişimin, Üretim zamanı ve toplam birim maliyet üzerindeki etkisi Tablo-3.1 ve Tablo-3.2'nin dikkatli şekilde incelenmesi sonucu elde anlaşılabilir.

#### 3.6.2.1. Üretim süresi ve kesme hızı ilişkisi

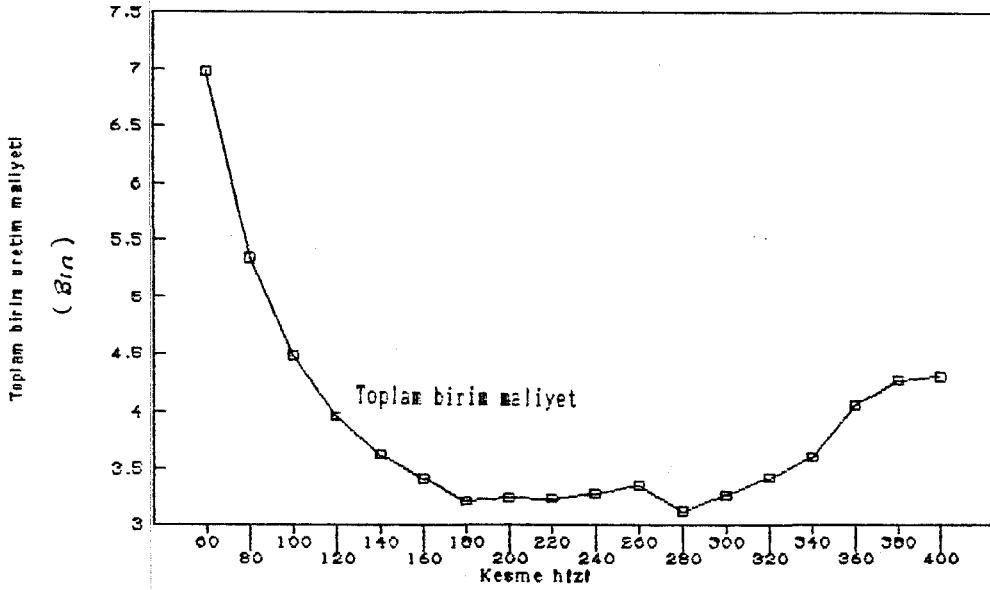
Kesme hızının 60 m/dk ile 400 m/dk arasında değiştirilmesi sırasında, en küçük üretim süresi 320 m/dk kesme hızına karşı, 12.55 dk olarak bulunmuştur. İlk beş sütündeki değerler zaman-kesme hızı eksenleri üzerinde işaretlenirse, kesme hızının artması ile işlem zamanının azalacağı görülebilir. Bununla birlikte, kesme hızındaki artış kesici takım ömrünü azaltır. Dolayısıyla kesici takım değiştirme sıklığının artmasına neden olur. Daha fazla kesici takım değiştirme daha fazla süre alacağından kesme hızının belli bir değerinden sonra üretim süresinde bir artış gözlenmektedir. Aslında en iyi üretim süresi, her ikisinde kesme hızına bağlı fonksiyonlar olan kesici takım değiştirme sıklığı ve işlem süresinin en iyi dengelendiği noktadadır. Şekil-3.6 bu durumu daha iyi açıklamaktadır.



Şekil-3.6. Kesme hızı Uretim süresi ilişkisi

### 3.6.2.2. Toplam birim maliyet ve kesme hızı ilişkisi

Tablo-3.1'in son sütunu kesme hızının 60 m/dk ile 400 m/dk arasında 20 şer br'lik artışlarla toplam birim maliyetteki değişimi gösterir. Bu sütun incelenirse en küçük maliyetin kesme hızının 280 m/dk olduğu konumda bulunduğu görülür. Toplam birim maliyet kesme hızı ilişkisi Şekil-3.7'de gösterilmektedir.



Şekil-3.7 Kesme hızı toplam birim maliyet arası ilişki

Böyle bir durumla karşılaşılmasının nedenini aşağıdaki maddelerle açıklayabiliriz.

i- Kesme hızının 280 m/dk'dan daha düşük değerleri ile 280 m/dk'ya karşı gelen değerlerinde birim sabit maliyetin aldığı değerler karşılaştırıldığında, 280 m/dk'lık kesme hızına karşı gelen birim sabit maliyetin daha düşük olduğu görülür. Bu noktada birim sabit maliyet 325 TL'dir. Birim sabit maliyet işlemler için gerekli olan yatırımlardan kaynaklanır. Tezgah sayısı sütununda gösterilen tamsayı değerler, pazarın talebini karşılamak için gerekli olan tezgah sayısını gösterir. Bu çalışmada pazarın talebi ve tezgah kapasiteleri sabit alınmıştır. Üretim süresinin artması, gerekli olan tezgah sayısının artmasına neden olur. Kesme hızı 280-300 m/dk arasında seçilirse, 1 tezgah talebi karşılamak için yeterli olacaktır. Farklı sayıda tezgah gerektiren iki ayrı kesme hızı göz önüne alalım. 260



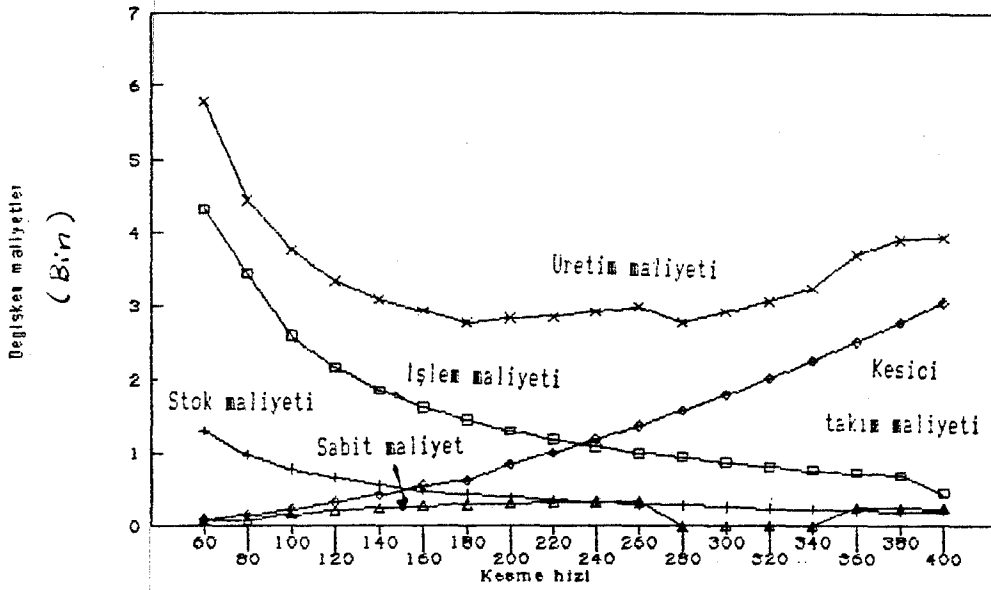
m/dk ve 280 m/dk kesme hızlarına karşı gelen aylık toplam sabit maliyeti hesaplırsak, 260 m/dk'lık kesme hızına karşı gelen aylık toplam sabit maliyet, iki tezgah gerektiğinden dolayı 1,200,000 TL, 280 m/dk kesme hızına karşı gelen aylık toplam sabit maliyet 1 tezgah gerektiğinden dolayı 600,000 TL olacaktır. Bu durum, iki farklı kesme hızında, iki farklı sabit maliyetle karşılaşılabileceğini gösterir. Tablo-3.2 dikkatli şekilde incelenirse, durum daha kolay farkedilebilir.

ii-Yine 280 m/dk kesme hızına karşı gelen birim değişken maliyetle, bu hızdan daha büyük kesme hızlarına karşı gelen birim değişken maliyet karşılaştırılırsa, 280 m/dk'lık kesme hızına karşı gelen maliyetin daha küçük olduğu görülür. Bu kesme hızına karşı gelen birim değişken maliyet 2777 TL.'dir. Kesme hızının daha yüksek değerlerine karşı gelen maliyetlere göre bu maliyetin daha küçük olmasının nedeni, bu hızdan daha büyük hızlarda kesici takım maliyetinin hızla artış göstermesindedir. Durum Tablo-3.2'den izlenebilir.

iii-Tablo-3.2'de en düşük stoklama maliyeti 280 m/dk kesme hızına karşı gelen satırda gerçekleşmiştir. Bunun nedeni ise, üretim zamanının tezgah kapasitelerine göre çok iyi belirlenmiş olmasındandır. Ayda 1706 parça üretildiğinden de, talep 1700 olduğu için 6 birim aylık temelde elde tutulur. Elde tutma maliyetinin azaltılması için, kalan bu parçalar daha sonraki dönemlerde talebi karşılamak için kullanılabilir.

Şekil-3.7 bu uygulamanın mali analizlerini grafik olarak vermektedir. Grafik üzerindeki uç eğri sabit, değişken ve toplam birim maliyetin değişimini göstermektedir. Grafik, yüksek kesme hızlarında, sabit

maliyetin azalışının hemen hemen durduğunu ve değişken maliyetin sık kesici takım değiştirme yüzünden hızla arttığını göstermektedir.



Şekil-3.6. Kesme Hızı ve Maliyetler Arası İlişki

### 3.6.2.3. Kâr ile kesme hızı arasındaki ilişki

Daha önce bildirildiği gibi, işlemleri gerçekleştirmenin enbüyük amacı, gelecekteki üretimler için yeni sermayeler oluşturmak ya da kar elde etmektir.

İşlemin kârı, sabit ve değişken maliyetlerin toplamı ile satış fiyatı arasındaki pozitif fark olarak tariflenir. Buradan, kesme hızının işlem kârını da etkilediği anlaşılmaktadır.

Kâr, üretilmiş parçaların sayısı ve ürünün fiyatı ile ilişkili olarak;

*BF* : Birim fiyat [TL/Adet]  
*TBM* : Toplam birim maliyet [TL/adet]  
*BTK* : Birim tezgah kapasitesi [Saat/Ay]  
*TS* : Tezgah sayısı [Adet]  
*US* : Birim üretim süresi [Adet/Ay]

olmak üzere,

$$K\hat{a}r = \frac{(BF - TBM) * BTK * TS}{US} \dots\dots\dots(3.14)$$

formülü ile hesaplanabilir.

Uygulamadaki parçanın 5000 TL/Adet olarak fiyatlandırıldığını düşünürsek, 260,280 ve 300 m/dk kesme hızlarında aylık kâr, sırası ile 5515535, 6384455 ve 5945585 TL. olarak bulunur.

Tanıtılan yöntemle benzer bir eniyileme yöntemi izleyen bölümde bir uzman sistem yaklaşımı çerçevesinde tartışılmaktadır.

#### 4. GELİŞTİRİLEN YUNTEM

Bu bölümde talaşlı imalat planlarında kullanılan kesme değişkenlerinin alacağı değerlerin ekonomik açıdan eniyilenmesine yönelik bir uzman sistem yaklaşımı ele alınarak incelenmiş ve yapılan çalışmalar sonucu, geliştirilen programın yapısı tanıtılmıştır.

##### 4.1. Yöntem Hakkında Genel Bilgi

İşlem bölgelerinin birbirinden bağımsız ve öncelik sıralarının bilindiği varsayımı altında, ekonomikliği sağlama açısından işlemlerin hangi tezgahlarda, kesme değişkenlerinin hangi değerlerinde ve hangi kesici takımlar kullanılarak gerçekleştirileceğinin bilinmesi istenir.

Çalışma sırasında, bu bilgileri oluşturan uzmanın kullanabileceği mantıksal yapıya benzer bir yapı oluşturulmaya çalışılmıştır. Geliştirilen yöntem hesaplamaların yapılması sırasında klasik arama yöntemlerini kullanmakla beraber, secilen malzemenin özelliklerine bağlı olarak hangi kesici takımın ve hangi tutucunun kullanılacağı, bu durumda kesici takımın omrunun ne olacağı gibi bilgileri bir uzman gibi belirler. Bu bilgileri belirlerken tüme varan bilgiler kullanılmaya çalışılmıştır. Bu amaçla SANDVIK CROMAT firmasının 8860 nolu katoloğundan yararlanılmıştır. (Sandvik,1980) Geliştirilen yapı aşağıda açıklanacaktır. Talep kısıtı altında eniyi işlem planı, toplam üretim maliyetinin

enkuçuk deęeri aldıęı kesme hızında gerçekteştięi Uçuncu Bölümde verilen sayısal örnek üzerinde gösterilmektedir. Bu noktada toplam birim maliyeti oluşturan bileşenleri; işleme maliyeti, kesici takım maliyeti, stoklama maliyeti, tezgah işletim maliyeti, tezgah sabit maliyeti, işçilik maliyeti ve kesici takım deęiştirme-tezgah yükleme boşaltma v.b. gibi işlerden kaynaklanan boş zaman maliyeti olarak verilebilir. Üretim maliyetlerini enkuçuklemek amacıyla geliştirilen program ise daha sonra tanıtılacaktır.

#### 4.2. Yapılan Varsayımlar ve Gözönüne Alınan Kısıtlar

Çalışma sırasında konunun boyutlarını daraltmak amacı ile bazı varsayımlar yapılmıştır. Ayrıca çalışmanın amacına uygun kısıtlar gözönüne alınmıştır. Varsayımlar ve kısıtlar aşağıda açıklanmaktadır.

##### **VARSAYIMLAR :**

1. Talep tam karşılanacak.
2. Talep aylık olarak bilinmektedir.
3. Talep gelecek devrelerde de olacaktır.
4. Parçaların bölgesel işlemleri birbirinden bağımsızdır.
5. İşlemlerin sırası biliniyor.
6. İşlenecek bölgenin yüzey düzgünlüğü Kesici takımın ilerleme hızına bağlıdır.
7. Kesici takım bilgileri seçilen kesici takım firması ile sınırlıdır
8. Parçanın talebi için gerekli tezgah kapasitesi, en az, seçilen tezgah gurubundaki tezgahlardan bir tanesinin ortalama devir hızındaki kapasitesinin %85'inden büyüktür.
9. Bir parçanın işlemlerinden herhangi biri bir tezgah

gurubunda yerine getirilir. Talep ayrı guruplara bölünemez.<sup>2</sup>

10. Kesici takımların tamamı tüm benzer tezgahlarda kullanılabilir.

11. Bulunacak çözümler mevcut imalat ortamının kaynaklarına bağlıdır.

12. Seçilen tezgahların tamamı bir çözüm araştırma sırasında kullanıma hazırdır. Bir parçanın herhangi bir işleminin atama yapıldığı tezgaha başka işlem atanamaz.

13. İşlemlerin işlem süreleri birbirinden bağımsızdır ve önceliği olan işlemler kendilerinden sonra gelen işlemlerin işlem sürelerini etkilemezler.

#### KISITLAR :

1. Her hangi bir çözüm aşamasında seçilen devir hızı tezgah gurubunun sahip olduğu enbüyük devir hızını geçemez.

2. Seçilen kesici takım ilerleme hızı tezgah gurubunun sahip olduğu enbüyük ilerleme hızını geçemez.

3. Seçilen ilerleme hızı tezgah gurubunun sahip olduğu enküçük ilerleme hızından daha küçük olamaz.

4. Seçilen kesici takımın uç yarıçapının %85'i kesici takım ilerleme hızından büyüktür.

5. Paso seçilen kesici takımın kesme kenarından daha küçüktür.

6. Her hangi bir çözüm aşamasında gerekli tezgah motor gücü tezgah gurubunun sahip olduğu motor gücünü aşamaz.

7. Kesici takımın ömrü bir bölgesel işlemin işlem süresinden büyüktür.

---

<sup>2</sup> Parçanın talebi kullanıcının deneyimi ile farklı guruplara bölünerek çözüm araştırılabilir.

8. Bulunacak üretim miktarları talebin iki katını aşamaz.

9. Bulunan üretim miktarı talepten büyük olmalıdır.

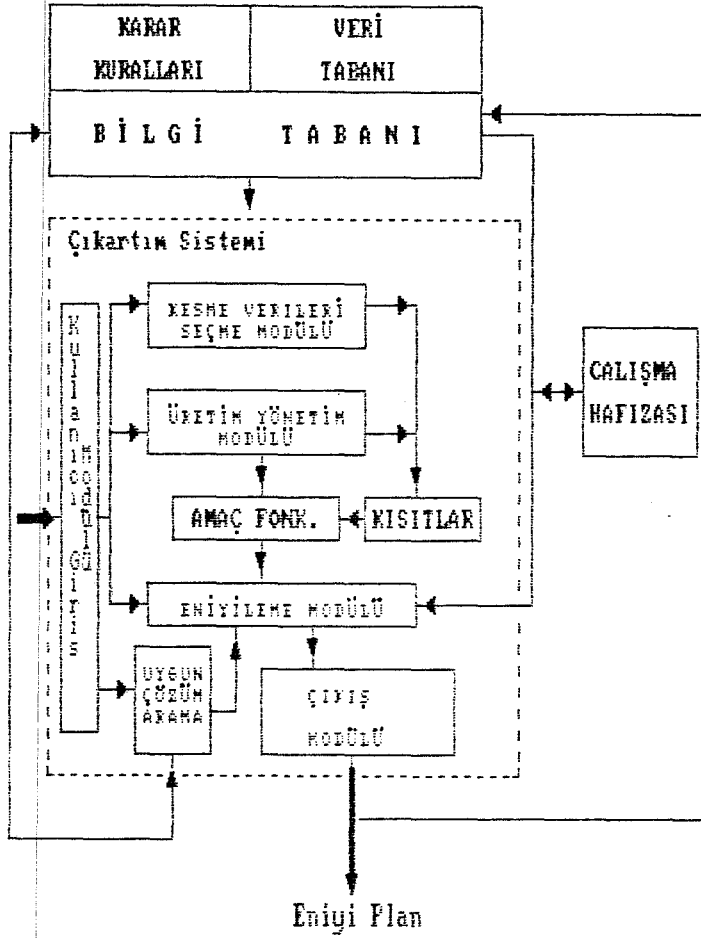
#### **AMAÇ :**

Üngörülen varsayımlar doğrultusunda ve belirtilen kısıtlar altında verilen talebi karşılayan enküçük maliyetli kesme planını oluşturan bilgilerin bulunması amaçlanmaktadır. Kesme planı başlıca kesme hızı, üretim zamanı, kesici takım ömrü, toplam birim maliyet v.b. bilgilerden oluşur.

#### **4.3. Geliştirilen Yöntemin Genel Yapısı**

Açıklamaları kolaylaştırması açısından, Bölüm 4.4'de ayrı ayrı açıklanan programı oluşturan modüller genel bir çerçevede ele alınmış ve Şekil-4.1 oluşturulmuştur.

Şekil-4.1 geliştirilen sistemin iki ana parçaya ayrıldığını gösterir. Bunlar bilgi tabanı ve çıkarım mekanizması diye adlandırılmıştır. Bilgi tabanı, geniş bir nedensel kurallar kümesi ve veri tabanından oluşur. Nedensel kurallar kümesi kesici takım, iş parçası, malzeme, tezgah vb. unsurların birbirleriyle ilişkilerinden oluşur. Bu küme öğrenme temelinde oluşturulmuştur. Yeni durumların bu kümeye eklenmesi kullanıcı giriş modülü ve programın çalıştırılması sonucu elde edilen bilgilerle mümkündür. Bu kümedeki diziler çıkartım mekanizmasını besler. Bilgi tabanı üzerinde çözümlene sonucu elde edilen bilgilerde bulunur. Çıkartım mekanizması ise, bağımsız yedi bölümden oluşur. Bu bölümler Şekil-4.1'de görülmektedir.



Şekil-4.1 Geliştirilen sistemin yapısı

Uygun çözüm arama modülü programın önceki çalıştırılmaları sırasında oluşturulmuş nedensel kurallar kümesi üzerinde benzer çözüm varlığını araştırır. Eğer çözüm varsa, bu çözüm çıkış modülüne gönderilerek kullanıcıya sunulur. Eğer giriş modülünden verilen bilgilere uygun çözüm, çözüm kümesinde bulunamazsa çözümün araştırılmasına gidilir ve gerekli yerlerde kesme verileri seçme modülü çalıştırılır.

Öncül kesme verileri seçme modülü bilgi tabanından uygun kural ve verileri bulma işlemini yerine getirir. Anlamlı ve kullanılabilir kural ve veriler bir bütün olarak ayrı bir yerde tutulur. Bazen bulunan bir bilgi programın çalışması sırasında sürekli aktif olarak tutulabilir.



Uretim yönetim modülünde ise, kullanıcı giriş modülü aracılığı ile çözümü araştırılacak işlemlere ilişkin finansal değerlere ait bilgiler bulunur. Böylece eniyileme modülüne hazırlık yapılmış olunur.

Eniyileme modülü benimsenen amaç doğrultusunda (Maliyet enküçükleme) sistemin kısıtlarını da göz önüne alarak, çıkartım mekanizması vasıtasıyla en iyi işlem planını bulmak için bir yaz boz işlemini yerine getirir. Sistemin imkanları tükeninceye kadar çözüm aramaya devam eder. İşlemler sona erdiğinde yaz boz üzerinde bulunan çözümlerlerin en iyisi kalacaktır. Bulunan bu çözüm, kullanıcıya sunulduktan sonra otomatik olarak bilgi tabanına çözüm bilgisi olarak aktarılır.

Sistemin amaç fonksiyonu, talebi karşılayacak en ekonomik işlem şartlarını bulma yönünde çalışır.

Sistemin çıktısını oluşturan en iyi plan, işlemlerin hangi tezgahlarda, hangi kesme hızlarında, hangi ilerleme, hangi kesici takım ve tutucu kullanılarak ne kadar maliyetle üretileceğini içerir. Programın oluşturduğu işlem planlarının içerdiği bilgiler, örnek bir çözümün sonucu olarak EK-4'de verilmiştir.

İlişkisel kurallara ilişkin örnekler aşağıda verilmektedir. Benzer örnekler çoğaltılabilir.

#### Örnek-1.

Parçanın malzemesi "Tavllanmış yumuşak çelik" ise,  
ve işlemin yönü "a1ın" ise,  
ve işlemin çıkış açısı "90 derece" ise,  
ve ilerleme hızı " $\leq$ " 0.4 ise,  
ve paso " $\leq$ " 1 ise,  
o zaman;

15 mm kesici kenar uzunluđu olan  
ve 0.8 burun yarı çaplı  
ve 4 kesici kenara sahip  
TNMM-71 kesici ucunu TPMR tutucusu ile  
kullanabilirsiniz.

Bu taktirde kesici takım maliyeti 3600 TL  
olacaktır.

#### Örnek-2.

Parçanın malzemesi "Tavllanmış yumuşak çelik" ise  
ömur denkleminin sağ taraf sabiti "=" 16'dır.

#### 4.4. Geliştirilen Programın İçeriđi

Geliştirilen program Turbo Prolog 2.0 programlama diliyle yazılmıştır. Turbo Prolog 2.0 büyük proje uygulamalarında modüler programlamaya izin verir. (Turbo,1989) Modüler programlama belli bir amacın birden fazla program tarafından yerine getirilmesidir. Uygulama programının geliştirilmesi sırasında ortaya çıkan bellek problemleri nedeniyle program bir kaç module ayrılmıştır. Bellek problemi işletim sisteminden kaynaklanmaktadır. Aşağıda bu modüller ve yerine getirdiđi işlevler açıklanmaktadır.

##### 4.4.1. Programın içerdđi modüller ve işlevleri

Bir bütün olarak Uzman Sistem yapısını oluşturan program modülleri aşağıdaki gibi verilebilir.

ANATAN.PRO	SABIT.PRO
CNCCOZS.PRO	SECIM2.PRO
CNCPAR.PRO	YENIUCSE.PRO
CNCTEZ.PRO	STATUS.PRO
COZUM.PRO	TPREAS.PRO

COZUM2.PRO	TDOMS.PRO
ENKUCU.PRO	COZARA.PRO
KAYITCI.PRO	LINEMENU.PRO
KULTEZ.PRO	YAZIM.PRO
MAKSAY.PRO	ENKUCU.PRO
PROJE.PRO	PROJE.PRJ

Bu program parçacıklarının herbiri ayrı bir işi yerine getirir. Bunun için her bir program parçacığı adı ile tanımlanan modüller olarak aşağıda açıklanacaktır.

#### ANATAN.PRO:

Bu modülde diğer modüllerde kullanılan genel yüklemeler, genel veri tabanları ve bu yüklemeler ve veri tabanlarının yapıları tariflenmektedir. Bu modülde tariflenen yüklemeler, veri tabanları ve yapılar diğer modüllerin herhangi birinde ihtiyaç duyulduğunda kullanılabilirler.

#### PROJE.PRO:

PROJE.PRO modülü aslında kullanıcının işlerini kolaylaştırmak amacıyla bir menü oluşturur. Kullanıcının bu menüden yaptığı seçime bağlı olarak diğer modüller çalıştırılır. Burada kullanıcının seçebileceği dört seçenek vardır. Bunlar "Tezgah", "Parça", "Çözüm" ve "Çıkış" 'dır. Kullanıcı "Tezgah" ya da "Parça" seçeneklerinden birini seçerse ilgili seçime bağlı olarak bilgi ekleme ve değiştirme işlemleri yapılır. Eğer "Çözüm" seçilirse, bir parçaya ilişkin çözüm bulma işlevi yerine getirilir.

#### TEZGAH.PRO:

Kullanıcı eğer PROJE.PRO modülünden "Tezgah" seçeneğini seçmişse bu modül çalıştırılır. Bu modül aracılığı ile

tezgahlara ilişkin bilgiler bir veri tabanına aktarılır. Ve çözüm arama sırasında bu bilgilerden yararlanılabilir. Tezgah bilgileri şunlardan oluşur;

- Tezgahın adı/kodu,
- Tezgahın aylık kapasitesi,
- Tezgahın verimi,
- Tezgah işletim maliyeti,
- Tezgah sabit maliyeti,
- Tezgaha yüklenebilir enbüyük parça çapı,
- Tezgaha yüklenebilir enbüyük parça boyu,
- Tezgah devir hızı başlama ve bitiş değerleri,
- Tezgah ilerleme hızı başlama ve bitiş değerleri,
- İlerleme ve devir hızı artış değerleri.

#### PARÇA.PRO:

Kullanıcı PROJE.PRO modulünden parça seçeneğini seçerse, parça bilgilerine ilişkin bilgilerin parça bilgileri kutuğune eklenmesi mümkün olur. Parça bilgileri aşağıdaki gibidir.

- Parçanın adı/kodu,
- Parçanın ilk aylık talebi.
- Parçanın birim stoklama maliyeti,
- Parçanın satış fiyatı,
  - Parçanın i. işleminin başlangıç çapı,
  - Parçanın i. işleminin son çapı,
  - Parçanın i. işleminin boyu,
  - Parçanın i. işleminin yüzey düzgünlüğüne bağlı olarak ilerleme hızının seçilmesi,
  - Parçanın i. işleminin için seçilen paso,
  - Parçanın i. işleminin işlem tipi,
  - Parçanın i. işleminin istenen çıkış açısı.

#### COZARA.PRO:

Eğer kullanıcı PROJE.PRO modulünde iken menüden çözüm

seceneğini seçerse, COZARA.PRO modulu aktif hale gelir. Bu modul çözümü araştırılacak parçanın adı ve bu parçanın yeni talebine ilişkin bilgileri kullanıcıdan ister. Bu bilgiler ışığında çözüm işlemi yönlendirilir. Bu modul vasıtası ile önce COZUM2.PRO modulünü çalıştırarak programın önceki çalışmaları sonucu oluşturduğu bilgi tabanı üzerinde uygun çözümün varlığı araştırılır. Eğer bu iş başarılırsa uygun çözüm üzerinde yeni düzenlemeler COZUM2.PRO yardımıyla yapılır. Ve sonuç kullanıcıya başka modüller aracılığı ile iletilir. Eğer COZUM2.PRO modulu başarılmazsa, CNCCOZS.PRO modulu çalıştırılır. Bu modulün işlevi yeri geldiğinde açıklanacaktır.

#### CNCCOZS.PRO:

Bu modul yeni çözümleri araştırır. Bu amaçla daha önce çözüm modulünden girilen parça ve talep bilgilerinden hareketle işlemler başlatılır. Bu modulde ilk aşama olarak, kullanıcıya mevcut tezgahları çözüm için kullanıp kullanmayacağını sormak amacıyla CNCKULTE.PRO modülü çalıştırılır. Bu modulün yerine getirdiği görev daha sonra anlatılacaktır. Kullanıcının seçtiği tezgahları kullanmak şartıyla çözüm işlemi başlatılır. CNCCOZS.PRO modulu içerisinde öncelikle, işlem göreceği parça ve bu parçanın işlemlerine ilişkin bilgilerin bulunduğu CNCPAR.DAT veri tabanından gerekli bilgiler alınır. Ve ilk işlemin çözümüne başlanır. Sonra, YENIUCSE.PRO modulu aracılığı ile, çözümü aranan işlem bilgilerine bağlı olarak uygun kesici takım ve tutucu belirlenir. Ayrıca tezgah bilgileri ve işparçası bilgilerinin uygunluğuna ilişkin kontroller yapılır. Buna ilişkin bir örnek, "iş parçasının enbüyük çapı kullanılacak tezgaha yüklenebilecek enbüyük çap değerinden küçük mü?" karşılaştırması şeklinde

verilebilir. Daha sonra, ENKUCU.PRO modulu aracılığı ile seçilen tezgah gurubunda bulunan tüm tezgahlar kullanıldığında talebi karşılamak için kesme hızının alacağı değer belirlenir. Ve bu hızın mümkün bir hız olup olmadığı kontrol edilir. Eğer mümkün değilse, başka bir seçilen tezgah gurubu denenir. Tüm tezgahlar denendikten sonra uygun tezgah gurubu bulunamazsa, çözümün olanaksız olduğu kullanıcıya iletilir. Eğer tezgah gurubu uygunsa, tezgah gücünün yeterliliği kontrol edilir. Bu da uygunsa, çözümü oluşturan bilgiler hesaplanır. Bu sırada başka bir takım kısıtlar da kontrol edilir. Eğer bunlardan herhangi biri sağlanamazsa, geri dönlür ve kesme hızı 10 birim arttırılır. "Kesme hızının onar birim arttırılmasının nedeni çözüme ulaşmada hızlılığı sağlamak içindir. Birbiri ardı sıra gelen onar birimlik artışlarda meydana gelen maliyet değişimi gözardı edilebilir." Bu kısıtlara bir örnek, üretim miktarının talepten küçük olmaması olarak verilebilir. Çözüm için bilgiler hesaplanırken kesici takımın ömrünün ne olacağını belirlemek için, SABIT.PRO modulu çalıştırılır. Hesaplamalar sonucu oluşan çözüm, KAYITCI.PRO modulu aracılığı ile geçici olarak tutulur. KAYITCI.PRO modulünde araştırma sonunda eniyi çözüm bulunur. Bu modulün çalışma şekli daha sonra açıklanacaktır. Kesme hızı, tezgah üzerinde mümkün en son kesme hızına kadar, onar birim arttırılarak uygun çözümler türetilir. Bu işlemler tüm tezgah gurupları üzerinde tüm işlemler bitinceye kadar yaptırılır. Sonuçta KAYITCI.PRO modulünde, eğer varsa eniyi çözümler tutulmuş olur. Bu eniyi çözümler parça üzerindeki her bir işlem için ayrı ayrı kullanıcıya ekran vasıtasıyla aktarılır. Bulunan tüm çözümler otomatik olarak çözümler bilgi tabanına

eklenir. Bu modülün çalışması sırasında çözüm oluşturulamamışsa, durum kullanıcıya bildirilir ve PROJE.PRO modülüne döner.

#### KAYITCI.PRO:

Bu modül CNCCOZS.PRO modülünün çalışması sırasında bulunan uygun çözümleri geçici olarak kara tahta görevini gören bir alana yerleştirme işini yapar. CNCCOZS.PRO modulünde araştırma işlemi devam ettikçe bulunan her uygun çözüm KAYITCI.PRO modülü tarafından değerlendirilir. Eğer kara tahta olarak tariflenen alan üzerindeki uygun çözüm bulunan son uygun çözümden daha iyi ise yeni bir araştırma CNCCOZS.PRO modülü tarafından başlatılır. Kara tahta üzerindeki uygun çözümden daha iyi bir uygun çözümle karşılaşırsa, kara tahta üzerindeki eski uygun çözüm silinir, yerine yeni uygun çözüm getirilir. Dolayısıyla araştırma işleminin sonunda kara tahta üzerinde eniyi çözümler kalmış olur.

#### SECIM2.PRO ve YENIUCSE.PRO:

Bu iki modül aslında aynı işlevi yerine getirir. Aynı ayrı tariflenmelerinin nedeni programın yazımı sırasında ortaya çıkan bellek problemlerini çözmek içindir. Bu iki modül CNCCOZS.PRO modulünde araştırma sırasında, parçanın işlemlerinin her birinin özelliklerini dikkate alarak kullanılabilen eniyi kesici takım ve tutucunun belirlenmesini işlevini yerine getirir. Kesici takım ve tutucu, işlem göreceğ bölgenin malzeme özelliği, işlem tipi, paso değeri ve ilerleme hızı değeri gözönüne alınarak belirlenir. Yukarıda sıralanan özellikleri sağlayan kesici takım ve tutucu takıma ilişkin bilgiler bu modüller aracılığı

ile belirlenir. Bu bilgileri, kesici takım maliyeti, kesici takım kenar uzunluğu, kesici takım uç yarıçapı ve kesici takım kenar sayısı olarak sıralayabiliriz.

#### COZUM2.PRO:

Bu modul bir çözüm isteği olduğunda CÜZUM.PRO modulu tarafından çalıştırılır. Bu modul çalıştırıldığında, "mevcut çözüm bilgi tabanı" nı aktif hale getirir. Çözümü istenen parçanın talebi ve önceki çözümlerde bulunan üretim miktarlarına bakar, eğer yeni talep miktarı eski çözümlerden birinin talep miktarından büyük ve eski çözüm de bulunan üretim miktarından küçükse stok maliyetleri üzerinde uygun değişiklikler yapılarak sonuç kullanıcıya ekrandan aktarılır. Yukardaki şartlara uygun bir çözüm CNCANACO.DAT bilgi tabanında yok ise COZUM.PRO modulüne geri dönlür.

#### SABIT.PRO:

Bu modulde işlem göreceğ olan parçanın malzemesine bağılı olarak kesici takım ömür denkleminin sağ taraf sabitinin alacağı değeri belirlenir. Modul. CNCCOZS.PRO modulünde araştırma yapılırken çalıştırılır. Module parçanın malzeme bilgisi gönderilir ve modulden ömür denkleminin sağ taraf sabiti değeri alınır.

#### YAZIM.PRO:

Bulunan çözümlerin ekranda görüntülenmesine yardımcı olma işlevini yerine getiren moduldür.

#### ENKUCU.PRO:

CNCCOZS.PRO modulünde, bir işleme uygun çözüm arama işlemi başlatıldığında, üzerinde çözüm aranan tezgah gurubunda bulunan tüm tezgahlar kullanılırsa, kesme hızının alacağı değeri belirlemek amacıyla bu modul oluşturulmuştur. Modulün amacı çözüm uzayını



daraltmaktır. Gerekli enkuçuk kesme hızı sıfırdan büyük olacağı için, çözüme bulunan enkuçuk kesme hızından başlanır.

STATUS.PRO, TPREDS.PRO, TDOMS.PRO, LINEMENU.PRO:

Bu programlar, programın ana menüsünü oluşturmak amacıyla Turbo Prolog Toolbox hazır programlarından alınmıştır. Bu programlar aracılığı ile programın ana menüsünün oluşturulmasında kolaylık sağlanmıştır.

#### 4.4.2. Programın içerdiği veri ve bilgi tabanları

Programın içerdiği veri ve bilgi tabanları toplu olarak aşağıdaki gibi verilebilir.

TEZGAH.DAT  
PARCA.DAT  
CNCKULTE.DAT  
CNCCOZ.DAT

TEZGAH.DAT:

Bu veri tabanında işletmede bulunan tezgahlara ilişkin bilgiler bulunur. İşletme yeni bir tezgah aldığıında bu tezgaha ilişkin bilgileri bu veri tabanına ekleyebilir. Veri tabanında bulunan bilgiler aşağıdaki gibidir:

- Tezgahın adı/kodu,
- Tezgahın aylık kapasitesi,
- Tezgahın verimi,
- Tezgah işletim maliyeti,
- Tezgah sabit maliyeti,
- Tezgaha yüklenebilir enbüyük parça çapı,
- Tezgaha yüklenebilir enbüyük parça boyu,
- Tezgah devir hızı başlama ve bitiş değerleri,
- Tezgah ilerleme hızı başlama ve bitiş değerleri,
- İlerleme ve devir hızı artış değerleri.

**PARCA.DAT:**

Bu veri tabanında çözümleri araştırılan ya da araştırılacak parça bilgileri bulunur. Yeni parça bilgileri bu veri tabanına eklenebilir. Veri tabanında bulunan bilgiler aşağıdaki gibidir:

- Parçanın adı/kodu,
- Parçanın ilk aylık talebi,
- Parçanın birim stoklama maliyeti,
- Parçanın satış fiyatı,
  - Parçanın i. işleminin başlangıç çapı,
  - Parçanın i. işleminin son çapı,
  - Parçanın i. işleminin boyu,
  - Parçanın i. işleminin yüzey düzgünlüğüne bağlı olarak kesici takım ilerleme hızı,
  - Parçanın i. işlemi için seçilen paso,
  - Parçanın i. işleminin işlem tipi,
  - Parçanın i. işleminin istenen çıkış açısı.

**CNCKULTE.DAT:**

Bu bilgi tabanındaki bilgiler, sürekli değişim gösterir. Bir çözüm araştırma işlemi sırasında çözümleri aranan parçanın işlenebileceği tezgahlar kullanıcıya seçtirilir. Seçilen tezgahlara ilişkin bilgiler bu bilgi tabanında geçici olarak saklanır. Yeni çözümler istendikçe bu bilgiler değişir. Bu bilgi tabanında saklanan bilgiler tezgah veri tabanında saklanan bilgilerle aynıdır.

**CNCCOZ.DAT:**

Bu bilgi tabanında yapılan çözümler sonucu oluşan bilgiler tutulur. Bu bilgi tabanının içerdiği bilgiler çözüm bilgileri ile aynıdır.

## 5. SONUÇ ve ÖNERİLER

Yapılan bu çalışma sırasında bilgisayarların kullanım faydasını arttırmaya yönelik çalışmalardan uzman sistemler incelenmiş, ve talaşlı imalat işlem planlarının oluşturulmasında bir uzman sistemin uygulanabilirliği gösterilmiştir.

Yapılan bu çalışmanın sonucu göstermiştir ki, talaşlı imalat işlem planlarında ekonomik analizin dikkate alınması gereklidir. Çalışma ile ayrıca, bu alanda bir uzman sistem uygulamasının getireceği yararlar tartışıldı.

Talaşlı imalat işlem planlarının geliştirilen program ile oluşturulmasının, atölye mühendisinin harcadığı süreye oranla oldukça kısa bir süre aldığı, programın çalıştırılması sırasında gözlenmiştir. Uzman Sistem, çözüm arama sırasında işlemlerin ekonomik yönünü dikkate almaktadır. Bunun için sistemin çalıştırılması sonucu elde edilecek plan, teknik ve ekonomik kısıtları sağlayan enküçük maliyet ölçütüne göre eniyi plandır.

Ayrıca geliştirilen program bulduğu çözümleri sakladığı için, uzun süre kullanıldığında, yeni çözüm isteklerini çok kısa zamanda cevaplayabilecektir.

Geliştirilen programın çalıştırılması sonucu elde edilecek işlem planlarındaki bilgiler üretim planlama ve atölye tezgah yükleme için veri niteliğindedir.

Çalışmanın bir başka yarar sağlayacak yönü de, atölyede kullanılan tezgahlar, kesici takımlar, tutucular v.b. alet ve edavata ilişkin bilgilere erişim kolaylığı sağlamasıdır.

Daha önceki bölümlerde açıklandığı gibi oluşturulan işlem planı, işlemin hangi tezgahlarda, hangi kesici takım kullanılarak, hangi kesme değişkenleri değerlerinde işleneceği vb. bilgilerden oluşur. Eğer yapılan çalışmaya zaman faktörü eklenirse, atama yapılan tezgahların boşalma zamanları, kesici takım sipariş verme zamanları vb. atölye çizelgelemeye destek olacak bilgiler elde edilebilir.

Yapılan bu çalışmanın devamı olarak, birden fazla parçanın işlenmesine aynı anda başlanacak ise, hangi parçaların çözümünün önce bulunması gerektiğini belirleme problemini çözmeye yönelik bir çalışma önerilebilir. Yani, parçalara ilişkin çözümlerin araştırılmasında izlenecek farklı sıralamaların toplam verim üzerine etkisinin varlığının analiz edilmesi gerekir.

Eğer atölye yükleme planları dönemler itibarı ile yapılmıyorsa, -iş geldiğinde planlanıyor- ve geliştirilen yöntem zaman faktörü etki ettirilmiş ise, dolu olan tezgahlara iş yükleme yapılamayacak ve dolayısıyla bir tür dinamik bir tezgah yükleme programı elde edilebilecektir.

Bu çalışma sırasında talep miktarının ve tezgah kapasitelerinin aylık olarak bilindiği varsayılmıştı. Tezgah kapasitelerinin günlük olarak hesaplanmaları mümkündür. Eğer talep miktarını da günlük olarak biliyorsak daha kısa üretim devreleri için çözüm araştırılması, geliştirilen program aracılığı ile yaptırılabilir.

## KAYNAKLAR

- Abuelnaga, A.M., Dardiry, M.A.El., 1984, Optimization metods for metal cutting, Int.Jour.Mach.Tool Des. Res., Vol.24, No.1, 11-18.p.
- Armerago, E.J.A., Russel, J.K., 1960, Maximum profit rate as a ciriterion for the selection of machining conditions, Int.J.Mach.Tool Res., Vol.6, 15-23.p.
- Baray, M., Doğaç, A., Fazlı, C., 1986 5. Kuşak bilgisayar sistemlerine genel bir bakış 3. Türkiye Bilgisayar Kongresi, 21-23.s.
- Brayko, I., 1987, Prolog Programing for Artificial Intelligence Addison-Wesley Publishing Company, 419.p, U.K.
- Boothroyd, C., Russek, P., 1976, Maximum rate of profit criteria in machining, Journal of Engineering for Industry, Trans., ASME, Series B, Vol.98, No.2, 217-220.p.
- Clarance, H.L., Goforth, R.E., 1981, Optimization of the second order logaritmik machining economics problem by extended geometric programing, AIIE Transaction, Vol.13, No.3.
- Ermer, D.S., Kromadihardjo, S., 1980, Optimization of multi-pass turning with constraints, ASME Produc., 80.p.
- Ermer, D.S., Shah, B.V., 1973 Analytical sensitivity studies of the optimum machining conditions for milling, reaming, drilling, and tapping, Journal of Engineering for Industry. Trans., ASME, Series B, Vol.95, No.1, 312-316.p.
- Ermer, D.S., 1970, A Bayesian model of machining economics for optimization by adaptive control, Journal of Engineering for Industry, Trans ASME, Series B, Vol.92, No.3, 628-632.p.
- Erol, D., 1985, Takım tezcaklarında kesme değişkenlerinin eniyi değerlerinin belirlenmesi, Anadolu U. Fen Bil. Eng., 105.s, Eskişehir (Yayımlanmamış Yüksek Lisans Tezi)
- Godethal, N., 1985 Turbo Prolog Programmer's Guide, Weber Language Series, 297.p, Chic, USA

- Guamming, Z., Stephen, G., Y., LU, 1990, An expert systems framework for economic evaluation of machining operation planing, J.Op.Res.Soc., Vol.41, No.5, 391-404.p, UK.
- Grover, M.P., 1975 Monte Carlo Simulation of the machining economics problem, Journal of Engineering for Industry, Trans., ASME, Series B, Vol.97, No.3, 931-938.p.
- Harmon, D., King, D., 1985, Experts Systems, John Wiley and Sons, inc, 279.p., USA.
- Hati, S.K., Rao, S.S., 1976 Determination of optimum machining condition-deterministic and probabilistic approaches, Journal of Engineering for Industry, Trans., ASME, Vol.98, No.1, 354-359.p.
- Hekmatpaur, S., 1989, Lisp A Portable Implementation, Prentice Hall ,176.p, UK.
- Iwata, K., and etc., 1977, Optimization of cutting condition for multi-pass operation considering probabilistic nature in machining process, Journal of Engineering for Industry, Trans., ASME, Series B, Vol.99, No.1, 210-217.p.
- Iwata, K., Murotsu, Y., 1972, Probabilistic approach to the optimum cutting condition, Journal of Engineering for Industry, Trans., ASME, Series B, Vol.94, No.4, 1089-1107.p.
- Kılıç, S.E., 1983, Optimization of cutting conditions for multi-pass turning operations, Journal of Pure and Applied Sciences, Vol.15, No.1, 21-39.p.
- Lambert, D.K., Walvekar, A.G., 1978, Optimization of multi-pass machining operation. Int.Journal of Production Research, Vol.16, No.4, 259-265.p.
- Malokoti, B., Devprasad, J., 1989, An interactive multiple criteria approach for parameter selection in metal cutting, Operation Research, 37.vol, 805-811.p, USA.
- Nilson, N., J., 1980, Prencipial of Artificial Intelligence, Tiago Publishing Company, 471.p, USA.
- Nişli, M.S., 1984, Optimization of single pass turning operations by genetic programming, M.Sc.Thesis, METU.
- Rich, K., Robinson, P.R., 1988, Using Turbo Prolog, McGraw-Hill, 253.p.

KAYNAKLAR (Devam)

- Rowe, N., C., 1988 Artificial Intelligence Through Prolog, Prentice Hall International Inc, 443.p, USA.
- Schildt, H., 1987, Advanced Turbo Prolog, McGraw-Hill, 300.p, USA.
- Dağlı, C., H., 1986, Uzman sistemlerin üretim planlamadaki yeri, II. Ulusal Makina Tasarım ve İmalat Kongresi, ODTU, Ankara.
- Solomentsov, Y. M., 1971, Optimizing the machining technology on batch production, Russian Engineering Journal, Vol.56, No.2, 55-58.p.
- Solomentsov, Y. M., 1971, Optimizing the machining of component using adaptive control systems, Russian Engineering Journal, Vol.51, No.9, 73-77.p.
- Timuroğlu, F., 1988, Yapay zeka ve uzman sistemler, 5. Türkiye Bilgisayar Kongresi Kitabı, 27-39.s.
- Tipnis, V. A., Friedman, M. Y., 1976, Cutting rate tool life characteristic function for material removal process, Journal of Engineering for Industry, Trans., ASME, Vol.98, No.2, 487-496.p.
- Tuzun, M. S., 1981, Optimization of cutting conditions for turning operations using the minimum cost criterion, M.Sc.Thesis, ODTU,
- Turban, E., 1990, Decision Support and Expert Systems Macmillian Publishing Company, 843.p, Newyork.
- Ulengin, B., 1987 Yapay zeka ve uzman sistemler XI. Ulusal Yıllık Araştırma Kongresi Bildiri kitabı, Vİİİ.h, 32-35.s.
- Williamson, A., 1986, Artificial Intelligence for Microcomputers, 171.p., Newyork.
- Wu, S. M., Ermer, D. S., 1966, Maximum profit as the criterion of the optimum cutting conditions, Journal of Engineering for Industry, Trans ASME, Series B, Vol.88, No.3, 435-443.p.
- , 1987 Yapay Zeka, Bilgisayar Ansiklopedisi, 7-8.c, 1681-1780.s.
- , 1989, Ortadoğu Teknik Üniversitesi Makina Elemanları ve Takım Tegahları ders notları, ODTU, Ank.
- , 1980, Choosing Turning Tools and Cutting Data, Sköve Offset, 60.p, Sweden.

<u>YIL</u>	<u>OLAY</u>
1642	Pascal'ın geliştirdiği hesap makinası.
1672	Leibniz'in geliştirdiği hesap makinası.
1832	Babage'ın geliştirdiği hesap makinası.
1847	Boole cebirinin Boole tarafından bulunması.
1937	Shannon'un Boole cebirinin elektronik devrelerin tarifinde kullanılabileceğini göstermesi.
1944	Pensilvanya Üniversitesinde ENIAC'ın geliştirilmesi.
1948	Winner'ın sibernetik hakkındaki yayını.
1950	Alan TURING'ın YZ konusu üzerine "Computing Machinery and Intelligence" adlı bilimsel makalesinin yayınlanması. Turing bu makalede insan zekasının benzetiminin yapılabileceğini savunuyordu.
1955	IPL-II (Information Processing Language II) adlı ilk A.I dili Allen Newel, J.C.Shaw ve Herbert Simon tarafından oluşturuldu. IPL bir liste işlemcisidir.
1955	McCarty LISP'i (List Programmig) geliştirdi.
1956	Dortmouth da Y.Z. hakkında yaz konferansları John McCarty, Marwin Minsky, Nathaniel Rochester ve Claude Shannon tarafından düzenlendi. Newell, Shaw ve Simon tarafından geliştirilen Logic Theorist (LT) bu konferansların birinde tartışıldı. Bu ilk A.I. programı Whitehead ve Russel'in Principia Mathematica adlı kitabındaki teoremleri ispatlayabiliyordu.
1957	Newell, Shaw ve Simon Genel Proble Çözer'i (General Problem Solver GPS) geliştirmeye başladılar.
1959	Arthur Samuel'in IEM Araştırma ve Geliştirme Dergisinde (IEM Journal of Research and Development) "Some Studies in Machine Learning the Using the Game of Checkers" adlı makalesi yayınlandı.
1960	Marvin Minsky ve Alan Turing yönetiminde Y.Z. üzerinde çalışmalara başlandı.
1963	Edward A.Figenbaum ve Julian Feldman "Bilgisayar ve Düşünce" adlı bir makale yayınladılar.
1965	Kimyasal bileşiklerin yapısını analiz eden ilk UZMAN SISTEM, DENDRAL, Stanford Üniversitesinde Joshua Lederberg, Edward A.Figenbaum, Bruce G.Buchanan, Dennis Smith ve Carl Djerassi tarafından geliştirildi.



- 1966 Joseph Weizenbaum ELIZA'yı geliřtirdi. ELIZA psikiyolojik rahatsızlıkları olan hastalarla karşılıklı diyalog kurarak tedavi amacı ile geliřtirilmiřtir.
- 1968 Marwin Minsky'nin "Semantik Information Processing" adlı kitabı yayınlandı.
- 1970 İlk mikro iřlemci INTEL 4004'un uretilmesi.
- 1970 Patrick H. Winston'un doktora çalıřması "Learning Structural Descriptions from Examples" yayınlandı.
- 1970 Alan Colmerauer ve arkadaşları PROLOG (Programming Logic) programlama dilini geliřtirme çalıřmalarına bařladılar.
- 1971 MACSYMA ilk olarak kullanıldı. MACSYMA türev ve integral hesapları yapıyor ve sembolik karşılařtırmalara izin veriyordu. Girdi ve çıktıyı sembolik ifadelerden oluřan bu program bir bilgi tabanına sahipti. Sistem Martin Moses ve Carl Engleman tarafından geliřtirildi.
- 1972 Colmerauer 'in ilk PROLOG derleyicisini tamamladı.
- 1973 Stanford Üniversitesinde SUMEX-AIM (Stanford University Medical Experimental Computer Project Artificial Intelligence in Medicine) çalıřmaları bařladı.
- 1973 Roger C. Schank ve Kenneth M. Colby tarafından "Computer Models of Thought and Language" adlı kitap yayınlandı.
- 1973 R.O. Duda ve P.E. Hart tarafından "Pattern Classification and Scene Analysis" adlı kitap yayınlandı.
- 1975 MYCIN'in ilk řekli SUMEX-AIM projesi dahilinde çalıřan Edward H. Shortliffe tarafından oluřturuldu. MYCIN menenjit ve diđer bakteriyel hastalıkların teřhisinde kullanılan bir uzman sistemdir.
- 1975 Patrick Winston'un "The Psychology of Computer Vision" adlı kitabı, Marwin Minsky'nin "A Framework for Representing Knowledge" adlı kitabı yayınlandı.
- 1975 Daniel G. Bobrow ve Allan Collins'in "Representation and Understanding" adlı kitabı yayınlandı. Bu kitap bilginin gösterilmesi konusunda önemli yayınlardan biridir.
- 1977 Budapeřte de Macaristan Bilgisayar Koordinasyon Enstitüsünde PROLOG dili ile kullanılabilir uzman sistemler geliřtirildi.

- 1978 R.O.Duda Stanford Ulusal Araştırma Enstitüsünün de PROSPECTOR hakkında bir makale yayınladı. PROSPECTOR jeolojik araştırmalara ilişkin bilgileri analiz etmeye yarayan bir uzman sistemdir.
- 1980 Digital Equipment Corporation adlı şirkette XCON adlı uzman sistem ilk olarak ticari amaçlı olarak uygulanmaya başlandı. XCON'un prototipi Carnegre-Melon Üniversitesinden John McDermont denetiminde oluşturuldu.
- 1981 "The Handbook of Artificial Intelligence" adlı kitabın ilk bölümü Avron Barr tarafından yayınlandı. Bu kitap üç bölümden oluşuyordu. Diğer iki bölümü izleyen yıllarda yayınlandı.
- 1981 Japonlar Beşinci Kuşak Bilgisayarlar Projesi adlı bir proje başlatacaklarını açıkladılar.
- 1982 PROSPECTOR ticari amaçla uygulamaya konuldu.
- 1982 Japonlar beşinci kuşak bilgisayarlar projesini TOKYO da başlattılar.
- 1982 Japonların beşinci kuşak bilgisayarlar projesine karşı ABD de "Bilgisayar ve Microelektronik şirketi (MMC) oluşturuldu.
- 1982 ABD de MMC altında "Alvey Program of Advanced Information Technology" adlı çalışma başlatıldı.
- 1983 Avrupa Ekonomik Topluluğu ESPRIT adı altında beşinci kuşak bilgisayarlar geliştirme yarışına katıldı.
- 1983 MMC çalışmalara başlamak amacıyla Austin, TEKSAS da açılış yaptı.
- 1983 Zeki makineler konusuna katkıda bulunmak amacıyla İskoçya, Edinburg da "Strathclyde Üniversitesinde Turing Enstitüsü açıldı.
- 1985 Kişisel bilgisayarlar da hafıza ve hız yönlü gelişmeler.

<u>YAZAR</u>	<u>YAKLAŞIM</u>	<u>YIL</u>
E.J.ARMEREGO J.K.RUSSELL	Karı enbüyükleyecek biçimde kesme değişkenlerinin seçilmesine ilişkin bir yöntem önermişlerdir. Bu yöntemde göre, pasoyu önceden seçerek ilerleme ve kesme hızlarının en iyi değeri kar enbüyüklemesi amacına göre saptanabilir (Armerego and Russell,1960).	1960
S.M.WU D.S.ERMER	Enbüyük kara karşı gelen kesme hızını saptama yönlü bir metod önermişlerdir. Bu yöntemde göre enbüyük kar marjinal gelirin marjinal maliyete eşit olduğu üretim düzeyinde gerçekleşir. Üretim düzeyinden kaynaklanan gelirler bilinebiliyorsa kesme hızı bir $V_{max}$ 'dan itibaren gelirdeki marjinal artış maliyetteki marjinal artışa denk olana kadar arttırılır. Bu yöntemde bir kesme hızı aramak yerine aralık aramak daha akıllıcadır (Wu and Ermer,1966).	1966
D.S.ERMER	Uyarlamalı denetim (Adaptiv Control) esasına dayanan bir eniyileme yöntemidir. Uyarlamalı denetim sistemleri sürecin istenen en iyi değerlerini süreç esnasında anında bulur. Bu yöntemde göre, kesicinin kesme kenarında gözlenen aşınma miktarına göre, kesme değişkenleri anında değiştirilir. Bu yaklaşımda, zaman ve maliyet verilerinin bilinebilen değerleri ile takım ömrü parametrelerinin kestirim değerleri birleştirilerek en küçük maliyete karşı gelen bir ilk kesme hızı belirlenebilir. Üretimin devamı esnasında takım ömründe gözlenen değerlere Bayesian modelinin uygulanması ile yeni takım ömrü parametreleri bulunur, ve yeni kesme hızı değeri hesaplanır. Bu işleme parametre değerleri arasında önceden belirlenen fark oluşuncaya kadar devam edilir (Ermer,1970).	1970
YU.M. SOLOMETSOV	YU.M.SOLAMETSOV'a göre parçaları sadece gerekli duyarlılıkta değil, aynı zamanda enbüyük üretkenlikte ve enküçük maliyetle üretmek gerekir. Önerilen yöntemin aşamaları aşağıdaki gibidir. a-İş parçasının tezçaha bağlanma ve ayarlanma yöntemi seçilir. b-Kesici takım geometrisi seçilir. c-İlerleme hızı, istenen yüzey kalitesini sağlayacak şekilde seçilir. d-Kesme hızı üretimde değişik faktörleri göz önüne alarak seçilir.	1971

- Seçilen kesme değişkenleri sürecin kısıtlarına göre kontrol edilir. Kısıtların tamamı sağlanıncaya kadar kesme koşulları değiştirilir (Solometsow, 1971)
- K.IWATA  
Y.MUROTSU  
T.IWATSUBA  
S.FUJII
- Rassal programlama kavramına dayalı analitik bir yöntem önermişlerdir. Bu yöntemde amaç fonksiyonunun ve kısıtların rassal karakterlerini göz önüne alarak kesme değişkenlerinin en iyi değerleri saptanır. Çalışmanın amacı, birim takım aşınma miktarı için talaş debisi ve üretim maliyetlerini en iyileyecek kesme değişkenleri saptamaktır (Iwata, 1972) 1972
- D.S.ERMER  
B.S.SHAH
- Çalışmada kesme değişkenlerinin en iyi değerlerinin belirlenmesinde duyarlılık analizleri için en küçük maliyet, en büyük üretim düzeyi ölçütüne göre bir analitik yöntem önerilmiştir. Çalışmanın ana teması, bir en iyi nokta yerine en iyi noktalardan oluşan, en iyi kesme değişkenleri aralığını belirleme yönündedir. Bu aralıkta en iyi değerlerin aranmasının duyarlılık açısından daha faydalı olacağı ileri sürülmüştür (Ermer and Shah, 1973). 1973
- M.P.GROOVER
- Monte Carlo benzetim tekniği esasına dayalı bir en iyileme metodu önermiştir. Kullanılan ölçüt, birim paça maliyetinin en küçüklenmesidir. Takım ömrü ve takım aşınması değişken varsayılarak, kısıt olarak yüzey pürüzlülüğünün belli bir düzeyde garantilenmesi temel amaç olarak alınmıştır. Bu çalışmada sabit ilerleme hızı değerine karşı en iyi kesme hızının belirlenmesi problemi hem geleneksel hem de benzetim tekniği kullanılarak çözülmüştür (Groover, 1973). 1973
- S.K.HATI  
S.S.RAO
- Klasik ve rassal yaklaşımın her ikisini de kullanarak kesme değişkenlerinin en iyi değerlerinin belirlenmesi probleminde matematiksel programlama tekniklerini uygulamışlardır (Hati and Rao, 1976) 1976
- YU.M.  
SOLOMETSOW
- Kitle üretimi teknolojisinde kesme değişkenlerinin en iyilenmesi ile ilgili bir çalışmadır (Solometsow, 1976). 1976
- M.Y.FRIEDMAN  
V.A.TIPNIS
- Araştırmacılar G-T karakteristik fonksiyonları diye adlandırılan yeni bir kavram yardımıyla probleme yaklaşmışlardır. Bu araştırmaya göre talaşlı üretim süreçlerinde iki temel parametre önemlidir. Bunlar talaş debisi (Kesme hızı) ve kesici kenar ömrüdür. Kesici kenarın ömrünün ekonomik değerleri için kesme 1976

- hızları ve ilerleme hızlarının değişim aralığını belirlemek için sabit takım ömrü ve sabit pasoya karşı gelen noktalar f-v düzleminde ve kendi aralarında birleştirilmek suretiyle talaş debisi-takım ömrü diyagramları elde edilir. Önerilen yöntem gere bu diyagramlar kullanılır. Çalışmaya göre herhangi bir işlemin karakteristik eğrisi hazırlanmış ise en iyi f ve v değerleri bulunabilir (Friedman and Tipnis,1976).
- G.BOOHROYD  
P.RUSEK Eniyileme ölçütü olarak, kar düzeyi (oran) 1976 alınmış olan bu çalışmada amaç, en iyi f ve v'yi belirlemek olmuştur (Boothroyd and Rusek,1976).
- K.IWATA  
Y.MUROTSU  
F.OBA Çalışmalarında araştırmacılar, çok pasolu 1977 işlemlerde kesme değişkenlerinin en iyilenmesi problemi ile ilgilenmişlerdir, işledikleri çözüm yöntemi dinamik programlama yaklaşımıdır (Iwata,1977)
- B.K.LAMBERT  
A.G.WALVEKAR Bu çalışmada, kesme değişkenlerinin en iyi 1978 değerini belirlemede analitik ve bilgisayar destekli arama yöntemleri kullanılmıştır (Lambert and Walvekar,1978).
- A.M.EI AWAN B.K.LAMBERT ve A.G.WALVEKAR'ın çalışmalarına 1979 devam etmiştir (Awan,1979).
- C.L.HOUGH  
E.E.GOFORTH Probleme geometrik programlama tekniği ile 1980 yaklaşmışlardır (Hough and Goforth,1980).
- D.S.ERMER  
S.  
KROMODIHADJO Çok pasolu işlemlerin en iyilenmesi problemi 1980 üzerinde çalışmışlardır (Ermer and Kromodihadjo,1980).
- M.S.TUZUN En küçük maliyet ölçütüne göre tornalamada 1981 kesme değişkenleri en iyilenmesi konulu bir çalışma yapmıştır. Çözüm tekniği olarak, bilgisayar destekli dinamik programlamayı kullanmıştır (Tuzun,1981).
- M.A.EI HAKIM  
A.M.EI AVAM  
A.S.EI SABAH Tornalamada kesme değişkenlerini en iyilemek 1981 amacıyla bir bilgisayar programı hazırlanmıştır. Fortran programlama dili ile yazılan programca izlenen yöntem gere, paso değeri ilkin alabileceği en yüksek değeri almakta sonra kademe olarak kısıtları sağlayacak değere indirilmektedir (Hakim, Awan, Sabah,1981).
- S.E.KILIÇ Çok pasolu tornalamada kesme değişkenlerinin en 1982 iyilenmesi amacıyla hazırlanmış bu çalışmada, aynı zamanda çok işler de gün ömrüne alınmıştır. Bu çalışmada benimsenen ölçüt maliyet en

kuçuklemesi olup; amaca uygun kesme deęişkenleri bilgisayar destekli dinamik programlama ile belirlenmiştir (Kılıç,1982).

- M.S.NIŞLI Çalışmada geometrik programlama teknięi 1984 kullanılarak, tek pasolu tornalama işleminde ilerleme ve kesme hızının en iyi deęerleri, en kuçuk maliyet ölçütüne göre belirlenir. Bilgisayar desteęinden yararlanılmıştır (Nişli,1984).
- D.EROL Yazar kesme deęişkenlerinin eniyi deęerlerini 1985 belirlemede sayımlama yöntemini kullanan bir eniyileme yöntemi geliştirmiştir. Yöntem, tezgahlar üzerinde sonlu sayıda var olan devir hızı ve ilerleme hızı deęerlerinin tamamına karşı gelen amaç fonksiyonu deęerlerini karşılaştırma şeklinde çalışır. Çalışmada, amacı en iyi saęlayan kesme deęişkenleri bir bilgisayar programı aracılığı ile bulunur (Erol,1985).
- B.MALAKGOTI Bu çalışmada çoklu amaç fonksiyonu altında 1987 J.DEVİPRASAO tornalamada kesme problemine çözüm amaçlanmıştır. Amaçlar birim parça üretim maliyetini, üretim zamanını ve yüzey kalitesini eniyilemektir. Çalışmada GRADIENT çoklu karar verme temelli bir heruistik, parametrelerin seçiminde kullanılmak üzere geliştirilmiştir. Kesme parametrelerinin seçimi amacıyla kişisel bilgisayarlarda kullanılabilecek bir yazılım paketi oluşturulmuştur. Bu paketin adı CAMPS'dır. (Computer aided machine parameter selection) (Malakgoti and Deviprasao,1987).
- I.YELLOWLEY Bu çalışmada kalem ömrü denkleminin 1989 E.A.GUNN kullanılması ile en iyi kesme büyüklüğünün frezeleme ve tornalama için bulunabileceęi ileri sürülmüştür (Yellowley and Gunn,1989).
- G.ZHANG Bu makalede işlen planına ekonomik bakış adı 1990 S.C.Y.LU altında bir çalışma sunulmuştur. Önerilen metod işlen sıralarının bir sıralamasını ve bu sıralama içerisinde en iyi planı oluşturur. Önerilen plan hem teknik hem de ekonomik kısıtları karşılar (Zhang and Lu,1990).

```

*****
*   Programda kullanılan genel yapıları   *
*   tarifleyen program                     *
*****

global domains
  son_bil=son(symbol,real,integer,symbol,symbol,symbol,real,real,
              real,real,real,real,real,real,integer,real,real,
              real,real,real,real,real)
  tez_bil=tez(symbol,real,real,real,real,real,integer,
              real,real,real,real,real,real,real,real)
  ss=symbol*
  rr=real*
  parbil=par(symbol,integer,integer,real,real,
              real,ss,rr,rr,rr,rr,rr,rr)

global predicates
  omur(integer,integer)-(i,o).
  kul_tezgah .
  maksay(integer,real,real,real)-(o,i,i,i).
  tablo.
  pkaydet.
  tezkaydet.
  cozara.
  bakara(integer,symbol,real,real,real,symbol,symbol,
          integer,integer,integer,real)-(i,i,i,i,i,o,o,o,o,o).
  cozumara1(symbol,real)-(i,i).
  cozumara2(symbol,real)-(i,i).
  kaydet(symbol,real,integer,symbol,symbol,symbol,real,real,
          real,real,real,real,real,integer,real,real,real
          ,real,real,real,real)
          (i,i,i,i,i,i,i,i,i,i,i,i,i,i,i,i,i,i,i,i,i,i)
  enk(real,real,integer,real,real,real,real,real,real,real)-
          (i,i,i,i,i,i,i,i,o).
  secir2(symbol,real,real,real,symbol,symbol,integer,integer,
          integer,real)-(i,i,i,i,o,c,c,o,o,o).

global database-parca
  parca(parbil).
global database-cozum
  cozun(son_bil).
global database-coz
  coz(son_bil).
global database-kultez
  kultez(tez_bil).
global database-tezgahlar
  tezgahlar(tez_bil).

```

```

*****
*Çözümlerin araştırılmasını yapan program *
*
*****

project"proje"
include"anatan.pro"

domains
sir=k(integer).

constants
pi=22/7

predicates
parcaci(symbol, integer).
irdele(integer, symbol, integer, real, real, real, ss, rr, rr, rr, rr, rr, rr).
islez(integer, symbol, integer, real, real, real, real, real, real,
      real, symbol, symbol, real, integer, real, real, real).
gonder(integer, symbol, integer, real, real, real, real, real,
      real, real, symbol, symbol, real, integer, symbol, real,
      integer, real, real, real, real, real, real, real).
kesme(integer, symbol, integer, real, real, real, real, real, real, real,
      symbol, symbol, real, integer, symbol, real, integer, real, real, real).
sil(symbol, integer).
ekle.
cozuyaz(integer).

database-ka
ka(sir).

clauses
parcaci(PADI, PTAL):-
  consult("CNCPAR.DAT", parca),
  makewindow(2, 14, 14, "ÇÖZÜMLER ", 0, 0, 25, 80),
  cursor(12, 12),
  write("C o z u m l e r   A r a s t i r i l i y o r . . . . ."),
  upper_lower(PADI, FP),
  parca(par(P, MALK, OZKESKUV, _, _, BRDEPMAL, ISYON, PASO,
           ILERLEME, BASCAP, SONCAP, ISBOY, CIKACI)), !,
  consult("CNCKULTE.DAT", kulte),
  A=PADI, B=MALK, C=OZKESKUV,
  D=PTAL, F=BRDEPMAL, G=ISYON, H=PASO,
  I=ILERLEME, J=BASCAP, K=SONCAP, L=ISBOY, M=CIKACI,
  retractall(_, parca),
  irdele(1, A, B, C, D, F, G, J, K, L, H, I, M).
parcaci(_, _).

irdele(K, PAD, PMAL, OZKES, PTAL, BRDEPM, [ISYON:T0], [BASCAP:T1],
      [SONCAP:T2], [ISBOY:T3], [PASO:T4], [ILERLEME:T5], [CIKACI:T6]):-
  K1=K+1,
  retractall(_, ka),
  asserts(ka(k(K))),
  bakara(PMAL, ISYON, CIKAC, ILERLEME, PASO, UC,
         TUT, KESMAL, KOSSAY, KENZU, RADUS),

```







```

        cursor(17,26),writef("%18.2",BOSZAMMAL),
        cursor(18,26),writef("%18.5",STOKMAL),
        cursor(19,26),writef("%18.2",BRSABMAL),
        cursor(20,26),writef("%18.2",BRDEGMAL),
        cursor(21,26),writef("%18.2",TOPBRMAL),
        cursor(22,16),write("Her hangi bir tusa basiniz")
        ,readchar(_),removewindow,P1=P-1,P1>0,cozumyaz(P1).
cozumyaz(P):-cozum(son(PAD,TAL,P,KES,TUT,TEZADI,DEV,V,TM,TL,TLE,
        BOS,URZ,MAKSAY,URMIK,ISMAL,KESMAL,BOSZAMMAL,STOKMAL,
        BRSABMAL,BRDEGMAL,TOFBRMAL)).

cozumara1(PAD,PTA):-
        parca(par(PAD,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_,_)),
        upper_lower(PAD,PP),
        cozumara2(PP,PTA),
        retractall(_,parca),
        retractall(_,coz),!.

cozumara1(PAD,PTA):-
        makewindow(1,2,3,"",0,0,25,80),
        cursor(12,15),
        write("D a h a   o n c e   c o z u l m e m i s   . . . . ."),
        cursor(16,12),
        write("C o z e y i m m i ?   [e/E] veya [h/H]...."),
        readln(0),
        upper_lower(0,01),
        O1="e",
        clearwindow,
        kul_tezgah,removewindow,
        parcaci(PAD,PTA),
        removewindow,
        ka(k(K)),
        cozumyaz(K),
        retractall(_,ka),
        ekle,
        !.

cozumara1(,_):-
        clearwindow,
        cursor(12,12),
        write("C o z u m   b u l u n a m a d i . . . . ."),
        readchar(_),
        removewindow,
        retractall(_,cozum),
        retractall(_,ka).

ekle:-
        cozum(son(PAD,TAL,KES,SI,TUT,TEZADI,DEV,V,TM,TL,TLE,EOS,URZ,
        MAKSAY,URMIK,ISMAL,KESMAL,BOSZAMMAL,STOKMAL,
        BRSABMAL,BRDEGMAL,TOFBRMAL)),
        assert(coz(son(PAD,TAL,KES,SI,TUT,TEZADI,DEV,V,TM,TL,TLE,EOS,
        URZ,MAKSAY,URMIK,ISMAL,KESMAL,BOSZAMMAL,
        STOKMAL,BRSABMAL,BRDEGMAL,TOFBRMAL))),
        fail.

```

```
ekle:-save("CNCANACO.DAT",coz),
clearwindow,
cursor(12,12),
write("Y e n i   c o z u m   k a y d e d i l d i....."),
retractall(_,cozum),
cursor(14,14),
write("Bir tusa basiniz..."),
readchar(_),
removewindow,!.
ekle:-clearwindow,
cursor(12,12),
write("Y e n i   c o z u m   k a y d e d i l e m e d i")
,readchar(_),removewindow.
```

```
project "proje"
include "anatan.pro"
```

```
predicates
```

```
sec(integer, real).
kont(integer).
par_tan_bil(symbol, integer, integer, real, real, real).
isl_tek_bil(integer, ss, rr, rr, rr, rr, rr, rr).
pkayit(parbil).
kalite(real).
islem_yon(symbol).
bak(integer, symbol).
kont1(integer).
parmal(integer, integer).
malz(integer, integer).
```

```
clauses
```

```
pkaydet:-makewindow(1,5,12,"PARÇA BİLGİLERİ",0,0,25,60),
makewindow(2,11,12,"PARÇA TANITIM BİLGİLERİ",1,1,11,65),
par_tan_bil(A,B,AB,C,D,E),A<>"",AB<>0,B<>0,C>=0,D>=0,E>=0,!,
makewindow(3,11,15,"İŞLEM TEKNİK BİLGİLERİ",4,7,15,50),
isl_tek_bil(1,W,K,L,M,N,P,R),removewindow,removewindow,
pkayit(par(A,B,AB,C,D,E,W,K,L,M,N,P,R)),!.
pkaydet:-write("Yanlis girdiniz.....devamı E/H"),readln(Z),
upper_lower(Z,ZZ),ZZ="e",
pkaydet,!.

```

```
par_tan_bil(A1,B1,BB1,C1,D1,E1):-clearwindow,
cursor(8,1),write("Yanliş girdinizse ESC basınız.."),
cursor(1,2),write("Parçanın adı :"),readln(A1),
parmal(B1,BB1),
cursor(2,2),write("Parçanın malzeme kodu :"),write(B1),
cursor(3,2),write("Üzgül kesme kuvveti kg/mm2 :"),write(BB1),
cursor(4,2),write("Parçanın aylık talebi :"),readreal(C1),
cursor(5,2),write("Parçanın satış fiyatı (TL) :"),readreal(D1),
cursor(6,2),write("Biriz depolara gideri (ay) :"),readreal(E1).
```

```
parmal(JJ,BB):-makewindow(7,2,9,"MALZEME SECİMİ",1,1,23,78),
cursor(1,1),write("-1- Alaşimsız düşük karbon çeliği"),
cursor(2,1),write("-2- Alaşimsız orta karbon çeliği"),
cursor(3,1),write("-3- Alaşimsız yüksek karbon çeliği"),
cursor(4,1),write("-4- Alaşimsız tem. sert. çelik"),
cursor(5,1),write("-5- Tavlan. yumuşak, alaşımli çelik"),
cursor(6,1),write("-6- Düşük alaşımli sert çelik"),
cursor(7,1),write("-7- Yüksek alaşımli tavlanmış çelik"),
cursor(8,1),write("-8- Çok sert yüksek alaşımli çelik"),
cursor(9,1),write("-9- Yüksek alaşımli takım çeliği"),
cursor(10,1),write("-10- Yüksek alaşımli sert çelik"),
cursor(11,1),write("-11-Çok sert temperlenmiş çelik"),
cursor(12,1),write("-12-Fastanmaz çelik,ferrit,tavli"),
cursor(13,1),write("-13-Fastanmaz çelik,austenik,tavli"),
cursor(14,1),write("-14-Alaşimsız dökme çelik"),
cursor(15,1),write("-15-Düşük alaşımli dökme çelik"),
```

```

cursor(16,1),write("-16-Yuksek alařımlı dökme çelik"),
cursor(17,1),write("-17-Paslanmaz ferritik dökme çelik"),
cursor(18,1),write("-18-Paslanmaz austenik dökme çelik"),
cursor(19,1),write("-19-Manganez dökme çeliđi"),
cursor(1,40),write("-20-Dövme demir ferrit yapı"),
cursor(2,40),write("-21-Dövme demir pearlitik yapı"),
cursor(3,40),write("-22-Beyaz dökme demir ferrit"),
cursor(4,40),write("-23-Beyaz dökme demir pearlitik"),
cursor(5,40),write("-24-Dökme demir ferrit yapı"),
cursor(6,40),write("-25-Dökme demir pearlitik yapı"),
cursor(7,40),write("-26-Sertleřtirilmiř dökme demir"),
cursor(8,40),write("-27-Tavlanarak yumuřatılmıř demirler"),
cursor(9,40),write("-28-Nikelli dökme demir"),
cursor(10,40),write("-29-Kobalt iđerikli dökümler"),
cursor(11,40),write("-30-Elektrolit bakır"),
cursor(12,40),write("-31-Kurřun alařım bronz, prinç"),
cursor(13,40),write("-32-Kırmızı prinç alařımlı bronz"),
cursor(14,40),write("-33-Isıl iřlemsiz alüminyumlar"),
cursor(15,40),write("-34-Isıl iřlemli alüminyumlar"),
cursor(16,40),write("-35-Isıl iřlemsiz alüminyum döküm"),
cursor(17,40),write("-36-Isıl iřlemli alüminyum döküm"),
cursor(19,39),
write("Malzemelerden uygun olanı seciniz..."),
cursor(20,55),
readint(JK),malz(JK,EBQ),JJ=JK,EB=EBQ,removewindow.

```

```

malz(1,190).
malz(2,210).
malz(3,230).
malz(4,330).
malz(5,210).
malz(6,250).
malz(7,275).
malz(8,200).
malz(9,250).
malz(10,300).
malz(11,320).
malz(12,280).
malz(13,260).
malz(14,200).
malz(15,180).
malz(16,240).
malz(17,235).
malz(18,230).
malz(19,360).
malz(20,280).
malz(21,180).
malz(22,185).
malz(23,175).
malz(24,195).
malz(25,198).
malz(26,295).
malz(27,125).

```

```

malz(28,200).
malz(29,235).
malz(30,110).
malz(31,70).
malz(32,175).
malz(33,50).
malz(34,70).
malz(35,75).
malz(36,90).

```

```

isl_tek_bil(F,[W1:T0],[K1:T1],[L1:T2],[M1:T3],
[N1:T4],[P1:T5],[R1:T6]):-
  cursor(1,1),write(F," İşleminin bilgilerinin girilmesi"),
  cursor(3,2),write(F," İşleminin başlangıç çapı (mm) :"),
  readreal(M1),M1>0,
  cursor(4,2),write(F," İşleminin bitiş çapı (mm) :"),
  readreal(N1),N1>0,
  cursor(5,2),write(F," İşleminin işlem boyu (mm) :"),
  readreal(P1),P1>0,
  cursor(6,2),islem_yon(W1),
  write(F," İşleminin islem yonu :"),
  write(W1),W1<>"",
  cursor(7,2),write(F," İşlemi için verilen paso (mm) :"),
  readreal(K1),K1>0,
  kalite(L1),
  cursor(8,2),write(F," İşlemi için seçilen ilerleme (mm) :"),
  write(L1),L1>0,
  cursor(9,2),write(F," İşleminin çıkış açısı 90 veya 0 :"),
  readreal(R1),
  cursor(10,2),
  write("Her hangi bir tuşa basınız..."),
  readchar(_),clearwindow,
  F1=F+1,
  cursor(11,4),write("İşlemler bitti ise enterle geçiniz.."),

```

```

isl_tek_bil(F1,T0,T1,T2,T3,T4,T5,T6).

```

```

isl_tek_bil(_,[],[],[],[],[],[],[]).

```

```

kalite(UJ):-

```

```

  makewindow(1,3,10,"KALITE SECIM",15,50,10,30),
  write(" 1 0 -5 mikron yüzey"),nl,
  write(" 2 5 -15 mikron yüzey"),nl,
  write(" 3 15-25 mikron yüzey"),nl,
  write(" 4 25-40 mikron yüzey"),nl,
  write(" 5 40-60 mikron yüzey"),nl,
  write(" 6 60- mikron yüzey"),nl,
  cursor(7,4),
  write("Birisini seçiniz "),
  readint(V),

```





```

project"proje"
include"anatan.pro"
domains
tezguh=tez(symbol, integer, real, real, real, real, integer,
            real, real, integer, integer, integer, real, real, real).
predicates
  consultfile.
  tkayit(tez_bil).
  tez_tan_bil(symbol, real, real, real, real, real, integer,
              real, real, real, real, real, real, real).

clauses
  consultfile:-existfile("CNCTEZ.DAT"),!,
               consult("CNCTEZ.DAT",tezguhlar).
  consultfile:-write("Tezguh bilgileri kutugu bulunamadi"),
               write("Yeni kutuk acayimmi? E/H"),
               readln(EB), upper_lower(EB,M),M="e".

tezkaydet:-
  makewindow(1,2,3,"***TEZGAH BILGILERI GIRISI***",0,0,25,80),
  consultfile,
  tez_tan_bil(A1,B1,C1,D1,E1,F1,G1,H1,J1,K1,L1,M1,N1,O1,P1),
  tkayit(tez(A1,B1,C1,D1,E1,F1,G1,H1,J1,K1,L1,M1,N1,O1,P1)),fail.
tezkaydet:-clearwindow,
           cursor(12,12),
           write("Devammi? E/e"),
           readln(Q),
           upper_lower(Q,Q1),
           Q1="e",tezkaydet.

tez_tan_bil(A1,B1,C1,D1,E1,F1,G1,H1,J1,K1,L1,M1,N1,O1,P1):-
  cursor(22,3),
  write("Sonlandirmak isterseniz entere basiniz..."),
  cursor(1,2),trace(off),
  write("Tezguh adini/nunerasi           ="),
  readln(A), upper_lower(A,A1),
  cursor(2,2),
  write("Tezguh verini                       ="),
  readreal(B1),
  cursor(3,2),
  write("Donerlik tez. kapasitesi           (saat/ay) ="),
  readreal(C1),
  cursor(4,2),
  write("Tezguh tahrik motor gucu           (KW) ="),
  readreal(D1),
  cursor(5,2),
  write("Tezgaha yuklenebilir ent cap       (mm) ="),
  readreal(E1),
  cursor(6,2),
  write("Tezgaha yuklenebilir ent boy       (mm) ="),
  readreal(F1),
  cursor(7,2),
  write("Bu tezgahtan eldeki sayi           (adet) ="),

```

```

readreal(G1),
cursor(8,2),
write("Tezgah maliyeti           (iřcilik+gen.gid)/dk ="),
readreal(H1),
cursor(9,2),
write("Tezgahin baslangic devir hizi degeri   (dev/dak) ="),
readint(J1),
cursor(10,2),
write("Tezgahin son devir hizi degeri        (dev/dak) ="),
readint(K1),
cursor(11,2),
write("Devir hizi artirim adim degeri        (dev/dak) ="),
readint(L1),
cursor(12,2),
write("Tezgahin ilerleme hizi baslama degeri (mm/devir)="),
readreal(M1),
cursor(13,2),
write("Tezgahin ilerleme hizi son degeri     (mm/devir) ="),
readreal(N1),
cursor(14,2),
write("Tezgahin ilerleme hizi artma degeri   (mm/devir) ="),
readreal(O1),
cursor(15,2),
write("Tezgah isletim maliyeti              (TL/Ay) ="),
readreal(P1),
cursor(16,2),
cursor(20,3),write("Girilen bilgiler dogrumu?..E/e      "),
readln(DY),upper_lower(DY,DY1),DY1="e".

```

```

tkayit(tez(A,B,C,D,E,F,G,H,J,K,L,M,N,O,P)):-
    tezgahlar(tez(A,B,C,D,E,F,G,H,J,K,L,M,N,O,P)).

```

```

tkayit(tez(A,B,C,D,E,F,G,H,J,K,L,M,N,O,P)):-
    asserta(tezgahlar(tez(A,B,C,D,E,F,G,H,J,K,L,M,N,O,P))),
    save("CNCTEZ.DAT",tezgahlar).

```

```
project"proje"  
include"anatan.pro"  
clauses  
cozara:-makewindow(1,6,15,"",0,0,25,80),  
consult("CNCPAR.DAT",parca),  
consult("CNCTEZ.DAT",tezgahlar),  
consult("CNCANACO.DAT",coz),  
makewindow(6,2,3,"COZUM ARAMA ICIN BILGI GIRISI",5,15,5,45),  
cursor(1,1),  
write("PARCA ADI/KODUNU GIRINIZ ="),  
cursor(1,28),  
readln(PAD),  
cursor(2,1),  
upper_lower(PAD,PAD1),  
write("PARCANIN TALEBINI GIRINIZ ="),  
cursor(2,28),  
readreal(PTAL),  
removewindow,trace(on),  
cozumara1(PAD1,PTAL),  
retractall(_,coz),  
retractall(_,cozum),  
retractall(_,kultez),  
retractall(_,tezgahlar),  
retractall(_,parca).
```





```
project"proje"  
include"anatan.pro"  
  
clauses  
  
kul_tezgah:-  
    makewindow(2,7,4,"KULLANILACAK MAKINALARIN SECIMI",10,10,4,60),  
    fail.  
  
kul_tezgah:-  
    tezgahlar(tez(TAD1,VER,TZKAP,MOTGUC,ENBCAP,ENBBOY,ADET,  
                TEZMAL,ENKDEV,ENEDEV,ART1,  
                ILERBAS,ILERSON,ART2,ISLMAL)),  
    cursor(1,1),  
    write(TAD1),  
    write(" tezgahini bu isde kullanacakmisiniz? E/H"),  
    readln(K),clearwindow,  
    upper_lower(K,K1),  
    K1="e",  
    assert(kultez(tez(TAD1,VER,TZKAP,MOTGUC,ENBCAP,ENBBOY,ADET,  
                    TEZMAL,ENKDEV,ENEDEV,ART1,  
                    ILERBAS,ILERSON,ART2,ISLMAL))),  
    fail.  
  
kul_tezgah:-  
    save("CNCKULTE.DAT",kultez),  
    removewindow,  
    retractall(_,kultez),!.
```

```
/*  
Turbo Prolog Toolbox  
(C) Copyright 1987 Borland International.  
In order to use the tools, the following domain declarations  
should be included in the start of your program  
*/  
DOMAINS  
ROW, COL, LEN, ATTR = INTEGER  
STRINGLIST = STRING*  
INTEGERLIST = INTEGER*  
KEY = cr; esc; break; tab; btab; del; bdel; ctrlbdel; ins;  
end ; home ; fkey(INTEGER) ; up ; down ; left ; right ;  
ctrlleft; ctrlright; ctrlend; ctrlhome; pgup; pgdn;  
ctrlpgup; ctrlpgdn; char(CHAR) ; otherspec
```

```

/*****
Turbo Prolog Toolbox
(C) Copyright 1987 Borland International.

This module includes some routines which are used in nearly
all menu and screen tools.
*****/

/*****/
/*repeat */
/*****/

PREDICATES
  nondeterm repeat

CLAUSES
  repeat.
  repeat:-repeat.

/*****/
/*miscellaneous*/
/*****/

PREDICATES
  maxlen(STRINGLIST, COL, COL)
  listlen(STRINGLIST, ROW)
  writelist(ROW, COL, STRINGLIST)
  reverseattr(ATTR, ATTR)
  min(ROW, ROW, ROW)
  min(COL, COL, COL)
  min(LEN, LEN, LEN)
  min(INTEGER, INTEGER, INTEGER)
  max(ROW, ROW, ROW)
  max(COL, COL, COL)
  max(LEN, LEN, LEN)
  max(INTEGER, INTEGER, INTEGER)

CLAUSES
  maxlen(H:T1, MAX, MAX1) :-
    str_len(H, LENGTH),
    LENGTH>MAX,!,
    maxlen(T, LENGTH, MAX1).
  maxlen(I_1T1, MAX, MAX1) :- maxlen(T, MAX, MAX1).
  maxlen(I, LENGTH, LENGTH).

  listlen(I, 0).
  listlen(I_1T1, N):-
    listlen(T, X),
    N=X+1.

  writelist(_,_, []).
  writelist(L1, ANTKOL, I:H:T1):-
    field_str(L1, 0, ANTKOL, H),
    L1=L1+1,
    writelist(L1, ANTKOL, T).

```



```
min(X,Y,X):-X<=Y,!.
min(_,X,X).
```

```
max(X,Y,X):-X>=Y,!.
max(_,X,X).
```

```
reverseattr(A1,A2):-
    bitand(A1,$07,H11),
    bitleft(H11,4,H12),
    bitand(A1,$70,H21),
    bitright(H21,4,H22),
    bitand(A1,$08,H31),
    A2=H12+H22+H31.
```

```
/******
/*      Find letter selection in a list of strings      */
/*      Look initially for first uppercase letter.      */
/*      Then try with first letter of each string.      */
/******
```

#### PREDICATES

```
upc(CHAR,CHAR) lowc(CHAR,CHAR)
try_upper(CHAR,STRING)
tryfirstupper(CHAR,STRINGLIST,ROW,ROW)
tryfirstletter(CHAR,STRINGLIST,ROW,ROW)
tryletter(CHAR,STRINGLIST,ROW)
```

#### CLAUSES

```
upc(CHAR,CH):-
    CHAR>='a',CHAR<='z',!,
    char_int(CHAR,C1), C1=C1-32, char_int(CH,C11).
upc(CH,CH).
```

```
lowc(CHAR,CH):-
    CHAR>='A',CHAR<='Z',!,
    char_int(CHAR,C1), C1=C1+32, char_int(CH,C11).
lowc(CH,CH).
```

```
try_upper(CHAR,STRING):-
    frontchar(STRING,CH,_),
    CH>='A',CH<='Z',!,
    CH=CHAR.
```

```
try_upper(CHAR,STRING):-
    frontchar(STRING,_,REST),
    try_upper(CHAR,REST).
```

```
tryfirstupper(CHAR,[W1_],N,N):-
    try_upper(CHAR,W),!.
```

```
tryfirstupper(CHAR,[_T1],N1,N2):-
    N3 = N1+1,
```

```
tryfirstupper(CHAR,T,N3,N2).
```

```
tryfirstletter(CHAR,[W1_],N,N):-
    frontchar(W,CHAR,_,!).
```

```
tryfirstletter(CHAR,[_T1],N1,N2):-
    N3 = N1+1,
    tryfirstletter(CHAR,T,N3,N2).
```

```

tryletter(CHAR,LIST,SELECTION):-
    upc(CHAR,CH),tryfirstupper(CH,LIST,0,SELECTION),!.
    tryletter(CHAR,LIST,SELECTION):-
        lowc(CHAR,CH),tryfirstletter(CH,LIST,0,SELECTION).

```

```

/*****/
adjustwindow takes a windowstart and a
window size and adjusts the windowstart
so the window can be placed on the screen.
adjframe looks at the frameattribute: if
it is different from zero, two is added to
the size of the window
/*****/

```

#### PREDICATES

```

adjustwindow(ROW,COL,ROW,COL,ROW,COL)
adjframe(ATTR,ROW,COL,ROW,COL)

```

#### CLAUSES

```

adjustwindow(LI,KOL,DLI,DKOL,ALI,AKOL):-
    LI<25-DLI,KOL<80-DKOL,!,ALI=LI,AKOL=KOL.
adjustwindow(LI,_,DLI,DKOL,ALI,AKOL):-
    LI<25-DLI,!,ALI=LI,AKOL=80-DKOL.
adjustwindow(,KOL,DLI,DKOL,ALI,AKOL):-
    KOL<80-DKOL,!,ALI=25-DLI,AKOL=KOL.
adjustwindow(,_,DLI,DKOL,ALI,AKOL):-
    ALI=25-DLI,AKOL=80-DKOL.

adjframe(0,R,C,R,C):-!.
adjframe(,R1,C1,R2,C2):-R2=R1+2,C2=C1+2.

```

```

/*****/
/* Readkey */
/* Returns a symbolic key from the KEY domain */
/*****/

```

#### PREDICATES

```

readkey(KEY)
readkey1(KEY,CHAR,INTEGER)
readkey2(KEY,INTEGER)

```

#### CLAUSES

```

readkey(KEY):-readchar(T),
    char_int(T,VAL),readkey1(KEY,T,VAL).
readkey1(KEY,_,0):-!,readchar(T),char_int(T,VAL),
    readkey2(KEY,VAL).
readkey1(cr,_,13):-!.
readkey1(esc,_,27):-!.
readkey1(break,_,8):-!.
readkey1(tab,_,9):-!.
readkey1(bdel,_,8):-!.
readkey1(ctribdel,_,127):-!.
readkey1(char(T),T,_) .

```

```
readkey2(btab,15):-!.
readkey2(del,83):-!.
readkey2(ins,82):-!.
readkey2(up,72):-!.
readkey2(down,80):-!.
readkey2(left,75):-!.

readkey2(right,77):-!.
readkey2(pgup,73):-!.
readkey2(pgdn,81):-!.
readkey2(end,79):-!.
readkey2(home,71):-!.
readkey2(ctrlleft,115):-!.
readkey2(ctrlright,116):-!.
readkey2(ctrlend,117):-!.
readkey2(ctrlpgdn,118):-!.
readkey2(ctrlhome,119):-!.
readkey2(ctrlpgup,132):-!.
readkey2(fkey(N),VAL):- VAL>58, VAL<70, N=VAL-58, !.
readkey2(fkey(N),VAL):- VAL>=84, VAL<104, N=11+VAL-84, !.
readkey2(otherspec,_).
```

```
project"proje"  
include"anatan.pro"  
clauses  
  tablo:-makewindow(5,2,3,"C O Z U M L E R",0,0,25,80),  
    write("PARCA ADI           ="),nl,  
    write("ISLEM NO           ="),nl,  
    write("TEZGAH ADI         ="),nl,  
    write("KESICI ADI         ="),nl,  
    write("TUTUCU ADI         ="),nl,  
    write("TALEP              ="),nl,  
    write("DEVIR HIZI         ="),nl,  
    write("KESME HIZI         ="),nl,  
    write("ISLEME ZAMANI       ="),nl,  
    write("TAKIM OMRU          ="),nl,  
    write("PARCA BASI TAKIM ZAMANI ="),nl,  
    write("BOSA GECEN ZAMAN     ="),nl,  
    write("URETIM ZAMANI        ="),nl,  
    write("GEREKLI MAKINA SAYISI ="),nl,  
    write("URETIM MIKTARI        ="),nl,  
    write("ISLETIM MALIYETI     ="),nl,  
    write("KESICI KALEM MALIYETI ="),nl,  
    write("BOS ZAMAN MALIYETI    ="),nl,  
    write("STOKLAMA MALIYETI     ="),nl,  
    write("BIRIM SABIT MALIYET   ="),nl,  
    write("BIRIM DEGİSKEN MALIYET ="),nl,  
    write("TOPLAM BIRIM MALIYET  =").
```

```

project "proje"
include "anatan.pro"

predicates
    kont1(integer).
    kont2(integer).
    kont3(integer).
    kont4(integer).
    kont5(integer).
    kont6(integer).
secim1(symbol,real,real,real,symbol,symbol,integer,integer,integer,real).
secim3(symbol,real,real,real,symbol,symbol,integer,integer,integer,real).
secim4(symbol,real,real,real,symbol,symbol,integer,integer,integer,real).
secim5(symbol,real,real,real,symbol,symbol,integer,integer,integer,real).
secim6(symbol,real,real,real,symbol,symbol,integer,integer,integer,real).
secim7(symbol,real,real,real,symbol,symbol,integer,integer,integer,real).
secim8(symbol,real,real,real,symbol,symbol,integer,integer,integer,real).
secim9(symbol,real,real,real,symbol,symbol,integer,integer,integer,real).
secim10(symbol,real,real,real,symbol,symbol,integer,integer,integer,real)
.
secim11(symbol,real,real,real,symbol,symbol,integer,integer,integer,real)
.

clauses

bakara(A,B,C,D,E,F,G,H,I,J,K):-
    kont1(A),!,secim2(B,C,D,E,F,G,H,I,J,K),!.
bakara(A,B,C,D,E,F,G,H,I,J,K):-
    kont2(A),!,secim6(B,C,D,E,F,G,H,I,J,K),!.
bakara(A,B,C,D,E,F,G,H,I,J,K):-
    kont3(A),!,secim7(B,C,D,E,F,G,H,I,J,K),!.
bakara(A,B,C,D,E,F,G,H,I,J,K):-
    kont4(A),!,secim8(B,C,D,E,F,G,H,I,J,K),!.
bakara(A,B,C,D,E,F,G,H,I,J,K):-
    kont5(A),!,secim9(B,C,D,E,F,G,H,I,J,K),!.
bakara(A,B,C,D,E,F,G,H,I,J,K):-
    kont6(A),!,secim10(B,C,D,E,F,G,H,I,J,K),!.
bakara(1,B,C,D,E,F,G,H,I,J,K):-
    secim1(B,C,D,E,F,G,H,I,J,K),!.
bakara(6,B,C,D,E,F,G,H,I,J,K):-
    secim3(B,C,D,E,F,G,H,I,J,K),!.
bakara(7,B,C,D,E,F,G,H,I,J,K):-
    secim4(B,C,D,E,F,G,H,I,J,K),!.
bakara(14,B,C,D,E,F,G,H,I,J,K):-
    secim5(B,C,D,E,F,G,H,I,J,K),!.
bakara(26,B,C,D,E,F,G,H,I,J,K):-
    secim11(B,C,D,E,F,G,H,I,J,K),!.

kont1(X):-X=2,!,X=3,!,X=4,!,X=5,!,X=6,!,X=9,!,X=10,!,
    X=11,!,X=12,!,X=13,!,X=15.
kont2(X):-X=16,!,X=17,!,X=18.
kont3(X):-X=19,!,X=20.

```



```

secim5("Diş yüzey",_,AA,BB,"CNMM-71","PCLN",10,4,18,2):-AA<1,BB<8,!.
secim5("Diş yüzey",_,AA,BB,"CNMM-71","PCLN",10,4,18,2):-AA<1,BB<16,!.
secim5("Alın",_,AA,BB,"TNMG-61","PTFN",14,6,18,1.5):-AA<0.3,BB<2,!.
secim5("Alın",_,AA,BB,"CNMM-71","PCLN",10,4,18,2):-AA<0.5,BB<4,!.
secim5("Alın",_,AA,BB,"CNMM-71","PCLN",10,4,18,2):-AA<1,BB<8,!.
secim5("Alın",_,AA,BB,"CNMM-71","PCLN",10,4,18,2):-AA<1,BB<16,!.
/*16. MALZEME*/
secim6("Diş yüzey",90,AA,BB,"TPMR","CTGP",6,3,15,1.5):-AA<0.3,BB<2,!.
secim6("Diş
yüzey",90,AA,BB,"TNMM-71","PTGN",14,6,18,1.5):-AA<0.5,BB<4,!.
secim6("Diş yüzey",90,AA,BB,"TNMM-71","PTGN",14,6,18,1.5):-AA<1,BB<8,!.
secim6("Diş yüzey",90,AA,BB,"TNMM-71","PTGN",14,6,18,1.5):-AA<1,BB<16,!.
secim6("Diş yüzey",_,AA,BB,"TPMR","CTEP",6,3,15,1.5):-AA<0.3,BB<2,!.
secim6("Diş yüzey",_,AA,BB,"CNMM-71","PCLN",10,4,18,2):-AA<0.5,BB<4,!.
secim6("Diş yüzey",_,AA,BB,"CNMM-71","PCLN",10,4,18,2):-AA<1,BB<8,!.
secim6("Diş yüzey",_,AA,BB,"CNMM-71","PCLN",10,4,18,2):-AA<1,BB<16,!.
secim6("Alın",_,AA,BB,"TPMR","CTFP",6,3,15,1.5):-AA<0.3,BB<2,!.
secim6("Alın",_,AA,BB,"CNMM-71","PCLN",10,4,18,2):-AA<0.5,BB<4,!.
secim6("Alın",_,AA,BB,"CNMM-71","PCLN",10,4,18,2):-AA<1,BB<8,!.
secim6("Alın",_,AA,BB,"CNMM-71","PCLN",10,4,18,2):-AA<1,BB<16,!.

secim7("Diş yüzey",90,AA,BB,"TNMM","PTGN",7,3,18,1.5):-AA<0.3,BB<2,!.
secim7("Diş yüzey",90,AA,BB,"TNMX","PTJN",12,6,15,1.5):-AA<0.5,BB<4,!.
secim7("Diş yüzey",90,AA,BB,"TNMM","PTGN",7,3,18,1.5):-AA<1,BB<8,!.
secim7("Diş yüzey",90,AA,BB,"TNMM","PTGN",7,3,18,1.5):-AA<1,BB<16,!.
secim7("Diş yüzey",_,AA,BB,"CNMM","PCLN",9,4,16,2):-AA<0.3,BB<2,!.
secim7("Diş yüzey",_,AA,BB,"CNMM","PCLN",9,4,16,2):-AA<0.5,BB<4,!.
secim7("Diş yüzey",_,AA,BB,"CNMM","PCLN",9,4,16,2):-AA<1,BB<8,!.
secim7("Diş yüzey",_,AA,BB,"CNMM","PCLN",9,4,16,2):-AA<1,BB<16,!.
secim7("Alın",_,AA,BB,"SNMM","PSKN",11,4,16,2):-AA<0.3,BB<2,!.
secim7("Alın",_,AA,BB,"SNMM","PSKN",11,4,16,2):-AA<0.5,BB<4,!.
secim7("Alın",_,AA,BB,"SNMM","PSKN",11,4,16,2):-AA<1,BB<8,!.
secim7("Alın",_,AA,BB,"SNMM-41","PSKN",12,4,16,2):-AA<1,BB<16,!.
/*21. MALZEME*/
secim8("Diş yüzey",90,AA,BB,"TNMG","PTGN",12,6,15,1.5):-AA<0.3,BB<2,!.
secim8("Diş yüzey",90,AA,BB,"TNMG","PTGN",12,6,15,1.5):-AA<0.5,BB<4,!.
secim8("Diş yüzey",90,AA,BB,"TNMG","PTGN",12,6,15,1.5):-AA<1,BB<8,!.
secim8("Diş yüzey",90,AA,BB,"TNMA","PTGN",12,6,15,1.5):-AA<1,BB<16,!.
secim8("Diş yüzey",_,AA,BB,"DNMG","PDJN",8,4,12,1.2):-AA<0.3,BB<2,!.
secim8("Diş yüzey",_,AA,BB,"DNMG","PDJN",8,4,12,1.2):-AA<0.5,BB<4,!.
secim8("Diş yüzey",_,AA,BB,"DNMG","PDJN",8,4,12,1.2):-AA<1,BB<8,!.
secim8("Diş yüzey",_,AA,BB,"ONMA","PCLN",12,8,16,1.4):-AA<1,BB<16,!.
secim8("Alın",_,AA,BB,"SNMG","PSKN",12,8,16,2):-AA<0.3,BB<2,!.
secim8("Alın",_,AA,BB,"DNMG","PSKN",8,4,12,1.2):-AA<0.5,BB<4,!.
secim8("Alın",_,AA,BB,"ONMG","PSKN",12,8,16,1.4):-AA<1,BB<8,!.
secim8("Alın",_,AA,BB,"ONMA","PSKN",12,8,16,1.4):-AA<1,BB<16,!.
/*26. MALZEME*/
secim11("Diş yüzey",90,AA,BB,"TNMA","PTGN",12,6,15,1.5):-AA<0.3,BB<2,!.
secim11("Diş yüzey",90,AA,BB,"TNMA","PTGN",12,6,15,1.5):-AA<0.5,BB<4,!.
secim11("Diş yüzey",90,AA,BB,"TNMA","PTGN",12,6,15,1.5):-AA<1,BB<8,!.
secim11("Diş yüzey",90,AA,BB,"TNMA","PTGN",12,6,15,1.5):-AA<1,BB<16,!.
secim11("Diş yüzey",_,AA,BB,"RNMG","PRGN",14,10,5,3):-AA<0.3,BB<2,!.
secim11("Diş yüzey",_,AA,BB,"RNMG","PRGN",14,10,5,3):-AA<0.5,BB<4,!.

```

```

secim11("Dış yUzey",_,AA,BB,"RNMG","PRGN",14,10,5,3):-AA<1,BB<8,!.
secim11("Dış yUzey",_,AA,BB,"RNMG","PRGN",14,10,5,3):-AA<1,BB<16,!.
secim11("Alın",_,AA,BB,"RNMG","PRGN",14,10,5,3):-AA<0.3,BB<2,!.
secim11("Alın",_,AA,BB,"RNMG","PRGN",14,10,5,3):-AA<0.5,BB<4,!.
secim11("Alın",_,AA,BB,"RNMG","PRGN",14,10,5,3):-AA<1,BB<8,!.
secim11("Alın",_,AA,BB,"RNMG","PRGN",14,10,5,3):-AA<1,BB<16,!.
/*27. MALZEME*/
secim9("Dış yUzey",90,AA,BB,"TNMM","PTGN",7,3,18,1.5):-AA<0.3,BB<2,!.
secim9("Dış yUzey",90,AA,BB,"TNMM","PTGN",7,3,18,1.5):-AA<0.5,BB<7,!.
secim9("Dış yUzey",_,AA,BB,"SNMM","PSDN",11,4,16,2):-AA<0.3,BB<2,!.
secim9("Dış yUzey",_,AA,BB,"SNMM","PSDN",11,4,16,2):-AA<0.5,BB<8,!.
secim9("Alın",_,AA,BB,"SNMM","PSKN",11,4,16,2):-AA<0.3,BB<2,!.
secim9("Alın",_,AA,BB,"SNMM","PSKN",11,4,16,2):-AA<0.5,BB<8,!.
/*30. MALZEME*/
secim10("Dış
yUzey",90,AA,BB,"TNMG-61","PTGN",14,6,18,1.5):-AA<0.3,BB<2,!.
secim10("Dış yUzey",90,AA,BB,"TNMM","PTGN",7,3,18,1.5):-AA<0.5,BB<4,!.
secim10("Dış yUzey",90,AA,BB,"TNMM-71","PTGN",14,6,18,1.5):-AA<1,BB<8,!.
secim10("Dış yUzey",90,AA,BB,"TKMA","PTGN",12,6,15,1.5):-AA<1,BB<20,!.

```





```
project"proje"  
include"anatan.pro"  
clauses  
/*2. MALZEME */  
secim2("Dış yüzey",90,AA,BB,"TPMR","CTGP",6,3,15,1.5):-AA<0.3,BB<2,!.  
secim2("Dış yüzey",90,AA,EB,"CNMM-71","PCLN",10,4,18,2):-AA<0.5,BB<4,!.  
secim2("Dış yüzey",90,AA,EB,"CNMM-71","PCLN",10,4,18,2):-AA<1,BB<8,!.  
secim2("Dış yüzey",90,AA,BB,"CNMM-71","PCLN",10,4,18,2):-AA<1,EB<16,!.  
secim2("Dış yüzey",_,AA,BB,"TPMR","CTBP",6,3,15,1.5):-AA<0.3,BB<2,!.  
secim2("Dış yüzey",_,AA,BB,"CNMM-71","PCBN",10,4,18,2):-AA<0.5,EB<4,!.  
secim2("Dış yüzey",_,AA,EB,"CNMM-71","PCBN",10,4,18,2):-AA<1,EB<8,!.  
secim2("Dış yüzey",_,AA,BB,"CNMM-71","PCBN",10,4,18,2):-AA<1,BB<16,!.  
secim2("Alın",_,AA,EB,"TPMR","CTFP",6,3,15,1.5):-AA<0.3,BB<2,!.  
secim2("Alın",_,AA,EB,"CNMM-71","PRGN",10,4,18,2):-AA<0.5,BB<4,!.  
secim2("Alın",_,AA,EB,"CNMM-71","PRGN",10,4,18,2):-AA<1,BB<8,!.  
secim2("Alın",_,AA,EB,"CNMM-71","PRGN",10,4,18,2):-AA<1,EB<16,!.  

```

```

project"proje"
include "anatan.pro"
include "tdoms.pro"
include "tpreds.pro"
include "linemenu.pro"
include "status.pro"

predicates
determ
menu.
determ
gid(integer)
parbak(integer)
tezbak(integer)
clauses

menu:-makewindow(1,23,0,"",0,0,24,80),
    makestatus(112," Seceneklerden birini seciniz.....
        Seciminizi -><- tuslari ile yapabilirsiniz..."),
    linerenu(0,3,82,["Parca","Tezgah","Cozme","Dosa_cik"],CH1),
    renovewindow,
    gid(CH1).

gid(1):-
    %makestatus(112," Birini seciniz.....Seciminizi -><- tuslari
        ile yapabilirsiniz..."),
    linerenu(2,11,82,["Kayit","Tarana","Duzeltme","Cikis"],CH1),
    parbak(CH1),renovewindow,!,menu.

gid(2):-
    %makestatus(112," Birini seciniz.....Seciminizi -><- tuslari
        ile yapabilirsiniz..."),
    linerenu(2,11,82,["Kayit","Tarana","Duzeltme","Cikis"],CH1),
    tezbak(CH1),renovewindow,!,menu.

gid(3):-cczara,!,menu.

gid(4):-!,exit.

gid(_):-menu.

parbak(1):-pkaydet,!.

parbak(_).

tezbak(1):-tezkaydet,!.

tezbak(_).

goal
    menu.

```

proje  
cnccozs  
cnpar  
cnctez  
cozum2  
enkucu  
cozara  
secim2  
sabit  
kayitci  
yazim  
maksay  
kultez  
yeniucse

## C O Z U M L E R

ADI	=mil	
NO	=1	
ADAH	=tor123	
ADI	=CNMM-71	
ADUCU	=PCLN	
	=	12000.00
HIZI	=	178.58
HIZI	=	52.76
ZAMANI	=	1.32
OMRU	=	453.64
BASI TAKIM ZAMANI	=	0.00290
GEÇEN ZAMAN	=	0.3288
ZAMANI	=	1.65
MAKINA SAYISI	=	1
MIKTARI	=	12385.27
MALİYETİ	=	1183.84
KALEM MALİYETİ	=	32.77
ZAMAN MALİYETİ	=	295.96
LAMA MALİYETİ	=	155.53679
SABIT MALİYET	=	117.07
DEĞİŞKEN MALİYET	=	1668.10
BİRİM MALİYET	=	1785.17

Her hangi bir tusa basiniz

## C O Z U M L E R

ADI	=mil	
NO	=2	
ADAH	=tor123	
ADI	=CNMM-71	
ADUCU	=PCLN	
	=	12000.00
HIZI	=	47.06
HIZI	=	13.31
ZAMANI	=	1.02
OMRU	=	12037.60
BASI TAKIM ZAMANI	=	0.00008
GEÇEN ZAMAN	=	0.2538
ZAMANI	=	1.27
MAKINA SAYISI	=	1
MIKTARI	=	16075.04
MALİYETİ	=	913.65
KALEM MALİYETİ	=	0.95
ZAMAN MALİYETİ	=	228.41
LAMA MALİYETİ	=	1267.50577
SABIT MALİYET	=	90.20
DEĞİŞKEN MALİYET	=	2410.53
BİRİM MALİYET	=	2500.73

Her hangi bir tusa basiniz

## C O Z U M L E R

A ADI	=kam	
I NO	=1	
H ADI	=ftmm998	
I ADI	=CNMM-71	
U ADI	=PCLN	
	=	5900.00
R HIZI	=	56.86
E HIZI	=	11.26
E ZAMANI	=	2.90
I OMRU	=	17939.81
BASI TAKIM ZAMANI	=	0.00016
GEÇEN ZAMAN	=	0.7243
M ZAMANI	=	3.62
LI MAKINA SAYISI	=	1
M MIKTARI	=	5964.26
İM MALİYETİ.	=	5794.26
I KALEM MALİYETİ	=	3.96
AMAN MALİYETİ	=	1448.57
AMA MALİYETİ	=	43.09493
I SABİT MALİYET	=	117.37
I DEĞİŞKEN MALİYET •	=	7289.88
M BİRİM MALİYET	=	7407.24

Her hangi bir tusa basınız