

İMALAT HÜCRESİNDE KULLANILAN BİR ROBOTUN İŞLEMLERİNİN SIRALANMASI

Yusuf USTA ve Yücel ERCAN*

Makina Mühendisliği Bölümü, Mühendislik Mimarlık Fakültesi, Gazi Üniversitesi, 06570 Maltepe Ankara,
uyusuf@gazi.edu.tr

*Makine Mühendisliği Bölümü, Mühendislik Fakültesi, TOBB-Ekonomi ve Teknoloji Üniversitesi, Söğütözü,
Ankara, yercan@etu.edu.tr

(Geliş/Received: 05.07.2005; Kabul/Accepted: 09.02.2006)

ÖZET

Hücreyel imalat sistemi son yıllarda yaygın olarak kullanılan ve parti bazındaki üretimlerde uygulanan bir tekniktir. Fiziksel olarak bir araya kümelenmiş makinalardan oluşan bu hücrelerde üretim verimini, kaliteyi ve iş emniyetini artırmak, sıkıcı işleri yorulmadan yaptırmak için robot kullanımı tercih edilmektedir. Bu çalışmada, en fazla on makinadan oluşan bir imalat hücresi ve bir tek robot olduğu kabul edilerek, en fazla on ayrı işin aynı anda üretilmesi durumunda, robotun bu işleri tezgahlara hangi zamanlarda ve hangi sıra ile yerleştirip-alması gerektiğini belirleyen algoritmalar geliştirilmiştir. Bu algoritmalar, imalat hücresini oluşturabilmek için gerekli kütüphane (tezgahlar, robotlar ve depolama alanları), tezgah yerleşimi ve iş parçaları bilgilerinin yer aldığı bir uzman sistemin alt modülü olarak yapılandırılmıştır. Giriş bilgileri uzman sistemin çalışması ile elde edilen veriler olmakta, çıkış bilgileri yine aynı uzman sistem tarafından kullanılmaktadır. Geliştirilen algoritmalar QUICKBASIC dilinde hazırlanmış bir yazılıma aktarılmış ve uzman sistemin “karar mekanizması” kısmını meydana getirmiştir. Örnek problemler üzerindeki testlerde geliştirilen yazılımın güvenilir sonuçlar ürettiği izlenmiştir.

Anahtar Kelimeler: İmalat hücresi, sıralama, işlem planlaması, uzman sistem.

SEQUENCING OF OPERATIONS OF AN INDUSTRIAL ROBOT USED IN A MANUFACTURING CELL

ABSTRACT

Cell production system is a technique which has been used widely for batch type of production in recent years. In a cell formed physically by machine-tools, robot usage is preferred to increase the production efficiency, quality and job security and to avoid boring work. In this study, algorithms have been developed to determine in which order and at what time work pieces will be loaded/unloaded from machine-tools by a robot. A maximum of ten machine-tools and a single robot in a cell, and production of at most ten work pieces simultaneously are assumed. The developed algorithms constitute the inference engine of an expert system. This expert system has been constructed in different modules and can generate manufacturing cells by using its library, machine tool layout data and workpiece data. Process planning module forms the inference engine of the expert system in which the developed algorithms are programmed by using QUICKBASIC language. The developed algorithms have been shown to give reliable results on sample problems.

Keywords: Manufacturing cell, sequencing, scheduling, expert system.

1. GİRİŞ

Esnek imalat sistemleri ve imalat hücrelerindeki modernleşme, iş parçasını yükleme, alma ve parçaları yerine götürme işlerinde, insanların yerini robotların

almasını sağlamıştır. İmalat hücresinde kullanılan bir robot, benzer ürünlerden bir kaç tanesini tezgahlara sırayla yerleştirip alacağından, belirli bir zaman diliminde kendini tekrarlayan bir yollar zincirini takip eder. Bu çalışmada, robotlu bir imalat hücresinde,

robotun bu taşıma işlemleri için hangi zamanlarda hangi işlemleri yapması gerektiğini belirleyen bir algoritma ve yazılım geliştirilmiştir. Geliştirilen algoritma, robotlu imalat hücreleri için geliştirilen bir uzman sistemin modüllerinden biri olarak yapılandırılmıştır. İmalat hücresinde en fazla 10 tezgah bulunabileceği ve aynı anda en fazla 10 iş parçasının işlem gördüğü varsayılmıştır. Hücrede taşıma ve yerleştirme işlemleri için sadece bir tek robot olduğu kabul edilmiştir.

İmalat hücrelerinde tezgahların bir kısmı birbiri ile aynı işleri yapabilecek özelliklerde olabilecekleri gibi, farklı tezgahlar da yer almaktadır. Dolayısıyla belirli operasyonların sistemdeki bir tek makinada yapılması zorunluluğu olabilmektedir. İmalat hücrelerinde kullanılan iş parçası yükleme boşaltma algoritmaları bu sebepten dolayı esnek imalat sistemlerinden farklılıklar gösterir.

Literatürde robotlu montaj sistemlerine ait çalışmalara sıklıkla rastlanmaktadır. Bu çalışmalarda, montajın robotla yapılabilmesi için montaj sıralamasının değiştirilmesi veya yeni baştan montaj sıralaması yapılması gibi çalışmalar dikkati çekmektedir [1,2]. Bu sıralamalarda hiyeristik algoritmalar kullanılmakta ve imalat hücrelerindeki işlem sıralamasıyla benzerlikler göstermektedir. İmalat hatlarında tekerlek üzerinde hareket edebilen robotlar kullanılmaktadır. Alexander J.C. ve arkadaşları [3] ile Desaulniers G. ve arkadaşları [4] tarafından yapılan çalışmalarda, hareketli robotlar için kullanılabilir en kısa yol algoritmaları geliştirilmiştir.

Robotlu imalat hücrelerindeki sıralama problemlerine ait çalışmalar son yıllarda da hızla devam etmektedir. Dawande M ve arkadaşları [5] yaptıkları literatür araştırması ile çoğunluğu 1990'dan sonra yapılan 70'in üzerindeki çalışmayı incelemişlerdir. Bu incelemelerden, bir ya da birden fazla robot bulunduran hücrenin, parça tiplerinin, robot erişim zamanlarının, robot türlerinin ve makine yerleşim tiplerinin dikkate alınarak çözümler geliştirilmeye çalışıldığı anlaşılmaktadır.

Üç makinadan oluşan robotlu bir imalat hücresinde parçaların hiç bekletilmediği bir sıralama problemi üzerinde Agnetis A. ve Pacciarelli D. bir çalışma yapmışlar [6] ve bazı çözümler ürettiklerini bildirmişlerdir. Yine benzeri bir çalışma Soukhal A. ve Martineau P. [7] tarafından yapılmış; burada da tezgahlar arasında bekleme alanlarının olmadığı varsayılmıştır. Bir diğer çalışmada ise [8], tezgah aralarında bekleme alanı yerine robot elinin çift çeneli olması önerisi sunulmuş ve önce iki makina için hiyeristik bir çözüm üretilmiş ve ardından bu çözüm dört makina için genişletilmiştir.

Yarı iletkenlerin inceltilmesi için kullanılacak ve büyük bir robotlu imalat hücresi için yapılmış bir

çalışma ise [9], çevrim zamanının optimizasyonu üzerine yapılmıştır.

Literatür araştırmaları ve ticari olarak piyasada bulunan imalat sistemlerinin incelenmesi sonucunda hem yükleme-boşaltma sistemlerinin ve hem de değişik sıralama yöntemlerinin var olduğu anlaşılmaktadır. Ancak, piyasada satılan sistemlerde, rekabet etme şartlarından dolayı hangi kriterler ve yöntemlerle karar verme mekanizmalarının kullanıldığı bilgilerine ulaşmak hemen hemen mümkün değildir. Yurdumuzda 1990'dan itibaren kullanılmaya başlanan CNC üretim tezgahları ve 1994'ten itibaren kullanılmaya başlanan esnek imalat hücreleri gün geçtikçe artmaktadır. Piyasa koşulları dünyadaki gibi yurdumuzda da bu tür üretim teknolojilerinin çoğalmasını ve imalat hatlarının esnekliğini giderek artırmayı zorunlu hale getirmektedir. Bu yöndeki gelişmeler doğrultusunda, yurdumuzda da yerli bir teknolojinin oluşmasına katkıda bulunmak bu çalışmanın başlıca amaçlarındandır.

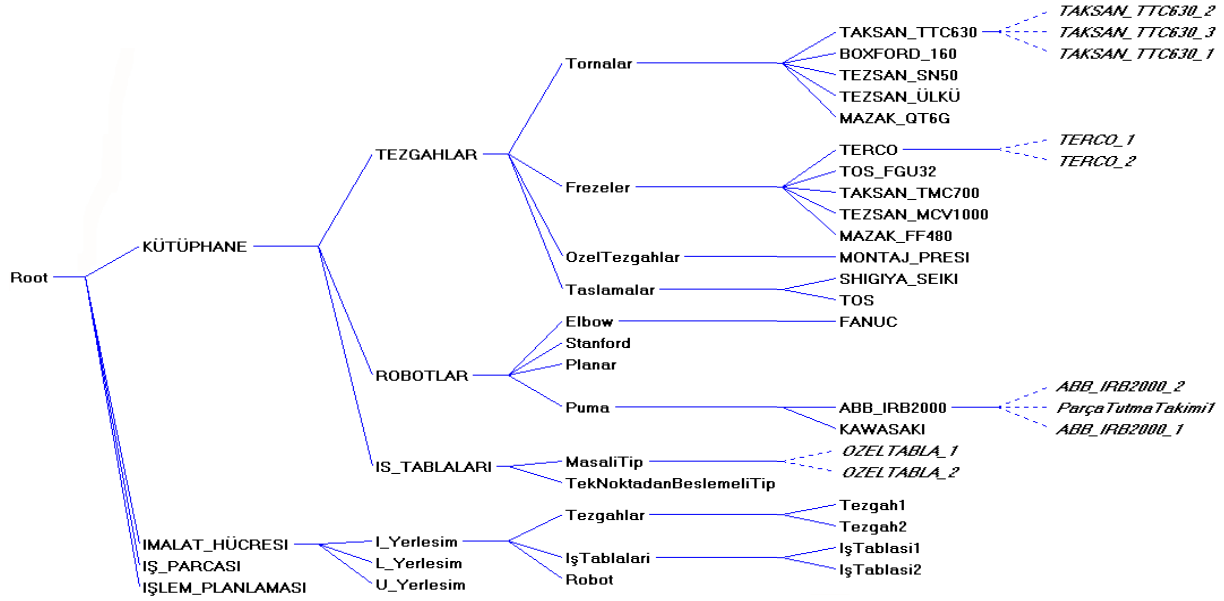
Bu çalışmada geliştirilen robot işlemleri sıralaması algoritması ve yazılımı bir uzman sistemin alt modülünü oluşturmaktadır. Uzman sistem KAPPA™ yazılımı kullanılarak geliştirilmiştir. Nesneye dayalı olarak geliştirilen bu uzman sistemin yapısı Şekil 1'de görülmektedir.

Şekilde görülen bağlama çizgileri nesnelerin her biri arasındaki hiyerarşinin nasıl ilerlediğini göstermektedir. Şekilde çizgiler boyunca soldan sağa doğru ilerledikçe bir önceki nesneden alınan bilgi ve özellikler de aktarılmaktadır. Yani, çizgilerin en sağında bulunan bir nesne kendisinden önceki nesnelerin bütün özelliklerini taşımakta ve gerekiyorsa fazladan bazı bilgi ve özellikleri de içerebilmektedir.

Uzman sistem dört ayrı modül altında toplanmıştır. Bunlardan birincisinde, bütün uzman sistemlerde olduğu gibi, bir "kütüphane" yer almaktadır. Kütüphanede, imalat hücresinin ihtiyacı olan ekipmanlar (tezgahlar, robotlar, iş tablaları) bulunmaktadır. Her bir ekipmanın fiziki boyutları, kapasiteleri ve haberleşme sinyalleri gibi bilgiler kütüphanede yer almaktadır.

Uzman sistemde, oluşturulan imalat hücresinde aynı anda imal edilecek parçalara ait bilgileri o parçaların operasyon planlarından çıkaran bir modül de bulunmaktadır. Bu modüle "iş parçası" modülü adı verilmiştir. İş parçası modülü, robot işlerinin sıralanması için gereken hazır bilgileri operasyon planı içerisinden almaktadır.

Uzman sistemin üçüncü kısmını "imalat hücresi" modülü oluşturmaktadır. Burada, kütüphanede bulunan ekipmanlardan bir kısmının seçilmesiyle imalat hücresi oluşturulmaktadır. Tezgahlar ve iş tablaları ile birlikte imalat hücresinin yerleşim planı



Şekil 1. Uzman sistemin genel yapısı

ve fiziki boyutları ortaya çıkmaktadır. Kütüphanedeki robotlardan birinin imalat hücresinde kullanımıyla da robotun hareket edebileceği alanlar ve çalışma hacmi ortaya çıkmış olmaktadır.

Uzman sistem yapısı içerisindeki dördüncü modül “işlem planlaması” adı verilen kısımdır. Bu modülün çalıştırılması sonucunda robotun yapacağı işlemler dizisi oluşturulmakta ve daha sonraki kullanımlar için bir kütükte saklanmaktadır. Bu çalışmada geliştirilen sıralama algoritması ve yazılımı uzman sistemin bu modülünde yer almaktadır.

2. ROBOT İŞLEMLERİNİN SIRALANMASI

Robotlu imalat hücresi uzman sisteminin karar mekanizmasını oluşturan iki kısımdan birisi, tezgah yükleme sırasını belirleyen Kayan Darboğazlı Sezgisel Algoritmasıdır (Shifting Bottleneck Heuristic Algorithm-SBH) [10]. Karar mekanizmasının ikinci kısmını ise robot işlemlerini sıralayan algoritmaların yerleştirildiği program blokları oluşturmaktadır. Kayan darboğazlı sezgisel algoritmanın (SBH) uygulanmasıyla her bir tezgahta operasyona girecek iş parçası sırası belirlenir. Ancak, SBH algoritmasını kullanan programın çıkış bilgilerine bakarak robotun yapması gereken işlemler hemen ortaya çıkmamaktadır. İş parçalarının tezgahlara yerleştirilip alınması işlemlerinin mantıklı bir sıra içinde takip edilmesi gereklidir. Bu çalışmada geliştirilen algoritmalar ve “SIRALAMA” adı verilen program, bu robot işlemleri zincirini belirlemektedir. Bu amaçla, literatürde bulunmayan algoritmalar geliştirilmiş ve tamamen kişisel tecrübeye dayalı karar mekanizmaları oluşturulmuştur. “SIRALAMA” programı QUICKBASIC (sürüm 7.1) ile hazırlanmıştır. Daha sonra “exe” hali, uzman sistemdeki “İşlem planlaması” modülüne yerleştirilmiş ve

modülün “metot” kısmında yazılan “exe” programı çalıştırma komutu ile çalışması sağlanmıştır.

SBH ve diğer sıralama algoritmaları iş parçalarının tezgahlarda hangi sıra ile işleme gireceğini belirlemede, ancak herhangi bir zaman bilgisi vermemektedir. Elde edilen sıralama bilgileri ve işlem zamanları dikkate alınarak her tezgahın işleri hangi zamanlarda operasyona alacağını belirlenmesine “çizelgeleme” denmekte [11,12] ve işlemler Gantt diyagramlarında gösterilmektedir. Burada da, robot işlerinin sıraya sokulmasında, hemen hemen bütün aşamalarda Gantt diyagramı üzerinde çalışılacaktır. Diyagramda işlerin başlangıç ve bitiş zamanları dikkate alınarak bir çok kararlar verilecektir. Gantt diyagramı burada hem bilgi kaynağı ve hem de sonuçların görüldüğü bir yer olacaktır.

2.1. Algoritmalar ve Programın Genel Yapısı

Geliştirilen algoritmalar ve SIRALAMA programı tanıtılırken aynı zamanda 4 makina ve 3 iş parçasından oluşan bir sıralama problemi ele alınacaktır. Bu tür problemlerin çözümüne ait yöntemler literatürde de bir örnekle anlatılmakta ve algoritmalar böylece daha kolay izlenebilmektedir [11].

2.1.1. Sıralama programının giriş bilgileri

SIRALAMA programının giriş bilgileri, SBH algoritması programının “ISSIRASI.TXT” isimli çıkış kütüğündeki bilgilerdir. Şekil 2’de bu kütüğün yapısı verilmiştir.

Şekil 2’de her bir makinanın karşısında yatay yönde ilerleyerek ilgili tezgahdaki iş parçası sıraları ve bunlara ait en erken işleme başlayabilme zamanı (serbest kalma zamanı- SKZ) ile proses süresi görülebilmektedir.

Hücredeki makina sayısı:

4

Hücrede aynı anda imal edilecek parça sayısı:

3

| Mak.No | IsNo | S.K.Z. | Pr.Sur. | IsNo | S.K.Z. | Pr.Sur. | IsNo | S.K.Z. | Pr.Sur. | | |
|--------|-------------------------|--------|---------|------|--------|---------|------|--------|---------|----|----|
| 1 | 1 | 0 | 10 | 2 | 10 | 3 | 3 | 13 | 4 | -1 | 17 |
| 2 | 2 | 0 | 8 | 1 | 10 | 8 | 3 | 18 | 7 | -1 | 25 |
| 3 | 1 | 18 | 4 | 2 | 22 | 6 | | | | -1 | 28 |
| 4 | 2 | 13 | 5 | 3 | 25 | 3 | | | | -1 | 28 |
| IsNo | İşin gideceği makinalar | | | | | | | | | | |
| 1 | 1 | 2 | 3 | | | | | | | | |
| 2 | 2 | 1 | 4 | | | | | | | | |
| 3 | 1 | 2 | 4 | | | | | | | | |

Mak.No: Tezgah numarası

IsNo: Tezgahta işlem göreceği iş parçası numarası

S.K.Z.: İş parçasının o tezgahta operasyona girebileceği en erken zaman

Pr.Sur.: İş parçasının o tezgahtaki işlem süresi

Şekil 2. SİRALAMA programının giriş bilgilerini içeren "ISSIRASI.TXT" kütüğünün görünümü

2.1.2. İşlemlerin kutularının oluşturulması ve ilk haliyle Gantt diyagramının elde edilmesi

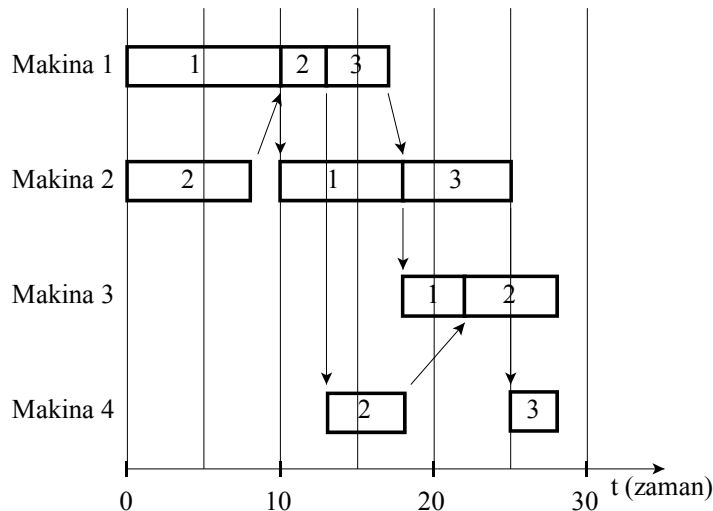
SİRALAMA programı, giriş kütüğünü okuduktan sonra bir Gantt diyagramı oluşturmaktadır. Şekil 3'te görülen diyagramdaki gibi, tezgahlardaki her bir operasyon bir kutu ile gösterilmektedir. Her bir kutuya ait dört temel bilgi mevcuttur. Ayrıca söz konusu operasyon kutusunun bir tezgaha mı yoksa bir iş tablasına mı ait olduğuna dair beşinci bir bilgiye de ihtiyaç duyulmaktadır. Bir kutuyu temsil eden bilgiler şöyledir:

1. KUTU(NN, 1): NN'inci kutuda kullanılan makinan n veya iş tablasının numarası.
2. KUTU(NN, 2): NN'inci kutuda işlem gören iş parçasının numarası.
3. KUTU(NN, 3): NN'inci kutuda işlem gören iş parçasının işlenmesine başlanma zamanı.
4. KUTU(NN, 4): NN'inci kutuda işlem gören iş parçasının bitiş zamanı.
5. KUTU(NN, 5): NN'inci kutu=0 ise imalat tezgahı, = -1 ise iş tablası.

Tanımlanmış kutularla her bir operasyonun yeri kolaylıkla belirlenebilmektedir. "NN" değeri kutu indisidir ve en büyük değeri imalat hücresinde yapılacak toplam operasyon sayısını gösterir. SİRALAMA programı ile robotun yapacağı işlemler dizisi oluşturulurken, tezgah araları ile iş parçalarının ilk ve son operasyona girdikleri tezgahların yanına iş tablaları yerleştirme işlemi de gerçekleştirilmektedir. Programın başında KUTU(NN,5) değerlerinin hepsi de sıfırdır. Çünkü bu aşamada henüz iş tablası yerleştirme işlemine karar verilmemiştir.

2.1.3. Aynı anda yapılan transfer operasyonlarının tespiti ve zaman kaydırmaları yapılması

Robot işlemleri sıralanırken ileride karşılaşılabilecek en büyük sorunlardan biri, robotun elinde iş parçası varken dolu bir tezgaha gitmesidir. Robotun mantıklı bir sıra ile çalışabilmesi için öncelikle iş parçası yükleyeceği tezgahın boş olması gereklidir. Dolayısıyla iş parçasının gideceği tezgah önceden boşaltılmış olmalıdır. Ayrıca, hücrede aynı anda birden fazla yükleme ve boşaltma işlemi yapılması



Şekil 3. Dört makina ve üç işe ait zaman bilgilerini gösteren örnek bir Gantt diyagramı

gerekiyorsa bunların önceliğine karar vermek son derece güç olabilmektedir.

Şekil 3'te, örneğin 10. zaman diliminde 1. tezgahtan 1. iş parçasının alınması, 2. tezgaha 1. iş parçasının yerleştirilmesi ve 1. tezgaha 2. iş parçasının yerleştirilmesi olmak üzere 3 robot işlemi görülmektedir. Burada ilk olarak 1. tezgaha 2. iş parçası koyulamaz; çünkü 1. tezgah doludur. Aynı şekilde 1. tezgah boşaltıldıktan sonra 2. iş parçası aynı tezgaha koyulamaz; bu defa da robot eli doludur. Bu kısmın çözümü için, ilk olarak 1. tezgahtan 1. iş parçası alınmalı, sonra aynı parça 2. tezgaha konulmalı ve son olarak 2. iş parçası bulunduğu yerden alınarak 1. tezgaha koyulmalıdır. Şekil 3'te benzeri bir durum da 18. zaman diliminde görülmektedir.

Şekil 3'teki örnek Gantt diyagramında ortaya çıkan bu aynı zamanlı işlem durumları, ilk bakışta kolayca çözülebilecek bir sıralama olarak gözükmemektedir. Ancak bu durumların çok karmaşık olanları ortaya çıkabilmektedir. Yapılan araştırmalar neticesinde çakışan işlemlerin beş değişik grupta toplandığı belirlenmiş ve bunların her biri için ayrı çözüm yöntemi geliştirilmiştir. Ayrıca ilerideki aşamaların doğru çalışabilmesi için bu aşamada ϵ gibi küçük zaman kaydırmaları yapılması ($\epsilon=0,01$) zorunluluğu ortaya çıkmıştır.

2.1.3.1. Normal ucu açık transfer çevrimi (PROBLEM-1)

Şekil 4a'da örneği görülen bu problem türünde, t_0 anında boş bir makina ve boşaltıldıktan sonra t_0 'da tekrar yükleme yapılmayan bir makina vardır. t_0 anında toplanmış bütün işlemlerde, her bir iş bir makinadan alınacak ve diğer bir makinaya

yüklenecektir. Yani bu anda sadece yüklenecek ya da sadece boşaltılacak bir iş parçası yoktur.

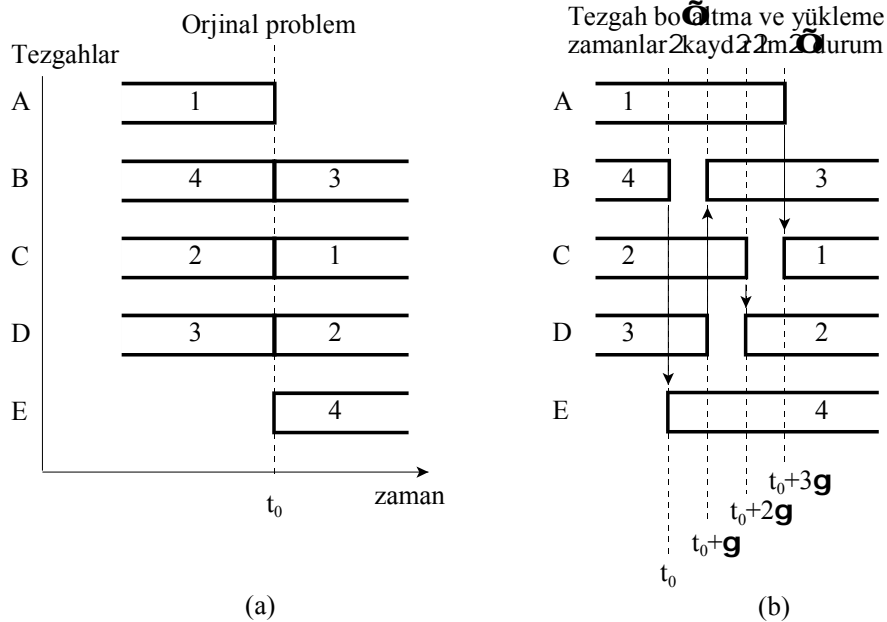
Problemin çözümüne transfer sırası belirlenerek başlanır. İlk transfer boş olan makinaya yapılarak her yeni boşalan makinaya transfere devam edilir ve tüm transfer işlemleri tamamlanır. Şekildeki örnekte transfer zinciri BE, DB, CD ve AC şeklindedir.

SIRALAMA programı çalışırken bu tür problemi tespit ettiğinde, işlemlerin hangi sırayla yapılacağını belirlemek amacıyla transfer zamanlarını aralarında ϵ kadar süre olacak biçimde Şekil 4b'deki gibi kaydırmaktadır. Burada görüldüğü gibi, aynı anda yapılacak işlemler, ϵ kadar farklı zaman aralıklarıyla sadece "önce boşalt - sonra yerleştir" şeklinde gruplara dönüşmektedir.

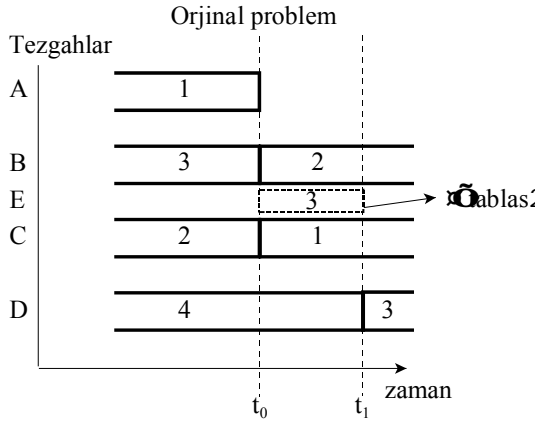
2.1.3.2. Dejenere ucu açık transfer çevrimi tip-1 (PROBLEM-2)

Bu sıralama problemi türünde, t_0 anında boşaltılan ve başka iş yüklenmeyen bir makina ile bir iş boşaltılıp başka bir iş yüklenen makinalar vardır. İş parçaları biri dışında, bir makinadan alınıp diğer makinaya t_0 anında yüklenmektedir. Parçalardan birisi ise t_0 anında herhangi bir makinadan alınmakta, ancak diğer makinaya t_0 'dan daha sonraki bir zamanda yüklenmektedir (Şekil 5).

Transfer işlemlerini gerçekleştirebilmek için t_0 zamanında B tezgahından boşaltılan 3 no.lu parçayı t_1 zamanına kadar bekletme zorunluluğu nedeniyle bir iş tablasının sisteme eklenmesi gereklidir. Yerleştirilen bu iş tablası hayali bir tezgah gibi düşünülerek şekildeki orijinal problem "Problem-1" tipindeki bir sıralama problemine dönüştürülür. Bu dönüşüm ile problem-1'in çözüm kuralları uygulanır ve hayali E



Şekil 4. Normal ucu açık transfer çevrimine (Problem-1) ait bir örnek



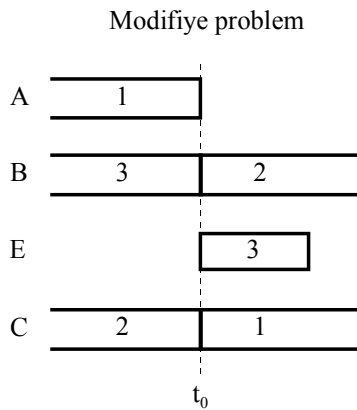
Şekil 5. Dejenere ucu açık transfer çevrimi tip-1'e ait bir örnek sıralama problemi

tezgahı t_0 'da boş olduğundan ilk yüklemeye buradan başlanır. Şekil 5'teki problemin modifiye edilmiş durumu ve çözümden sonra oluşacak yeni Gantt diyagramı Şekil 6'da verilmiştir. Görüldüğü gibi, işlem sıralamalarını belirlemek amacıyla transfer zamanları, aralarında ε kadar süre olacak biçimde kaydırılmıştır. Çözülmüş problemde transfer zinciri BE, CB ve AC olacaktır.

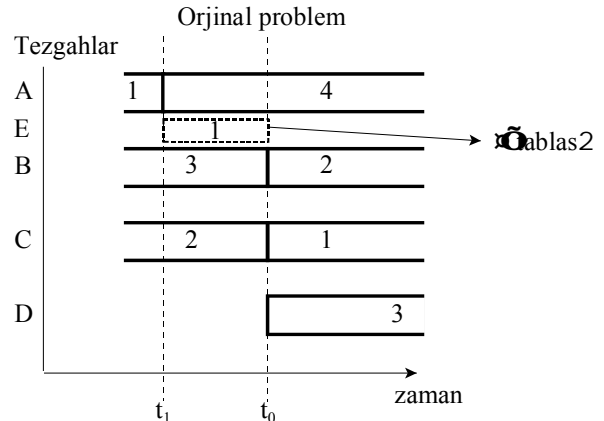
2.1.3.3. Dejenere ucu açık transfer çevrimi tip-2 (PROBLEM-3)

Bu sıralama problemi türünde, t_0 anında yüklenen ve önceden boş olan bir makina ile bir iş boşaltılıp başka bir iş yüklenen makineler vardır. Biri dışında bütün iş parçaları bir makinadan alınıp diğer bir makina t_0 anında yüklenmektedir. Parçalardan birisi ise t_0 anında herhangi bir makina yüklenmekte ancak diğer makinadan t_0 'dan daha önceki bir zamanda (t_1) alınmaktadır (Şekil 7).

Bu problemde transfer işlemlerini yapabilmek için t_1 zamanında A tezgahından alınan 1 no.lu iş parçasını stoklamak üzere bir iş tablasının tezgah çıkışına yerleştirilmesi gereklidir. Yerleştirilen bu iş tablası yine hayali bir tezgah gibi düşünülerek şekildeki orijinal problem "Problem-1" tipindeki bir sıralama



Şekil 6. Problem-2 türü modifiye edilmiş sıralama problemi ve çözümden sonraki Gantt diyagramı



Şekil 7. Dejenere ucu açık transfer çevrimi tip-2'ye ait bir örnek sıralama problemi

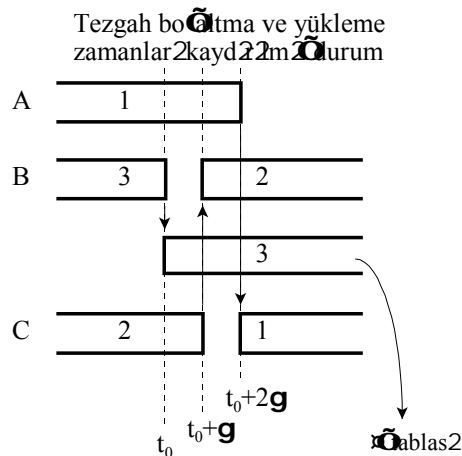
problemine dönüştürülür. Bu dönüşümden sonra problem-1'in çözüm kuralları uygulanır ve ilk yükleme önceden boş olan D tezgahına yapılır. Çözülmüş problemde transfer zinciri BD, CB ve EC olacaktır. Şekil 7'deki problemin modifiye edilmiş durumu ve çözümden sonra oluşacak yeni Gantt diyagramı Şekil 8'de verilmiştir.

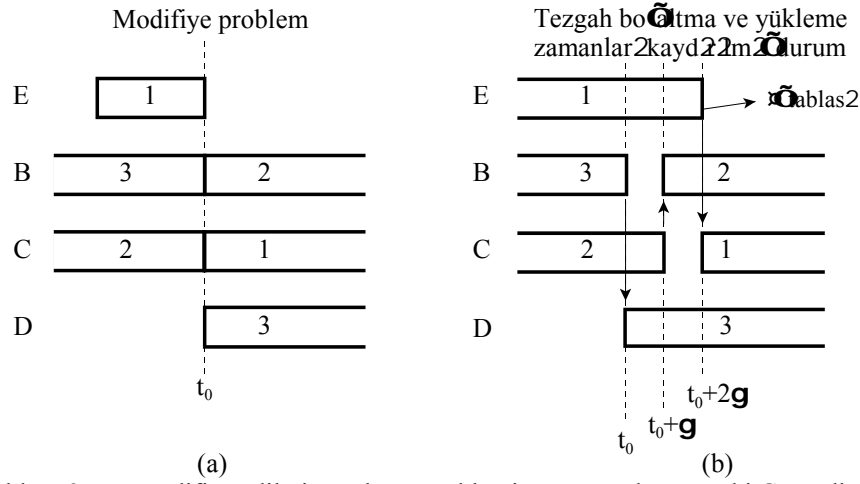
Görüleceği gibi, D tezgahına ilk yüklemenin yapılmasıyla B tezgahı boşalır ve bu tezgaha bir sonraki iş parçasını yükleme zamanı ε kadar kaydırılır. Bu şekilde sırayla boşalan her tezgahın yükleme zamanı ε kadar ileri kaydırılarak Şekil 8b'deki Gantt diyagramı elde edilir.

2.1.3.4. Kapalı transfer çevrimi (PROBLEM-4)

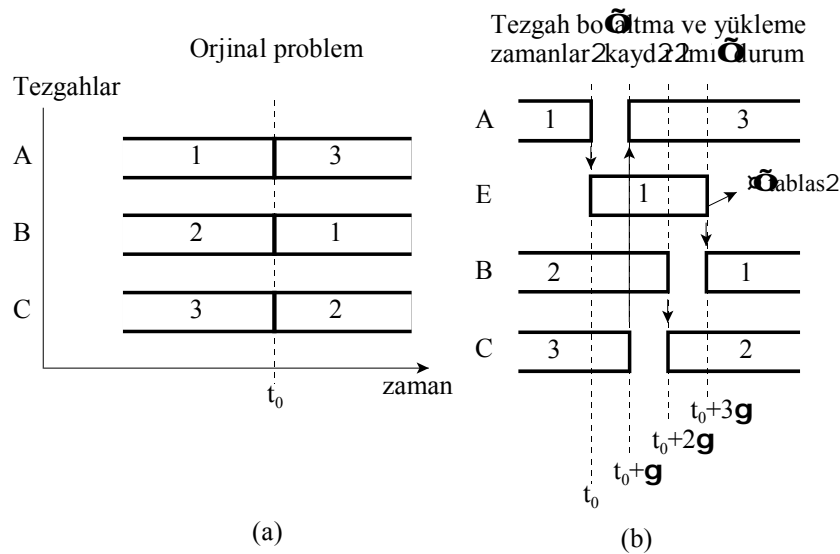
Şekil 9'da örneği görülen bu problem türünde, t_0 zamanında, aynı anda boşaltılan ve yüklenen tezgahlar mevcuttur. Yani, boş bir makina veya boşaltıldıktan sonra tekrar yükleme yapılmayan bir makina yoktur. t_0 anında toplanmış bütün işlemlerde, her bir parça bir makinadan alınacak ve diğer bir makina yükleneyecektir.

Bu problem herhangi bir tezgahın çıkışına bir iş tablası yerleştirilerek Problem 1'e dönüştürülebilir.





Şekil 8. Problem-3 türü modifiye edilmiş sıralama problemi ve çözümüden sonraki Gantt diyagramı hali



Şekil 9. Kapalı transfer çevrimine (Problem-4) ait bir örnek

Şekil 9'da böyle bir iş tablası A tezgahının çıkışına koyulmuştur. Böylece işlem başlangıcında bir boş tezgah (iş tablası) elde edilmiş olur. Önce A tezgahından bu boş tezgaha yükleme yaparak ve her boşalan tezgahın yükleme zamanını ϵ kadar ileri atarak transfer işlemleri tamamlanır. Buna göre Şekil 9'daki örnek için oluşacak transfer zinciri AE, CA, BC ve EB şeklindedir. Yükleme zamanlarının kaydırılması sonucu elde edilen yeni Gantt diyagramı ise Şekil 9b'de verildiği gibidir.

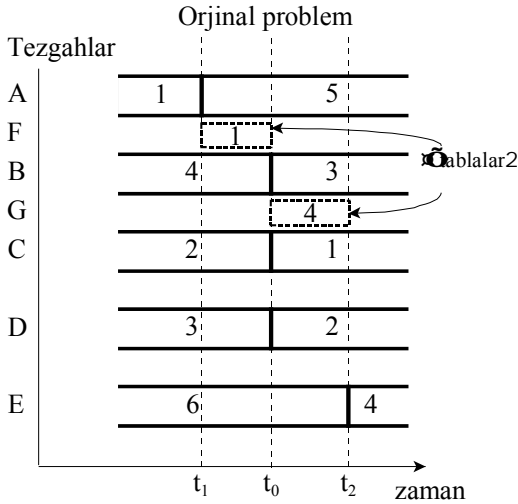
2.1.3.5. Kompozit dejenere ucu açık transfer çevrimi (PROBLEM-5)

Bu problem türü Problem 2 ve Problem 3'ün özelliklerini bir arada taşır. t_0 anında sadece boşaltılan veya sadece yüklenen tezgah yoktur. Problemde t_0 zamanında bir iş herhangi bir makinaya yüklenecektir, ancak bu iş t_0 'dan önceki bir t_1 zamanında başka bir makinadan alınmıştır. Aynı şekilde, t_0 zamanında bir iş herhangi bir makinadan alınacaktır, ancak bu işin bir başka makinaya yüklenmesi t_0 'dan daha sonraki bir t_2 zamanında olacaktır.

Bu transfer probleminin bir örneği Şekil 10'da verilmiştir. Şekilden anlaşılacağı gibi, 1 no.lu iş parçası A tezgahından alındıktan sonra C tezgahına yüklenebilmek için beklemektedir. Yine aynı şekilde, 4 no.lu parça B tezgahından alındıktan sonra E tezgahına yüklenebilmek için bekleyecektir.

Bu transfer işleminin gerçekleştirilebilmesi için 1 no.lu iş parçasını A tezgahından aldıktan sonra stoklamak için bir iş tablası (F), 4 no.lu iş parçasını B tezgahından aldıktan sonra stoklamak için de diğer bir iş tablası (G) kullanılması gerekmektedir. Bu iş tablaları koyulursa A ve E tezgahlarının Problem 5'in çözümünde dikkate alınmasına gerek kalmamaktadır. Elde edilen modifiye problem Şekil 11'in solundaki gibidir. Elde edilen yeni problem, Problem 1'e benzemektedir.

Problemin çözümüne boş olan tezgaha (iş tablası G) ilgili iş parçasını yükleyerek başlanır. Bunun sonucunda boşalan tezgahın (B) yükleme zamanı ϵ kadar ileri kaydırılarak, işlemlere benzer biçimde bütün transferler tamamlanıncaya kadar devam edilir.



Şekil 10. Kompozit dejenere ucu açık transfer çevrimine (Problem-5) dair bir örnek

Şekil 11'deki problem için bu yöntemle elde edilecek transfer zinciri BG, DB, CD ve FC biçimindedir. Yükleme zamanlarının ileriye doğru kaydırılması neticesinde, çözümlenen problemin Gantt diyagramı yapısı Şekil 11'in sağ tarafında görüldüğü gibi olacaktır.

2.1.4. İmalat hücresine iş tablalarının yerleştirilmesi

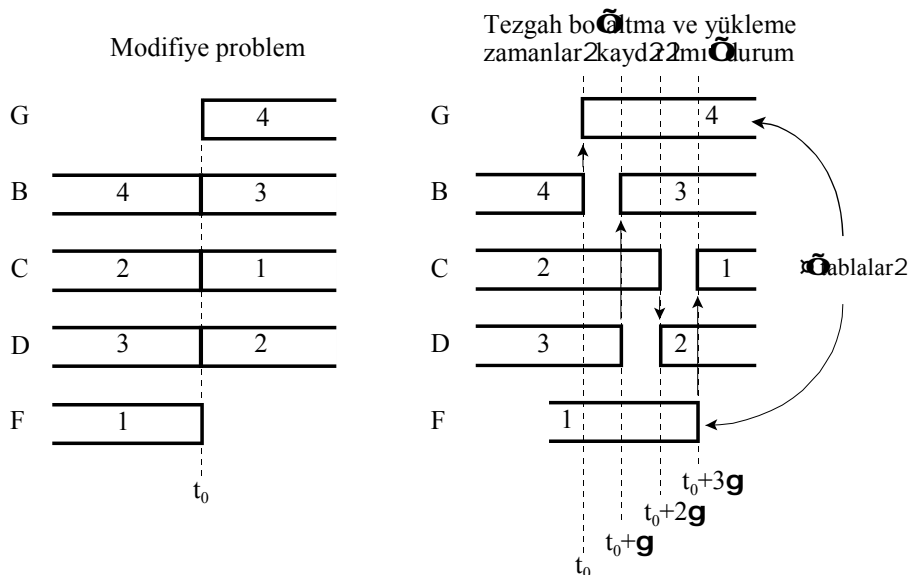
Yukarıda kısım 2.1.3'te verilen örneklerden de görüldüğü gibi, tezgah yükleme zamanlarının boşaltma zamanlarına göre ϵ kadar süreyle ileri kaydırılması ve çeşitli tezgahların boşaltma/yükleme sıralamalarının belirlenmesi için iş tablalarının fiili olarak yerleştirilmesi gerekli değildir. Önemli olan, problem türünün tanımlanması, boşaltma işlemine doğru tezgahla başlanması ve sırasıyla boşaltılan her tezgahın yükleme zamanının boşaltma zamanına göre ϵ kadar ileri kaydırılmasıdır. SIRALAMA programı ilk aşamada sadece transfer sıralaması ve yükleme

zamanı kaydırma işlemlerini yapmaktadır. İkinci aşamada ise bu şekilde düzenlenmiş ve boşaltma/yükleme zamanları kaydırılmış Gantt diyagramlarına dayanarak, yukarıdaki özel problemlere çözüm getirecek ve kilitlemeleri önleyecek biçimde, tezgah yanlarına iş tablaları yerleştirilmektedir.

İş tablaları üç farklı noktaya yerleştirilmektedir. Bunlar operasyon aralarına, her işin ilk operasyonun olduğu tezgah yanına ve her işin son operasyonunun yapıldığı tezgah yanına şeklindedir.

Operasyon aralarına yerleştirilecek iş tablalarına karar verebilmek için tezgahlardaki bekleme sürelerinin göz önüne alınması gerekmektedir. Bekleme sürelerinin belirlenmesi için Gantt diyagramına bakılmalıdır. Şekil 3'e bakıldığında, işlerin bir tezgahtan çıktuktan sonra nereye gideceğini bildiren oklar görülecektir. Eğer imalat sırasında herhangi bir bekleme söz konusu değilse bu oklar dikey olarak yer alacaklardır. Ama iş parçası diğer tezgaha gitmek için bekliyorsa ok eğimli olacaktır. Gantt diyagramında böyle eğimli okların bulunduğu zamanlarda iş tablası bulundurma ihtimali vardır. Diğer zamanları kontrol etmeye gerek yoktur. Şekil 3'te 8-10, 17-18 ve 18-22. zaman dilimleri, iş tablası yerleştirilebilecek yerlerdir.

Ayrıca Bölüm 2.3.3'te anlatılan aynı zamanlı parça transferlerinde, Problem 1 dışındaki durumlarda, belirli tezgahlardan sonra iş tablası koyulması gerekmektedir. Aşağıda verilen "iş tablası yerleştirme kriteri" hem eş zamanlı parça transferi yapılan bu özel durumlarda hem de genel durumlarda gereken iş tablalarını sisteme ilave etmektedir. Eğer Bölüm 2.3.3'te anlatılan aynı zamanlı parça transferleri varsa, aşağıdaki kural uygulanmadan önce ilgili kısımlarda açıklandığı gibi boşaltma ve yükleme zamanları kaydırmaları yapılarak modifiye edilmiş Gantt diyagramları elde edilmelidir.



Şekil 11. Modifiye edilmiş problem-5 türü çevrim ve çözümlenmiş halde Gantt diyagramı yapısı

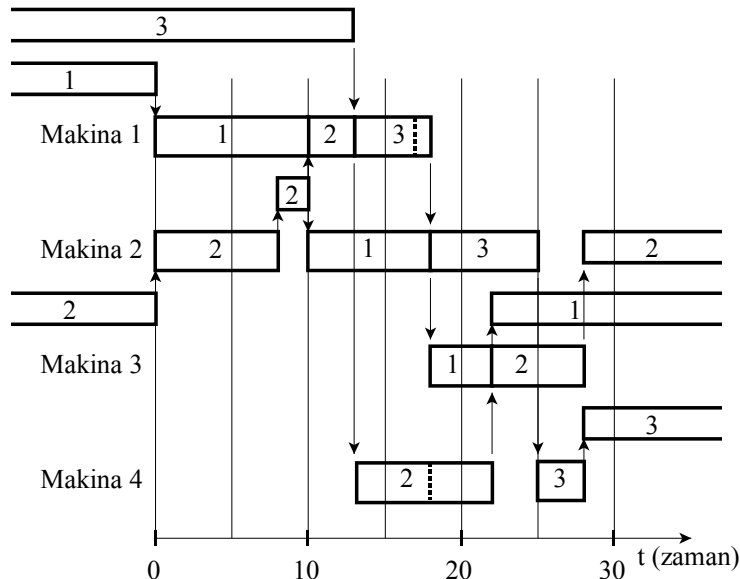
Kural: *Eğer bir işin herhangi bir tezgahdaki bitme zamanı ile, bir sonraki tezgahdaki başlama zamanı arasında süre varsa ve boşaltılacak tezgaha daha önce başka bir iş koyuluyorsa, bu işi bu süre içinde tutmak için tezgahın çıkışına iş tablası koy.*

Yukarıdaki iş tablası yerleştirme kriteri SIRALAMA programına yerleştirilmiştir. Yapılan testlerde, bu kriterin Bölüm 2.3.3'teki özel durumlarda doğru olarak çalıştığı görülmüştür. Ayrıca Şekil 3'teki örnek Gantt diyagramı üzerinde çalıştırıldığı zaman, 8. zaman diliminde 2. tezgahta işi biten 2. parçayı 10. zaman diliminde 1. tezgaha koyabilmek için 2. tezgahın çıkışına iş tablası yerleştirilmesi gerektiğini belirlemiştir. Çünkü 10. zaman diliminde hem 1 ve hem 2. tezgah ayrı bir iş parçasını işleyeceğinden, 2. tezgahın boşaltılmış olması gerekmektedir. Program, 17 ve 18. zaman dilimlerinde ise bir iş tablası yerleştirmemiştir. Çünkü burada, işlemi biten iş parçası diğer tezgahta operasyona başlayana kadar boşalan tezgaha bir başka iş parçası gelmemektedir. Dolayısıyla işi biten parça tezgahta bekleyebilir. Bu durumda, bekleme olduğu halde iş tablası yerleştirilmeyen kısımlarda, daha sonraki program aşamalarının doğru çalışabilmesi için kutuların bitiş zamanlarını kaydırmak gereği vardır. Aksi takdirde, örneğin, 17. zaman diliminde robot 1. tezgahtan 3. parçayı alacak, ancak nereye götüreceğini bilemeyecektir. Yapacağı en mantıklı işlem, 3. parçanın 2. tezgaha yüklendiği 18. zaman dilimine kadar parçayı elinde tutmaktır. Ancak bu kez de 18. zaman diliminde 2. tezgahı boşaltması gerekmektedir. Yani robot, eli dolu iken bir tezgaha iş almak için gelecektir. Bu kilitlenme durumunu ortadan kaldırmak için operasyonu biten parçanın kutusunun bitiş zamanını ($KUTU(NN, 4)$) diğer tezgahta işleme gireceği zamana kadar kaydırmak yeterli olacaktır. SIRALAMA programına sözü geçen bu işlemi yapan kısımlar yerleştirilmiştir.

İmalat sisteminde iş tablası yerleştirilecek diğer kısımlar ise, her bir işin ilk ve son operasyona girdiği tezgahların yanındır. Robot, imalatı başlatabilmek için ham malzeme tablasından iş parçasını alacak ve ilk tezgaha yükleyecektir. Ham malzeme ise iş tablasına başka bir kaynaktan konulmaktadır. Aynı şekilde robot, imalatı biten parçayı tezgahtan alıp bitmiş ürün tablasına koyacaktır. Bu tabla da bir başka kaynak tarafından boşaltılacaktır. Buna göre, imalat hücresinde hem giriş ve hem çıkışa, imal edilecek parça türü kadar iş tablası yerleştirilmelidir. Bu ihtiyacı sağlamak için SIRALAMA programına bu iş tablalarını koyan satırlar da yerleştirilmiştir.

İş tablaları program içinde tezgahlar gibi tanımlanmakta ve işlem görmektedir. Yani bunların da kutuları vardır. Başlangıca ve sona yerleştirilen iş tablalarının durumu ise biraz farklıdır. Başlangıçtaki iş tablaları robot tarafından yüklenmediği için bunları temsil eden kutuların başlama zamanı "-1" gibi imkansız bir zaman olarak tanımlanmıştır. Aynı şekilde bitişteki iş tablaları da robot tarafından boşaltılmadığından, bunların boşalma zamanı "999" gibi toplam imalat süresinin çok üstünde bir değerle tanımlanmıştır. Robot işlemlerinin sıralanması esnasında, program bu "-1" ve "999"lu değerlere sahip zamanları dikkate almamaktadır. Şekil 3'te görülen imalat sistemine ilave edilmiş iş tablaları ve zamanları kaydırılmış kutularla oluşan yeni Gantt diyagramı Şekil 12'de verilmiştir.

Tezgah kutularının 5. değeri daha önceden bir şey tanımlanmadığı için sıfırdır. İş tablası yerleştirme kısmında yapılan diğer bir işlem ise, iş tablalarına ait kutuların 5. değerinin "-1" yapılarak ($KUTU(NN, 5) = -1$) bunların iş tablası olduğunun işaretlenmesidir.



Şekil 12. İş tablaları ilave edilmiş ve zamanları kaydırılmış kutulardan oluşan imalat hücresinin Gantt diyagramı

2.1.5. Yerleştirme ve alma işlemlerinin artan zamana göre sıralanması

Robot işlemlerinin ilerleyen zamana göre sıralanması gerekmektedir. Gantt diyagramında görülen her bir kutunun başlangıç ve bitiş zamanları birer robot operasyonunu belirtmektedir. Dolayısıyla programda kutuların başlangıç ve bitiş zamanları küçükten büyüğe doğru sıralanacaktır.

Sıralama işleminin yapılabilmesi için her bir kutunun başlangıç ve bitiş değerlerinin bir tek dizide toplanması gerekmektedir. Bunun için iki boyutlu yeni bir dizi daha program içerisinde tanımlanmıştır. Bunun tanıtımı şöyledir:

$SIRALIZAMAN1(NNN, 1) \equiv$ Kutu numarası.
 $SIRALIZAMAN1(NNN, 2) \equiv$ Başlangıç veya bitiş bilgisi (3: yükleme (başlangıç), 4: boşaltma (bitiş)).

Programın başlangıcında $SIRALIZAMAN1(NNN, 2)$ değeri sıfırdır. Daha sonra zamanların sıraya sokulmasında, önce *alma* ve sonra *koyma* işlemleri ayrı ayrı sıralanacak ve o anda $SIRALIZAMAN1(NNN, 2)$ değeri 3 veya 4 olacaktır.

Bundan sonra $SIRALIZAMAN1$ dizisi artan zamana göre yeniden sıralanacaktır. $SIRALIZAMAN1$ dizisindeki kutuların önce başlangıç değerleri dikkate alınarak sıralama yapılmakta ve dizinin 2. boyutuna "3" değeri verilmektedir. Daha sonra ise kutuların bitiş değerleri dikkate alınarak sıralama yapılmakta ve bu kez de dizinin 2. boyutuna "4" değeri verilmektedir. Böylece $SIRALIZAMAN1$ dizisinde önce 2. değeri "3" ve başlangıç zamanları küçükten büyüğe sıralanmış kutu numaraları, ardından da "4" değerine sahip 2. değerler ve bitiş zamanları küçükten büyüğe sıralanmış kutu numaraları yer almaktadır.

Başlangıç ve bitiş değerlerini gösteren zamanların aynı anda sıralanması mümkün olmamaktadır. Aynı anda sıralama yapıldığında, sıraya giren kutunun başlangıcının mı, yoksa bitişinin mi yer aldığı bilinmemektedir. Dolayısıyla programın doğru çalışabilmesi için iki ayrı sıralama yapıp daha sonra bunların artan zamana göre birleştirilmesi gerekmektedir. Bu birleşik sıralama ile elde edilen yeni diziye " $SIRALIZAMAN$ " adı verilmiştir ve bu dizi de tıpkı $SIRALIZAMAN1$ dizisi gibi olup iki boyutludur. Dizinin birinci değeri kutu numarasını ikinci değeri ise başlangıç (yükleme) veya bitiş (alma) bilgisini taşır.

2.1.6. Aynı iş parçasını alan ve koyan işlemlerin önce alma sonra koyma şeklinde ard arda getirilmesi

Yukarıdaki işlemlerle elde edilmiş yeni dizinin

($SIRALIZAMAN$), kutuların sadece başlangıç ve bitiş zamanları dikkate alınarak sıraya sokulması ve aldığı değerlere karşılık gelen kutu bilgileriyle elde edilen birkaç satır

$SIRALIZAMAN(7,1)=11$ $SIRALIZAMAN(7,2)=4$
 $KUTU(11,4)=10$ $KUTU(11,2)=2$ $KUTU(11,1)=2$
 $SIRALIZAMAN(8,1)=5$ $SIRALIZAMAN(8,2)=3$
 $KUTU(5,3)=10$ $KUTU(5,2)=1$ $KUTU(5,1)=2$
 $SIRALIZAMAN(9,1)=2$ $SIRALIZAMAN(9,2)=3$
 $KUTU(2,3)=10$ $KUTU(2,2)=2$ $KUTU(2,1)=1$
 $SIRALIZAMAN(10,1)=1$ $SIRALIZAMAN(10,2)=4$
 $KUTU(1,4)=10$ $KUTU(1,2)=1$ $KUTU(11,1)=1$

şeklinde. Bu satırlara bakıldığında, 10. zaman diliminde (üçüncü sütun) 4 operasyonun varlığı anlaşılmaktadır. İkinci sütuna bakıldığında görülen "4" değeri kutunun bitiş değerini gösterir ve iş parçasının tezgahtan veya iş tablasından alınacağını gösterir. Aynı şekilde "3" kutunun başlangıç değerini gösterir ve iş parçasının tezgaha veya iş tablasına yerleştirileceğini gösterir. Dördüncü sütunda iş parçası numaraları vardır. Beşinci sütunda ise tezgah numaraları sıralanmıştır.

Bu sıralamaya göre ilk bakışta, önce 2. tezgahtan 2. işin alınacağı, sonra 2. tezgaha 1. işin koyulacağı, daha sonra 1. tezgaha 2. işin koyulacağı ve en sonunda 1. tezgahtan 1. işin alınacağı bilgilerine ulaşılmaktadır. Görüldüğü gibi, robot 2. tezgahtan 2. işi alacak fakat bunu bırakmadan önce 1. işi 2. tezgaha koymaya çalışacaktır. Yani eli dolu iken bir başka işi taşımaya kalkışacaktır. Bu mantıksız işlemler zinciri verilen örnekte 4. satırda da sürmektedir. Halbuki, robot eline aldığı parçayı boş olan bir tezgaha yüklemelidir. Ancak bu şekilde mantıklı işlemler dizisi elde edilebilir. Buna göre, **robot önce eline parçayı alacak ve sonra aynı parçayı önceden boş olan bir tezgaha koyacaktır.** Bu sırayı sağlayabilmek için önce aynı iş numarasına sahip satırların (dördüncü sütun) peş peşe getirilmesi gerekmektedir. Bunun ardından aynı işe sahip her iki satır kendi içinde *önce alma (4) - sonra koyma (3)* (ikinci sütun) şeklinde yeniden düzenlenmelidir. Bu şartlara göre düzenlenmiş yukarıdaki dört satır şöyle olacaktır:

| | | | | |
|---------------------|---------------------|---------------|--------------|--------------|
| SIRALIZAMAN(7,1)=11 | SIRALIZAMAN(7,2)=4 | KUTU(11,4)=10 | KUTU(11,2)=2 | KUTU(11,1)=2 |
| SIRALIZAMAN(8,1)=2 | SIRALIZAMAN(8,2)=3 | KUTU(2,3)=10 | KUTU(2,2)=2 | KUTU(2,1)=1 |
| SIRALIZAMAN(9,1)=1 | SIRALIZAMAN(9,2)=4 | KUTU(1,4)=10 | KUTU(1,2)=1 | KUTU(11,1)=1 |
| SIRALIZAMAN(10,1)=5 | SIRALIZAMAN(10,2)=3 | KUTU(5,3)=10 | KUTU(5,2)=1 | KUTU(5,1)=2 |

Sayılan bu kurallara göre SIRALIZAMAN dizisini yukarıdaki gibi düzenleyen program satırları SIRALAMA programında yer almıştır.

2.1.7. Robot işlem sıralamasının makina ve iş tablası arasındaki öncelik sırasına göre yeniden sıraya sokulması

SIRALAMA programında, yukarıda anlatılan işlemleri yapan program bloklarıyla, iş parçalarının önce alınması sonra da koyulması şeklinde işlemler sıraya sokulmuş olmaktadır. Yapılan bu sıralama robotun doğru zamanda doğru işlemleri yapabilmesi için yeterli değildir. Ayrıca, *robotun elindeki işin gideceği tezgahın önceden boş olması gereklidir*. Yani robot dolu tezgaha iş parçası yerleştirmeye kalkışmamalıdır. Buradan, aynı anda yapılan işlemler arasında, öncelikle “tezgahların boşaltılması” işleminin yapılması gerektiği şartı ortaya çıkmaktadır. Boş olan tezgahlara iki kaynaktan iş parçası gelebilir. Bu kaynaklar ya bir tezgah, ya da bir iş tablası olabilir. Yani, ya tezgahtan tezgaha, ya da iş tablasından tezgaha şeklinde iki tür iş parçası transferi olabilir. Burada da bir öncelik söz konusudur. Bir parçayı tezgahtan alıp bir başka tezgaha götürmek zaman kazandıracaktır. Çünkü bir tezgahtan çıkan parça iş tablasına uğramadan diğer tezgaha geçecek ve sadece robot intikal süresi kadar bir süre sonunda iş parçası operasyonlarına devam edebilecektir. Tezgahtan tezgaha transferin diğer bir amacı da aynı anda bir tezgahın daha boşaltılmış olmasıdır ki, boşaltılan bu tezgah bir başka işi yüklenmeye hazırdır. Anlatılan bu bilgilere göre sıralamada ikinci önceliği “tezgahtan tezgaha transferler” olacaktır.

Aynı anda yükleme yapılan işlemler arasında, geriye kalan robot işlemleri ise iş tablasından tezgaha yüklenen parçaların transferi işlemleridir. Dolayısıyla transferde son öncelik “iş tablasından tezgaha” parçaların taşınmasıdır. Yukarıda sayılan bu üç öncelik şöyle sıralanıp özetlenebilir:

| <u>Öncelik sırası</u> | <u>İşlemin adı</u> |
|-----------------------|-------------------------|
| 1 | Makinadan iş tablasına |
| 2 | Makinadan makineye |
| 3 | İş tablasından makineye |

Bu aşamada öncelik kazandırılmayan tek şey aynı tür işlemlerin kendi içinde sıralandırılmasıdır. Programda aynı tür taşıma işlemlerinin sırası rasgeledir. Ancak bu durum kilitlenmeye yol açmamaktadır. Sadece intikal zamanları nedeniyle toplam imalat süresi etkilenebilir.

SIRALAMA programına yerleştirilen bu kurallarla robotun yapacağı işlemler mantıklı bir sıraya dizilmiş

olmaktadır. Şekil 12'deki Gantt diyagramına bakıldığında, 0, 10, 13, 18, 22 ve 28. zaman dilimlerinde aynı anda yapılacak işlemler görülmektedir. Burada 0. zamanda iki adet iş tablasından tezgaha transfer vardır ve yeni yapılan bir şey yoktur. 10. zaman diliminde ise 1 adet tezgahtan tezgaha ve 1 adet de iş tablasından tezgaha transfer vardır. Sıralama önceliği sonucunda, önce 1. tezgahtan 2. tezgaha 1. iş parçası taşınacak ve böylece 1. tezgah boşalmış olacaktır. Sonra da iş tablasında bulunan 2. iş parçası boşalmış olan 1. tezgaha yüklenecektir. 10. zaman dilimindeki çözümde ilginç bir nokta yakalanmıştır: Burada, yukarıda sayılan önceliklerle herhangi bir kilitlenme olmadan sıralama yapılabilen ve 1. tezgaha 2. iş parçası son olarak yüklenmektedir. Fakat programın başında yapılan beş tür problem tipi ve buna bağlı ϵ zaman kaydırmaları neticesinde, program bu zaman diliminde bir adet problem-3 türü transfer çevrimi tespit etmiş ve bunun çözümü sonucunda 1. tezgahın yükleme zamanını ϵ kadar ileri almıştır. Böylece 10. zamanda tezgahtan tezgaha bir tek işlem ve $10+\epsilon$. zamanda da iş tablasından tezgaha bir tek işlem olarak sıralama çıkmıştır. SIRALAMA programında ϵ zaman kaydırması işlemleri aynı anda tek işlem esasına göre yapıldığında, öncelik belirlemede ise tezgah boşaltılmasına ilk öncelik verilmiştir. Görüldüğü gibi, her iki çözüm yöntemi farklı bir esasa göre yürütülmekte fakat 10. zamanda her ikisinin de verdiği sonuç aynı olmaktadır. Bu, programda geliştirilen algoritmaların güvenilir olduğunu göstermektedir.

Şekil 12'deki diyagramda, 18. zaman diliminde 2 adet tezgahtan tezgaha yükleme işlemi mevcuttur. Burada öncelikli yöntem bir sonuç vermemektedir. Ancak programın başında 2. ve 3. makinadaki transferler arasında problem-3 türü bir çevrim bulunmuş ve bunun sonucunda 3. iş parçasının yüklenme zamanı ϵ kadar ileri alınmıştır. Dolayısıyla burada önce 1. iş parçasının sonra da 3. iş parçasının transferi yapılarak akıcılık sürdürülmüştür.

22. zamanda yine 1 adet tezgahtan tezgaha ve 1 adet tezgahtan iş tablasına transfer vardır. Burada da önce tezgahtan iş tablasına transfer yapıldığından bir kilitlenme söz konusu değildir. Gantt diyagramının sonunda yer alan 28. zamanda 2 adet tezgahtan iş tablasına taşıma vardır ve bunların sırası rasgeledir.

2.1.8. Sonuçların çıkış kütüğüne kaydedilmesi

SIRALAMA programının çıktısı yine bir çıkış kütüğünde toplanmıştır. Şekil 13'te, incelenen örnek problemin çıkış kütüğü görülmektedir. Burada, ikinci

| İşlem sırası | Zaman | ROBOT İŞLEMLERİ SIRASI | | Açıklamalar |
|-----------------|-------|------------------------|-------|-------------------------|
| | | Tezgah no | İş no | |
| 1 | 0 | 2 | 2 | İş tablasından al. |
| 2 | 0 | 2 | 2 | Tezgaha yerleştir. |
| 3 | 0 | 1 | 1 | İş tablasından al. |
| 4 | 0 | 1 | 1 | Tezgaha yerleştir. |
| 5 | 8 | 2 | 2 | Tezgahtan al. |
| 6 | 8 | 2 | 2 | İş tablasına yerleştir. |
| 7 | 10 | 1 | 1 | Tezgahtan al. |
| 8 | 10 | 2 | 1 | Tezgaha yerleştir. |
| 9 | 10 | 2 | 2 | İş tablasından al. |
| 10 | 10 | 1 | 2 | Tezgaha yerleştir. |
| 11 | 13 | 1 | 2 | Tezgahtan al. |
| 12 | 13 | 4 | 2 | Tezgaha yerleştir. |
| 13 | 13 | 1 | 3 | İş tablasından al. |
| 14 | 13 | 1 | 3 | Tezgaha yerleştir. |
| 15 | 18 | 2 | 1 | Tezgahtan al. |
| 16 | 18 | 3 | 1 | Tezgaha yerleştir. |
| 17 | 18 | 1 | 3 | Tezgahtan al. |
| 18 | 18 | 2 | 3 | Tezgaha yerleştir. |
| 19 | 22 | 3 | 1 | Tezgahtan al. |
| 20 | 22 | 3 | 1 | İş tablasına yerleştir. |
| 21 | 22 | 4 | 2 | Tezgahtan al. |
| 22 | 22 | 3 | 2 | Tezgaha yerleştir. |
| 23 | 25 | 2 | 3 | Tezgahtan al. |
| 24 | 25 | 4 | 3 | Tezgaha yerleştir. |
| 25 | 28 | 4 | 3 | Tezgahtan al. |
| 26 | 28 | 4 | 3 | İş tablasına yerleştir. |
| 27 | 28 | 3 | 2 | Tezgahtan al. |
| 28 | 28 | 3 | 2 | İş tablasına yerleştir. |

İMALAT HÜCRESİNE YERLEŞTİRİLMESİ GEREKEN İŞ TABLALARI VE YERLERİ
Operasyon aralarında ihtiyaç duyulan iş tablaları:
 Tezgah no Koyulacak iş parçası
 2 2
Ham iş parçalarının ilk operasyonuna ait tezgah girişindeki iş tablaları:
 Tezgah no Koyulacak iş parçası
 1 1
 2 2
 1 3
İş parçalarının son operasyonunun yapıldığı tezgah çıkışındaki iş tablaları:
 Tezgah no Koyulacak iş parçası
 3 1
 3 2
 4 3

Şekil 13. SIRALAMA programının çıkış bilgilerini içeren "SIRALAMA.TXT" kütüğünün görünümü

sütunda her bir işlemin yapılacağı zamanlar görülmektedir.

Üçüncü sütunda ise işlemin hangi tezgahta (ya da tezgah yanındaki iş tablasında) gerçekleşeceği bilgisi yer almaktadır. Dördüncü sütunda ise taşınacak parçanın numarası yer almıştır.

"Açıklamalar" sütununda yer alan bilgiler önemlidir. "Tezgah No" sütununda yer alan rakamlar, işlemin tezgahta mı ya da iş tablasında mı gerçekleşeceği bilgisini vermemektedir. Bu bilgi için "açıklamalar" sütunundaki bilgilere bakılmalıdır. Örneğin, şekilde birinci sıradaki işleme bakıldığında, 2. tezgahtan 2. işin alınması ya da koyulması işlemi anlaşılmaktadır.

Ancak açıklamalara bakıldığında 2. tezgahın yanındaki iş tablasından 2. iş parçasının alınması işleminin yapılacağı anlaşılmaktadır. İkinci satırdaki bilgilerde de tezgah numarasının 2 ve iş numarasının 2 olduğu görülmektedir. Açıklamaya bakıldığında ise, 1. satırdaki işlemle robotun elinde bulunan 2 no.lu iş parçasının 2. tezgaha konulması işlemi anlaşılmaktadır.

Bir iş parçasının herhangi bir yerden alınıp başka bir yere konulması işleminin izlenebilmesi için iki satıra birlikte bakılmalıdır. Zaten dördüncü sütunda iş parçası numaraları bunu doğrulayacak şekilde iki satırda peş peşe yerleşmiştir. Aynı sebeple,

“açıklamalar” sütununda işlemlerin “al” ve “yerleştir” şeklindeki bir sırayı takip ettiği görülmektedir.

SIRALAMA programının çalıştırılması ile, imalat hücrelerinde bulunması gereken iş tablaları, bu tablaların nerede yer alacağı ve üzerinde hangi parçayı bulunduracağına dair kararlar alınmaktadır. Çıkış kütüğünde bu bilgilere de yer verilmiştir. Kütüğe bakıldığında, operasyon aralarında ihtiyaç duyulan iş tablaları ile, ham malzeme ve bitmiş ürünün koyulacağı iş tablalarına ait bilgiler görülmektedir. Tezgah no. sütununda görülen bilgiler iş tablasının hangi tezgahın yanına yerleştirilmesi gerektiğini vermektedir.

2.2. Örnek Problemler

Uzman sistemin karar mekanizmaları anlatılırken 4 makina ve 3 işten oluşan örnek bir sıralama problemi dikkate alınarak hem geliştirilen algoritmalar açıklanmış ve hem de bu arada adı geçen problemin çözümü ortaya çıkmıştır. Şekil 2 ve 13'te görülen SBH algoritması ve SIRALAMA programı çıkış kütükleri söz konusu 4 makina ve 3 işli problemin çözüm kütükleridir. Bunun yanı sıra, başka sıralama problemlerinin SBH algoritması ve SIRALAMA programlarında çözümleriyle elde edilen örnek çözüm sonuçları da incelenmiştir [14]. İncelenen örnek problemlerde de, SIRALAMA programının sonuçlarının hiç bir tıkanmaya sebep olmadan robot işlemlerini sıraladığı görülmüştür.

BNECK ve SIRALAMA programının çalışma hızları örneklerin çözümü sırasında izlenmiştir. 4 makina, 4 iş parçası ve toplam 16 operasyona sahip problemin en uzun çözüm süresini verdiği görülmüş ve bunun çözümü için BNECK programının yaklaşık 10 sn., SIRALAMA programının ise yaklaşık 8 sn. zaman harcadığı gözlenmiştir. Bu hızlar 266 MHz hızına sahip bir bilgisayarda kaydedilmiştir.

3. SONUÇ VE ÖNERİLER

Geliştirilen algoritmalar robotlu bir imalat hücreleri için yapılandırılan bir uzman sistemin karar mekanizmasını oluşturmaktadır. Karar mekanizmasının birinci kısmını tezgah yükleme sıralarını belirleyen Kayan Darboğazlı Sezgisel (SBH) algoritma oluşturmaktadır. Diğer kısmı ise, robotun hangi zamanlarda hangi işlemleri yapacağına karar veren ve bu çalışmada geliştirilen algoritmalarlardır. Bu algoritmalara dayanarak hazırlanan programın çalıştırılmasıyla, tezgahlar arasında yükleme - boşaltma yapabilmek için iş parçalarının ham malzeme alanından alınıp bitmiş ürün alanına kadar götürülmesini izleyen bir işlemler zinciri oluşturulmaktadır. Program, kullanılan algoritmalar ile, tezgahları mümkün olan en yüksek dolulukta tutabilmek için operasyon aralarında gerekirse parçaları iş tablalarına (buffer) yerleştirme işlemine de karar vermektedir. SIRALAMA adı

verilen bu uzman sistem yazılımının çalıştırılması sonucunda, belirli bir zamanda, robotun hangi parçayı hangi tezgahtan (veya hangi iş tablasından) alacağı ve bunu nereye koyacağı (diğer tezgah, iş tablası veya bitmiş ürün alanı) bilgileri tablo halinde elde edilmektedir (Şekil 13).

Bölüm 2'de uzman sistemin karar mekanizmaları anlatılırken 4 makina ve 3 işten oluşan örnek bir sıralama problemi dikkate alınarak hem geliştirilen algoritmalar açıklanmış ve hem de bu arada adı geçen problemin çözümü ortaya çıkmıştır. Şekil 2 ve 13'te görülen SBH algoritması ve SIRALAMA programı çıkış kütükleri söz konusu 4 makina ve 3 işli problemin çözüm kütükleridir. Bunun yanı sıra, diğer bazı örnek problemlerin çözümleri de elde edilmiş ve tıkanmaya sebep olmadan robot işlemlerini başarıyla sıraladığı gösterilmiştir.

Bu çalışmada geliştirilen algoritmalar ile imalat hücrelerine yerleştirilecek iş tablaları ve robotun söz konusu yükleme-boşaltma işleri için yapması gereken işlemler ortaya çıkmaktadır. Bundan sonraki aşamada robotun bu işlemleri yapabilmek için izleyeceği yolların belirlenmesi incelenmelidir.

Yapılacak bu işlemler için robotun zamana göre izlemesi gereken yollar çıkarılırken imalat hücrelerinin fiziki boyutları dikkate alınmalı ve çarpmaları önlemek için ara noktalar tarif edilmelidir.

TEŞEKKÜR

KAPPA™ yazılımın kullanımı için ODTÜ bilgisayar imkanlarından yararlanılmıştır. Teşekkürü bir borç biliriz.

KAYNAKLAR

1. Su, C.T. & Fu, H.P., “A Simulated Annealing Heuristic for Robotics Assembly Using The Dynamic Pick-And-Place Model”, **Production Planning & Control**, Vol.9, No.8, 795-802, 1998.
2. Son, C., “An Optimal Planning Technique for Intelligent Robot's Part Assembly in Partially Unstructured Environments”, **Robotica**, Vol.16, 47-57, 1998, UK.
3. Alexander, J.C., Maddocks, J.H. & Michalowski, B.A., “Shortest Distance Paths for Wheeled Mobile Robots”, **IEEE Transactions on Robotics and Automation**, Vol.14, No.5, 657-662, 1998.
4. Desaulniers, G., Soumis, F. & Laurent, J.C., “A Shortest Path Algorithm for A Carlike Robot in A Polygonal Environment”, **The Int. Journal of Robotics Research**, Vol.17, No.5, 512-530, 1998.
5. Dawande, M., Geismar, H.N., Sethi, S.P., & Sriskandarajah, C., “Sequencing and Scheduling in

- Robotic Cells: Recent Developments”, **Journal of Scheduling**, Vol.8, No.5, 387-426, 2005.
6. Agnetis, A., & Pacciarelli, D., “Part Sequencing in Three-Machine No-Wait Robotic Cells,” **Operations Research Letters**, Vol.27, No.4, 185-192, 2000.
 7. Soukhal, A., & Martineau, P., “Resolution of a Scheduling Problem in a Flowshop Robotic Cell”, **European Journal of Operational Research**, Vol.161, No.1, 62-72, 2005.
 8. Sriskandarajah, C., Drobuuchevitch, I., Sethi, S.P., & Chandrasekaran, R., “Scheduling Multiple Parts in a Robotic Cell Served by a Dual-Gripper Robot”, **Operations Research**, Vol.52, No.1, 65-82, 2004.
 9. Kumar, S., Ramanan, N., & Sriskandarajah, C., “Minimizing Cycle Time in Large Robotic Cells”, **IIE Transactions (Institute of Industrial Engineers)**, Vol.32, No.2, 123-136, 2005.
 10. Adams, J., Balas, E. & Zawack, D., “The Shifting Bottleneck Procedure for Job Shop Scheduling”, **Management Science**, Vol.34, No.3, 391-401, 1988.
 11. Pinedo, M., **Scheduling - Theory, Algorithms, and Systems**, Prentice Hall Englewood Cliffs, New Jersey, USA, 1995.
 12. Acar, N., **Üretim Planlaması Yöntem ve Uygulamaları**, 2. Baskı, Milli Produktivite Merkezi Yayınları, No:280, Ankara, 1985.
 13. Usta, Y., “Robotlu İmalat Hücresindeki Taşıma İşlemleri İçin Modele Dayalı Bir Uzman Sistem Geliştirilmesi”, **Gazi Üniv. Müh. Mim. Fak. Dergisi**, Cilt.20, No.2, 275-288, 2005.
 14. Usta, Y., **Robotlu İmalat Hücreleri için Uzman Sistem Geliştirilmesi**, Doktora Tezi, Gazi Üniv. Fen Bilimleri Enst. Ankara, 1999.