

**T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**TORNALAMADA İŞLEM PARAMETRELERİNİN
OPTİMİZASYONUNDA, GENETİK ALGORİTMA
ÇALIŞMALARI ve UYGULAMASI**

**Serdar DÜZGÜN
Teknik Öğretmen**

**YÜKSEK LİSANS TEZİ
MAKİNE EĞİTİMİ ANABİLİM DALI**

**DANIŞMAN
Yrd.Doç.Dr. Ferhat GÜNGÖR**

İSTANBUL, 2006

**T.C.
MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**TORNALAMADA İŞLEM PARAMETRELERİNİN
OPTİMİZASYONUNDA, GENETİK ALGORİTMA
ÇALIŞMALARI ve UYGULAMASI**

Serdar DÜZGÜN

Teknik Öğretmen

141101720020132

**YÜKSEK LİSANS TEZİ
MAKİNE EĞİTİMİ ANABİLİM DALI**

**DANIŞMAN
Yrd.Doç.Dr. Ferhat GÜNGÖR**

İSTANBUL, 2006

ÖNSÖZ

Ünlü matematikçi Jacques Salomon HADAMARD “Gerek matematikte, gerek başka alanlarda buluş ve icatlar farklı alanlardan düşüncelerin bir araya getirilmesiyle gerçekleşir” demiş. Bu söz bize genetik algoritmaların (GA) nasıl ortaya çıktığını anlatır gibi. Karınca koloni sistemi, yapay sinir ağları ve genetik algoritmalar gibi sistemler sayesinde de görebileceğimiz gibi tabiat da bu konuda bize hayli yardımcı olmaktadır. Ben de bu şekilde düşünerek, GA çalışma mantığı ve bir uygulama alanı üzerine yorucu ama bir o kadar da keyifli bir çalışma yaptığımıza inanıyorum. Bu tezde GA'nın uygulama alanı olarak çok aşamalı tornalama işlemlerindeki parametrelerin optimizasyonuna dayalı bir çalışma seçtim. Yaptığım araştırmalarda bu konudaki çalışmaların ne denli az olduğunu gördüm. Bu çalışma sayesinde müteakip çalışmalara ışık tutacağım inancındayım. Bu konuda okulumuzda yapılacak çalışmalara kaynak oluşturması açısından tezimi, ekleri ve kaynakçamı geniş tuttum. Ayrıca İnternet’te bulduğum kaynakları da kaynakçada belirterek gelecek çalışmalara ışık tutması açısından faydalı olmaya çalıştım.

Bu çalışma sırasında benden yardımlarını esirgemeyen, takıldığım konularda bana öneri ve tecrübesiyle yön veren ve sürekli işimi kolaylaştırıp beni cesaretlendiren, hocamız sayın Yrd.Doç.Dr. Ferhat GÜNGÖR’e ve Yrd.Doç.Dr.Vedat TOPUZ’a teşekkürlerimi sunarım.

TEMMUZ, 2006

Serdar DÜZGÜN

İÇİNDEKİLER

	SAYFA
ÖNSÖZ	i
İÇİNDEKİLER	ii
ÖZET	iv
ABSTRACT	vi
SEMBOL LİSTESİ	viii
KISALTMALAR	xii
ŞEKİL LİSTESİ	xiii
TABLO LİSTESİ	xv
BÖLÜM I. GİRİŞ VE AMAÇ	1
I.1. GİRİŞ ve AMAÇ	1
BÖLÜM II. GA NEDİR, NASIL VE NERELERDE KULLANILIR?	4
II.1. GİRİŞ	4
II.2. GENETİK ALGORİTMALARIN TEMEL TEOREMİ	7
II.3. BASİT GENETİK ALGORİTMA	8
II.3.1. Çözümlerin Kodlanması	9
II.3.2. İlk Populasyonun Oluşturulması	9
II.3.3. Uygunluk Değerinin Hesaplanması	9
II.3.4. Çoğalma İşleminin Uygulanması	9
II.3.5. Çaprazlama İşleminin Uygulanması	10
II.3.6. Mutasyon İşleminin Uygulanması	10
II.3.7. Yeni Kuşağın Oluşması ve Döngünün Durdurulması	11
II.4. GENETİK ALGORİTMALARDA PARAMETRE SEÇİMİ	11
II.5. GENETİK ALGORİTMANIN BİR FONKSİYON OPTİMİZASYONUNDA UYGULANMASI	13

II.6. GENETİK ALGORİTMALARIN UYGULAMA ALANLARI	16
III.6.1. Genel Uygulama Alanları	16
III.6.2. İşletmelerdeki Uygulama Alanları	19
BÖLÜM III. ÇOK AŞAMALI TORNALAMA OPERASYONLARININ GENETİK ALGORİTMA KULLANILARAK OPTİMİZASYONU	26
III.1. GİRİŞ	26
III.2. TEK ve ÇOK AŞAMALI TORNALAMANIN SEÇİM KRİTERLERİ ve BU OPERASYONLARDA YAPILAN OPTİMİZASYON ÇALIŞMALARININ TARİHSEL GELİŞİMİ	27
III.3. ÇOK AŞAMALI TORNALAMA OPERASYONLARI KESME MODELİ	30
III.3.1 Matematiksel Modelleme	30
III.3.1.1 Kesme Maliyeti	30
III.3.1.2 Makine Boşta Kalma Zamanı	31
III.3.1.3 Takım Değiştirme Maliyeti	31
III.3.1.4 Takım Maliyeti	31
III.3.1.5 Birim Üretim Maliyeti	32
III.3.2 Parametre Sınırları ve Kesme Koşulları Sınırlamaları	33
III.3.3 Son Kesme Modeli	34
III.4 OPTİMİZASYON ALGORİTMASI	36
III.4.1 Temsil	37
III.4.1.1 İkili Sistem Planlaması	38
III.4.2 Başlangıç	39
III.4.3 Evrim	40
III.4.4 Seçim ve yeniden kopyalama	40
III.4.5 Genetik Operatörler: Çaprazlama ve Mutasyon	41
III.4.6 Değiştirme Stratejisi	42
III.4.7 En iyi çözümün kaybını önlemek	43
III.5 CEZA FONKSİYONU	46
III.6 BİR UYGULAMA MODELİ	48
III.6.1 Sonuç	50
III.7. ELEŞTİRİ ve ÖNERİLERİ	50
III.7.1 İşleme Modeli ve Dizi Sunumu	51
BÖLÜM IV. BİLGİSAYAR PROGRAMI	54
IV.1 BİLGİSAYAR PROGRAMI	54
IV.2 PROGRAMIN İÇERİĞİ	56
IV.3 PROBLEMİN PROGRAMDA İŞLEYİŞİ	57
IV.4 SONUÇ	92
KAYNAKLAR	96

ÖZGEÇMİŞ

ÖZET

TORNALAMADA İŞLEM PARAMETRELERİNİN OPTİMİZASYONUNDA, GENETİK ALGORİTMA ÇALIŞMALARI ve UYGULAMASI

Bu tez, çok aşamalı tornalama operasyonlarında kesme parametrelerinin belirlenmesi için genetik algoritmaya dayalı yeni bir optimizasyon tekniği önermektedir. Kesme prosesi çok aşamalı kaba ve son paso tornalamayı eş zamanlı olarak ele alır. Optimum işleme parametreleri pratik işleme sınırları göz önüne alınarak birim üretim maliyetinin minimizasyonu ile belirlenir. Formüle edilmiş kesme modeli, 20 işleme parametre sınırlaması olan bir doğrusal olmayan sınırlanmış programlama (NCP) problemidir. Deneysel sonuçlar NCP problemini çözmeye yönelik genetik algoritma tabanlı prosedürün hem etkili hem de verimli olduğunu ve kompleks işleme optimizasyon problemlerini çözmeye yönelik akıllı üretim sistemlerine katılabileceğini gösterir.

Akıllı üretim, eğer verimli otomatikleştirilmiş proses planlama modülüyle, üretim, ulaştırma, montaj vb. gibi diğer otomatikleştirilmiş sistemleri birleştirebilirse, para ve zaman bakımından önemli tasarruflar sağlar. Proses planlama, uygun makinelerin, işlenecek parçalara ait takımların, kesme alanı içindeki ortalama sıcaklığı düşürecek kesme sıvısının ve belirli bir işlenmiş parçaya ait her bir operasyon için belli kesme koşulları altındaki işleme parametrelerinin belirlenmesine gerek duyar. İşleme ekonomisi problemi objektif bir fonksiyonu optimize etmek için genellikle kesme hızı, devir oranı ve kesme derinliği olmak üzere, proses parametrelerinin belirlenmesi ile ilgilidir. İşleme koşullarının optimumluğunu ölçmeye yarayacak birkaç objektif fonksiyon şunları içerir: (1) minimum birim üretim maliyeti, (2) maksimum üretim oranı, (3) maksimum kar oranı, (4) birçok objektif fonksiyonun ağırlıklı kombinasyonu. İşleme ekonomisinde ele alınması uygun olacak birçok kesme sınırlaması şunları içerir: takım ömrü sınırlaması, kesme kuvveti sınırlaması, güç, stabil kesme alanı sınırlaması, talaş-

takım ara yüzeyi sıcaklığı sınırlaması, yüzey bitiş sınırlaması, ve kaba ve son paso parametre ilişkileri. Klasik olarak, problemde iki temel yaklaşım kullanılarak bahsedilmiştir: tek ve çok pasolu.

Tek pasolu yaklaşımda, toplam kesme derinliği sadece tek bir pasoda halledilebilir olarak ele alınmıştır. Dolayısıyla, kesme derinliği ve paso sayısı fazla basitleştirmeye götürecek optimizasyon problemi değişkenlerinden elenir. Tek pasolu yaklaşımın toplam kesme derinliğinin sadece tek bir pasoda halledildiği durumlara uygulanabilmesine rağmen, pratikte bu nadiren gerçekleşir. Bu yüzden, işleme ekonomisi problemine çok aşamalı bir yaklaşım ekonomik sebeplerden dolayı ele alınmak zorundadır. Tezde çok aşamalı ve tek pasoya dayanan yaklaşımların birbirlerine üstünlükleri ve nasıl seçim yapılacağına dair detaylı bir analiz yapılmıştır.

Bu tezin ana amacı, belli kesme sınırlamalarına dokunmadan birim üretim maliyetini minimize edecek optimal işleme parametrelerini belirlemektir. Dolayısıyla, çok aşamalı işleme optimizasyon problemine ait matematiksel formülasyon, Chen ve Tsai'nin (1996) 20 kesme sınırlaması olanına benzerlik göstermektedir. Genetik algoritma yaklaşımına dayalı yeni bir yerel arama optimizasyonu işleme optimizasyon modelini çözmek için geliştirilmiştir. Kati ve kati olmayan limitleri proseste kullanmak için ikili verilere dönüştüren bir teknik optimizasyon formüllerinin kati formülasyonları tarafından meydana gelen zorlukları aşmak için genetik algoritma yaklaşımı ile bir arada kullanılır.

Tez şu şekilde düzenlenmiştir. Bölüm 1 tez konusu hakkında giriş bilgisi ve amacı hakkında bilgi vermektedir. Bölüm 2, tek ya da çok aşamalı tornalama operasyonlarına ait seçim kriterlerinden, Genetik Algoritma'nın tarihçesinden, gelişiminden, kullanım alanlarından, operatörlerinden ve teorik bilgisini ele alır. Çok aşamalı tornalama operasyonlarındaki işleme parametrelerinin Genetik Algoritma kullanılarak optimizasyonu Bölüm 3'te kapsamlı bir şekilde anlatılmıştır. Bölüm 4'te çok aşamalı bir tornalama operasyonundaki işleme parametrelerinin genetik algoritma ile optimizasyonunu yapan, Delphi dilinde yazılmış bir bilgisayar programının, bir probleme uygulanması ve elde edilen sonuçların mevcut verilerle karşılaştırılması ve maliyet minimizasyonuna dair kapsamlı bir araştırması yer almıştır.

ABSTRACT

IN MACHINING PARAMETERS OPTIMIZATION OF TURNING, GENETIC ALGORITHM STUDIES and ITS APPLICATION

The paper proposes a new optimization technique based on genetic algorithms for the determination of the cutting parameters in multipass machining operations. The cutting process simultaneously considers multipass rough machining and finish-machining. The optimum machining parameters are determined by minimizing the unit production cost subject to practical machining constraints. The cutting model formulated is a non-linear-constrained programming (NCP) problem with 20 machining parameter constraints. Experimental results show that the proposed genetic algorithm-based procedure for solving the NCP problem is both effective and efficient, and can be integrated into an intelligent manufacturing system for solving complex machining optimization problems.

Intelligent manufacturing achieves substantial savings in terms of money and time if it integrates an efficient automated process-planning module with other automated systems such as production, transportation, assembly, etc. Process planning involves determination of appropriate machines, tools for machining parts, cutting fluid to reduce the average temperature within the cutting zone and machining parameters under certain cutting conditions for each operation of a given machined part. The machining economics problem consist in determining the process parameter, usually cutting speed, feed rate and deep of cut, in order to optimize an objective function. A number of objective functions by which to measure the optimality of machining conditions include: (1) minimum unit production cost, (2) maximum production rate, (3) maximum profit rate, (4) weighted combination of several objective functions. Several cutting constraints that should be considered in machining economics include: tool-life

constraint, cutting force constraint, power, stable cutting region constraint and roughing and finishing parameter relations. Classically, the problem has been dealt with by using two basic approaches: single and multipass.

In the single-pass approach, the total deep of cut is considered achievable in just one pass. Consequently, the deep of cut and the number of passes are eliminated from the variables of the optimization problem leading to oversimplification. Although the single-pass approach can be applied in cases where the total depth of cut can be achieved in just one pass, in practice this rarely happens. Therefore, a multipass approach to the machining economics problem has to be considered for economics reasons. In the paper, the superiority to each other of single-pass and multipass approaches and how to make the optimal choice were analyzed in detail.

The main objective of the present paper is to determine the optimal machining parameters that minimize the unit production cost without violating any imposed cutting constraints. Consequently, the mathematic formulation of the multipass machining optimization problem is similar to that of Chen and Tsai (1996) having 20 cutting constraints. A new local search optimization based on genetic algorithm approach is developed to solve the machining optimization model. A technique that converts crisp and non-crisp limits into binary data for processing is used in conjunction with the genetic algorithm approach in order to overcome the difficulties caused by the crisp formulation of the optimization problem.

The paper is organized as follows. Section1 gives us information about introduction and aims. Section 2 deals with the selection criteria of single or multipass turning operations, the historical background, development, usage field, operators and theoretical information of Genetic Algorithm. Optimization of multipass turning operations using genetic algorithms is mentioned comprehensively in section 3. However in Section 4, a program, which computes the optimization of multipass turning operations using genetic algorithm, written in Delphi, is considered. Results and constraints are discussed comprehensively, too.

JULY, 2006

Serdar DÜZGÜN

SEMBOL LİSTESİ

I. Tornacılıkta İşlem Ekonomisi

C	: Taylor'un takım ömrü denklemindeki sabiti
D	: İş çapı (mm)
K	: Genel işgücü ve genel giderler maliyeti M_d+M_e (TL/saat)
K_d	: Aşınmış takımın değiştirilme maliyeti (TL)
K_e	: Endirekt maliyet (TL)
K_f	: Kesici ucun ilk satınalma maliyeti (fatura fiyatı) (TL)
K_h	: Esas işlem için tezgahı hazırlama zamanı (TL)
K_i	: Her bir parçanın işlenme maliyeti (TL)
K_m	: Her bir parçanın malzeme maliyeti (TL)
K_t	: Takım maliyeti (TL)
L	: İşleme boyu (mm)
M_b	: Kesiciyi tekrar bilemek için bir dakikadaki maliyet (TL)
M_d	: Direkt iş gücü maliyeti (TL)
M_e	: Birim zamandaki endirekt maliyet (TL)
M_k	: Kesicide bir kesme ağzının maliyeti (TL)
M_t	: Her bir parçanın toplam maliyeti (TL)
M_{tmax}	: Her bir parçanın maksimum üretim miktarındaki maliyeti (TL)
M_{tmin}	: Her bir parçanın minimum maliyetteki fiyatı (TL)
n	: Taylor takım ömrü denkleminin üst sabiti
s	: İlerleme (mm/dev)
t_h	: İşlem süresi (dak)
t_h	: Minimum maliyet durumlarında işlem süresi (dak)
t_r	: Gerekli hazırlık zamanı
T	: Takım ömrü (dak)

- T_{\min} : Her bir parçanın minimum maliyeti için takım ömrü (dak)
 T_{\max} : Maksimum üretim miktarı için takım ömrü (dak)
 T_{τ} : Bir parçanın işlenebilmesi için gereken toplam zaman (dak)
 U_s : Bir takma uç üzerindeki kesici kenar sayısı
 U_b : Muhtemel tekrar bileme sayısı
 U_d : Kesici takımını değiştirme ve ayarlama zamanı (dak)
 U_g : Tekrar bilemek için gereken süre (dak)
 V_{\max} : Maksimum üretim miktarı için keme hızı (m/dak)
 V_{\min} : Her bir parçanın minimum maliyeti için kesme hızı (m/dak)

II.Kesme Modelinde Kullanılan Simgeler

- UC : Malzeme maliyeti hariç birim üretim maliyeti (TL/parça)
 C_m : Kesme operasyonundaki asıl zamana ait kesme maliyeti
 C_1 : Parça indirme bindirme operasyonları ve takımın boşta bulunma zamanı yüzünden makinenin boşta kalma zamanı, hareket zamanı (TL/parça)
 C_R : Takım değiştirme maliyeti (TL/parça)
 C_T : Takım maliyeti (TL/parça)
 $V_r V_s$: Kaba ve bitiş tornalamadaki kesme hızı (m/dak)
 $V_{rL} V_{rU}$: Kaba tornalamada kesme hızının alt ve üst sınırı (m/dak)
 $V_{sL} V_{sU}$: Bitiş tornalamada kesme hızının alt ve üst sınırı (m/dak)
 $f_r f_s$: Kaba ve bitiş tornalamadaki ilerleme miktarı (mm/dev)
 $f_{rL} f_{rU}$: Kaba tornalamadaki ilerleme miktarının alt ve üst sınırı (mm/dev)
 $f_{sL} f_{sU}$: Bitiş tornalamadaki ilerleme miktarının alt ve üst sınırı (mm/dev)
 $d_r d_s$: Kaba ve bitiş tornalamada her bir paso için talaş derinliği (mm)
 $d_{rL} d_{rU}$: Kaba tornalamada kesme derinliğinin alt ve üst sınırı (mm/dev)
 $d_{sL} d_{sU}$: Bitiş tornalamada kesme derinliğinin alt ve üst sınırı (mm/dev)
 N : Kaba kesme işlemlerinin sayısı (tamsayı)
 d_1 : Malzemenin işlenecek derinliği (mm)
 DL : İş parçasının çapı ve uzunluğu (mm)
 k_0 : Tam işçilik zamanı + bütün giderler (parça/YTL)
 k_t : Kesme kenarı maliyeti (kenar/YTL)
 $t_{mr} t_{ms} t_m$: Kaba, bitiş ve gerçek işleme zamanı (dak)

$t_c t_v t_i$: İndirme/bindirme için hazırlık zamanı, Takım dolaşımı, takımın yaklaşma ayrılma zamanı ve toplam makine boş geçirme zamanının gibi takım hareketlerindeki boş zaman (dak)
$t_e t_r$: Takım değişimi ve takım değiştirme zamanları (dak)
$h_1 h_2$: Takım dolaşımı ve yaklaşma/ayrılma zamanına ile ilgili sabitler
$TT_r T_s$: Takım ömrü, kaba tornalama için öngörülen takım ömrü ve bitiş tornalama için öngörülen takım ömrü (dak)
T_p	: T_r ve T_s 'nin (dak) ağırlıklı kombinasyonuna ait takım ömrü, T_p için ağırlık [0,1]
$T_U T_L$: Takım ömrünün alt ve üst sınırları (dak)
$\alpha\beta\gamma C$: Takım Ömrü denklemi sabitleri
$pqrC_o$: $p=1/\alpha$; $q=\beta/\alpha$; $r=\gamma/\alpha$ ve $C_o=C^{1/\alpha}$
SR	: Azami müsaade edilebilir yüzey pürüzlülüğü (mm)
R	: Kesme takımının burun yarıçapı (mm)
$F_r F_s$: Kaba ve bitiş tornalama esnasındaki kesme kuvvetleri
F_U	: Azami müsaade edilebilir kesme kuvveti (kgN)
$K_1\mu\nu$: Kesme kuvveti denklemi sabitleri
$P_r P_s$: Kaba ve bitiş tornalama esnasındaki kesme gücü (kW)
P_U	: Azami müsaade edilebilir kesme gücü (kW), güç verimi
$\lambda\nu$: Sabit kesme alanının ifadesine ilişkin sabitler
SC	: Sabit alan kesme sınırlaması limiti
$Q_r Q_s$: Talaş-Takım ara yüzeyi kaba ve bitiş işleme sıcaklıkları ($^{\circ}C$)
Q_U	: Azami müsaade edilebilir talaş-takım ara yüzeyi sıcaklığı ($^{\circ}C$)
$k_2T\Phi\delta$: Talaş-takım ara yüzeyi sıcaklığı denklemine ilişkin sabitler
X	: $\{x_1, x_2, \dots, x_m\}$, işleme parametre ayarı
n_i	: Verilmiş bir X değeri altında tahmini n değeri
d_{r_i}	: Verilmiş bir X değeri altında tahmini d_r
$N_L N_U$: n'nin alt ve üst sınırları
$k_3 k_4 k_5$: Kaba ve bitiş parametreleri ilişkilerine ait sabitler $k_3, k_4, k_5 \geq 1$

III. Optimizasyon Algoritmasında Kullanılan Simgelemeler

$P_{popsize}$: Çözüm boşluğu ebadı
P_{pcross}	: Çaprazlama olasılığı
$P_{pmutate}$: Mutasyon olasılığı

- S**_(t) : Çözüm boşluğu
g_(t) : Bir diziye ait objektif fonksiyon
f_{max} : Maksimum objektif fonksiyon
f : Skor fonksiyonu
e_i : Öngörülen sayım
P_(i) : Seçim olasılığı

KISALTMALAR

GA	:	Genetik Algoritma
EA	:	Evrimsel Algoritma
GP	:	Genetik Programlama
TS	:	Tabu Arama
SA	:	Tavlama Benzetimi
UC	:	Birim üretim fonksiyonu
NCP	:	Doğrusal olmayan sınırlanmış programlama
AS	:	Karınca sistemi
SUS	:	Sıralı Seçim Yöntemi
SGA	:	Basit Genetik Algoritma
FEGA	:	Genetik Algoritma Kodlaması

ŞEKİL LİSTESİ

	<u>SAYFA NO</u>
Şekil II.1 $F(x)=2\cos(50x) + 3\sin(30x)$ Grafiği	14
Şekil II.2 Genetik Algoritma Akış Diyagramı	15
Şekil III.1 Maliyet Grafiği	49
Şekil IV.1 Programın GA ve Grafik Sekmeleri	54
Şekil IV.2 Programın Sonuç ve Problem Değer Girişi Sekmeleri	55
Şekil IV.3 Programın Sınır Belirle Sekmesi	55
Şekil IV.4 Programın Kısıtlar Sekmesi	56
Şekil IV.5 $V_r=500$ için parça başına maliyet değeri hesaplaması	59
Şekil IV.6 $V_r=450$ için parça başına maliyet değeri hesaplaması	60
Şekil IV.7 $V_r=550$ için parça başına maliyet değeri hesaplaması	61
Şekil IV.8 $f_r=0,8$ için parça başına maliyet değeri hesaplaması	63
Şekil IV.9 $f_r=1,0$ için parça başına maliyet değeri hesaplaması	64
Şekil IV.10 $f_r=1,1$ için parça başına maliyet değeri hesaplaması	65
Şekil IV.11 $f_r=1,2$ için parça başına maliyet değeri hesaplaması	66
Şekil IV.12 $d_r=2,8$ için parça başına maliyet değeri hesaplaması	67
Şekil IV.13 $d_r=3,2$ için parça başına maliyet değeri hesaplaması	68
Şekil IV.14 $d_r=3,4$ için parça başına maliyet değeri hesaplaması	69
Şekil IV.15 $d_r=3,6$ için parça başına maliyet değeri hesaplaması	70
Şekil IV.16 $d_r=3,8$ için parça başına maliyet değeri hesaplaması	71
Şekil IV.17 $d_s=2,8$ için parça başına maliyet değeri hesaplaması	72
Şekil IV.18 $d_s=3,2$ için parça başına maliyet değeri hesaplaması	73
Şekil IV.19 $d_s=3,4$ için parça başına maliyet değeri hesaplaması	74
Şekil IV.20 $d_s=3,8$ için parça başına maliyet değeri hesaplaması	75
Şekil IV.21 $f_s=0,8$ için parça başına maliyet değeri hesaplaması	76
Şekil IV. 22 $f_s=1,0$ için parça başına maliyet değeri hesaplaması	77
Şekil IV.23 $f_s=1,1$ için parça başına maliyet değeri hesaplaması	78

Şekil IV.24 $f_s=1,2$ için parça başına maliyet değeri hesaplaması	79
Şekil IV.25 $V_s=450$ için parça başına maliyet değeri hesaplaması	80
Şekil IV.26 $V_s=550$ için parça başına maliyet değeri hesaplaması	81
Şekil IV.27 Generasyon sayısı 1000 için parça başına mal.değ.hesp.....	82
Şekil IV.28 Generasyon sayısı 10000 için parça başına mal.değ.hesp.....	83
Şekil IV.29 $L=325$ mm için parça başına maliyet değeri hesaplaması ...	85
Şekil IV.30 $d=55$ mm için parça başına maliyet değeri hesaplaması	86
Şekil IV.31 Rulet seçim yöntemi için maliyet değeri hesaplaması	87
Şekil IV.32 Turnuva seçim yöntemi için maliyet değeri hesaplaması ...	88
Şekil IV.33 Gray kodlama ile parça başına maliyet değeri hesaplaması ..	89
Şekil IV.34 Dinamik Oran ile parça başına maliyet değeri hesaplaması ..	90
Şekil IV.35 Random ile parça başına maliyet değeri hesaplaması	91
Şekil IV.36 V_r kaba tornalama kesme hızına ait girilen sınır değerlerin maliyet değeri grafiği	93
Şekil IV.37 f_r kaba tornalama devir oranına ait girilen sınır değerlerin maliyet değeri grafiği	93
Şekil IV.38 d_r kaba tornalama talaş derinliğine ait girilen sınır değerlerin maliyet değeri grafiği	94
Şekil IV.39 d_s bitiş tornalama talaş derinliğine ait girilen sınır değerlerin maliyet değeri grafiği	94
Şekil IV.40 f_s bitiş tornalama devir oranına ait girilen sınır değerlerin maliyet değeri grafiği	95
Şekil IV.41 V_s bitiş tornalama kesme hızına ait girilen sınır değerlerin maliyet değeri grafiği	95

TABLO LİSTESİ

	<u>SAYFA NO</u>
Tablo II.1 İyileşmenin Görüldüğü Kuşaklar ve Uygunluk Değerleri ...	14
Tablo III.1 Optimizasyon prosesinin sonuçlarından elde edilen Paretian noktaları	49
Tablo IV.1 V_r 'ye ait veri değişiklikleri ve maliyet değerine etkisi	58
Tablo IV.2 Parça ebadına bağlı değişimlerde ortaya çıkan maliyet değerleri	84
Tablo IV.3 Genetik Algoritma seçim yöntemi ve kodlamalarındaki değişimlere ait maliyet	84

BÖLÜM I

I.1. GİRİŞ ve AMAÇ

Akıllı üretim, eğer verimli otomatikleştirilmiş proses planlama modülüyle üretim, ulaştırma, montaj vb. gibi diğer otomatikleştirilmiş sistemleri ile birleştirebilirse, para ve zaman bakımından önemli tasarruflar sağlar. Proses planlama, uygun makinelerin, işlenecek parçalara ait takımların, kesme alanı içindeki ortalama sıcaklığı düşürecek kesme sıvısının ve belirli bir işlenmiş parçaya ait her bir operasyon için belli kesme koşulları altındaki işleme parametrelerinin belirlenmesine gerek duyar. İşleme ekonomisi problemi objektif bir fonksiyonu optimize etmek için genellikle kesme hızı, ilerleme miktarı ve talaş derinliği olmak üzere, proses parametrelerinin belirlenmesi ile ilgilidir.

İşleme koşullarının optimumluğunu ölçmeye yarayacak birkaç objektif fonksiyon şunları içerir:

- Minimum birim üretim maliyeti,
- Maksimum üretim miktarı,
- Maksimum kar,
- Birçok objektif fonksiyonun ağırlıklı kombinasyonu.

İşleme ekonomisinde ele alınması uygun olacak birçok kesme sınırlaması şunları içerir: takım ömrü sınırlaması, kesme kuvveti sınırlaması, güç, sabit kesme alanı sınırlaması, talaş-takım ara yüzeyi sıcaklığı sınırlaması, yüzey bitiş sınırlaması, ve kaba ve son paso parametre ilişkileri.

Klasik olarak, problemden iki temel yaklaşım kullanılarak bahsedilmiştir: tek ve çok pasolu. Tek pasolu yaklaşımda, toplam talaş derinliği sadece tek bir pasoda

halledilebilir olarak ele alınır. Dolayısıyla, paso sayısı optimizasyon problemi değişkenlerinden çıkarılır. Tek pasolu yaklaşımın toplam talaş derinliğinin sadece tek bir pasoda halledildiği durumlara uygulanabilmesine rağmen, pratikte bu nadiren gerçekleşir. Bu yüzden, işleme ekonomisi problemine çok aşamalı bir yaklaşım ekonomik sebeplerden dolayı ele alınmak zorundadır.

Geçmiş çalışmalar, tek pasolu operasyonlarla sınırlanmıştır (Ermer ve Wu 1967, Ermer ve Morris 1969, Iwata et al. 1977). Müteakip çalışmalar optimal işleme parametrelerini belirleyen çok aşamalı operasyonları keşfetmiştir (Iwata et al. 1972, Ermer ve Kromodihardo 1981, Alberti 1986, Shin ve Joo 1992, Tan ve Creese 1995). Klasik olarak, işleme ekonomisi problemi optimizasyon algoritmaları kullanılarak çözülmüştür, bu optimizasyonlar geometrik programlamayı (Ermer ve Kromodihardo 1981, Gopalkrishan ve Faiz 1991), dinamik programlamayı (Iwata et al. 1977, Lambert ve Walvekar 1978) ve lineer programlamayı (Gupta et al. 1995) içerir. Diğerleri dinamik programlamayı ve Fibonacci aramayı (Shin ve Joo 1992), onları takip edenleri ise sınırlanmamış minimizasyon tekniklerini (Hati ve Rao 1976) içerir ve arama stratejisi (Alberti 1986) üzerine odaklanır [2].

Evvelki çalışmaları daha önce bahsedilen bütün kesme sınırlamalarını ele almaz çünkü çoğu sınırlamalar mevcut metotları kullanan işleme optimizasyon problemini güçleştirir. İlave değişkenler ve paso sayısı çok aşamalı problemi NP-sert yapar. Evvelki optimizasyon teknikleri sadece spesifik bir problem için kullanışlı olma ihtimaline sahiptir ve yerel bir optimal çözüm elde etme eğilimini göstermektedir (Chen ve Tsai 1996). Dolayısıyla son zamanlarda tahmin algoritmaları çeşitli işleme problemleri tiplerini çözmek için uygulanmaktadır. Yerel ve yöresel arama teknikleri birleştirilebilen optimizasyon problemlerini çözmekte güçlü olan tahmin algoritmalarıdır. Çok aşamalı tornalama operasyonuna ait optimal koşulların belirlenmesi birleştirilebilen bir optimizasyon problemidir. Dolayısıyla yerel arama teknikleri son zamanlarda çok aşamalı işleme ekonomisi problemine uygulanmaktadır. Yerel arama teknikleri tavlama benzetimi (SA) algoritmasını (Kirkpatrick et al. 1983), genetik algoritma (GA) yaklaşımını (Goldberg 1989), yasaklı araması (TS) yaklaşımını (Glover 1989) ve karınca sistemi (AS) yaklaşımını (Dorigo et al. 1996) içermektedir. Chen ve Tsai (1996) işleme optimizasyon problemini çözmek için model araması (PS) tekniğini ve tavlama benzetimini birleştirmişlerdir. Model arama tekniğini tavlama benzetimi tekniğine bir giriş olarak

bir çekirdek çözüm üretmek için kullanmışlardır. Böylece tavlama benzetimi tekniği daha iyi çözümler bulur ve global optimal çözümlere doğru ilerler. Formüle edilmiş kesme modeli, Shin ve Joo (1992) tarafından ele alındığı üzere çok aşamalı kaba tornalama ve bitiş tornalama olmak üzere çok aşamalı bir tornalama operasyonudur. İşleme parametrelerinin ikisinin de aslında birbirine bağımlı olmasından dolayı, Chen ve Tsai (1996) kaba ve son bitiş operasyonlarındaki işleme parametrelerini iki ayrı gruba bölmektense beraber ele alarak analiz etmişlerdir. Alberti ve Perrone (1999) klasik çok aşamalı tornalama operasyonlarına ait bir fuzzy olanaksız formülasyonunu ele almışlardır ve sonuç veren olanaksız modeli genetik algoritma kullanarak optimize etmişlerdir. Problemi sınırlı sayıda kesme sınırlamaları ile ele almışlardır.[27]

Bu tezin ana amacı birim üretim maliyetini, dayatılmış kesme sınırlamalarına dokunmadan birim üretim maliyetini minimize edecek optimal işleme parametrelerini belirlemektir. Bunu yaparken, yapacağımız işleme ait matematiksel modelleme ortaya konulmakta ve uygun optimizasyon algoritması seçilerek maliyet değeri minimize edilmektedir. Bu algoritmayı uygulayan bir program yardımıyla bu tezde ki probleme ait sonuçlar ayrıntılı olarak irdelenmektedir.

BÖLÜM II

GA NEDİR, NASIL VE NERELEERDE KULLANILIR?

II.1. GİRİŞ

Genetik algoritmalar, evrim mekanizmasını örnek alan bir arama yöntemidir ve bir veri grubundan özel bir veriyi bulmak için kullanılır. Amacı, problemler için doğada geçerli olan en iyinin yaşaması kuralına dayanarak sürekli iyileşen çözümler üretmektir. Genetik algoritma ile oluşturulan seçim, doğal topluluklara benzer bir şekilde bilgisayar hafızasına depo edilmiş kromozomlar üzerinde icra edilmektedir.[30]

Genetik algoritmalar yapay zekânın gittikçe genişleyen bir kolu olan evrimsel hesaplama tekniğinin bir parçasını oluşturmaktadır. Adından da anlaşıldığı üzere, evrimsel hesaplama tekniğinin bir parçası olan genetik algoritma Darwin'in evrim teorisinden esinlenerek oluşturulmuştur. Herhangi bir problemin genetik algoritma ile çözümü, problemi sanal olarak evrimden geçirmek suretiyle yapılmaktadır.[3]

Genetik Algoritma çalıştırılmadan önce bir problemin çözümü için değişkenlerin belirli bir düzende sıralanmasından oluşturulmuş kromozomların bir gurubu başlangıç popülasyonu olarak seçilir. Daha sonra bu popülasyon, kodu çözülmüş biçimde değerlendirilir, çaprazlama ve mutasyondan oluşan evrim süreci başlar ve en uygun kromozomlar seçilir. Çaprazlama, iki kromozomun bir araya gelerek genetik bilgi değişimi yapmasıdır. Bunun yanında iki ebeveynden yeni kromozomlar üreten bir operatör olan çaprazlama, GA'ların temel bir özelliğidir. Mutasyon veya değişim ise bir kromozomun taşıdığı genetik bilgide bir neden bağlı

olmaksızın deęişme olmasıdır. Mutasyon, aramada kısır döngüye girilmemesini sağlamak, toplulukta çözüm olmayan birbirine benzer bireylerden kurtulmak ve yeni alt optimum çözümler bulunmasını sağlamak için kullanılır. Bulunan en uygun kromozomlar yeni oluşturulan eşleşme havuzunda bu genetik işlemlerden geçirilerek bir alt popülasyon, yani yeni nesil oluşturulur. Bu yeni nesil de benzer şekilde değerlendirilip, bilinen bir yerine geçme stratejisiyle eski neslin yerine geçer. Çevrim uygun birey veya bireyler bulunana kadar tekrarlanır.[30]

Evrimsel hesaplama ilk olarak 1960’larda I.Rechenberg tarafından “Evrimsel Stratejileri (Evolutions strategie)” isimli eserinde tanıtılmıştır. Onun fikri daha sonra başka araştırmacıların da ilgisini çekmiş ve geliştirilmiştir. John Holland evrim sürecinin bir bilgisayar yardımıyla kullanılarak, bilgisayara anlayamadığı çözüm yöntemlerinin öğretilebileceğini düşündü. Genetik algoritmalar, John Holland ve Michigan Üniversitesi’ndeki koleji tarafından yürütülen “Hüresel Otomatikleştirme” çalışmasından çıkmıştır.[8]

Böylece, Genetik Algoritma (GA), John Holland tarafından bu düşüncenin bir sonucu olarak bulundu. Onun öğrencileri ve arkadaşları tarafından geliştirildi ve bu sayede Holland’ın kitabı “Doğal ve Yapay Sistemlerde Adaptasyon (Adoption in Natural and Artificial Systems)” 1975 yılında yayınlandı. Genetik algoritma çalışmaları, 1980’lerin ortalarında Illinois Üniversitesinde Genetik Algoritmalar ile ilgili ilk uluslar arası konferans gerçekleştirilinceye kadar oldukça kuramsal kalmıştır.[24]

1992 yılında John Koza genetik algoritmayı kullanarak çeşitli görevleri yerine getiren programlar geliştirdi. Bu metoda Genetik Programlama adını verdi.

Genetik algoritma geleneksel yöntemlerle çözümü zor veya imkânsız olan problemlerin çözümünde kullanılmaktadır. Çok genel anlamda genetik algoritmanın üç uygulama alanı bulunmaktadır. Bunlar deneysel çalışmalarda optimizasyon, pratik endüstriyel uygulamalar ve sınıflandırma sistemleridir.[3]

Mühendislik problemlerinde optimizasyon amaçlı olarak kullanılmaya başlanmıştır. Özellikle mekanizma tasarımında çok iyi sonuçlar verdiği bilinmektedir. Bunlardan başka otomatik programlama, öğrenme kabiliyetli makinalar, ekonomi, çevrebilim, planlama, üretim hattı yerleşimi gibi alanlarda da uygulanmaktadır. Ayrıca dijital resim işleme tekniğinde de çokça uygulama alanı bulmuştur.[4]

Bu problemlerin hemen hepsi çok geniş bir çözüm havzasının taranmasını gerektirmektedir. Bu çözüm havzasının geleneksel yöntemlerle taranması çok uzun sürmekte, genetik algoritmayla ise kısa bir sürede kabul edilebilir bir sonuç alınabilmektedir.[3]

Günümüzün karmaşık ve zor koşulları problemlere hızlı ve kolay çözüm veren yeni çözüm yöntemleri arayışına neden olmuştur. Özellikle katı (hard) optimizasyon teknikleri yerine, yumuşak hesaplama (soft computing) ve evrimsel algoritma (evolutionary algorithm) kullanımı ön plana çıkmıştır. Evrimsel yaklaşımlardan olan genetik algoritmalar da, bu arayışlar içinde önemli bir yer tutmaya başlamıştır. Uygulama başarıları artan ve sürekli geliştirilmeye çalışılan genetik algoritmalar diğer yumuşak hesaplama yöntemleri ile birlikte kullanılarak hibrid (hybrid) çözümler geliştirilmesine çalışılmaktadır. Genetik algoritmalar, doğal seçim ilkelerine dayanan bir arama ve optimizasyon yöntemidir. Temel ilkeleri John Holland tarafından ortaya atılmıştır. Temel ilkelerinin ortaya atılmasından sonra, genetik algoritmalar hakkında birçok bilimsel çalışma yayınlanmıştır. Ayrıca, genetik algoritmaların teorik kısmı ve uygulamaları hakkında birçok uluslararası konferans da düzenlenmektedir. Genetik algoritmaların, fonksiyon optimizasyonu, çizelgeleme, mekanik öğrenme, tasarım, hücresel üretim gibi alanlarda başarılı uygulamaları bulunmaktadır. Geleneksel optimizasyon yöntemlerine göre farklılıkları olan genetik algoritmalar, parametre kümesini değil kodlanmış biçimlerini kullanırlar. Olasılık kurallarına göre çalışan genetik algoritmalar, yalnızca amaç fonksiyonuna gereksinim duyar. Çözüm uzayının tamamını değil belirli bir kısmını tararlar. Böylece, etkin arama yaparak çok daha kısa bir sürede çözüme ulaşırlar.[3]

Diğer bir önemli üstünlükleri ise çözümlerden oluşan popülasyonu eş zamanlı incelemeleri ve böylelikle yerel en iyi çözümlere takılmamalarıdır. “Reeves; ulusal hükümetler ve organizasyonlar tarafından genetik algoritma tabanlı projelere, tavlama benzetimi (simulated annealing) ve yasaklı arama (tabu search) tabanlı projelere göre daha fazla kaynak ayrılmakta olduğunu belirtmektedir”. Bu ifadeden anlaşılacağı üzere, genetik algoritmaların geleneksel optimizasyon yöntemlerine olduğu gibi sözü edilen yapay zeka yöntemlerine göre de çeşitli alanlarda üstünlükleri bulunmaktadır. Bu üstünlükler genetik algoritmaların arama yapısı ile ilgilidir. Genetik algoritmaların arama yapısı ise, alt diziler teoremi ve yapı blokları hipoteziyle açıklanmaktadır. [3]

II.2.GENETİK ALGORİTMALARIN TEMEL TEOREMİ

Algoritma ilk olarak populasyon diye tabir edilen bir çözüm(kromozomlarla ifade edilir) seti ile başlatılır. Bir populasyondan alınan sonuçlar bir öncekinden daha iyi olacağı beklenen yeni bir populasyon oluşturmak için kullanılır. Yeni populasyon oluşturulması için seçilen çözümler uyumluluklarına göre seçilir. Çünkü uyumlu olanların daha iyi sonuçlar üretmesi olasıdır. Bu istenen çözüm sağlanıncaya kadar devam ettirilir[4].

Genetik algoritmaların nasıl arama yaptığı alt dizi kavramıyla açıklanmaktadır. Alt diziler, genetik algoritmaların davranışlarını açıklamak için kullanılan teorik yapılardır. Bir alt dizi, belirli dizi kümeleri arasındaki benzerliği tanımlayan bir dizidir. Alt diziler, {0, 1, *} alfabeti kullanılarak tanımlanır. Örneğin H alt dizisi, ilk konumunda 0, ikinci ve dördüncü konumunda 1 değeri olan kromozomlar kümesi içindir.

$$H = 0 1 * 1 *$$

* sembolü dizinin o konumunun hangi değeri alıp almadığının önemli olmadığı anlamındadır. Dizi o konumda 0 veya 1 değeri alabilir. Eğer bir x dizisi, alt dizinin kalıbına uyarsa x dizisine “H”nin bir örneğidir” denir. Alt dizilerin iki özelliği mevcuttur. Bu özellikler aşağıda verilmiştir.

1. Alt dizi derecesi: Bir H alt dizisinin derecesi $o(H)$ ile gösterilir ve mevcut alt dizi kalıbında bulunan sabit konumların sayısıdır. Bu sayı ikili alfabede 0 ve 1 değerlerinin sayısının toplamına eşittir.

2. Alt dizi uzunluğu: Bir H alt dizisinin uzunluğu $\bar{a}(H)$ ile gösterilir ve mevcut alt dizi kalıbında bulunan belirli ilk ve son konumlar arasındaki uzaklıktır.

Alt dizi derecesi ve alt dizi uzunluğu kavramlarının genetik algoritmaların temel teoreminde son derece önemli bir yeri değer. Alt dizi derecesi düşük, alt dizi uzunluğu kısa olan diziler “yapı blokları” olarak adlandırılır. John Holland, genetik algoritmaların işleyişinde uygun yapı bloklarının tanımlanmasını ve bu yapı bloklarının daha uygun yapı blokları elde etmek amacıyla birleştirilmesini önermektedir. Bu fikir yapı blokları hipotezi olarak bilinmektedir. Genetik algoritmanın temel teoremi ise şöyle açıklanmaktadır:

Popülasyon ortalamasının üstünde uyum gücü gösteren, kısa uzunluğa ve düşük dereceye sahip alt diziler zamanın ilerlemesiyle üstsel olarak çoğalırlar. Bu çoğalma, genetik işlemler aracılığı ile gerçekleşmektedir ve sonucunda ana-babadan

daha üstün özellikler taşıyan bireyler ortaya çıkmaktadır. Bu çözüm kalitesinin kuşaktan kuşağa artması iki nedene bağlanmaktadır. Bu nedenler şöyle açıklanabilir. Başarısız olan bireylerin üreme şansları azaltıldığı için kötüye gidiş zorlaşmaktadır.

Genetik algoritmaların yapısı kötüye gidişi engellemekle kalmamakta, genetik algoritmaların temel teoremi uyarınca, zaman içinde hızlı bir iyiye gidiş de sağlayabilmektedir. Genetik algoritmaların işleme adımları incelendiğinde bu nedenler daha iyi anlaşılacaktır. Genetik algoritmalar yapısı gereği, kötü bireyleri yani uygun olmayan çözümleri, operatörleri sayesinde elemektedir. Bu işlemler bir döngü içerisinde durdurma kriteri sağlanana kadar devam etmektedir.[3]

II.3. BASİT GENETİK ALGORİTMA

Birçok alanda uygulama imkânı ve uygulamaları olan genetik algoritmaların işleme adımları şöyle açıklanabilir. Arama uzayındaki tüm mümkün çözümler dizi olarak kodlanır. Genellikle rastsal bir çözüm kümesi seçilir ve başlangıç popülasyonu olarak kabul edilir. Her bir dizi için bir uygunluk değeri hesaplanır, bulunan uygunluk değerleri dizilerin çözüm kalitesini gösterir.

Bir grup dizi belirli bir olasılık değerine göre rastsal olarak seçilip çoğalma işlemi gerçekleştirilir. Yeni bireylerin uygunluk değerleri hesaplanarak, çaprazlama ve mutasyon işlemlerine tabi tutulur. Önceden belirlenen kuşak sayısı boyunca yukarıdaki işlemler devam ettirilir. İterasyon, belirlenen kuşak sayısına ulaşıncaya kadar işlem sona erdirilir. Amaç fonksiyonuna göre en uygun olan dizi seçilir. Genetik algoritmalar bir çözüm uzayındaki her noktayı, kromozom adı verilen ikili bit dizisi ile kodlar. Her noktanın bir uygunluk değeri vardır. Tek bir nokta yerine, genetik algoritmalar bir popülasyon olarak noktalar kümesini muhafaza eder. Her kuşakta, genetik algoritma, çaprazlama ve mutasyon gibi genetik operatörleri kullanarak yeni bir popülasyon oluşturur. Birkaç kuşak sonunda, popülasyon daha iyi uygunluk değerine sahip üyeleri içerir. Bu, Darwin'in rastsal mutasyona ve doğal seçime dayanan evrim modellerine benzemektedir. Genetik algoritmalar, çözümlerin kodlanmasını, uygunlukların hesaplanmasını, çoğalma, çaprazlama ve mutasyon operatörlerinin uygulanmasını içerir.[3]

II.3.1. Çözümlerin Kodlanması

Bir problemin çözümü için genetik algoritma geliştirmenin ilk adımı, tüm çözümlerin aynı boyutlara sahip bitler dizisi biçiminde gösterilmesidir. Dizilerden her biri, problemin olası çözümler uzayındaki rastsal bir noktayı simgeler (Yeniay, 2001: 38). Parametrelerin kodlanması, probleme özgü bilgilerin genetik algoritmanın kullanacağı şekle çevrilmesine olanak tanır.

II.3.2. İlk Popülasyonun Oluşturulması

Olası çözümlerin kodlandığı bir çözüm grubu oluşturulur. Çözüm grubu popülasyon, çözümlerin kodları da kromozom olarak adlandırılır. İkili alfabenin kullanıldığı kromozomların gösteriminde, ilk popülasyonun oluşturulması için rastsal sayı üreticileri kullanılabilir. Rastsal sayı üreticisi çağrılır ve değer 0,5'den küçükse konum 0'a değilse 1 değerine ayarlanır. Birey sayısının ve kromozom uzunluğunun az olduğu problemlerde yazı-tura ile de konum değerleri belirlenebilmektedir. Genetik algoritmalarda ikili kodlama yöntemi dışında, çözümü aranan probleme bağlı olarak farklı kodlama yöntemleri de kullanılmaktadır.

II.3.3. Uygunluk Değerinin Hesaplanması

Bir kuşak oluşturulduktan sonraki ilk adım, popülasyondaki her üyenin uygunluk değerini hesaplama adımıdır. Örneğin, bir maksimizasyon problemi için i . üyenin uygunluk değeri $f(i)$, genellikle o noktadaki amaç fonksiyonunun değeridir. Çözümü aranan her problem için bir uygunluk fonksiyonu mevcuttur. Verilen belirli bir kromozom için uygunluk fonksiyonu, o kromozomun temsil ettiği çözümün kullanımıyla veya yeteneğiyle orantılı olan sayısal bir uygunluk değeri verir. Bu bilgi, her kuşakta daha uygun çözümlerin seçiminde yol göstermektedir. Bir çözümün uygunluk değeri ne kadar yüksekse, yaşama ve çoğalma şansı o kadar fazladır ve bir sonraki kuşakta temsil edilme oranı da o kadar yüksektir.

II.3.4. Çoğalma İşleminin Uygulanması

Çoğalma operatöründe diziler, amaç fonksiyonuna göre kopyalanır ve iyi kalıtsal özellikleri gelecek kuşağa daha iyi aktaracak bireyler seçilir. Üreme operatörü yapay bir seçimdir. Dizileri uygunluk değerlerine göre kopyalama, daha

yüksek uygunluk değerine sahip dizilerin, bir sonraki kuşaktaki bir veya daha fazla yavruya daha yüksek bir olasılıkla katkıda bulunması anlamına gelmektedir. Çoğalma, bireyleri seçme işleminden, seçilmiş bireyleri bir eşleme havuzuna kopyalama işleminden ve havuzda bireyleri çiftler halinde gruplara ayırma işleminden oluşur. Uygunluk değerinin hesaplanması adımından sonra mevcut kuşaktan yeni bir populasyon yaratılmalıdır. Seçim işlemi, bir sonraki kuşak için yavru üretmek amacıyla hangi ailelerin yer alması gerektiğine karar vermektedir. Bu doğal seçimdeki en uygunun yaşaması durumuna benzerdir. Bu yöntemin amacı, ortalama uygunluğun üzerindeki değerlere çoğalma fırsatı tanımaktır. Bir dizinin kopyalanma şansı, uygunluk fonksiyonuyla hesaplanan dizinin uygunluk değerine bağlıdır. Seçim yöntemlerine rulet tekerleği seçimi, turnuva seçimi ve sıralama seçimi gibi seçim yöntemleri örnek verilebilir.

II.3.5. Çaprazlama İşleminin Uygulanması

Mevcut gen havuzunun potansiyelini araştırmak üzere, bir önceki kuşaktan daha iyi nitelikler içeren yeni kromozomlar yaratmak amacıyla çaprazlama operatörü kullanılmaktadır. Çaprazlama genellikle, verilen bir çaprazlama oranına eşit bir olasılıkla seçilen aile çeşitlerine uygulanmaktadır.

Genetik algoritmanın performansını etkileyen önemli parametrelerden biri olan çaprazlama operatörü doğal populasyonlardaki çaprazlamaya karşılık gelmektedir. Çoğalma işlemi sonucunda elde edilen yeni populasyondan rastsal olarak iki kromozom seçilmekte ve karşılıklı çaprazlama işlemine tabi tutulmaktadır. Çaprazlama işleminde dizi uzunluğu L olmak üzere, $1 \leq k \leq L-1$ aralığında k tamsayısı seçilmektedir. Bu tamsayı değerine göre dizi çaprazlamaya uğratılır. En basit çaprazlama yöntemi tek noktalı çaprazlama yöntemidir. Tek noktalı çaprazlama yapılabilmesi için her iki kromozomun da aynı gen uzunluğunda olması gerekir. İki noktalı çaprazlamada ise kromozom iki noktadan kesilir ve karşılıklı olarak pozisyonlar yer değiştirilir.

II.3.6. Mutasyon İşleminin Uygulanması

Çaprazlama mevcut gen potansiyellerini araştırmak üzere kullanılır. Fakat populasyon gerekli tüm kodlanmış bilgiyi içermez ise, çaprazlama tatmin edici bir çözüm üretmez. Bundan dolayı, mevcut kromozomlardan yeni kromozomlar üretme

yeteneğine sahip bir operatör gerekmektedir. Bu görevi mutasyon gerçekleştirir. Yapay genetik sistemlerde mutasyon operatörü, bir daha elde edilemeyebilir iyi bir çözümün kaybına karşı koruma sağlamaktadır. İkili kodlama sisteminin kullanıldığı problemlerde mutasyon, düşük bir olasılık değeri altında bir bit değerini (0 veya 1 olabilir) diğer bit değerine dönüştürür. İkili kodlama sisteminin kullanılmadığı problemlerde ise daha farklı mutasyon yöntemleri kullanılmaktadır. Hangi yöntem kullanılırsa kullanılsın, mutasyonun genel amacı, genetik çeşitliliği sağlamak veya korumaktır.

II.3.7. Yeni Kuşağın Oluşması ve Döngünün Durdurulması

Yeni kuşak çoğalma, çaprazlama ve mutasyon işlemlerinden sonra tanımlanmakta ve bir sonraki kuşağın ebeveynleri olmaktadır. Süreç yeni kuşakla çoğalma için belirlenen uygunluk ile devam eder. Bu süreç, önceden belirlenen kuşak sayısı kadar veya bir hedefe ulaşıncaya kadar ya da başka bir durdurma kriteri sağlanana kadar devam eder. İstenen hassasiyet derecesine göre de maksimum iterasyon sayısı belirlenebilmekte ve iterasyon bu sayıya ulaştığında döngü durdurulabilmektedir. Durdurma kriteri iterasyon sayısı olabileceği gibi hedeflenen uygunluk değeri de olabilmektedir. [3]

II.4. GENETİK ALGORİTMALARDA PARAMETRE SEÇİMİ

Parametreler, genetik algoritma performansı üzerinde önemli etkiye sahiptir. Optimal kontrol parametreleri bulmak için birçok çalışma yapılmıştır fakat tüm problemler için genel olarak kullanılacak parametreler bulunamamıştır. Bu parametreler, kontrol parametreleri olarak adlandırılmaktadır. Kontrol parametreleri popülasyon büyüklüğü, çaprazlama Olasılığı, mutasyon Olasılığı, kuşak aralığı, seçim stratejisi ve fonksiyon ölçeklemesi olarak sayılabilir. Bu parametreler aşağıda açıklanmıştır.

- **Popülasyon Büyüklüğü:** Genetik algoritma kullanıcısı tarafından verilen en önemli kararlardan birisidir. Bu değer çok küçük olduğunda, genetik algoritma yerel bir optimuma takılabilmektedir. Popülasyonun çok büyük olması ise çözüme ulaşma zamanını arttırmaktadır. Bu konuda Goldberg 1985’de, yalnızca kromozom

uzunluđuna bađlı bir populasyon byklđ hesaplama yntemi nermiřtir. Ayrıca Schaffer ve arkadařları 1989'da ok sayıda test fonksiyonları zerinde yaptıkları arařtırmalar sonucunda, 20-30 arası bir populasyon byklđnn iyi sonular verdiđini belirtmiřlerdir.

- **aprazlama Olasılıđı:** aprazlamanın amacı, mevcut iyi kromozomların zelliklerini birleřtirerek daha uygun kromozomlar yaratmaktır. Kromozom iftleri $P(c)$ Olasılıđı ile aprazlamaya uđramak zere seilirler. aprazlamanın artması, yapı bloklarının artmasına neden olmakta fakat aynı zamanda bazı iyi kromozomların da bozulma Olasılıđını arttırmaktadır.

- **Mutasyon Olasılıđı:** Mutasyonun amacı populasyondaki genetik eřitliliđi korumaktır. Mutasyon $P(m)$ Olasılıđı ile bir kromozomdaki her bitte meydana gelebilir. Eđer mutasyon Olasılıđı artarsa, genetik arama rastsal bir aramaya dnřr. Fakat bu aynı zamanda kayıp genetik malzemeyi tekrar bulmada yardımcı olmaktadır.

- **Kuřak Aralıđı:** Her kuřaktaki yeni kromozom oranına kuřak aralıđı denmektedir. Genetik operatrler iin ka tane kromozomun seildiđini gsterir. Yksek bir deđer birok kromozomun yer deđiřtirdiđi anlamına gelmektedir.

- **Seim Stratejisi:** Eski kuřađı yenilemenin eřitli yntemleri mevcuttur. Kuřaksal stratejide, mevcut populasyondaki kromozomlar tamamen yavrular ile yer deđiřtirir. Populasyonun en iyi kromozomu da yenilendiđinden dolayı bir sonraki kuřađa aktarılamaz ve bu yzden bu strateji en uygun (elitist) stratejisiyle beraber kullanılmaktadır. En uygun stratejisinde, populasyondaki en iyi kromozomlar hibir zaman yenilenmemektedir, bundan dolayı ođalma iin en iyi zm her zaman elveriřlidir. Denge durumu stratejisinde ise, her kuřakta yalnızca birkaç kromozom yenilenmektedir. Genellikle, yeni kromozomlar populasyona katıldıđında en kt kromozomlar yenilenir.

- **Fonksiyon leklemesi:** Dođrusal lekleme, stsel lekleme gibi yntemler mevcuttur. Probleme gre en uygun lekleme ynteminin seilmesi genetik algoritmanın etkin iřlemesi aısından nem tařımaktadır.

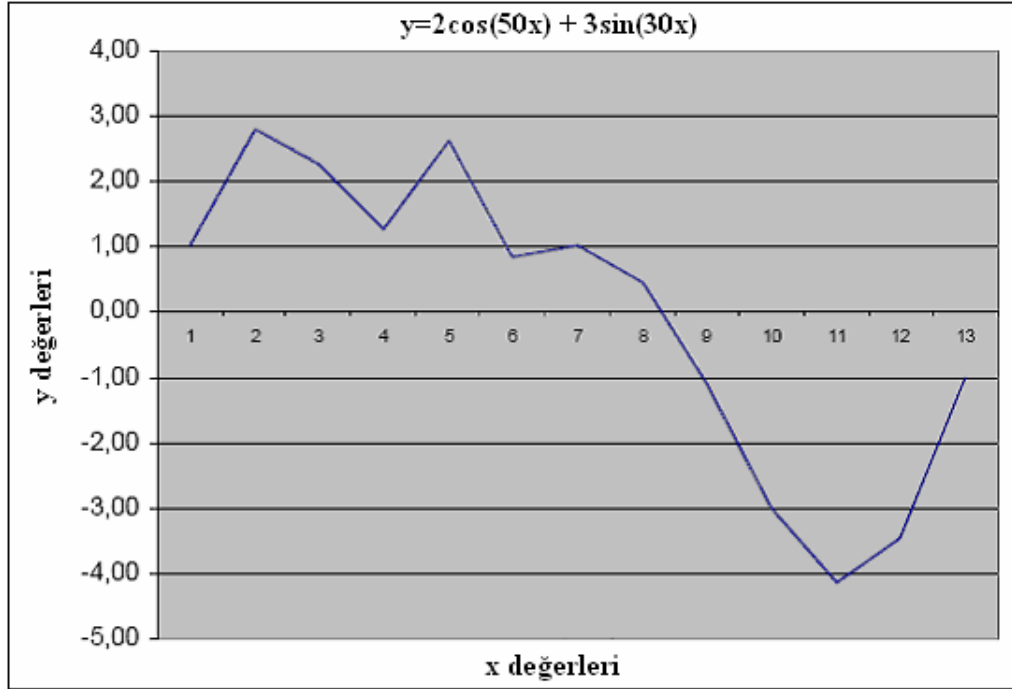
II.5. GENETİK ALGORİTMANIN BİR FONKSİYON OPTİMİZASYONUNDA UYGULANMASI

Burada, genetik algoritmaların nasıl çalıştığını göstermek için tek değişkenli bir fonksiyonun optimizasyonu ele alınacaktır. Örnek fonksiyon aşağıdaki gibi olup, grafiği Şekil II.1’de gösterilmiştir:

$$y = 2\cos(50x) + 3\sin(30x) \quad 0 \leq x \leq 12$$

Amaç, 0 ile 12 aralığında fonksiyonu minimize eden x^* değerini bulmaktır. Çözümü zor olmayan bu problem çeşitli yöntemlerle ve elle çözülebilir. Bu yöntemlerden biri ile çözüldüğünde, $x^*=10,2$ 'ye karşılık gelen $f(x^*)$ değeri -4.15 olarak bulunmaktadır. Elde edilen değer, burada sadece genetik algoritmayla yapılan çözümün sonucu ile karşılaştırma amacıyla hesaplanmıştır. Genetik algoritmayla fonksiyonun optimum değerini bulmak için ise Turbo Pascal’da kodlanmış Goldberg’in basit genetik algoritma kodu örneğe uyarlanarak kullanılmıştır. Problemin çözümü için ikili kodlama kullanılmış ve kromozomlar 30 bitlik dizilerden oluşturulmuştur. Kromozom uzunluğu gerçek uygulamalarda uzun alınmaktadır. İlk popülasyon oluşturulmuş, minimum değer olarak $-4,9039264020$ elde edilmiştir. Kullanılan genetik algoritmada, çaprazlama ve mutasyon olmak üzere iki temel genetik operatör kullanılmıştır. Parametreler ise şöyle seçilmiştir:

Popülasyon büyüklüğü	: 100
Çaprazlama Olasılığı	: 0,70
Mutasyon Olasılığı	: 0,01
Durdurma kriteri	: 15 kuşak
Seçim stratejisi	: kuşaksal strateji



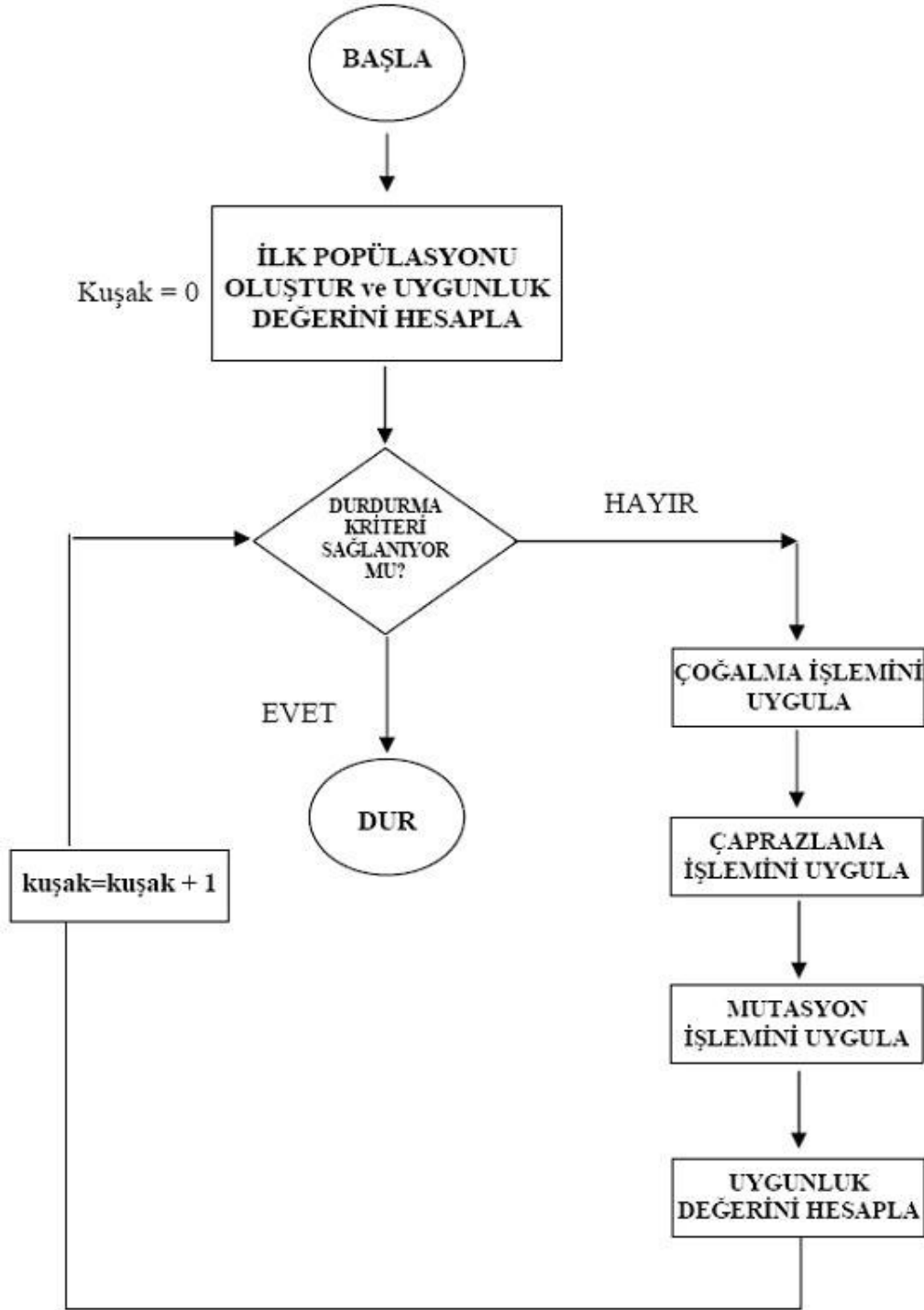
Şekil II.1. $F(x)=2\cos(50x) + 3\sin(30x)$ Grafiği

Deney sonuçları: İlk populasyon yaratıldıktan sonra program, Pentium 120 mHz bir PC’de çalıştırılmıştır. Hesaplama zamanı 13 saniye olarak ölçülmüştür. Tablo II.1’de iyileşmenin görüldüğü kuşaklar ve minimum değerler özetlenmiştir. Genetik algoritmanın çalıştırılması sonucunda 12. kuşakta minimum değer bulunmuştur. Bu değer, optimal çözüme çok yakın olup kısa bir sürede elde edilmiştir. Kullanılan parametreler ise, yapılan çeşitli deneysel çalışmalara ve literatürde karşılaşılan benzer örneklere dayanılarak belirlenmiştir.

Örnekte uygulanan genetik algoritmanın akış diyagramı şekil II.2’de verilmiştir. [3]

Tablo II.1. İyileşmenin Görüldüğü Kuşaklar ve Uygunluk Değerleri

Generasyon Sayısı	Minimum Uygunluk Değeri
0	- 4,9039264020
1	- 4,6593255932
2	- 4,5596444645
9	- 4,1082673943
12	- 4,1131100874



Şekil II.2. Genetik Algoritma Akış Diyagramı

II.6. GENETİK ALGORİTMALARIN UYGULAMA ALANLARI

Karmaşık problemleri hızlı ve optimale yakın olarak çözebilen genetik algoritmalar, çeşitli problem tiplerine uygulanabilmektedir. Büyük çözüm uzaylarının geleneksel yöntemlerle taranması hesaplama zamanını arttırmaktadır. Ancak bu tip problemlere, genetik algoritmalar ile kısa sürede, kabul edilebilir çözümler bulunabilmektedir.

Genetik algoritmalar özellikle çözüm uzayının geniş, süreksiz ve karmaşık olduğu problem tiplerinde başarılı sonuçlar vermektedir. Genetik algoritmaların uygulama alanları bu çalışmada genel uygulama alanları ve işletmelerdeki yaygın uygulama alanları olmak üzere iki sınıfa ayrılarak incelenmiştir.

II.6.1. Genel Uygulama Alanları

Genetik algoritmaların genel uygulama alanları aşağıdaki gibi verilebilir:

• Optimizasyon

Bir arama yöntemi olan genetik algoritmalar, farklı bilim dallarındaki optimizasyon problemlerini çözmeye kullanılmaktadır. Genetik algoritmaların uygulandığı optimizasyon problemleri, fonksiyon optimizasyonu ve birleşik (combinatorial) optimizasyonu altında toplanabilir.

Genetik algoritma araştırmalarının önemli bir bölümü fonksiyon optimizasyonu ile ilgilidir. Genetik algoritmalar, geleneksel optimizasyon tekniklerine göre zor, süreksiz ve gürültü (noisy) içeren fonksiyonları çözmeye daha etkindirler. Optimize edilecek amaç fonksiyonunun süreksiz olması halinde, süreksizlik noktalarında fonksiyonun türevi alınamayacağından, türev almaya dayalı optimizasyon yöntemleri kullanılamamaktadır. Oysa genetik algoritmalar, problemlerin çözümü için türev veya diğer yardımcı bilgilere gereksinim duymadığından özellikle bu tip problemlerin çözümünde geleneksel yöntemlere göre önemli bir üstünlük sağlamaktadır.

Genetik algoritmaların uygulandığı diğer bir optimizasyon problem sınıfı olan birleşik optimizasyon problemleri ise, istenen amaçlara ulaşmak üzere, sınırlı kaynakların etkin tahsis edilmesiyle ilgilidir. Bu sınırlar genel olarak, işgücü, tedarik veya bütçe ile ilgilidir. Sözü geçen “birleşik” kelimesi, yalnızca sonlu sayıda

alternatif uygun çözümün mevcut olması ile ilgilidir. Birleşik optimizasyon, iyi tanımlanmış bir problem uzayında bir veya daha fazla optimal çözüm bulma sürecidir. Bu tip problemler yönetim biliminin tüm dallarında da (finans, pazarlama, üretim, stok kontrolü, veri-tabanı yönetimi vb.) ortaya çıkmaktadır. Gezgin satıcı problemi, araç rotalara problemi, Çinli postacı problemi, iş atölyesi çizelgeleme problemi, atama problemi, yerleşim tasarımı problemi ve sırt çantası problemi birleşik optimizasyon problemlerine örnektir. Birleşik optimizasyon problemlerinde, incelenen değişken sayısı arttıkça, çözüme ulaşma zamanı üstsel olarak artmaktadır. Çözüm uzayının tamamının taranmasını gerektiren geleneksel çözüm yöntemlerinde problem çözümü değişken sayısının artmasıyla imkânsız hale gelebilmektedir. Genetik algoritmalar ise çözüm uzayının yalnızca belirli bir kısmını taradığı ve eş zamanlı arama yaptığı için, bu tip problemlerde çözüme daha kısa sürede ulaşabilmektedir. Çeşitli avantajlarına rağmen genetik algoritmaların uygulamalarında bir takım sorunlarla da karşılaşmaktadır. Bu sorunları aşmak için çeşitli yöntemler geliştirilmiştir. Buna kısıtların ele alınmasındaki soruna karşı ceza fonksiyonu yönteminin kullanılması örnek verilebilir. Ancak, bulunan çeşitli yöntemlere rağmen bu konuda yeni yaklaşımlara gereksinim duyulmaktadır.

• Otomatik Programlama ve Bilgi Sistemleri

Genetik algoritmaların yaygın olarak kullanıldığı alanlardan biri, belirli ve özel görevler için gerekli olan bilgisayar programlarını geliştirmedir. Ayrıca, diğer hesaplama gerektiren yapıların tasarımı için de kullanılmaktadır. Bunlara örnek olarak, bilgisayar çipleri tasarımı, ders programı hazırlanması ve ağların çizelgelenmesi verilebilir. Genetik algoritmalar kullanılarak dağıtılmış bilgisayar ağlarının tasarımı da gerçekleştirilmektedir. Bu problem tipinde ağ güvenilirlik parametrelerini (çap, ortalama uzaklık ve bilgisayar ağ güvenilirliği gibi) optimize etmek için birden fazla amaç fonksiyonu kullanılmaktadır. Genetik algoritmalar ile 100 düğüme kadar olan ağlar başarıyla tasarlanmıştır. Ağ tasarımında genetik algoritmaların kullanılması, tasarım sürelerinin ve maliyetlerinin azalmasında önemli bir katkı sağlamıştır. Özellikle, maksimum miktardaki verinin minimum iletişim hattıyla taşınmasında yüksek bir performans göstermiştir. Ayrıca genetik algoritmaların kullanımıyla, çeşitli alanlara dağıtılmış bir sistem için en uygun dosya tahsisatı gerçekleştirilmektedir.

• Mekanik Öğrenme

Mekanik öğrenme; ilki, gözlenmiş bir veri takımını anlamak ve yorumlamak, ikincisi de görülmemiş objelerin özelliklerini tahmin etmek olan iki temel amaç için model kurmayı amaçlar. Parametrik istatistikten ziyade çok büyük veri takımlarının yönetimi üzerinde çalışır. Kullandığı metotların Çoğu dağılımdan bağımsız metotlar olarak sınıflanabilir. Uygun model seçimi için işe problem hakkındaki varsayımlarla başlamaz. Onun yerine uygun model yapısını belirlemek için doğrudan mevcut veriden hareketle bir araç kutusu yaklaşımı kullanır. Sınıflama sistemi, genetik algoritmaların mekanik öğrenme alanında bir uygulamasıdır. Basit dizi kurallarını öğrenen bir mekanik öğrenme sistemi olan sınıflama sisteminin kural ve mesaj sistemi, özel bir üretim sistemi olarak adlandırılabilir. Bu üretim sistemi, “eğer-sonra” kural yapısını kullanır. Bir üretim kuralı, “eğer” yapısından sonra belirtilen durum için, “sonra” yapısından sonra gelen faaliyetin gerçekleştirilmesini içerir. Genetik algoritmalar, sınıflama sistemlerinde kural-bulma mekanizması olarak kullanılmaktadırlar. Genetik algoritmalar Ayrıca, sinir ağlarında ve proteinin yapısal analizinde de kullanılmaktadır.

• Ekonomik ve Sosyal Sistem Modelleri

Bir sistemi ölçen ampirik olarak gözlenmiş değişkenler arasındaki matematiksel ilişkiyi keşfetme problemi ekonomide en önemli problemlerden biridir. Pratikte gözlenmiş veri gürültü içerebilir ve kapsanan ilişkileri kesin ve açık bir şekilde açıklayacak bir yol bilinmeyebilir. Bu tip problemler, sembolik sistem tanımlama, kara kutu, veri madenciliği ve modelleme problemleri olarak bilinir. Eğer keşfedilen model, sistemin durum değişkenlerinin gelecek değerlerini tahmin etme için kullanılacaksa problem öngörüleme problemi adını alır. Geleneksel doğrusal, kuadratik ve üstsel regresyon modellerinde sapma hataları minimize edilerek fonksiyonlara uygun sayısal katsayılar bulunur. Buradaki yaklaşım, model seçildikten sonra uygun sayısal katsayıların aranmasıdır. Gerçek problem ise verinin değerlendirilmesi için hangi tip modelin uygun olduğunun kararıdır. Keyfi bir matematiksel ilişkiyi açıklamada bilgisayarlar, bu ilişkiyi formüller ve denklemler aracılığı ile açıklamaktan daha esnekler. Bu nedenle, bu tip ilişki açıklamaları için sembolik regresyon kullanılabilir. Sembolik regresyonlar, hem fonksiyon formunu hem de o fonksiyondaki uygun katsayıyı araştırmaktadır. Bunu bulma ise, verilen girdiler için arzu edilen çıktıları üreten özel bir hesaplama programı ile

program uzayında arama yapmaya benzemektedir. Genetik algoritmaların kullanıldığı genetik programlamayla bu tip problemlere tatmin edici çözümler çok daha kolay getirilebilmektedir. Genetik algoritmalar yenilik sürecinin modellenmesi amacıyla da kullanılmaktadır. Ayrıca genetik algoritmaların, fiyat verme stratejilerinin gelişim süreçlerini ve kazanç getiren pazarların ortaya çıkış süreçlerini modelleme alanlarında da kullanımları oldukça yaygındır. Genetik algoritmalar sosyal sistemlerin evrimsel yönlerini anlamak amacıyla kullanılmaktadır. Bunlara örnek olarak işbirliğinin evrimi, iletişimin evrimi ve karıncalardaki iz takibi davranışının evrimi verilmektedir.

II.6.2. İşletmelerdeki Uygulama Alanları

Genetik algoritmalar; başta üretim/işlemler olmak üzere finans ve pazarlama gibi işletmelerin fonksiyonel alanlardaki birçok farklı iş probleminin çözümü için kullanılmaktadır. Genetik algoritmaların özellikle, kaynak tahsisi, iş atölyesi çizelgelemesi, makine parça gruplaması ve bilgisayar ağ tasarımı gibi çeşitli alanlarda uygulamaları mevcuttur. İşletmelerdeki en yaygın kullanım alanları aşağıdaki gibi verilebilir:

• Finans

Genetik algoritmalar, finansal modelleme uygulamaları için son derece uygundur ve amaç fonksiyonu odaklıdır. Finans problemlerinde genel olarak, amaç fonksiyonları tahmin etme gücüne veya bir kıyaslama sonucuna bağlı getirilerdeki gelişmeleri içerir. Kullanılan araç ve problemler arasında mükemmel bir eşleşme mevcuttur. Özellikle hisse senedi fiyatlarındaki değişim kalıplarını tahmin etmede ve bulmada, kaynak tahsisi ve uluslararası sermaye tahsisi stratejilerini belirlemede genetik algoritmalar kullanılabilir. Bu yaklaşımla, kısıtlanmış portföy optimizasyonu, endeks izleme, işlem maliyetleri ve risk tercihleri kısıtlarının da katıldığı çok dönemli portföy yönetim sistemlerinin kurulması, yine minimum işlem lotlu portföy seçimi problemlerin çözümü yapılabilmektedir. Daha yüksek getiriler elde etmek için FX piyasalarındaki ticari kuralları geliştirmede (al-tut stratejilerinden daha karlı olanları bularak) genetik algoritmalar kullanılabilir. Ayrıca, müşterilerinin kredi değerliliğini ölçmede, yatırım araçlarının performansını belirlemede, işletmedeki mali kayıpların araştırılmasında, finansal opsiyonların geliştirilmesinde kullanılan veri madenciliğine uygulanabilmektedir. Müşteri kredi

değerliliğini ölçme, kredi kartı puanlama, piyasalar ile ilgili tahminleri ve şirketlerdeki iflas tahminlerini yapma genetik algoritmaların en sık uygulandığı finans problemlerindedir. Finans problemlerinin çözümünde genetik algoritmalar, bulanık ve yapay sinir ağları yaklaşımlarıyla birlikte kullanılmaktadır. Yumuşak hesaplama ve hibrid genetik algoritma yaklaşımı sık görülmektedir. Ayrıca, çözüm performansı açısından finans problemlerindeki genetik algoritma çözümleri yasaklı arama, tavlama benzetimi arama metotları ile karşılaştırılmakta ve o probleme uygun çözüm yöntemi önerilmektedir. Genetik algoritmaların optimal kaynak tahsisi problemlerine uygulanması ile ortalama-varyans optimumundan farklı çözüm yöntemi geliştirilmiş ve kuadratik optimizasyona genetik algoritmalar uygulanmış olmaktadır.

• Pazarlama

Tüketicilere ait verileri analiz etmek, çeşitli tüketici kalıpları çıkarmak ve bu kalıplara dayanarak pazarlama stratejileri uygulamak, pazarlamanın en önemli fonksiyonlarından biridir. Tüketicilerin profilleri çıkarılarak, belirli satın alma kalıpları yakalanabilmektedir. Ancak tüketici profilini çıkarabilmek için, çok büyük veri tabanlarını işletme amaçları doğrultusunda hızlı ve etkin biçimde kullanmak gerekmektedir. Burada kullanılan teknik veri madenciliğidir. Veri madenciliği, çok geniş veri tabanlarından veriyi süzme tekniğidir. Pazarı ve tüketiciyi tanımada son derece önemli rol oynayan veri madenciliği, veriyi bilgiye bilgiyi de güvenli kararlara dönüştürür. Veri madenciliğinin verimlilik, karlılık, müşteri tatmini ve rekabet edebilme yeteneği gibi yaşamsal konularda işletme üzerinde çok önemli etkileri bulunmaktadır. Rekabet edebilme yeteneği karar alma kalitesine bağlıdır ve bundan dolayı işletmeler sürekli karar kalitelerini geliştirmeye çalışırlar. Veri madenciliğinde kullanılan tekniklerden birisi de genetik algoritmadır. Genetik algoritma tabanlı yaklaşım kullanılarak veri yığınlarından modeller elde edilmektedir. Bu konuda Siddhartha Bhattacharyya'nın 1999 yılında ve Marshall'in da aynı yılda yayınlanmış çalışması bulunmaktadır.

• Üretim/İşlemler

Genetik algoritmaların en çok uygulandığı alanların başında üretim/işlemler gelmektedir. Burada üretim/işlemler alanıyla ilgili çeşitli problemler ele alınacaktır.

• Montaj Hattı Dengeleme Problemi

Montaj işlemi endüstrilerde çok önemli bir rol oynamaktadır. Nof ve arkadaşlarının 1997'de yayınlanan çalışmalara göre üretilen mamullerin montajı, toplam üretim zamanının %50'sine, toplam birim üretim maliyetinin %20'sine ve işçilik maliyetlerinin %30 - %50'sine karşılık gelmektedir. Bundan dolayı montaj hattı dengeleme problemi, firmalar açısından yaşamsal öneme sahiptir. Bu konuda; Leu, Motheson ve Rees'in 1994'de, Rubinovitz ve Levitin'in 1995'de, Tsujimura, Gen ve Kubota'nın 1995'de, Sabuncuoğlu, Erel ve Tanyer'in 1999'da, Ponnambalam, Aravindan ve Naidu'nun 2000'de yayınlanmış çalışmaları bulunmaktadır. Bu çalışmalardan Tsujimura, Gen ve Kubata'nın 1995 yılında yayınlanan çalışmasında; her bir iş istasyonundaki toplam işlem zamanlarını minimize etmeyi hedefleyen amaç fonksiyonunun çözümü, genetik algoritma ile bulanık küme mantığı birlikte kullanılarak gerçekleştirilmiştir.

• Çizelgeleme Problemi

Genetik algoritmaların çizelgeleme problemine ilk uygulama çalışması, Davis tarafından 1985 yılında yapılmıştır. 1987'de Liepins ve arkadaşları, belirli teslim tarihleri ve işlem süreleri olan işlerin çizelgelenmesi problemini araştırmışlardır. Bu problem en basit çizelgeleme problemi adlandırılmaktadır. 1993'de Gupta ve arkadaşları, akış zamanını minimize etme amacını taşıyan tek makine modeli üzerindeki çalışmalarını yayınlamışlardır. Lee ve Kim 1995'de gecikme ve sarkma cezalarını da modele katan çalışmalarını sunmuşlardır. Cheng ve arkadaşları gene aynı yıl, özdeş paralel makinelerden oluşan model üzerindeki çalışmalarını yayınlamışlardır. Bunun dışında; iş atölyesi çizelgelenmesi problemi için Biegel ve Davern'nin 1990'da, akış atölyesi problemi için Badami ve Parks'in 1991'de, süreç planlama problemi için Vancza ve Markus'un 1991'de yayınlanmış çalışmaları bulunmaktadır. Genel olarak genetik algoritmalar, çizelgeleme problemlerine optimale yakın çözüm bulmuşlardır. Fakat çözüm bulma süreleri diğer çözüm yöntemlerine göre oldukça hızlı olmuştur.

• Tesis Yerleşim Problemi

Tesis yerleşim problemleri araç/gereçleri veya diğer kaynakları belirli bir kritere göre optimum performans sağlayacak şekilde yerleştirme kararını içermektedir. Bu gibi kararlar, araç/gereçlerin genellikle farklı ürünleri üretme

esnasında kullanılmasından dolayı karmaşık hale gelmektedir. Her ürünün kendine özgü gereksinimleri olabilir ve tüm ürünler için toplam üretim maliyetinin optimum olması sağlanacak şekilde yerleşim tasarlanabilir. Yerleşim kararları hızlı ve doğru verilmelidir. Çünkü kararların zayıflığı üretim esnasında ortaya çıkmakta ve bu da artı maliyetlere yol açmaktadır. Örneğin, üretimde robot kullanan işletmelerin tesis yerleşimi tasarımında karmaşıklık söz konusudur. Tek bir robot bir makineden diğerine parçalar taşıırken hareketsiz bir noktada sabitlenir ve yalnızca bir eksen etrafında hareket eder. Robotun hareketine göre, makineler tek-sıra, doğrusal çift-sıra, dairesel tek-sıra ve çoklu-sıra gibi dört farklı yerleşim şekliyle yerleştirilebilir. Burada, dairesel tek-sıra, doğrusal tek-sıranın özel bir durumudur. Ayrıca doğrusal çift-sıra da çoklu sıra probleminin bir alt kümesidir. Tesis yerleşim problemleri bunun gibi birçok zorluğu içermektedir. Genetik algoritmalar, bu tip problemlerin çözümünde uygun bir çözüm yöntemi olabilmektedir. Bu alanda Tam'ın 1992 yılında, Chan ve Tansri'nin 1994 yılında, Tom ve Chan'nin 1998 yılında, İşlier'in 1998 yılında ve Al-Hakim'in 2000 yılında yayınlanmış çalışmaları bulunmaktadır.

• Atama Problemi

Genel olarak atama problemi; n elemanın, n farklı göreve atanması problemidir. I . kişinin, j . işi yapma maliyeti c_{ij} 'dir. Bu durumda problem amaç fonksiyonunu minimize edecek $\{\pi_1, \dots, \pi_n\}$ atama kümesinin bulunması şeklinde tanımlanabilir. Burada problem çözümü, $\{1, \dots, n\}$ sayılarının $\{\pi_1, \dots, \pi_n\}$ permütasyonu olarak gösterilmektedir. Genetik algoritmaların atama problemlerine uygulanması konularında, Huntley ve Brown'un 1991'de, Nissen'nin 1992'de, Tate ve Smith'in 1995'de, Ahuja, Orlin ve Tiwari'nin 2000'de yayınlanmış çalışmaları mevcuttur. Ayrıca 1996 yılında Chu ve Beasley, minimum maliyetli atamanın hedeflendiği problem için genetik algoritmaların kullanıldığı bir çözüm önermiştir. 1995 yılında Zhao, Tsujimura ve Gen'nin yayınlanan makalelerinde, iş istasyonu atama problemi için genetik algoritma kullanılmıştır.

• Hücresel Üretim Problemi

Hücresel üretim kavramı, üretim sistemlerinin verimliliğini arttırmada anahtar faktörlerden biridir. Hücresel üretim, parça ailelerini belirledikten sonra, her parça ailesini ayrı bir üretim hücresinde imal ederek hücreler arası taşımaları en aza indirmeyi amaçlamaktadır. Genetik algoritmalar, hücreler arası taşımanın minimum

olduğu bir hücre kuruluşu amaçlanmasında kullanılabilir. Bu konuda Tate ve Smith'in, Kamrani ve Parsai'nin yayınlanmış çalışmaları bulunmaktadır. Ayrıca, Joines'in 1996'da yayınlanmış çalışması mevcuttur. Venugopal'in 1999'daki çalışması, hücresel üretim konusu için uygulanmış çözüm tekniklerinin genel bir değerlendirmesini içermektedir. İşler'in bu konudaki çalışmasında ise, üretim hücrelerinin yapısını temsil eden bilgilerin gösterimi, iki bölüm halindedir. Birinci bölümde, tezgah-hücre ilişkileri, bunun devamı olan ikinci bölümde de parça-hücre ilişkileri yer almaktadır. Genetik algoritmalar ile birden fazla çözüm aynı anda ele alınmakta ve bu sayede farklı bölgeler eş zamanlı olarak taranmaktadır. Bunun sonucunda da daha kısa zamanda daha uygun sonuçlar elde edilmektedir.

• Sistem Güvenilirliği Problemi

Bir sistemin güvenilirliği, belirli koşullar altında belirli bir zaman aralığında sistemin başarılı olarak çalışma Olasılığı olarak tanımlanmaktadır. Çoğu sistem, çeşitli işlemlerde kritik bir role sahiptir ve eğer sistemde arıza olursa sonuçları oldukça ciddi olmaktadır. Bu alanda optimizasyon, etkisiz parçaların sisteme en iyi şekilde tahsis edilebilme veya yararlanabilme yolunu bulmayı içermektedir. Parçalara, güvenilirliklerinin etkin olarak ölçülebilmesi için olasılıklar atanmaktadır (Dong, 2000: 7). Bu konuda; Painton ve Campbell'in 1995'de, Sasaki, Gen ve Yamashiro'nun 1995'de, Coit ve Smith'in 1996'da, Dengiz, Altıparmak ve Smith'in 1997 ve 2000'de yayınlanmış çalışmaları bulunmaktadır.

• Taşıma Problemi

Taşıma problemi; tedarikçilerden tüketicilere, talebi karşılamak üzere, minimum maliyetle tek tipte mamul gönderilmesini içermektedir. M tane tedarikçi ve n tane de tüketici mevcuttur. Tek tedarikçiden her bir tüketiciye bir birim mamul ulaştırma maliyeti bilinmektedir. Problem, tüm talebin karşılanması ve maliyet minimizasyonu şartıyla mamulün arz yerinden talep yerine optimum tahsisini sağlamaktır. Son zamanlarda, çeşitli taşıma problemlerinin çözümü için evrimsel (evolutionary) yaklaşımlarla çözüm önerileri sunulmaktadır. Michalewicz ve arkadaşları, doğrusal ve doğrusal olmayan taşıma problemleri için genetik algoritma kullanımını ilk öneren araştırmacılarıdır. Ayrıca, Gen ve Li de genetik algoritmaları taşıma problemlerinin çözümü için kullanmışlardır.

• Gezgın Satıcı Problemi

Genetik algoritmaların, birleşik optimizasyon problemlerine uygulamaları ile ilgili çeşitli çalışmalar mevcuttur. En yoğun yapılan çalışmalardan biri de gezgin satıcı problemleri için yapılmaktadır. Gezgın satıcı probleminde amaç, kastedilen toplam mesafeyi minimize eden bir yolculuk planı oluşturmaktır. Bir çok problem tipi gezgin satıcı problemi gibi modellenilebilmektedir. Bunlara örnek olarak; devre tasarımı, posta taşıyıcılarının, havayolu uçaklarının, okul otobüslerinin rotalarının bulunması verilebilir. Gezgın satıcı probleminin bir özelliği de değişken sayısı artıkça üstsel artış gösteren zaman ihtiyacı içinde çözüme ulaştırılabilmesidir. Bu durum bir örnekle şöyle açıklanabilir; bir satış görevlisinin ziyaret etmek durumunda olduğu n tane şehir olsun. Burada tüm şehirler arasındaki maksimum izlenecek rota sayısı $(n-1)!$ dır. Tüm mümkün rotaları basitçe inceleyen ve en kısa olan rotayı bulan bir algoritma kullanılır. Fakat şehir sayısı artıkça algoritmanın hesaplama için gereksinim duyduğu zaman daha da büyük bir oranda artmaktadır. Ziyaret edilmesi gereken 25 şehir varsa, algoritmanın inceleyeceği rota sayısı $24!$ 'dir. Bu da yaklaşık $6,2 \times 10^{23}$ sayısına karşılık gelmektedir. Saniyede bir milyon rota inceleme kapasitesine sahip bir bilgisayar, bu problemi, $6,2 \times 10^{17}$ saniyede yani, $1,96 \times 10^{10}$ yılda çözebilmektedir. Herhangi bir problem için kullanılan algoritmanın en yaygın performans ölçütü, algoritmanın çözüme ulaşma süresidir. Gezgın satıcı gibi değişken sayısı artıkça çözüm zamanı üstsel olarak artan problemlerde bu daha da önemlidir. Genetik algoritmalar birleşik optimizasyon problemlerini klasik yöntemlere göre çok daha kısa sürede çözmektedir. Sonuçta optima yakın ve kabul edilebilir bir çözüm bulunmaktadır.

• Araç Rotalama Problemi

Birleşik optimizasyon problemlerinin örneklerinden biri de araç rotalama problemidir. Temel araç rotalama problemi, talebi belirli olan müşterileri kapsar. Tek bir depodan araçlar ayrılmakta ve müşteri taleplerini karşılayarak tekrar depoya dönmektedir. Her aracın kapasite kistti vardır. Bu temel probleme Ayrıca, her aracın alacağı yol da mesafe kesiti olarak eklenebilir. Her bir müşterinin talebini yalnızca bir araç karşılamaktadır. Problem, bu kısıtlar altında minimum toplam maliyeti veren rotaları bulmaktır. Daha karmaşık bir araç rotalama problemi olan zaman pencereli rotalama probleminde ise amaç müşteri talebini belirli zaman aralıkları içerisinde minimum toplam maliyetle karşılamaktır. Genetik algoritmalar özellikle zaman

pencereli araç rotalama problemlerinin çözümü için kullanılmaktadır. Thangiah, Blanton ve Wainwright, Prinetto bu konuda çalışmalarda bulunan araştırmacılardan bazılarıdır.

• Minimum Yayılan Ağaç Problemi

Minimum yayılan ağaç problemi, grafik teorisinde klasik bir problemidir. Coğrafi bir alanda dağılmış çeşitli şehirlerarasında fiber-optik kablo döşeme problemi bu tip problemler için uygun örneklerden birisidir. Her şehir arasında kablo döşeme maliyeti bilinmektedir. Problemin amacı, tüm şehirleri fiber-optik kablo ağına bağlayan yerleşimi en düşük maliyetle bulabilmektir. Yerleşim herhangi iki şehir arasında veri paketinin gidip gelmesine, uzaklık ne olursa olsun olanak tanınmalıdır. Grafik teorisinde, şehirler vertisler, kablolar da kenarlar olarak tanımlanır. Burada her kenarın bir maliyeti vardır. Bu problem şöyle formüle edilmektedir. $G = (V, E)$ bağlantılı ve yönlendirilmemiş bir grafik olsun. $V = \{v(1), v(2), \dots, v(n)\}$ vertisler kümesi ve $E = \{e(1), e(2), \dots, e(m)\}$ kenarlar kümesi olsun. Her kenarın negatif olmayan w ağırlığı mevcuttur. Bu da $w = \{w(1), w(2), \dots, w(m)\}$ olarak gösterilmektedir. Yayılan ağaç, grafikteki tüm vertisleri birbirine bağlayan minimum değerli kenarlar kümesidir. Buradaki “ağaç” tanımı, döngü kuran vertisleri veya yalnızca tek bir vertisi kapsamaz. Kısacası, minimum yayılan ağaç problemi her bir olay veya nokta çiftleri arasındaki kenarların (dalların) en kısa olanının bulunarak ağaç içindeki kenarların birbirleriyle toplam en kısa yolu bulacak şekilde ilişkilendirilmesidir.

GA’ları minimum yayılan ağaç problemini çözmek için kullanırken önemli olan nokta, ağacın nasıl kodlanacağıdır. Kodlama; kenar kodlaması, vertis kodlaması veya her iki kodlama birleşik kullanılarak yapılabilmektedir. Genetik algoritmaların etkin sonuç verebilmesi için kodlamanın doğru şekilde yapılması gerekmektedir. Kodlamada, tüm ağaçlar temsil edilmeli ve her ağaç aynı sayıda kodlanmalıdır.[3]

BÖLÜM III

ÇOK AŞAMALI TORNALAMA OPERASYONLARININ GENETİK ALGORİTMA KULLANILARAK OPTİMİZASYONU

III.1. GİRİŞ

Paso sayısı, her bir paso için talaş derinliği, ilerleme miktarı ve kesme hızı gibi optimal kesme parametrelerinin seçimi her işleme prosesi için çok önemli bir konudur. Atölye uygulamasında kesme parametreleri işleme veritabanlarından veya özelleştirilmiş el kitaplarından seçilir, ancak bu kaynaklardaki aralıklar başlangıç değerleridir, optimal değerler değildir.

Kesme parametrelerinin optimizasyonu şu yönleri gerektiği için genellikle zor bir çalışmadır; işleme bilgisi, takım ömrüne ilişkin deneysel denklemler, kuvvetler, güçler, bitiş yüzeyi vs., gerçekçi sınırları geliştirmek, makine takım kapasitelerinin özellikleri, etkin bir optimizasyon kriterinin gelişimi ve matematiksel ve nümerik optimizasyon tekniklerinin bilinmesi.

Her optimizasyon prosedüründe, usta etkisinin verimini, sözde optimizasyon hedefini ve optimizasyon kriterini tanımak kritik bir durumdur. Üretim proseslerinde en sık kullanılan optimizasyon kriteri belli bir maliyettir ki, bu dalda yapılan çalışmalarda başlangıçtan bu yana bir çok yazar tarafından ele alınmıştır.

Bazen, işleme zamanı gibi diğer kriterler, talaş kaldırma oranı yada takım ömrü de kullanılmıştır. Bununla birlikte bu tek hedef yaklaşımların işleme proseslerinin karmaşık doğası yüzünden optimal kesme koşullarını düzeltecek bir

sınır değeri vardır. Burada birçok farklı ve çelişkili hedefler eş zamanlı bir şekilde optimize edilmelilerdir.

Bazı çok amaçlı yaklaşımlar kesme parametreleri optimizasyonunda anlatılmıştır, ancak esasen bir öncelik tekniği kullanırlar ki burada karar alıcı farklı hedefleri bir maliyet fonksiyonu ile birleştirir. Bu da çok hedefli problemi, optimizasyon öncesi tek hedefli problem haline dönüştürür.

Öte yandan sonraki tekniklerde, karar alıcı bir grup baskın olmayan optimal aday çözümleri ile sunulur ve o gruptan seçim yapar. Bu çözümler, kapsamlı anlamda optimaldırlar ki bütün optimizasyon hedefleri eş zamanlı olarak ele alınırken, arama boşluğundaki diğer hiç bir çözüm onlara üstün değildir. Bunlar aynı zamanda pareto-optimal çözümü olarak da bilinmektedir.

Alıntıları teknik olarak kıyaslamak, geçmiş yıllarda daha sonraki tekniklerin popülaritesini gösterir. Çok hedefli optimizasyon problemi ile uğraşmada, klasik optimizasyon metotları (ağırlıklı özet metotları, hedef programlama, minimum-maksimum metotları, vs.) etkili değildir, çünkü tek bir koşulda devamlı çözümler bulamazlar, bundandır ki istenilen Pareto-optimal çözümlerinin sayısı kadar uygulanmaları gerekmektedir.

Buna karşılık, evrimsel algoritmalar üzerine çalışmalar bu metotların yukarıda bahsedilen klasik metotların zorluklarını elemekte etkili olarak kullanılabileceğini göstermiştir.[7]

III.2. TEK ve ÇOK AŞAMALI TORNALAMANIN SEÇİM KRİTERLERİ ve BU OPERASYONLARDA YAPILAN OPTİMİZASYON ÇALIŞMALARININ TARİHSEL GELİŞİMİ

Tornalama imalat sektöründe kullanım alanı çok geniş olan bir işleme prosesidir. Bu yüzden tek yada çok aşamalı tornalama operasyonlarının sınırlamaları içinde ekonomik bir amacı tatmin edecek kesme parametrelerinin optimal seçimi çok önemli bir görevdir. Tek aşamalı bir tornalama operasyonunda parçadan kaldırılması istenilen talaş derinliği bir pasoda gerçekleştirilir. Bu tip bir operasyonda zamandan kazanılır fakat kesici daha çabuk yıpranır. Çok aşamalı bir tornalama operasyonunda ise paso sayısı en az iki olmaktadır. Bu operasyonda daha iyi bir yüzey pürüzlülüğü

elde edilmesine karşın daha fazla zaman yitirilir. Öte yandan kesici ömrü daha uzun olur. Ancak burada dikkate alınması gereken şudur ki, tek aşamalı operasyon her zaman için çok aşamalıdan daha ekonomik değildir. Esas amaç, üretilecek parçaların tezgâhta işlenmesindeki istenen özellikleri (ölçü toleransları, yüzey kalitesi, fiziksel özellikleri vb.) yerine getirmek, böylece parçaları mümkün olan en düşük maliyette veya en az zamanda gerekli özellikleri sağlayacak şekilde üretmektir.[14]

Tek aşamalı tornalama, sadece işlem müsaade edilen en yüksek devir ile sınırlandırılmışsa -ki genel de bu söz konusu değildir- optimum olur. Çoğunlukla durum operasyonunun, beygir, istenilen yüzey pürüzlülüğü, minimum takım ömrü, maksimum müsaade edilebilir devir ve müsaade edilebilir kesme hızı aralığı gibi pratik sınırlamalara bağlı olması şartından ibaret olacaktır. Dahası iki ve hatta üç aşamalı operasyonlar bile tek aşamalı operasyonlardan daha ucuza ya da daha az üretim zamanının mal olabilmektedir. [6]

Makinede işleme operasyonlarının, malzeme, yatırım ve işgücü maliyetlerinin her geçen gün fiyatlarının artması nedeniyle yüksek maliyetli oluşu, fiyatların ve uygun işlem şartlarının dikkatli analizini gerektirir. Bu nedenle tek yada çok aşamalı tornalama operasyonunun seçim kriterlerine ilişkin iyi bir bilgi edinilmesi gerekir.

İşleme ekonomisinin orijinal çalışması Gilbert'e aittir. 2 ekonomik hedef üzerine çalışılmıştır; maksimum üretim oranı ve minimum üretim maliyeti. Literatürde, tornalama operasyonlarına ait kesme parametreleri üzerine bir çok optimizasyon çalışması hazırlanmıştır. Optimal kesme parametrelerini belirlemek için kesme parametrelerini kesme performansı ile ilişkilendirmek için güvenilir matematik modellerin formüle edilmesi gereklidir. Bunun yanı sıra şu iyice bilinir ki, güvenilir matematik modelleri elde etmek kolay değildir. Son yıllarda, kesme parametrelerini kesme performansı ile ilişkilendirecek uygun öğrenme gereçlerinin kullanımı aşama aşama güvenilir ve efektif bir modelleme tekniği olarak ün kazanıyor.

Tornalama operasyonları için güvenilir bir modelin kurulmasıyla, bir optimizasyon algoritması optimal kesme parametrelerini belirlemek için bu modele uygulanır. İşlenmiş kısmın geometrik modeli ve çeşitli zamanlar ile çok aşamalı tornalama operasyonunun maliyet elemanları ayrıca bu optimizasyon prosesinde verilmiştir. Bu optimal kesme parametreleri yüzey pürüzlülüğünün ve kesme kuvvetinin müsaade edilebilir bir limitinin sınırlamasına sahip ya maksimum üretim oranı ya da minimum üretim maliyetine ait bir hedef fonksiyonuna bağlıdır.[14]

Geçmiş çalışmalar, tek pasolu operasyonlarla sınırlanmıştır (Ermer ve Wu 1967, Ermer ve Morris 1969, Iwata et al. 1977). Müteakip çalışmalar optimal işleme parametrelerini belirleyen çok aşamalı operasyonları keşfetmiştir (Iwata et al. 1972, Ermer ve Kromodihardo 1981, Alberti 1986, Shin ve Joo 1992, Tan ve Creese 1995). Klasik olarak, işleme ekonomisi problemi optimizasyon algoritmaları kullanılarak çözülmüştür, bu optimizasyonlar geometrik programlamayı (Ermer ve Kromodihardo 1981, Gopalkrishan ve Faiz 1991), dinamik programlamayı (Iwata et al. 1977, Lambert ve Walvekar 1978) ve lineer programlamayı (Gupta et al. 1995) içerir. Diğerleri dinamik programlamayı ve Fibonacci aramayı (Shin ve Joo 1992), onları takip edenleri ise sınırlanmamış minimizasyon tekniklerini (Hati ve Rao 1976) içerir ve arama stratejisi (Alberti 1986) üzerine odaklanır.

Evvelki çalışmaları daha önce bahsedilen bütün kesme sınırlamalarını ele almaz çünkü çoğu sınırlamalar mevcut metotları kullanan işleme optimizasyon problemini güçleştirir. İlave değişkenler ve paso sayısı çok aşamalı problemi NP-sert yapar. Evvelki optimizasyon teknikleri sadece spesifik bir problem için kullanışlı olma ihtimaline sahiptir ve yerel bir optimal çözüm elde etme eğilimini göstermektedir (Chen ve Tsai 1996). Dolayısıyla son zamanlarda tahmin algoritmaları çeşitli işleme problemleri tiplerini çözmek için uygulanmaktadır. Yerel ve yöresel arama teknikleri birleştirilebilen optimizasyon problemlerini çözmekte güçlü olan tahmin algoritmalarıdır. Çok aşamalı tornalama operasyonuna ait optimal koşulların belirlenmesi birleştirilebilen bir optimizasyon problemidir. Dolayısıyla yerel arama teknikleri son zamanlarda çok aşamalı işleme ekonomisi problemine uygulanmaktadır. Yerel arama teknikleri tavlama benzetimi (SA) algoritmasını (Kirkpatrick et al. 1983), genetik algoritma (GA) yaklaşımını (Goldberg 1989), yasaklı araması (TS) yaklaşımını (Glover 1989) ve karınca sistemi (AS) yaklaşımını (Dorigo et al. 1996) içermektedir. Chen ve Tsai (1996) işleme optimizasyon problemini çözmek için model araması (PS) tekniğini ve tavlama benzetimini birleştirmişlerdir. Model arama tekniğini tavlama benzetimi tekniğine bir giriş olarak bir çekirdek çözüm üretmek için kullanmışlardır. Böylece tavlama benzetimi tekniği daha iyi çözümler bulur ve global optimal çözümlere doğru ilerler. Formüle edilmiş kesme modeli, Shin ve Joo (1992) tarafından ele alındığı üzere çok aşamalı kaba tornalama ve bitiş tornalama olmak üzere çok aşamalı bir tornalama operasyonudur. İşleme parametrelerinin ikisinin de aslında birbirine bağımlı olmasından dolayı,

Chen ve Tsai (1996) kaba ve son bitiş operasyonlarındaki işleme parametrelerini iki ayrı gruba bölmektense beraber ele alarak analiz etmişlerdir. Alberti ve Perrone (1999) klasik çok aşamalı tornalama operasyonlarına ait bir fuzzy olanaksı formülasyonunu ele almışlardır ve sonuç veren olanaksı modeli genetik algoritma kullanarak optimize etmişlerdir. Problemi sınırlı sayıda kesme sınırlamaları ile ele almışlardır.[2]

III.3. ÇOK AŞAMALI TORNALAMA OPERASYONLARI KESME MODELİ

Oluşturulan optimizasyon probleminde, üç karar değişkeni ele alınır: kesme hızı v , devir f , ve talaş derinliği a . Bunlar prosesin gerçek kesme parametreleridir.[7]

III.3.1 Matematiksel Modelleme

Birim üretim fonksiyonu, UC, malzeme maliyeti hariç, çok aşamalı tornalama operasyonu problemi dört temel maliyet ögesinden oluşmaktadır.

- Kesme operasyonundaki asıl zamana ait kesme maliyeti, C_m .
- İndirme ve bindirme operasyonları ve boş takım hareketi sebebiyle makine boşta kalma maliyeti, C_1 .
- Takım değişiminin getirdiği maliyet, C_R .
- Takım maliyeti, C_T .

Chen ve Tsai (1996), Shin ve Joo (1992) tarafından tartışılan dört temel maliyet ögesini ele almışlardır. Sunum çalışması üç ana maliyeti ele almaktadır çünkü çok aşamalı tornalama operasyonları probleminin modellenmesinde yeterli görünürler. Bu ögeleri şimdi özetleyeceğiz.

III.3.1.1 Kesme Maliyeti

Kesme prosesi çok aşamalı kaba ve bitiş olmak üzere ikiye ayrılmaktadır. Kesme maliyeti, C_M , şöyle ifade edilmektedir.

$$C_M = k_o \left(\frac{\pi DL}{1000V_r f_r} \left(\frac{d_t - d_s}{d_r} \right) + \frac{\pi DL}{1000V_s f_s} \right)$$

Denklem III.1

III.3.1.2 Makine Boşta Kalma Zamanı

Makine boşta kalma zamanı, indirme ve bindirme operasyonları ve takımın boşta hareketi değişken bir terim sebebiyle sabit bir terime bölünmüştür ve şöyle ifade edilmektedir:

$$C_I = k_o \left[t_c + (h_1 L + h_2) \left(\frac{d_t - d_s}{d_r} \right) + 1 \right]$$

Denklem III.2

III.2.1.3 Takım Değişirme Maliyeti

Kullanılan Taylor takım ömrü denklemi şöyle verilmiştir:

$$T = \frac{C_o}{V^p f^{qr}}$$

Denklem III.3

Farklı işleme koşullarına nedeniyle, kaba ve bitiş tornalama arasındaki yıpranma oranı genellikle değişmektedir. Böyle bir durumda takım ömrü şöyle ifade edilebilir:

$$T_p = \theta T_r + (1 - \theta) T_s$$

Denklem III.4

Burada:

$$T_r = \frac{C_o}{V_r^p f_r^q d_r^r}, T_s = \frac{C_o}{V_s^p f_s^q d_s^r}$$

Denklem III.5

Takım değişim zamanı, takım ömrü (T_p), bir takımı değiştirmek için gereken zaman (t_c) ve kesme zamanı (Shin ve Joo 1992) bakımından yazılabilir. Şu şekilde verilmiştir: [23]

$$C_R = k_o \frac{t_e}{T_p} \left[\frac{\pi DL}{1000 V_r f_r} \left(\frac{d_t - d_s}{d_r} \right) + \frac{\pi DL}{1000 V_s f_s} \right]$$

Denklem III.6

III.3.1.4 Takım Maliyeti

Takım ömrü, Taylor'un genişletilmiş yasası olan kesme parametreleriyle ilişkilidir;

$$T = C_T v^a f^b a^{\gamma}$$

Takım Maliyeti, C_T şöyle ifade edilebilir:[26]

$$C_T = \frac{k_t}{T_p} \left[\frac{\pi DL}{1000V_r f_r} \left(\frac{d_t - d_s}{dr} \right) + \frac{\pi DL}{1000V_s f_s} \right]$$

Denklem III.7

III.3.1.5 Birim Üretim Maliyeti

III.1, III.2, III.6 ve III.7 denklemlerini kullanarak, birim üretim maliyeti, UC, şöyle tanımlanır:

$$UC = C_M + C_I + C_R + C_T$$

Denklem III.8

D	L	D_t	T_L	T_u	F_u	P_u	S_r	Q_u	S_c	R_a	q
50	300	6	25	45	5	200	10	1000	140	1.2	0.5
k₁	k₂	k₃	k₄	k₅	p	q	r	m	v	h	
108	132	1	2.5	1	5	1.75	0.75	0.75	0.95	0.85	
l	n	t	f	δ	ko	k_t	h₁	h₂	t_c	t_e	C_o
2	-1	0.4	0.2	0.105	0.5	2.5	7E-04	0.3	0.75	1.5	6E+11
V_r	F_r	D_r	D_s	F_s	V_s						
205.24	0.2626	2.763	1.181	0.451	202.3						

Maliyet

C_m 1.0207

C_i 1.0747

C_r 0.2145

C_t 0.715

UC 3.025 YTL

Tr 7.98

Ts 6.295

Tp 7.137

Kısıt Değişkenleri

F_r 104

P_r 4.105

Q_r 945.6

F_s 69.6

P_s 2.707

Q_s 958.1

Yukarıdaki excel programında, sabit değerler girildiğinde, mevcut formüller doğrultusunda UC Birim Üretim Maliyeti hesaplanmaktadır. Maliyet hesabının yapılmasıyla birlikte kısıt değişkenleri ve sınır değerleri de hesaplanır ve istenilen sınır değerleriyle karşılaştırma imkânı bulunur.

III.3.2 Parametre Sınırları ve Kesme Koşulları Sınırlamaları

Optimal kesme koşullarının seçimini etkileyen ve hesaba katılması gereken bazı sınırlar vardır. İlk olarak, takım üreticisi tarafından verilen ve alt ve üst müsaade edilebilir değerler olarak ifade edilen kesme parametrelerini ele alırsak;

$$V_{\min} \leq V \leq V_{\max}$$

$$f_{\min} \leq f \leq f_{\max}$$

$$a_{\min} \leq a \leq a_{\max}$$

Ayrıca makine özelliklerine ilişkin bazı sınırlar da vardır. Kesme kuvveti F_c , Makinenin ve kesici takımın güç ve dengesi tarafından verilen belli bir maksimum değerden daha fazla olmamalıdır, $F_{c-\max}$. Kesme kuvveti deneysel denklemlerden şu şekilde hesaplanır;[23]

$$F_c = C_F v^{\alpha'} f^{\beta'} a^{\gamma'}$$

Burada C_F , α' , β' , γ' deneysel olarak elde edilen katsayılardır. Bir diğer makineye ilişkin sınır ise kesme kuvveti için maksimum müsaade edilebilir P değeridir, makine motor gücünü geçmemesi gereken P_{mot} (iletimdeki sürtünme kayıpları ele alınarak) [7]:

$$P = \frac{v F_c}{6 \times 10^4} \leq \frac{P_{\text{MOT}} \eta}{100}$$

Yüzey bitiş pasosu işleme kalitesini direkt olarak etkiler ve bu yüzden bitiş operasyonlarındaki en önemli faktörlerden biridir. Bitiş yüzeyi ilerleme, hız, talaş derinliği, takım geometrisi ve kesici takım materyali gibi çeşitli parametrelerden etkilenmektedir. Bunun yanı sıra takım değiştirme, çapak ve pah kırma, bozuk yüzey gibi istenmeyen durumlar da meydana gelebilmektedir ama burada sadece ilerleme ve burun yarıçapı ele alınacaktır, çünkü bitiş yüzeyinde daha dominant bir etkisi vardır. Bu yüzden yüzey bitiş sınırlaması şöyle olacaktır:

$$\frac{f_s^2}{8r} \leq R_{\max}$$

burada R_{\max} maksimum müsaade edilebilir yüzey pürüzlülüğünü ve r , kesici takımın burun yarıçapını ifade etmektedir.

Ele alınan çok sayıda parametre sınırı şunlardır: hem kaba hem de bitiş koşulları için kesme hızı, ilerleme miktarı ve talaş derinliğidir. İşleme ekonomisindeki birim üretim maliyeti optimizasyonunda ele alınan çok sayıda pratik kesme sınırlamaları şunları kapsamaktadır: takım ömrü sınırlaması, kesme kuvveti sınırlaması, güç, sabit kesme alanı sınırlaması, talaş-takım ara yüzeyi sıcaklığı sınırlaması, yüzey bitiş sınırlaması, kaba ve bitiş parametre ilişkileri ve paso sayısı. Belli işleme parametreleri (X) altında, paso sayısı (n) ve talaş derinliği (d_r), n_i 'ye ayarlanmıştır ve asgari birim üretim maliyeti $[UC(X, n_i, d_r)]$ olur, böylece denklem III.8'deki ifade edilen objektif fonksiyon şöyle olur:[23]

$$F(x) = \min[UC(X, n_i, d_r)], i = N_L, \dots, N_U$$

Denklem III.9

III.3.3 Son Kesme Modeli

Çok aşamalı tornalama operasyonlarına ait optimizasyon modeli evvelki konu kullanılarak formüle edilebilir.

$$Y_{min} = f(X)$$

Denklem III.10

kaba işleme ile birlikte:

$$V_{rL} \leq V_r \leq V_{rU} \quad \text{Denklem III.11}$$

$$f_{rL} \leq f_r \leq f_{rU} \quad \text{Denklem III.12}$$

$$d_{rL} \leq d_r \leq d_{rU} \quad \text{Denklem III.13}$$

$$T_L \leq T_r \leq T_U \quad \text{Denklem III.14}$$

$$F_r \leq F_U \quad \text{Denklem III.15}$$

$$P_r \leq P_U \quad \text{Denklem III.16}$$

$$Q_r \leq Q_u \quad \text{Denklem III.17}$$

$$V_r^\lambda f_r d_r^v \geq SC \quad \text{Denklem III.18}$$

bitiş işleme ile birlikte:

$$V_{sL} \leq V_s \leq V_{sU} \quad \text{Denklem III.19}$$

$$f_{sL} \leq f_s \leq f_{sU} \quad \text{Denklem III.20}$$

$$d_{sL} \leq d_s \leq d_{sU} \quad \text{Denklem III.21}$$

$$T_L \leq T_s \leq T_U \quad \text{Denklem III.22}$$

$$F_s \leq F_U \quad \text{Denklem III.23}$$

$$P_s \leq P_U \quad \text{Denklem III.24}$$

$$Q_s \leq Q_u \quad \text{Denklem III.25}$$

$$V_s^\lambda f_s d_s^v \geq SC \quad \text{Denklem III.26}$$

$$\frac{f_s^2}{8R} \leq SR_U \quad \text{Denklem III.27}$$

ilişkiler:

$$V_s \geq k_3 V_r \quad \text{Denklem III.28}$$

$$f_r \geq k_4 f_s \quad \text{Denklem III.29}$$

$$d_r \geq k_5 d_s \quad \text{Denklem III.30}$$

burada

$$F_r = k_1 f_r^\mu d_r^v, \quad P_r = \frac{F_r V_r}{6120\eta} = \frac{k_1 f_r^\mu d_r^v V_r}{6120\eta}$$

ve

$$Q_r = k_2 V_r^t f_r^\Phi d_r^\delta \leq Q_u$$

olur.

Benzer tanımlar sırasıyla F_s , P_s ve Q_s 'ye uygulanır.

Denklem III.15'e ait kesme kuvveti sınırlaması ve denklem III.16'ya ait güç sınırlaması Shin ve Joo (1992) formülasyonunu takip eder. Denklem III.17'ye ait talaş-takım ara yüzeyi sınırlaması Hati ve Rao (1976) formülasyonunu takip eder. Denklem III.18'e ait sabit kesme alan sınırlaması ve denklem III.27'ye ait bitiş yüzey sınırlaması Narang ve Fischer (1993) formülasyonunu takip eder.

Yukarıda formüle edilen kesme modeli işleme parametrelerinden bahsettiği gibi devamlı değişken katsayıya sahip doğrusal olmayan sınırlanmış programlama (NCP) problemidir. Kaba ve bitişteki işleme parametreleri esasında bağımlıdır, bu nedenle aynı anda analiz edilirler. Son zamanlarda kombine optimizasyon problemlerini çözmek için meydana çıkmış yerel arama tekniği haricinde burada gözden geçirilen tüm bu teknikler çoğu sınırlamalardan meydana gelen komplekslik dolayısıyla kesme modelini çözmek için pek uygun değildir. Kompleks problemleri çözme kapasitesine sahip anlatılan genetik algoritma optimizasyon tekniği aşağıda tanımlanmaktadır.[2]

III.4 OPTİMİZASYON ALGORİTMASI

Ele alınan GA, mikro-GA temellidir. Yeterli popülasyon çeşitliliğini sürdürecektir bir tekniği olan en zarif algoritmadır. Sınırları ele almak için, aşağıdaki prensiplere dayalı bazı işlemler eklenmiştir:

- Belli bir popülasyonda, uygulanabilir çözümler uygulanamaz çözümlere tercih edilir, yani uygulanabilir çözümler uygulanamazlardan daha iyi dereceye sahiptir.
- Uygulanamazlık miktarı (yada sınır ihlali boyutu), bilginin önemli bir parçasıdır ve sınırlarla uğraşılırken ihmal edilmemesi gerekir.
- İhlal edilen sınırların sayısı da bilginin önemli bir parçasıdır ve onlarla uğraşılırken dikkate alınması gerekir.[7]

Geniş ölçekli geleneksel birleşik optimizasyon problemlerini çözmek için iki metot vardır: optimizasyon algoritmaları ve tahmini algoritmalar. Optimizasyon algoritmaları, muhtemel aşırı bir bilgisayar zamanı sürecindeki optimal çözümü global bir şekilde elde ederek, birleşik optimizasyon problemini çözme kapasitesine sahiptir. Tahmini algoritmalar iki kategoriye ayrılabilir: yapısal ve yöresel yada yerel arama. Güncel olarak geniş ölçekli geleneksel birleşik optimizasyon problemlerini çözmekte kullanılan yerel arama teknikleri tavlama benzetimini (SA), genetik algoritmaları (GA), tabu arama (TS) yaklaşımını ve karınca sistemini (AS) kapsamaktadır.

GA tabanlı çok aşamalı tornalama operasyonlarına ait gelişim için kullanılan terminoloji ve tasarım konuları devam eden bölümlerde ele alınmaktadır.

Bu tezde rapor edilen iş için görevlendirilen genetik algorithmada, bir aday çözüm her bir ürün tipine ait çok aşamalı tornalama operasyonlarını gösterecek rakam dizileri tarafından temsil edilirler. T zamanındaki bir dizi grubu çözüm boşluğu $S(t)$ olarak bilinir. Belli bir zamandaki çözüm boşluğu durum olarak bilinir. Bir birey dizisinin potansiyeli bir skor (objektif) fonksiyon tarafından belirlenir. Skor (objektif) fonksiyon her bir dizinin nispi değeri için özel ilgi alanı verir. Dizileri müteakip dizileri için orta durumu üretmekte kullanılacak dizileri belirlemede kullanılan gerçek sayılara dönüştürür. Genetik algoritma prosedürü, belli bir ilgi alanı problemini en uygun çözümü bulacak şekilde çözüm boşluğunu kullanır. Her bir çözüm boşluğu için, en iyi çözüm bulunur ve seçim, çaprazlama ve mutasyon gibi bazı genetik algoritma operatörlerini uygulayarak gelişim eğilimli çözümlere götürecek yeni çözüm boşlukları bulunur. Her bir peş peşe çözüm boşluğuna ait durumları ile bireysel çözümlerin kalitesindeki gelişmeler kazanılma eğilimindedir. Algoritmaların temel görüşü şöyle temsil edilebilir:[2]

Prosedür GA

başla $t=0$;

çözüm boşluğunu hazırla $S(0)$;

değerlendir $S(0)$;

tekrarla

$t = t + 1$;

$S(t)$ 'yi $S(t-1)$ 'den seç;

$S(t)$ 'yi çeşitlendir;

$S(t)$ 'yi yoğunlaştır;

$S(t)$ 'yi değerlendir;

buraya kadar (durma koşulları karşılanmıştır).

son.

III.4.1 Temsil

İşleme bilgilerini şifre ile yazarak aşağıdaki gibi dizi-bit bloğu yapılı: kaba işleme ve bitiş işleme parametreleri altı çözüm dizi-bit bloğundaki kodlanmış değerleri belirten değişkenlerdir. Hem kaba hem de bitiş işleme koşulları için kesme

hızı tam sayılardan oluşurken, ilerleme miktarı ve talaş derinliği kaba işleme için de bitiş işleme için de reel sayılardan oluşmaktadır. Hem kaba hem de bitiş işleme koşulları kesme hızları (V_r , V_f), devir oranları (f_r , f_f) ve talaş derinliği (d_r , d_f) tornalama operasyonlarında birincil parametreleri temsil etmektedir. Bu birincil değişkenlerin her biri ikili bir diziyeye dönüştürülür ve her bir çözüm dizisi için toplam 132 bit verilerek, 22 bitli bir bloğa tahsis edilir.

1-22	23-44	45-66	67-88	89-110	111-132
V_r	f_r	d_r	d_s	f_s	V_s

Tamsayıları ve reel sayıları ikili dizilere çevirmedeki detaylar sıradaki altbölümlerde ele alınacaktır. İkili düzen bilgisi genetik operatörler tarafından yürütülür ve tekrar tamsayı ve reel sayılara dönüştürülür.

III.4.1.1 İkili Sistem Planlaması

İkili bir dizi x değişkeninin gerçek değerlerini temsil etmek için çözüm dizisi olarak kullanılır. Dizi genişliği gerekli hassasiyete bağlıdır, tornalama operasyonlarında ondalık sayısı olarak 6 hane kullandık. x değişkeninin genişlik=4 ilgi alanı vardır, böylece hassasiyet gereksinimi $[-2,-1,1,2]$ alanının en az 4×10^6 eşit boyut alanı bölünmesi gerektiğine işaret eder. Simetrik alan kaba ve bitiş işleme koşullarını sağlamak için seçilmiştir. Bunun anlamı 22 bitin ikili bir dizi (çözüm dizisi) olarak istenmesi olduğudur:

$$2\,097\,152 = 2^{21} \leq 4\,000\,000 \leq 2^{22} = 4\,194\,304$$

İkili bir diziyeye $\{b_{21}, b_{20}, \dots, b_0\}$ ayrılan alandaki bir x reel sayısı ile yapılan planlama iki adımda tamamlanır.

Adım 1. Aynı x reel sayısını bul:

$$x = -2.0 + x \times \frac{4}{(2^{22} - 1)}$$

burada 2.0 ilgi alanının sol sınırındadır ve 4 ilgi alanının genişliğidir.

Adım 2. İkili diziyi 2 tabanından 10 tabanına aşağıdaki gibi çevir:

$$(\{b_{21}, b_{20}, \dots, b_0\})_2 = \left(\sum_{i=0}^{21} b_i \times 2^i \right)_{10} = x'$$

Örneğin, 0.729 mm/dev'lik ilerleme miktarı için bir çözüm dizisi bloğu, bu değeri Adım 1'e x değeri olarak yerleştirip x' değerini çözerek bulunur. Sonra bu x' Adım 2'deki bir ikili diziyi aşağıdaki gibi dönüştürülür:

$$0,729 = -2 + x' \times \frac{4}{(2^{22} - 1)}$$

$$x' = (0.729 + 2) \times (2^{22} - 1) / 4 = 2861560.5 = (2861561)_{10}$$

$$\text{İkili düzende, } x' = (1010111010100111111001)_2$$

Operasyonel olarak, rasgele meydana gelmiş altı işleme parametresi çözüm boşluğunun her dizisi için reel yada tamsayılar gibi 10 temelindedir. Dâhili olarak, bilgi ikili rakamlara dönüştürülür ve genetik operatörlerle üzerinde işlem yapılır. Bunlar geçici çözüm boşluklarında depolanır ve yeniden ikili planlama tekniği kullanılarak tamsayılar veya reel sayılara dönüştürülür.

III.4.2 Başlangıç

Başlangıç esnasında, *populasyon boyutu* çözüm dizilerine ait bir çözüm boşluğu ilerleme miktarı ve talaş derinliği için rasgele olarak meydana gelir. Kesme oranları sırasıyla kaba ve bitiş koşulları için $[f_{rL}, f_{rU}]$ ve $[f_{fL}, f_{fU}]$ limitleri arasında rasgele meydana gelen reel sayılardır. Talaş derinliği değerleri, sırasıyla kaba ve bitiş koşulları için $[d_{rL}, d_{rU}]$ ve $[d_{fL}, d_{fU}]$ limitleri arasında rasgele meydana gelen reel sayılardır. Takım ömrü değerleri $[T_L, T_U]$ limitleri arasında rasgele meydana gelen reel sayılardır. Takım ömrü denklemini kullanarak, sırasıyla kaba ve bitiş koşulları için kesme hızları sırasıyla kaba ve bitiş koşulları için kabul edilmiş $[V_{rL}, V_{rU}]$ ve $[V_{fL}, V_{fU}]$ limitleri arasında yatması olayı ile ilgili olarak türetilmiştir. Her bir çözüm dizisi, kesme parametrelerinin kodlanmış konfigürasyonuna dayalı olarak çok aşamalı tornalama operasyonu performansını benzeterek elde edilen bir maliyet fonksiyonu değeri ile ilgilidir. Bu başlangıç prosesi rasgele meydana gelmiş çözüm boşluğu ile yerine getirilir. Arzu edilen boyuta ait (populasyon boyutu) bir başlangıç

çözüm boşluğu $S(0)$, çözüm boşluğu boyutu popülasyon boyutu=20 ile rasgele meydana gelmiştir.

III.4.3 Evrim

Genetik algoritma çözüm boşluğuna ait her bir dizi için skor (objektif) fonksiyonu hesaplar ki dizi maksimum skor fonksiyon değeri olan dizi belirlensin. Optimizasyon problemlerinin amacı bazı maliyet fonksiyonlarını azaltmaktır. Genetik algoritma yaklaşımında, optimize edilen maliyet fonksiyonu genellikle bir skor fonksiyonunu planlar. i 'nci çözümün rekabet edilebilirliği $f^i(t)$, şöyle elde edilir:

$$f^i(t) = f_{\max}^i - g^i(t)$$

Denklem III.31

burada $g(t)$ bir diziye ait objektif fonksiyondur ve f_{\max} mevcut çözüm boşluğundaki en düşük objektif fonksiyon değeridir. Aynı seçim olasılığı $P(i)$ şuna eşittir:

$$P(i) = \frac{f^i(t)}{\sum_{k=1}^{popsize} f^k(t)}$$

Denklem III.32

En rekabete edilebilir çözüm dizileri, müteakip durumlara ilerlemek için daha yüksek bir örnekleme ihtimali sebebiyle etkilenmektedir.

III.4.4 Seçim ve yeniden kopyalama

Goldberg'te (1989) 6 alternatif seçim şeması vardır, belirleyici örnekleme, değiştirmesiz kalan stokastik örnekleme, değiştirmeli kalan stokastik örnekleme, değiştirmesiz stokastik örnekleme, değiştirmeli stokastik örnekleme ve stokastik turnuva. Değiştirmesiz kalan stokastik örnekleme diğer 5 stratejiden daha üstün niteliklidir ve burada sadece bu çalışılacaktır. Bu stratejide, beklenen sayım e_i her zamanki gibi hesaplanır;

$$\left(e_i = \frac{f(t)}{\sum f^i(r) / popsize} \right)$$

e_i 'nin kesirli bölümü olasılık olarak ele alınır. Kesirli bölümü başarı olasılıklarımış gibi kullanarak yazı tura atımı gerçekleştirilir. Diziler e_i 'nin bütün bölümüne eşit kopyalar alır.

III.4.5 Genetik Operatörler: Çaprazlama ve Mutasyon

Çaprazlama genetik algoritmayı daha önce görülenlerden farklı çeşitli akılda kalır yollardan farklı çözümler meydana getirmesini ve uğranmamış alanları sınamasını teşvik eden bir *çeşitlendirme* mekanizmasıdır. Çaprazlanacak diziler ve çaprazlayan noktalar rasgele seçilir. *İki noktalı* çaprazlama tekniği bu çalışmada benimsenmiştir. Dizi uzunluğu için, l ve x_p -noktası operatörü, $\begin{pmatrix} 1 \\ X_p \end{pmatrix}_{x_p}$ çaprazlama noktasını seçmenin farklı yolları vardır. Bu nedenle bir *iki noktalı* çaprazlama bir *tek noktalı* çaprazlamaya tercih edilmektedir. Bununla birlikte, x_p arttıkça, her bir operatör çaprazlama sırasında daha aza seçilme şansına sahiptir ve daha az yapı korunabilir. Çaprazlama bir çaprazlama olasılığı (pcross) ile yapılır. Çaprazlama mekanizması çözüm dizisinin 6 bloğunun her biri içindeki ikili rakamlarla çalışır. Çaprazlama kavramı muhtemel çözümü temsile eden birinci ikili bloğun (kesme hızı için) on altıncı ve on dokuzuncu dizilerini kullanarak örneklenmektedir. Çaprazlama operatörü aşağıdaki gibi iki yeni dizi oluşturur (burada $\hat{\quad}$ çaprazlama noktasını temsil etmektedir)

Dizi 1: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 $\hat{\quad}$ 0 0 1 $\hat{\quad}$ 1 1 0

Dizi 2: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 $\hat{\quad}$ 1 0 1 $\hat{\quad}$ 1 0 0

Yeni Dizi 1: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 0

Yeni Dizi 2: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0

Görülebildiği üzere, kesme hızı değeri sırasıyla 78 ve 108 m/dak olan ilk bloğa ait iki dizi, çaprazlama operasyonundan sonra, 110 ve 76 m/dak olmuşlardır.

Mutasyon rasgele seçilmiş bir dizinin rasgele bir değişimidir. Bırakılacak minimum yerel alan bir alanı müsaade edecek herhangi bir başlangıç çözüm boşluğu için çözüm boşluğunu inceleme olasılığını garanti eder. Mutasyon, mutasyon ihtimali (pmutate) ile yapılır. İki rasgele tamsayı r_1 , ve r_2 sırasıyla 1 ve 2 dizilerinden seçilir, öyle ki $1 \leq r_1, r_2 \leq n$ (blok-boyutu) ve $r_1 \neq r_2$ elde edilir. Genetik algoritma prosedürü sonra r_1 ve r_2 pozisyonları tarafından belirtilmiş dizi bitlerini tersine çevirir (0'dan 1'e, yada 1'den 0'a). Örneğin, eğer $r_1 = 17$ ve $r_2 = 18$ ise, sonra önceki yeni 1 ve 2 dizileri (mutasyona uğratılmış pozisyonların altı çizilmiştir) şöyle olur:

Yeni Dizi 1: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0

Yeni Dizi 2: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0

Bu çaprazlama ve mutasyon operatörlerini uygulayarak, 1 ve 2 dizilerinden farklı yeni diziler meydana getirmek mümkün olmuştur. Genetik algoritmanın gücü bu karakteristikleri bünyesinde bulundurmasından gelir. Mutasyon operasyonu aslında bir *yoğunlaşma* mekanizmasıdır. Yoğunlaşma esnasında, arama olası çözümlerin komşularını sınıma üzerinde odaklanmıştır. Başarılı çaprazlama ve mutasyonlar yöresel duruma ön ayak olurlar. Bu örnekte, yeni 1 ve 2 dizileri sırasıyla 1 ve 2 dizilerinin komşularıdır. Adapte edilen çaprazlama oranları her bir operasyon için 0.3'tür. Adapte edilen mutasyon oranları her bir operasyon için 0.001'dir.

III.4.6 Değişirme Stratejisi

Her çaprazlama ve mutasyon operasyonunda, yeni diziler yaratılır. Bunun anlamı, uygulayıcı diziler bir değişirme stratejisi kullanarak yenilenmektedir. Geçmişte tavsiye edilen birçok strateji Liepins ve Hillard'ın (1989) çalışmalarını da kapsamaktadır. Bunlar aşağıdakiler içermektedir.

- Olanaksı yer değişirme: Bir önceki durumdaki başarısız uygulayıcı dizileri olanak dahilinde kapsayan metodu kullanır.
- Sıkıştırma stratejisi: Bir önceki durumdaki en benzerleri olanaksı olarak yeniler.
- En seçin strateji: Bir önceki durumun en iyi uygulayıcı dizilerini şu an ki duruma ilişitir. Bu daha iyi performansı olan dizilerin bir sonraki durumda yaşayacaklarından emin olmaya çalışmak için tasarlanmıştır.

Bu çalışmada, bu metotların kombinasyonu formüle edilmiş ve aşağıda taslağı çizilmiştir.

- Her çaprazlama ve mutasyondan sonra, yeni meydana getirilmiş bir dizi havuzu bir seçim havuzu biçimlendirmektedir. Seçim havuzu boyutu, havuz boyutu = çapraz çözüm boşluğu boyutu olasılığı olarak hesaplanır. Bu yüzden, havuz boyutu çözüm boşluğu boyutundan daha düşüktür.
- Diziler objektif fonksiyon kullanılarak değerlendirilir. Sonra objektif fonksiyon bir performans ölçümü olarak kullanılır.

- Yeni ve eski diziler arasında yakın bir kıyas yapılır ve daha iyi performans gösterenler bir sonraki duruma ilerler.
- Havuz boyutu çözüm boşluğu boyutundan daha az olduğuna göre, çözüm boşluğunu dolduracak geriye kalan diziler beta denilen belli bir önceden kararlaştırılmış ihtimalini kullanarak rasgele seçilir.

III.4.7 En iyi çözümün kaybını önlemek

Eski çözüm boşluğundaki en iyi dizilerin korunması çok önemlidir. Çeşitlendirilmiş çözüm boşluğu objektif fonksiyon kullanılarak değerlendirilir ve en iyi uygulayıcı diziler önceki çeşitlendirilmemiş çözüm boşlukları ile kıyaslanır. Çeşitlendirilmiş ve çeşitlendirilmemişler arasındaki en iyi bir sonraki duruma ilerletilir. Bu en iyi çözümün kaybını önler. Bu problemi çözmek için, en iyi çözümün kaybını önleyecek bir prosedür optimizasyon algoritmasında bulunmaktadır. “Isbest” stratejisi denilen bir mekanizma Genetik Algoritma prosedürünün yürütümünde bulunmaktadır, bu mevcut durumdaki en iyi çözümün sonraki durumlara “itildiğini”, böylece “daima daha iyi çözümümüzün olacağını” temin eder. Sonraki aşamalardaki muhtemel ikinci derece koşulu konusu genetik algoritmanın yürütümündeki en büyük zorluklardan biridir. Bu yüzden, “isbest stratejisi”nin dahil edilmesi, bu tezdeki genetik algoritmanın yürütümündeki bizim ana katkılarımızdan biri olacaktır.

“Isbest stratejisi” için matematiksel temeller burada verilmiştir. Mevcut çözüm boşluğunun S_{ij}^t aynı objektif fonksiyon değerleri ile olduğunu belirtildiği gibi görelim:

$$Z_i' = f(S_{ij}')$$

Denklem III.33

Burada tornalama operasyonları için, i çözümleri ve j başlangıç parametreleri vardır. Çözüm boşluğunda N dizilerinin (çözümlerinin) olduğu varsayılır. Şimdi tanımlayalım;

$$S_a = \{1,2,\dots,N\}$$

Denklem III.34

Mademki çoğu dizi için objektif fonksiyonların aynı değerine sahip olmak mümkün, o zaman çözüm boşluğundaki bu tür dizilerin toplam sayısı şöyle tanımlanır:

$$N = \sum_{i \in S_a} \arg \min \{Z_i^t\}$$

Denklem III.35

Şimdi tanımlayalım;

$$S_b = \{1, 2, \dots, n\}$$

Denklem III.36

$$S_c = \{1, 2, \dots, N - n\}$$

Denklem III.37

Burada, S_b en iyi çözüme sahip bir grup dizidir. Bazı durumlarda çoğu diziler en iyi çözümün tamamıyla aynı değerine sahip olabilirler. Bu iki grup en iyi çözüm çalışmasını nasıl koruduklarına ait stratejiyi tanımlamakta gereklidir. En iyi çözümü olan diziler ($S_i^{f_{ij}} \in S_b$) herhangi bir genetik operatörün bir sonraki duruma ilerlemesi için uygulanmasından önceki mevcut tekrarlamada korunur (geçici olarak R_{ij}^t 'de saklanır). Bu şöyle gerçekleşir:

$$R_{ij}^t \leftarrow S_{i \in S_b}^{f_{ij}}$$

Denklem III.38

Mevcut çözüm S_{ij}^t daha sonra mutasyona uğrattılır ve yeni bir çözüm boşluğu $S_{ij}^{\sim t+1}$ elde etmek için bir sonraki duruma ilerletilir. Yeni çözüm boşluğundaki başarısız çözümler şu operasyon ile elenirler:

$$V_{ij}^{t+1} \leftarrow S_{i \in S_c}^{\sim t+1}$$

Denklem III.39

Bu durumda, mutasyona uğratılmış (N-n) en iyi çözüme sahip diziler geçici olarak V_{ij}^{t+1} 'de saklanırlar. Son n çözümü düşürülür. Sonra, bu çözümler en iyi çözüme sahip olan bir önceki tekrarlamalardan diziler ile birleştirilir. Çözüm boşluğunun boyutu sonra şöyle sürdürülür:

$$S_{ij}^{\sim t+1} \leftarrow V_{ij}^{t+1} \cup V_{ij}^t$$

Denklem III.40

$$S_{ij}^t \leftarrow S_{ij}^{t+1} \text{ eğer ilerlemenin sonu değilse}$$

Denklem III.41

Çok aşamalı tornalama operasyonları problemi için genetik algoritma tabanlı yaklaşım simgeleme III bölümünden alınarak aşağıdaki gibi sunulmaktadır:

- Adım 1. Bir başlangıç çözüm boşluğu *oldpop* al {Başlangıç}
- Adım 2. Son generasyonda değilken yap
 - Yer değiştirmesiz stokastik örnekleme kullanarak diziler seç;
 - Diziler objektif fonksiyon, fonksiyon kalitesi ve beklenen sayım hesaplaması ile değerlendir;
 - Beklenen sayımın tamsayı kısmını kullanarak geçici bir havuz (*temppop*) yarat ve kesirli kısımları başarı olasılıkları olarak kullan;
- Adım 3 Çok çaprazlı operatörleri *temppop*'a yeni bir seçim havuz boyutu (*poolsize* = *popsiz* * *pcross*) yaratmak için uygula {Çaprazlama/Yeniden Kopyalama}
 - Yer değiştirmesiz kalan seçim ile çaprazlama için rasgele adaylar seç;
 - Çok çaprazlı operatörleri seçilmiş iki diziye bir Olasılık ile uygula;
 - Sınırlamayı kale almaksızın mutasyonu yeni dizilere uygula;
 - Sınırlamaları test et; eğer bozulduysa, mutasyonu sınırlamalar kafi gelene kadar uygula;
 - Objektif fonksiyon değerlerini hesaplayarak değerlendir;
 - Seçim havuzunu objektif fonksiyon değerlerinin artan sırasına göre sınıfla;
- Adım 4 Aynı eski çözüm boşluğu dizileri ile sınıflanmış seçim havuzu dizilerinin her birini kıyasla; {Yer değiştirme stratejisi}
 - Seçim havuzu ve eski çözüm boşluğundaki aynı dizileri kıyasla ve döngüdeki en iyiyi al;
 - Her bir döngüde diğerinden daha iyi sonuç vereni al;
 - Diğer diziler için, rasgele bir seçim 0.3 ihtimali ile yapılmaktadır;
- Adım 5 Mevcut çözüm boşluğu eski çözüm boşluğu olur;
 - Süreci sonlandır;
- Adım 6 Mevcut çözüm boşluğundaki en küçük objektif fonksiyon değerine sahip çözüme geri dön.

III.5 CEZA FONKSİYONU

Evrimsel hesaplamada kullanılan ceza fonksiyonları, sabit, statik, dinamik ve uyarlı olarak tanımlanan ana tiplerden oluşmaktadır. Ceza fonksiyonları yıllardır sınırlı optimizasyondaki literatürün bir parçası olmuştur. Ceza fonksiyonları mümkün olmayan çözümleri cezalandıran harici ceza fonksiyonu ve mümkün olan çözümleri cezalandıran dahili ceza fonksiyonları olmak üzere iki ana tipten oluşmaktadır. Yukarıda harici fonksiyonların hangi tiplerden oluştuğunu söyledik. Dahili ceza fonksiyonlarının alanı ise evrimsel hesaplamadaki potansiyel araştırmalardan ibarettir. Dahili ceza fonksiyonlarının ana amacı şudur; bir optimal çözüm bir sınırın aktif olmasını gerektirir ki, optimal çözümler uygulanabilir ve uygulanamaz arasındaki sınıra dayansın. Bunun ışığında, “dahili çözümler” denen sınır aktif değilken, bir ceza mümkün çözümlere uygulanır. Tek bir sınır için, bu yaklaşım apaçıktır (evrimsel hesaplama literatüründe görünmemesine rağmen), bununla birlikte devamlı sınırların daha yaygın halleri için, dahili ceza fonksiyonlarının yürütülmesi dikkate degecek kadar daha karmaşıktır.

Harici ceza fonksiyonlarının 3 derecesi vardır: (1) uygulanamaz hiçbir çözümün ele alınmadığı bariyer metotları, (2) uygulanabilirlik sınır yakınında bir ceza uygulanan kısmi ceza fonksiyonları, ve (3) uygulanamaz alan boyunca uygulanan global ceza fonksiyonları.

Genelde bir ceza fonksiyonu yaklaşımı aşağıdaki gibidir. Belli bir optimizasyon probleminde, aşağıdaki fonksiyon sınırların en genel formülasyonudur:

$$\begin{array}{ll} \min f(x) & \\ \text{s.t} & x \in A \\ & x \in B \end{array} \quad \text{Denklem III.42}$$

burada x karar değişkeni vektörüdür, “ $x \in A$ ” sınırlarını karşılaması bağıl olarak basittir, ve “ $x \in B$ ” sınırlarını karşılaması bağıl olarak zordur, problem şu şekilde yeniden formüle edilebilir:

$$\begin{array}{ll} \min f(x) + p(d(x, B)) & \\ \text{s.t} & x \in A \end{array} \quad \text{Denklem III.43}$$

burada $d(x, B)$ B alanından x vektörü çözüm mesafesini tanımlayan metrik bir fonksiyondur, ve $p(\cdot)$, B 'nin dışında yeteri kadar çabuk büyür, denklem III.42'nin optimal çözümü aynı zamanda denklem III.43 için de optimal olacaktır. Dahası,

denklem III.43'ün herhangi bir optimal çözümü ,(1) için optimumda bir üst sınır sağlayacaktır, ve bu sınır genelde $f(x)$ 'i A'ya basitçe optimize ederken elde edilenden daha sıkı olacaktır.

Pratikte, “ $x \in B$ ” sınırları eşitsizlik ve eşitlik sınırları olarak şu şekilde ifade edilir:

$$i = 1, \dots, q \text{ için } g_i(x) \leq 0$$

$$i = q + 1, \dots, m \text{ için } h_i(x) = 0$$

burada

q =eşitsizlik sınırları sayısı

$m - q$ = eşitlik sınırları sayısı

$p(\cdot)$ ve $d(\cdot)$ fonksiyonlarının çeşitli türleri, sınırları ikili hale getirecek evrimsel optimizasyonlar için çalışılmıştır. Farklı muhtemel mesafe metrikleri, $d(\cdot)$, bir çok sayıda bozulmuş sınırlamaları içerir. Bu yaklaşımlara ait çeşitlemeler farklı başarı dereceleriyle denenmiştir. Daha dikkate değer örneklerin bazıları sıradaki bölümde anlatılmaktadır.

Kayıp sınırlamalar için etkili ve etkin bir vekil olan bir ceza fonksiyonu bulmak zor olabilir. Ceza fonksiyonunu belli bir probleme bir defa veya tekrarlı bir şekilde uygulamak için gerekli efor, arama nihai çözüm kalitesindeki bütün kazanımları yok edebilecekken hesaplama yapar. Bir çok problem meydana gelir çünkü optimal çözüm sık sık uygulanabilir alanın sınırlarına dayanacaktır. Optimum çözümün gentipine çok benzer bu çözümlerin çoğu uygulanamaz olacaktır. Bu yüzden, aramayı sadece uygulanabilir çözümlere kısıtlamak ya da çok sert cezalara dayatmak, popülasyonu Smith ve Tate (1993), Anderson ve Ferris (1994), Coit at al. (1996) ve Michalewicz'in (1995) araştırmasında gösterilen optimumluğa götürecektir şemayı vermeyi zorlaştıracaktır. Öte yandan, eğer ceza yeteri kadar sert değilse, o zaman daha geniş bir alan taranacak ve arama zamanının çoğu uygulanabilir alandan daha fazla alanlar çıkarmak için kullanılacaktır. Bu durumda arama uygulanabilir alanın dışında devam edecektir. [10]

III.6 BİR UYGULAMA MODELİ

Bir CNC torna tezgâhında, çelik bir silindir, bir P20 karbür takımıyla işlenecek. Torna tezgâhının 10kW'lık bir motoru var ve iletim verimliliği %75. Parça-takım-makine sistemi maksimum 5000N'luk bir kesme kuvvetine müsaade ediyor. Ampirik modeller, takım ömrü ve kesme kuvveti için deneysel olarak elde edilmiştir ve şu sonucu vermiştir;

$$T = \frac{5,48 \times 10^9}{v^{3,46} \cdot f^{0,696} \cdot a^{0,460}}$$
$$F_c = \frac{6,56 \times 10^3 \cdot f^{0,917} \cdot a^{1,10}}{v^{0,286}}$$

Seçilen kesici takımlar, müsaade edilebilir kesme parametrelerinde aşağıdaki limitlere sahiptir;

$$a_{\text{MIN}}=0,5 \text{ mm}, a_{\text{MAX}}=6,0 \text{ mm},$$
$$f_{\text{MIN}}=0,15 \text{ mm/dev}, f_{\text{MAX}}=0,55 \text{ mm/dev},$$
$$v_{\text{MIN}}=250 \text{ m/dak}, v_{\text{MAX}}=400 \text{ m/dak}.$$

Mevcut operasyonda, 219,912 mm³'lük hacim kaldırılacak. Hazırlık zamanı, takım değiştirme zamanı ve kesicinin boşta kalma zamanı sırasıyla 0,15 dak., 0,20 dak. ve 0,05 dak. olarak varsayılmıştır.

Genetik algoritma uygulaması için aşağıdaki parametreler tespit edilmiştir: statik popülasyon genişliği: 500; dinamik popülasyon genişliği: 10; maksimum Paretian büyüklüğü: 20; çağ sayısı: 25; evrimsel periyotların sayısı: 100.

TabloIII.1'de, evrimsel proseslerden sonra elde edilen Paretian noktaları gösterilmiştir. Görülebildiği üzere, kısıtlayıcı sınır kesme gücüdür.

Şekil III.1'de, elde edilen baskın olmayan noktalar, analizi ve karar alıcı prosesi daha kolaylaştıran Pareto önünü oluşturmak için çizilir.

Yukarıdaki şekli analiz edersek, bazı ilginç yönler not edilebilir. Birincisi, Pareto hududunun, bariz bir şekilde iki adet tanımlı bölgesi olan, tipik bir asimptotik profili vardır.

1. 1 noktasından 5'e: Kullanılan takım ömründe kayda değer bir düşüş ve toplam operasyon zamanında buna karşılık bir artış vardır. Bu zaman artışın (%12) kullanılan takım ömrü düşüşünden daha ılımlı olması gerektiği vurgulanmalıdır.
2. 5 noktasından 14'e: Operasyon zamanı yaklaşık %10 artar ve kullanılan takım ömründe küçük bir düşüş vardır.

Ek bilgi, yukarıda Pareto grafiğinden yapılan maliyet grafiğinden özetlenebilir. İki ana maliyet hesaplanmalıdır; takım ömrü, Z_T ki şöyle hesaplanır;

$$Z_T = \xi \cdot z_T$$

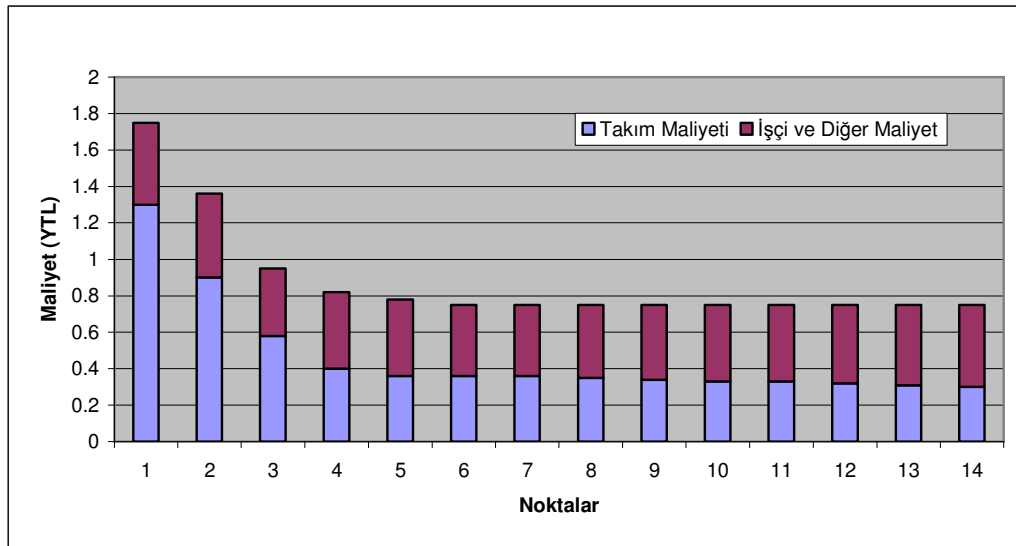
Burada z_T her bir takım ucu için maliyettir ve işçi ve genel giderler için maliyet, $Z_L + O$, şöyle hesaplanır;

$$Z_{L+O} = (z_L + z_O) \cdot \tau$$

Burada z_L ve z_O özel işçi ve diğer giderlerdir. Bu modelde, her bir takım ucu için maliyet, 14,17 YTL olarak seçilmiş; işçi ve diğer giderler ise sırasıyla 0,26 YTL ve 0,18 YTL/dak olarak seçilmiştir.

Tablo III.1. Optimizasyon prosesinin sonuçlarından elde edilen Paretian noktaları

No.	τ (min)	ξ (%)	a (mm)	f (mm/dev)	v (m/dak)	M (mm ³ /dak)	T (dak)	F_C (N)	P (kW)
1	0,85	9,22	1,6	0,55	400	346,520	6,9	1125	7,5
2	0,87	6,75	1,7	0,55	358	333,372	9,8	1257	7,5
3	0,91	3,90	1,9	0,55	294	311,495	18,1	1528	7,5
4	0,94	2,77	2,1	0,55	261	298,575	26,6	1727	7,5
5	0,95	2,46	2,1	0,55	250	294,281	30,3	1800	7,5
6	0,96	2,45	2,2	0,53	250	292,619	30,6	1800	7,5
7	0,98	2,39	2,5	0,45	250	284,831	32,3	1800	7,5
8	1,00	2,33	2,9	0,38	250	277,270	34,0	1800	7,5
9	1,02	2,28	3,3	0,32	250	270,135	35,7	1800	7,5
10	1,03	2,24	3,7	0,29	250	265,117	37,0	1800	7,5
11	1,06	2,18	4,3	0,24	250	256,782	39,4	1800	7,5
12	1,08	2,13	4,9	0,20	250	250,600	41,2	1800	7,5
13	1,10	2,08	5,5	0,18	250	244,608	43,2	1800	7,5
14	1,12	2,05	6,0	0,16	250	240,880	44,5	1800	7,5



Şekil III.1. Maliyet Grafiği

Bu grafikte (şekil III.1’de), takım maliyeti, işçi ve diğer gider maliyetleri, her bir Paretian noktası için, çizilmiştir. Açıkça, 1 noktasından 14’e, takım maliyetinde devamlı (ama aynı değil) bir düşüş vardır. Buna zıt olarak, bu yönde, işçi ve diğer giderde bir artış not edilebilir. Dahası, 5 ve 6 noktalarında, en düşük toplam maliyet değerleri başarılmıştır.

Sağ vuruş grafiği üzerinde çalışırsak, gerçek koşullara dayanılarak, birçok kararlara varılabilir. Örneğin, eğer söz konusu atölye, sıradan aletlerle çalışıyorsa, en uygun kesme parametreleri, düşük takım ömrü kullanımına ve kabul edilebilir bir operasyon zamanına sahip 5 ya da 6 noktasına tekabül eder. Dahası, bu noktalarda en düşük maliyete ulaşılmıştır.

Buna karşılık, eğer atölye küçükse ve süreksiz üretimleri varsa, bu her zaman işe yaramaz, o zaman düşük operasyon zamanlarını kullanmaya gerek yoktur. Bu durumda, uzun bir takım ömrü elde etmek daha önemlidir, bu yüzden, 1 noktasıyla eşleşen kesme parametreleri seçilmelidir.

Sonuç olarak, savaş zamanı ve ya diğer başka acil durumlar gibi özel koşullarda, üretim hacmi, takım ömrü ve operasyon zamanı ihmal edilerek, kesme parametrelerini seçmede en önemli elementtir. Sonuç olarak, bu durumda, artan takım tüketimi ve yüksek maliyet yerine, en düşük kesme zamanını sunan 14 noktasına tekabül eden kesme parametreleri seçilmelidir. [7]

III.6.1 Sonuç

Ele alınan modelde fark edileceği üzere, müteakip bir çok hedefli optimizasyon, tornalamadaki kesme parametrelerini seçmede bir karar almak için çok geniş bilgi önerir. Pareto hudut grafikleri vasıtasıyla, herhangi bir koşul için doğru parametre seçimini kolaylaştıran birçok farklı durum ele alınabilir.

Anlatılan mikro GA, mantıklı düşük hesaplanabilir bir maliyette Pareto önünü ayarlamak için, eşit yayılmış birçok noktayı elde etmeyi göstermiştir. Çeşitlilik sürdürümü ve sınırları işleme tarzı gibi yönler, çalışılan problem için başarılı bir şekilde sınıflanmıştır.

Maliyet analizi Pareto önü bilgisini tamamlayabilir ve karar alıcı prosese yardımcı olur. Önerilen model, kesme yüzeyi sıcaklığı gibi, daha fazla sınırı içermesi için genişletilmelidir.[7]

III.7. ELEŞTİRİ ve ÖNERİLER

Onwubolu ve Kumalo'nun (2001) bu dergideki tezleri işleme koşulları optimizasyon problemlerini çözmeye çeşitli çözüm algoritmaları ile genetik algoritmaların etkinliğini kıyaslamıştır. Chen ve Tsai'nin (1996) problemini kullanarak genetik algoritmaların tavlama benzetiminden bariz bir şekilde daha iyi performans verdiği sonucuna varmışlardır. Bu tez Onwubolu ve Kumalo'nun işleme modelini yanlış ele aldıklarını göstermektedir. Ayrıca Onwubolu ve Kumalo tarafından kullanılan genetik algoritmanın dizi sunumunu da tartışır ve bazı baskı hatalarını da düzeltir.

İşleme koşulu optimizasyon problemlerinde, başlangıç hedefi, belli işleme koşullarını karşılarken, üretim maliyetlerini asgari düzeye çekecek olan optimal işleme parametrelerini (ilerleme miktarı, kesme hızı, talaş derinliği, paso sayısı) belirlemektir. İşleme koşulu optimizasyon problemi genellikle, doğrusal olmayan bazı optimizasyon metodolojileri için zorluklar gösterebilecek yüksek doğrusallığı olmayı içermektedir. Bu probleme çözüm için bir çok metod tavsiye ve test edilmiştir.

Araştırmacılar ve pratisyenler bir kompleks doğa problemleriyle uygun bir optimizasyon algoritması seçmek zorundadırlar. Metaheuristic alanındaki son araştırmalar (tavlama benzetimi, genetik algoritma, tabu arama ve karınca koloni sistemi vb.) çok amaçlı optimizasyon metodolojilerini geliştirecek karmaşık bir hırs ortaya koymuştur. Tavlama benzetimi ve genetik algoritma gibi non-Gradient tabanlı metaheuristicler, doğrusal olmayan optimizasyon problemlerini çözmeye güvenilir ve güçlülerdir. Khan *et al.* (1997) işleme koşulu optimizasyon problemlerinde tavlama benzetimi ve genetik algoritmanın gradient tabanlı metotlara göre daha belirgin avantajlar sunduğunu rapor etmiştir. Buna ilaveten, Khan *et al.* tavlama benzetiminin genetik algoritmalar ile oldukça rekabete dayalı olduğunu göstermiştir.

Onwubolu ve Kumalo'da (2001), genetik algoritma en iyi çözüm yaklaşımı olarak ortaya çıkmaktadır. Bununla birlikte Onwubolu ve Kumalo, genetik algoritmaların üstünlüğünü aşırı ifade etmişlerdir çünkü Chen ve Tsai'de (1996) sunulan işleme modelini yanlış ele almışlardır. Bu notun amacı bu yanlış ele alımı düzeltmektir. Buna ilaveten, bu not Onwubolu ve Kumalo'da sunulan genetik algoritmaların dizi sunumunun gereksiz yere uzun olduğunu göstermektedir. Dahası bazı baskı hataları da düzeltilmiştir.[5]

III.7.1 İşleme Modeli ve Dizi Sunumu

Chen ve Tsai'de (1996) sunulan işleme modeli sınırlandırılmış doğrusal olmayan bir optimizasyon problemidir. Bu modelde, işleme parametreleri bir grup pratik işleme sınırlarını altında asgari birim üretim maliyeti bakımından kesme koşullarını iyilemek için belirlenmiştir. İşleme parametreleri kaba paso sayısını, kaba talaş derinliğini, kaba kesme hızını, kaba ilerleme miktarını, son paso derinliğini, son paso hızını ve son paso ilerleme miktarını (n , d_r , V_r , f_r , d_s , V_s , f_s) kapsamaktadır.

Chen ve Tsai (1996) işleme modellerini aşağıdaki kaynağa göre geliştirmişlerdir: “Bu tezde, birçok kaba pasosu ve tek bir bitiş pasosu olan çok aşamalı bir tornalama modeli formüle edilmiştir. Onwubolu ve Kumalo (2001) genetik algorithmadan tavlama benzetimine (Chen ve Tsai'nin 1996) elde edilen sonuçlarını topladıklarından itibaren Chen ve Tsai'de tarif edilen kaynağı takip etmeliydiler. Tornalamada, toplam talaş derinliği kaldırılacak gerekli miktarı belirler. Bu yüzden bir denklem (Chen ve Tsai'deki s.2811), bitiş talaş derinliğine (d_s) ve toplam kaba talaş derinliğine (nd_r) eşit toplam talaş derinliğini (d_t) sınırlamada kullanılır. Bu eşitlik sınırlaması şöyle tarif edilebilir:

$$d_r = \frac{d_t - d_s}{n}$$

Kaba paso sayısının (n) bir tamsayı olduğunu not ediniz. Çözüm prosedürünün kolaylığı için, Chen ve Tsai iki karar değişkenini (n , d_r) ve bazı matematiksel düzeltmeler tarafından eşitlik sınırlamasını elemiştir. Pratik olarak, kaba paso sayısına ait geçerli aralık dardır ve böylece geniş kapsamlı ve ayrıntılı bir tavırla belirlenebilir. n değeri (kaba paso sayısı), d_s değeri (bitiş paso derinliği) ve d_r değeri (kaba paso derinliği) verileri denklem sınırlamasından elde edilir. Bu yüzden, tavlama benzetimi tabanlı optimizasyon algoritması (Chen ve Tsai 1996) sadece optimal olarak işleme modelindeki beş işleme parametresini (V_r , f_r , V_s , f_s , d_s) belirlemiştir.

Onwubolu ve Kumalo'da (2001) genetik algoritmalar, altı işleme parametresini (d_r , V_r , f_r , V_s , f_s , d_s) optimal olarak belirlemiştir.

Onwubolu ve Kumalo'da (2001) genetik algoritmalar, altı işleme (d_r , V_r , f_r , d_s , V_s , f_s) parametresini ikili düzene kodlandırılmalarıyla optimal olarak belirlemiştir (Onwubolu ve Kumalo, s.3734). Bu arada, Onwubolu ve Kumalo üst eşitlik

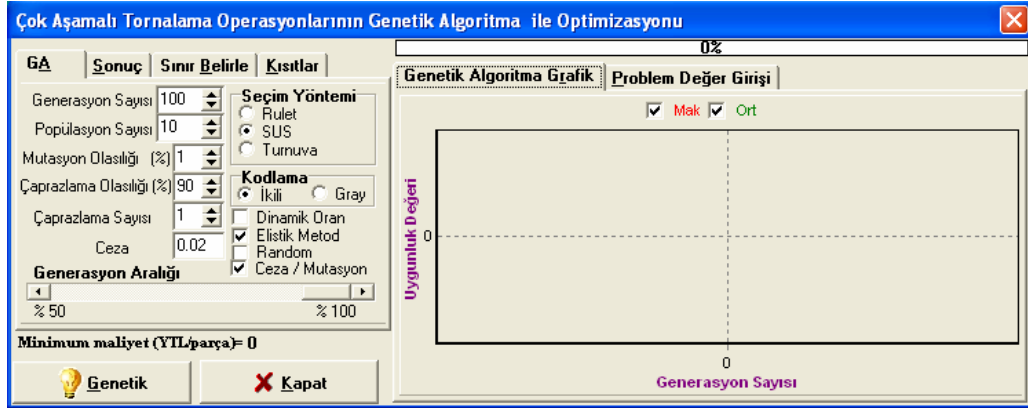
sınırlamasını dikkate almamışlardır. Genetik algoritma ile elde ettikleri çözümlerinde, toplam talaş derinliği (d_t) bitiş pasosu derinliğine (d_s) ve toplam kaba paso derinliğine (nd_r) eşit olmayabilir. Optimal objektif fonksiyon değerleri (birim üretim maliyeti, YTL/parça) işleme sınırlaması karşılanırsa daha iyi olabilir, mesela tornalanarak kaldırılan miktar gerekenden azdır. Dahası, objektif fonksiyon değerleri hesabı, $(d_t - d_s) / d_r$ (Onwubolu ve Kumalo, denklem 8) bir tamsayı değerine sınırlanmadığı sürece yanlış olabilir (mesela kaba paso sayısı n yanlışlıkla devamlı olabilir). Çözümleri, işleme koşulları karşılanmadığı sürece kullanışsız olabilir.

İşleme modelinde yanlış işlem müteakip iki şekilde düzeltilebilir. Onwubolu ve Kumalo'nun altı işleme parametresinin kodlama şemasının kullanılması koşuluyla, işleme modeli (Onwubolu ve Kumalo, s.3732) eşitlik sınırlamasını kapsamalıdır. Öte yandan çözümü Chen ve Tsai'de anlatılan matematiksel işlemleri uygulayarak, beş işleme parametresi ile sunmak yeterlidir (V_r, f_r, d_s, V_s, f_s). Kaba paso sayısı (n) ve kaba paso derinliği (d_r), sırasıyla geniş kapsamlı bir şekilde ve denklemden (1) elde edilebilir. Yukarıdaki tartışma Onwubolu ve Kumalo tarafından kullanılan dizi sunumunun gereksiz yere uzun olduğunu göstermektedir.[3]

BÖLÜM IV

IV.1 BİLGİSAYAR PROGRAMI

Bu tezin III. Bölümünde, verilen probleme ait hesaplamaları yapan ve probleme ait kısıtlara bağlı kalınmak kaydıyla, maliyetin hangi metotlar denenerek minimuma çekilebileceğini ve sınır değerlerle ne kadar oynanırsa maliyetin ne kadar değişeceğini takip edebileceğimiz, Delphi dilinde yazılmış bir bilgisayar programı yaparak problemin işleyişini test etmiş oluyoruz. Verilen örneğin programda işleyişini anlamak için programa kısaca bir göz atalım.



Şekil IV.1 Programın GA ve Grafik Sekmeleri

Bu sekmede (GA), belli genetik operatörler ve genetik yöntem seçenekleri verilerek hesabı yapılacak probleme ait maliyet değeri bulunur. Problemimizde generasyon sayısı 100, popülasyon sayısı 10, Mutasyon olasılığı 1, çaprazlama olasılığı 90, çaprazlama sayısı 1 ve ceza fonksiyonu 0,02 alınmıştır. Problemi ceza fonksiyonu ile çözmek optimum sonucu vermiştir. Kodlama ikili alfabede yapılp, seçim yöntemi SUS olarak belirlendiğinde, minimum maliyet değerini verecek yöntemler seçilmiş olur. Bu opsiyonların hemen altında generasyon aralığı bir kaydırma çubuğu kolaylığı ile verilmiştir. Bu kaydırma çubuğunu %100 yaptığımızda, bir önceki generasyondaki en kötü birey yeni generasyona aktırılmaz

ki basit genetik algoritmanın prensibi de buna dayanır. Bunu yanı sıra, en kötü bireyden her ikisinden birini yeni kuşağa aktarma olasılığı da kaydırma çubuğu %50 kısmına getirerek sağlanabilir. Bütün bu opsiyonlar tamamlandıktan sonra Genetik butonuna basılarak gerekli hesaplama yapılır ve parça başına minimum maliyet sonucu çıkar. Sağ taraftaki diyagramda ise generasyon sayısı kadar her bir iterasyonda elde edilen uygunluk değerini gösteren bir grafik elde ederiz.

Genetik Algoritma Grafik		Problem Değer Girişi	
D (mm)	50	k1	108
L (mm)	300	k2	132
dt (mm)	6	k3	1
Tl (dak)	25	k4	2.5
Tu (dak)	45	k5	1
Fu (kg-f)	5	p	5
Pu (KWatt)	200	q	1.75
Sr (mikron)	10	r	0.75
Qu (C)	1000	μ	0.75
SC	140	v	0.95
Ra (mm)	1.2	nü	0.85
Teta	0.5	lamda	2
		ve	-1
		to	0.4
		fi	0.2
		delta	0.105
		ko (YTL/dak)	0.5
		kt (YTL/dak)	2.5
		h1	0.0007
		h2	0.3
		tc (dak/parça)	0.75
		te (dak/kenar)	1.5
		Co	600000000000

Şekil IV.2 Programın Sonuç ve Problem Değer Girişi Sekmeleri

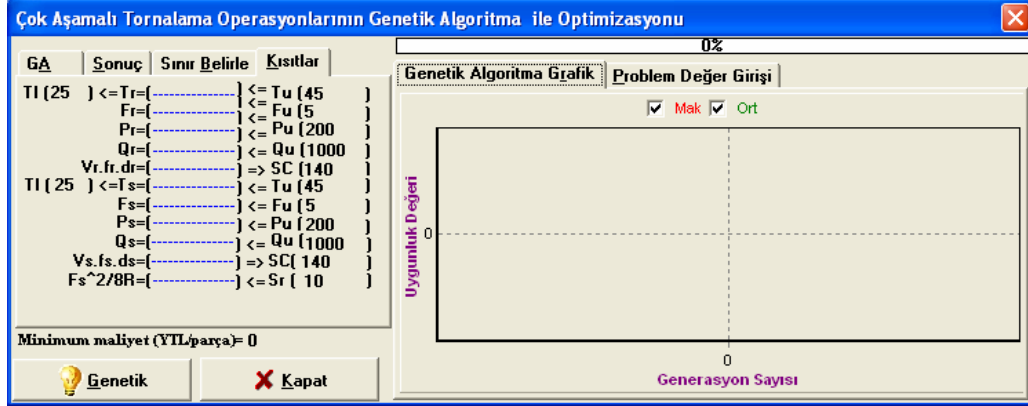
İkinci sekmede (Sonuç) ise, genetik butonuna basmamızla beraber yapılan hesaplamaların ardından elimize geçen sonuçları bildiren kutucuklar vardır. Bunlar, kaba ve bitiş tornalamada kesme hızı (V_r , V_s); kaba ve bitiş tornalamada ilerleme miktarı (F_r , F_s); kaba ve bitiş tornalamada talaş derinliği (d_r , d_s)'dir. Bu hesaplamalarda kullanılacak sabitleri ise sağdaki Problem Değer Girişi kısmından yapmaktayız.

Sınır Belirle	
Vr min	50
Vr mak	500
fr min	0.1
fr mak	0.9
dr min	1
dr mak	3
ds min	1
ds mak	3
fs min	0.1
fs mak	0.9
Vs min	50
Vs mak	500

Şekil IV.3 Programın Sınır Belirle Sekmesi

Üçüncü sekmede (Sınır Belirle), tezimize konu olan bu 3 parametreye ait (kesme hızı V, ilerleme miktarı f, talaş derinliği d) sınır değerler vardır. Bu sınır

değerlerini belli oranlarda değiştirdiğimizde, çalıştırdığımız genetik algoritma ile minimum maliyetin ne kadar değiştiğini bir sonraki bölümde inceleyeceğiz.



Şekil IV.4 Programın Kısıtlar Sekmesi

Dördüncü sekmede (Kısıtlar), probleme ait değerler girildiğinde elde edilen sonuçlar ile problemde istenen kısıtların kıyaslamasını bulunduğu bir tablo ekrana gelmektedir.

IV.2 PROGRAMIN İÇERİĞİ

```
// object fonksiyonu
function objfunc(var chrom : chromosome;j:integer; var
pop:population):extended;
var cm,ci,cr,ct,uc,val:real; tr,ts,tp,ffr,pr,qr,ffs,ps,qs:extended;
begin
with pop[j] do
begin
Vr:=(((vrmax-vrmin)/(LdExp(1,22)-1))*x1)+vrmin;
fr:=(((frmax-frmin)/(LdExp(1,22)-1))*x2)+frmin;
dr:=(((drmax-drmin)/(LdExp(1,22)-1))*x3)+drmin;
ds:=(((dsmax-dsmin)/(LdExp(1,22)-1))*x4)+dsmin;
fs:=(((fsmax-fsmin)/(LdExp(1,22)-1))*x5)+fsmin;
vs:=(((vsmax-vsmin)/(LdExp(1,22)-1))*x6)+vsmin;

tr:=(co/(power(vr,p)*power(fr,q)*power(dr,r)));
ts:=(co/(power(vs,p)*power(fs,q)*power(ds,r)));
tp:=teta*tr+(1-teta)*ts;

cm:=ko*(((pi*d*1)/(1000*Vr*fr))*((dt-ds)/dr)+ ((pi*d*1)/(1000*vs*fs)));
ci:=ko*(tc+(h1*1+h2)*(((dt-ds)/dr)+1));
cr:=(ko*te)/tp*(((pi*d*1)/(1000*Vr*fr))*((dt-ds)/dr)+ ((pi*d*1)/(1000*vs*fs)));
ct:=(kt/tp)*(((pi*d*1)/(1000*Vr*fr))*((dt-ds)/dr)+ ((pi*d*1)/(1000*vs*fs)));
uc:=cm+ci+cr+ct;
val:=1/uc; // fitness değeri bulunuyor

// kısıt değişkenleri hesaplanıyor
ffr:=k1*power(fr,mu)*power(dr,v);
pr:=(ffr*vr)/(6120*nu);
```

```

qr:=k2*power(vr,tos)*power(fr,fi)*power(dr,delta);
ffs:=k1*power(fs,mu)*power(ds,v);
ps:=(ffs*vs)/(6120*nu);
qs:=k2*power(vs,tos)*power(fs,fi)*power(ds,delta);
begin
// ceza yaklaşımına göre kısıtlar hesap ediliyor
if ((tr > tu)or(tr < tl)) then val:=val-ceza;
if (ffr > fu) then val:=val-ceza;
if (pr > pu) then val:=val-ceza;
if (qr > qu) then val:=val-ceza;
if (((power(vr,lamda)*fr*power(dr,v)) < sc)) then val:=val-ceza;

if ((ts > tu)or(ts < tl)) then val:=val-ceza;
if (ffs > fu) then val:=val-ceza;
if (ps > pu) then val:=val-ceza;
if (qs > qu) then val:=val-ceza;
if (((power(vs,lamda)*fs*power(ds,v))<sc)) then val:=val-ceza;

// if (((Power(fs,2)/(8*ra)) < (s*r)))then val:=val-ceza;

if( (vs < k3*vr) or( fr < k4*fs) or (dr < k5*ds)) then val:=val-ceza;
end

```

IV.3 PROBLEMİN PROGRAMDA İŞLEYİŞİ

Programın sağ sütunundaki ikinci sekmesinde bulunan Problem Değer Girişi kısmından probleme ait değerler girilip, sol sütundaki Sınır Belirle sekmesinden de sınır değerleri girildikten sonra, ilk sekmeye (GA) geçilir ve uygun yöntem seçilerek, Genetik butonuna basıldığında en uygun değeri buluruz. Burada dikkat etmemiz gereken nokta, hangi yöntemi seçeceğimizdir. Problem değerleri ve kısıtlar girildikten sonra ilgili seçim yöntemleri sırayla seçilir ve en uygun değeri veren yöntem kullanılır. Ele aldığımız problemdeki formül ve verilere göre parça başına minimum maliyeti veren seçim yöntemi SUS (Sıralı Seçim Yöntemi) olmuştur. Kodlamayı ikili alfabe yöntemi ile yaptık ve uygunluk değerini bulmak için ceza fonksiyonunu kullandık. Hesaplamayı yaptırdığımızda Şekil IV.5 deki tablo ile karşılaştık. Yukarıda ele aldığımız problemdeki değerler girildiğinde ve en uygun parça başı maliyet değerini verecek yöntem seçildiğinde çıkan sonuç, 1,71 YTL olmuştur. Parçamızın çapı, boyu gibi değerler ve probleme ilişkin sabitlere dokunmaksızın bu maliyeti aşağı çekebilmenin tek yolu, sınır değerlerle oynamaktır. Bunun için sınır değerleri sırayla değiştirerek en uygun değeri bulmaya çalıştım. Burada dikkat ettiğim nokta, sınır değerleri mümkün olduğu kadar az değiştirmeye çalışmaktır. Bu bağlamda üçüncü sekme olan “Sınır Belirle” kısmındaki değerlerden

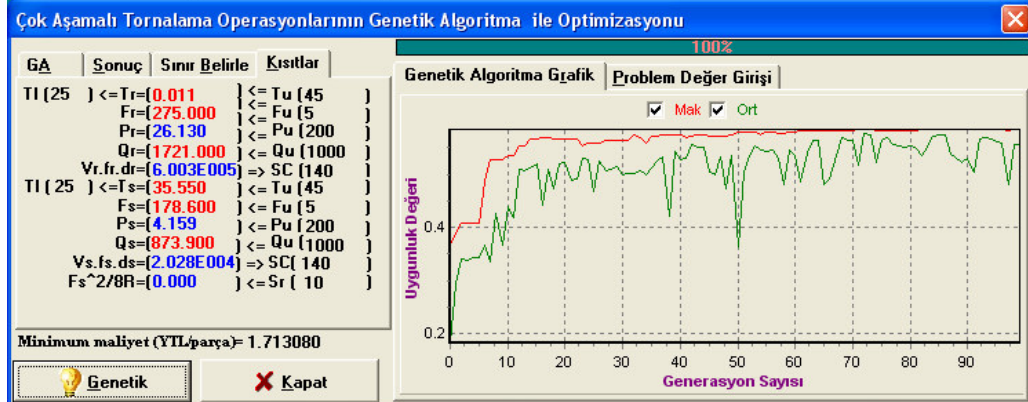
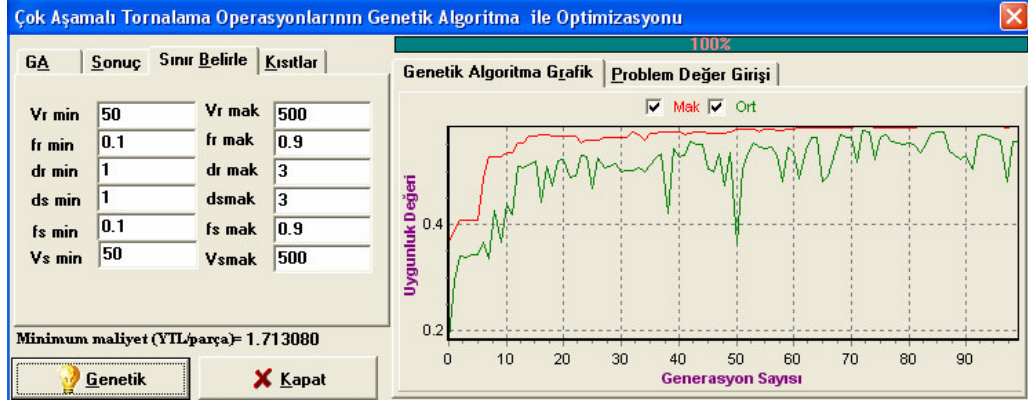
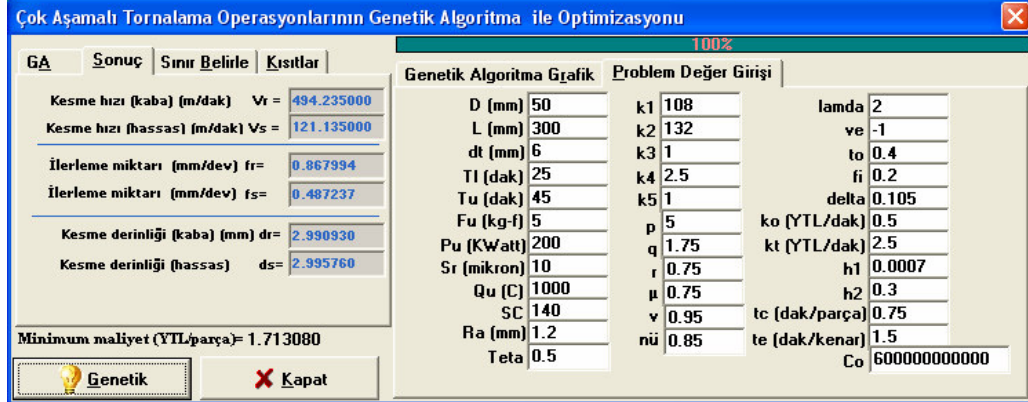
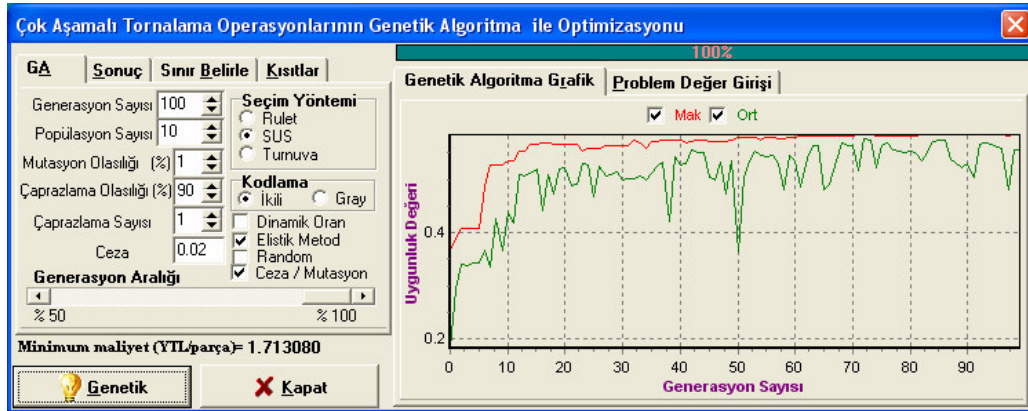
V_r 'nin üst sınırı olan 500 değerini 450 yaptığımda karşıma Şekil IV.6'deki; 550 yaptığımda ise IV.3'teki tablo çıkmıştır. V_r ile ilgili bu sınır değişikliklerinde gördüğümüz üzere parça başına minimum maliyet değeri 450 de 1,51 çıkmıştır ve bu en uygun olamıdır. O halde V_r 'yi 450 kabul edip, diğer sınırlarda değişiklik yapalım.

Sınır değerlere ait veri değişikliklerinde ortaya çıkan sonuçlara ilişkin tablo şöyledir:

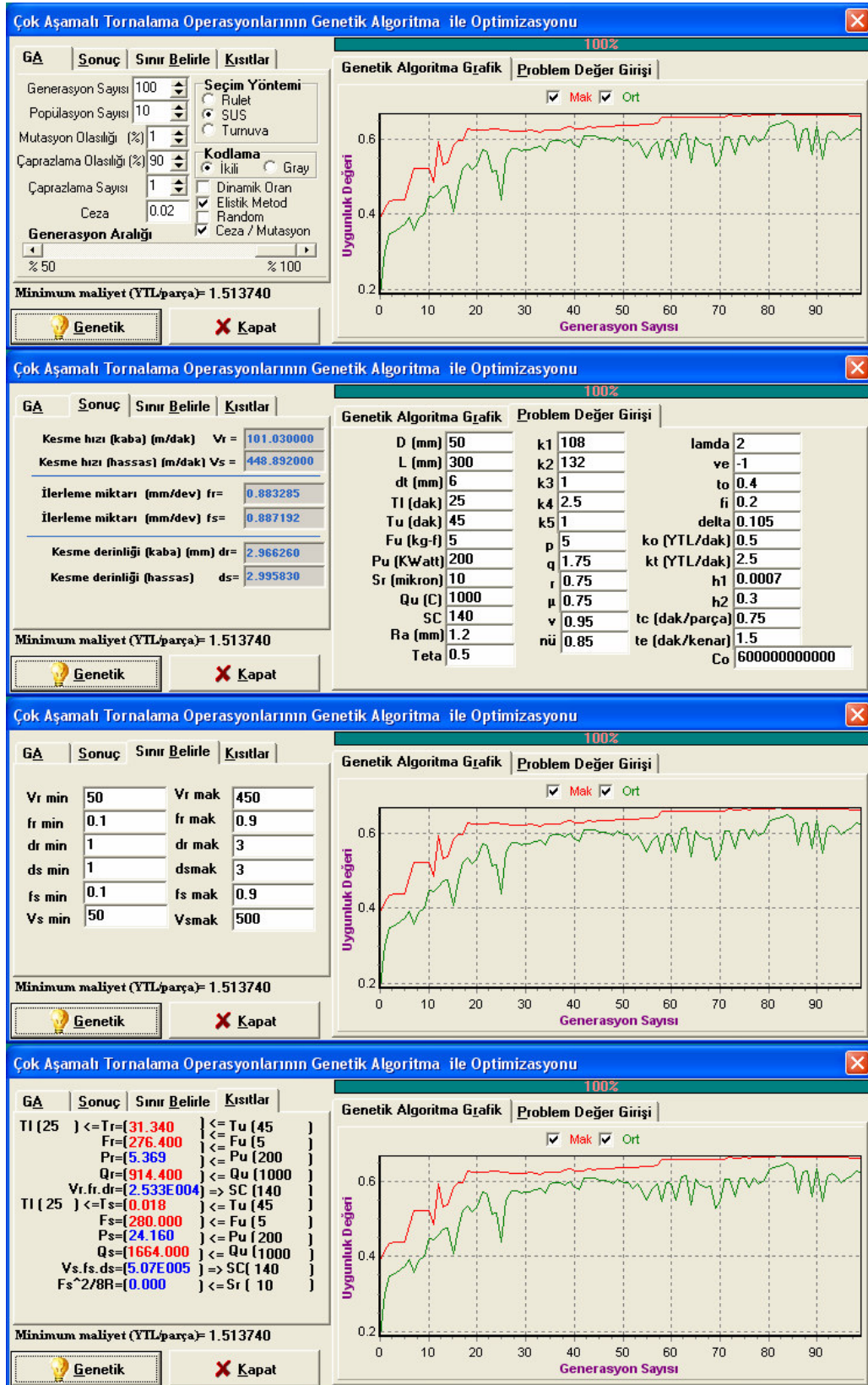
Tablo IV.1 V_r 'ye ait veri değişiklikleri ve maliyet değerine etkisi

Kıstas	Sınır Değeri	Parça Başına Maliyet Değeri	Şekil no
V_r	500	1,71 YTL	Şekil IV.5
V_r	450	1,51 YTL	Şekil IV.6
V_r	550	1,65 YTL	Şekil IV.7
f_r	0,8	1,55 YTL	Şekil IV.8
f_r	1	1,53 YTL	Şekil IV.9
f_r	1,1	1,49 YTL	Şekil IV.10
f_r	1,2	1,48 YTL	Şekil IV.11
d_r	2,8	1,54 YTL	Şekil IV.12
d_r	3,2	1,51 YTL	Şekil IV.13
d_r	3,4	1,38 YTL	Şekil IV.14
d_r	3,6	1,39 YTL	Şekil IV.15
d_r	3,8	1,29 YTL	Şekil IV.16
d_s	2,8	1,33 YTL	Şekil IV.17
d_s	3,2	1,28 YTL	Şekil IV.18
d_s	3,4	1,23 YTL	Şekil IV.19
d_s	3,8	1,13 YTL	Şekil IV.20
f_s	0,8	1,13 YTL	Şekil IV.21
f_s	1	1,14 YTL	Şekil IV.22
f_s	1,1	1,11 YTL	Şekil IV.23
f_s	1,2	1,25 YTL	Şekil IV.24
V_s	450	1,14 YTL	Şekil IV.25
V_s	550	1,12 YTL	Şekil IV.26

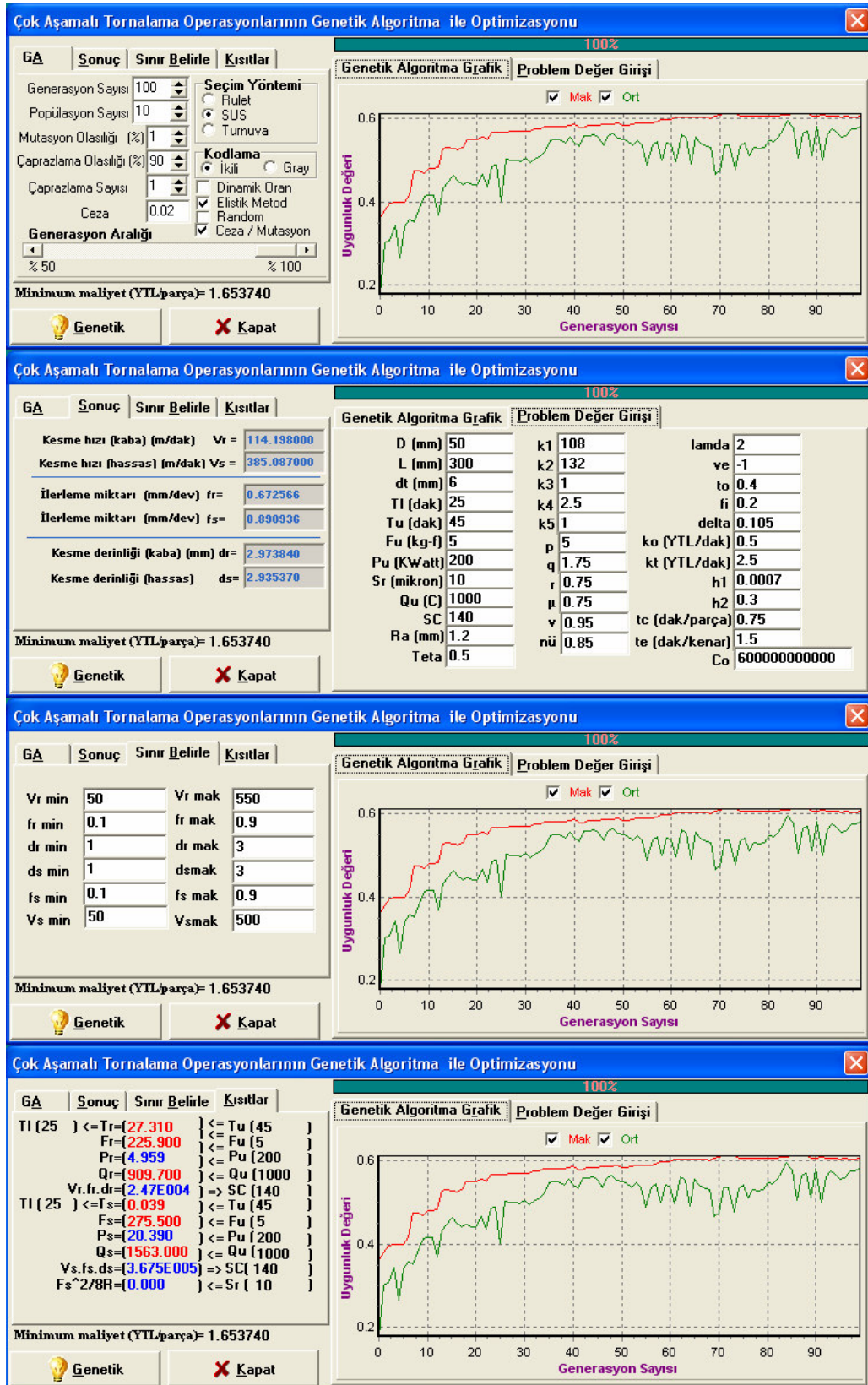
Kalın puntuyla belirtilmiş satırlar en uygun sonucu veren sınır değerine ait değişikliğe dalalet eder.



Şekil IV.5 Vr=500 için parça başına maliyet değeri hesaplaması



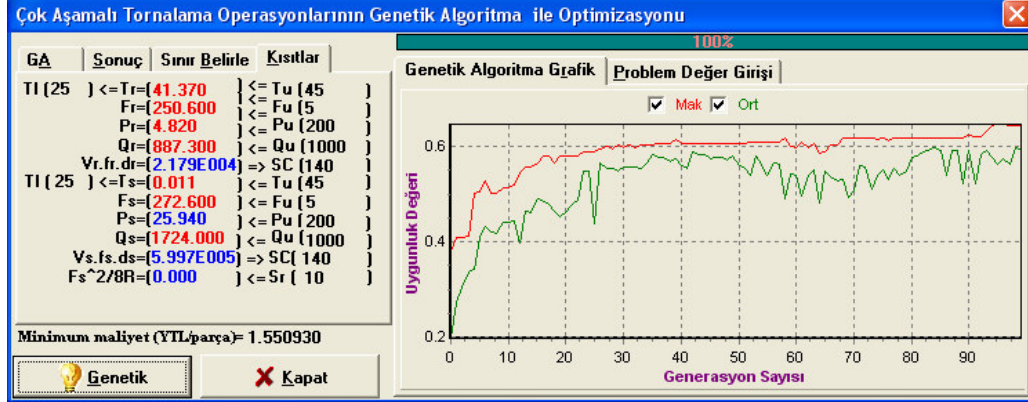
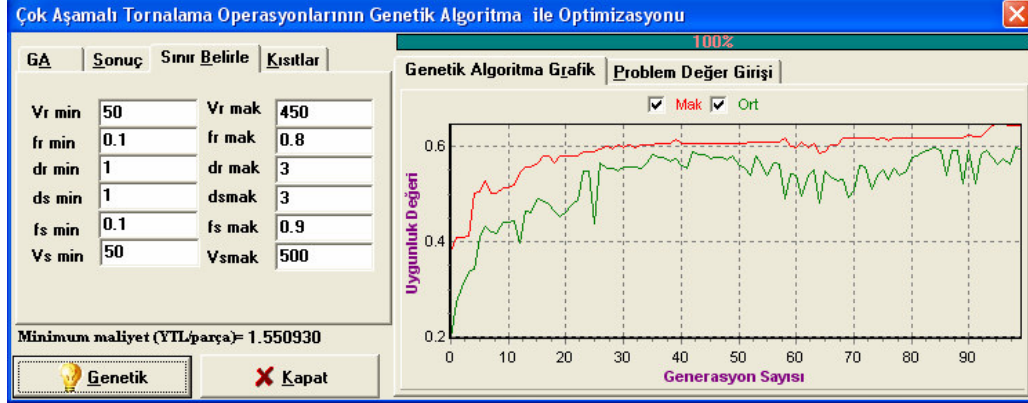
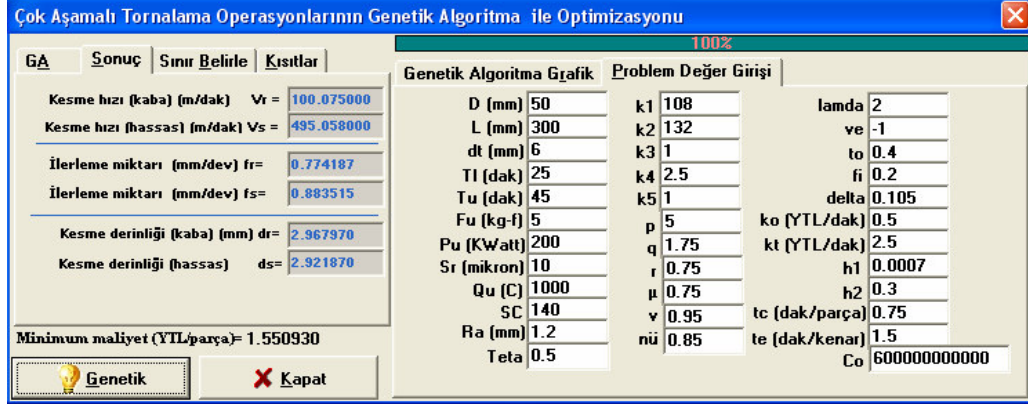
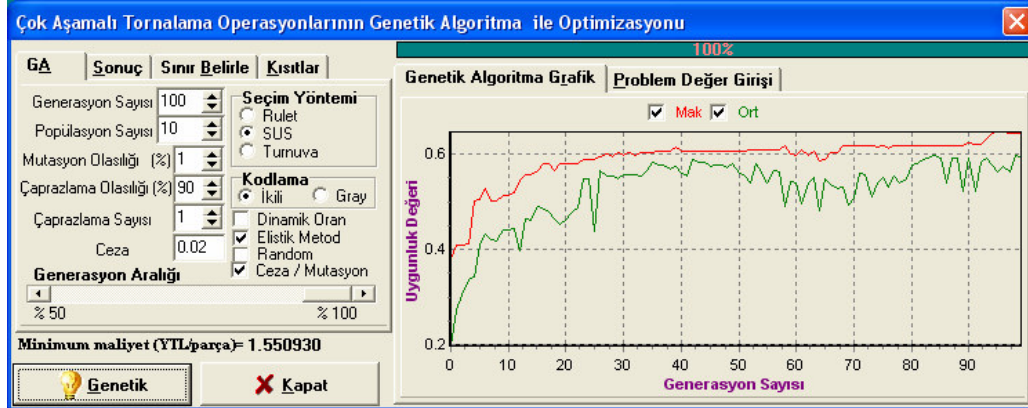
Şekil IV.6 Vr=450 için parça başına maliyet değeri hesaplaması



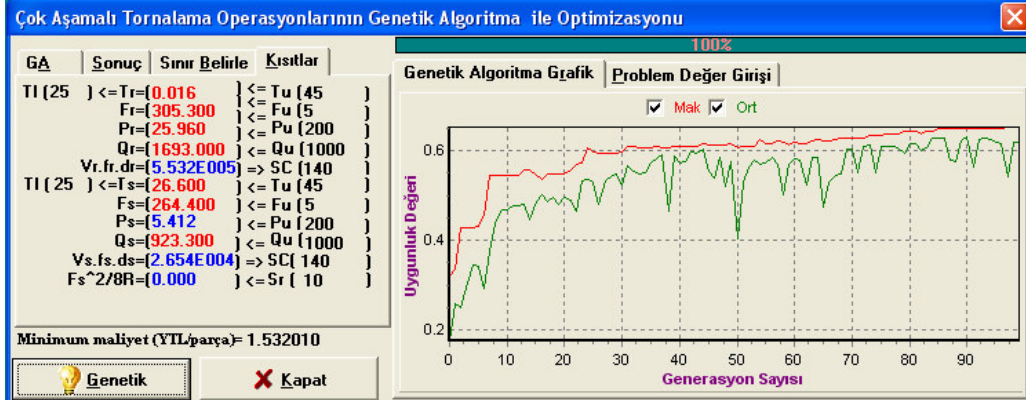
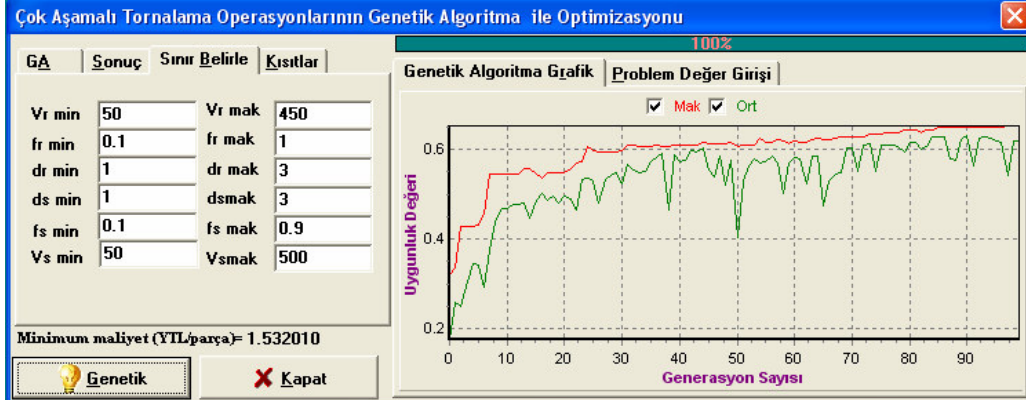
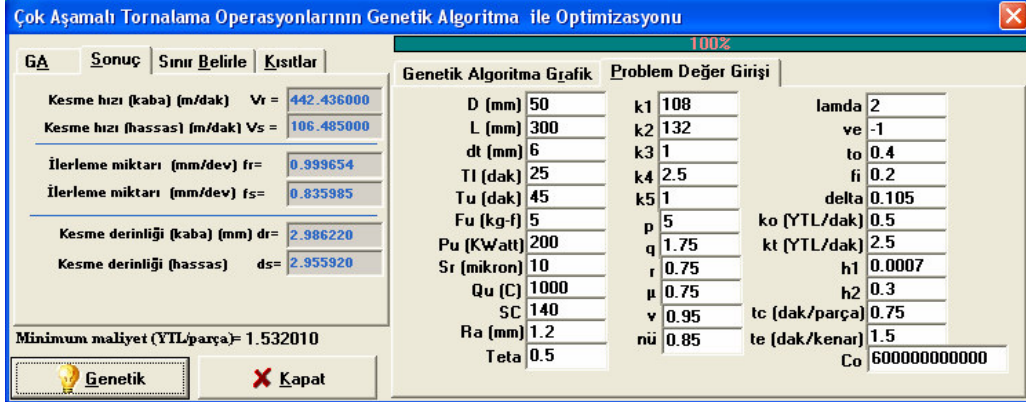
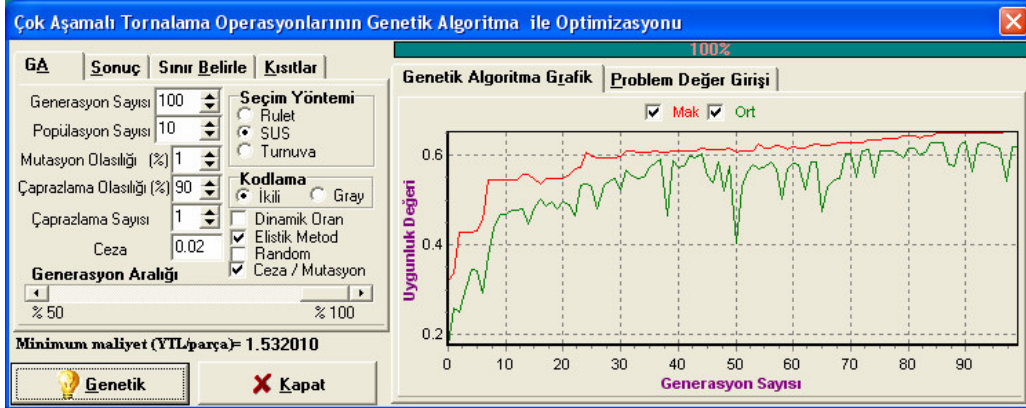
Şekil IV.7 Vr=550 için parça başına maliyet değeri hesaplaması

f_r 'nin üst sınırını sırasıyla 0,8; 1; 1,1 ve 1,2 yaptığımda elde ettiğim parça başı minimum maliyet değerleri sırasıyla, 1,55 YTL; 1,53 YTL; 1,49 YTL ve 1,48 YTL olmuştur. Gördüğümüz üzere kaba tornalamada ilerleme miktarını düşürmek maliyeti artırmıştır, hâlbuki ilerleme miktarının üst sınırını 1,2 yaptığımızda, girilen değerler arasında en uygun parça başı maliyet değerini bulmuş olduk. f_r 'ye ait bu sınır değişiklikleri Şekil IV.8, IV.9, IV.10 ve IV.11'de gösterilmiştir. Buradan anlaşılacağı üzere f_r 'yi 1,2 alarak diğer parametrelerimizde değişiklik yapmaya devam edeceğiz.

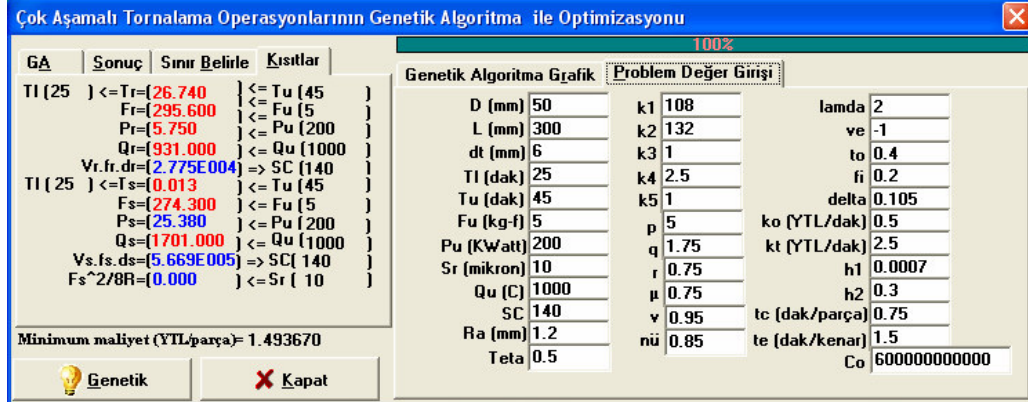
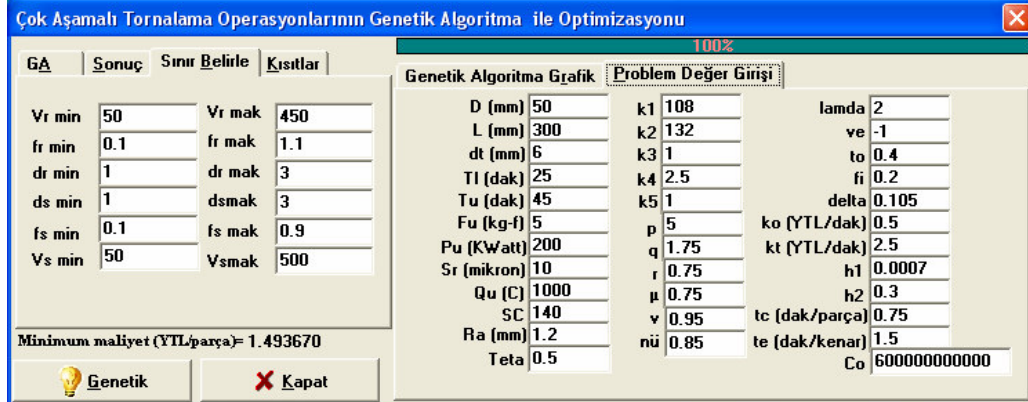
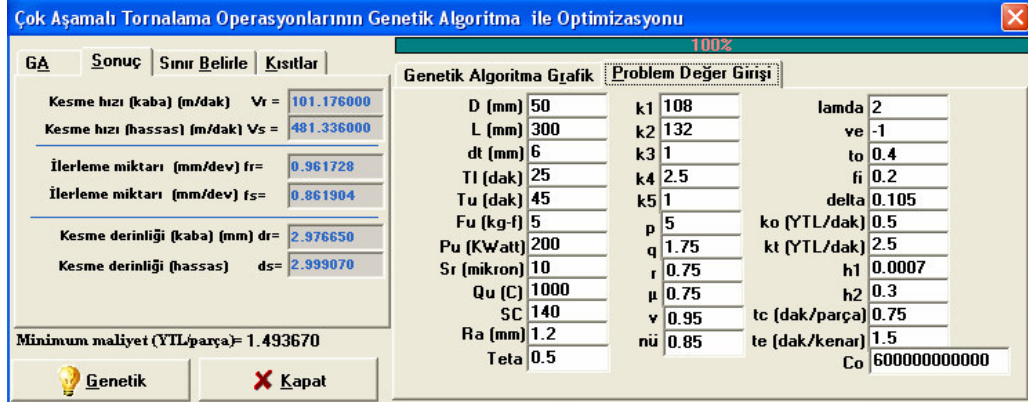
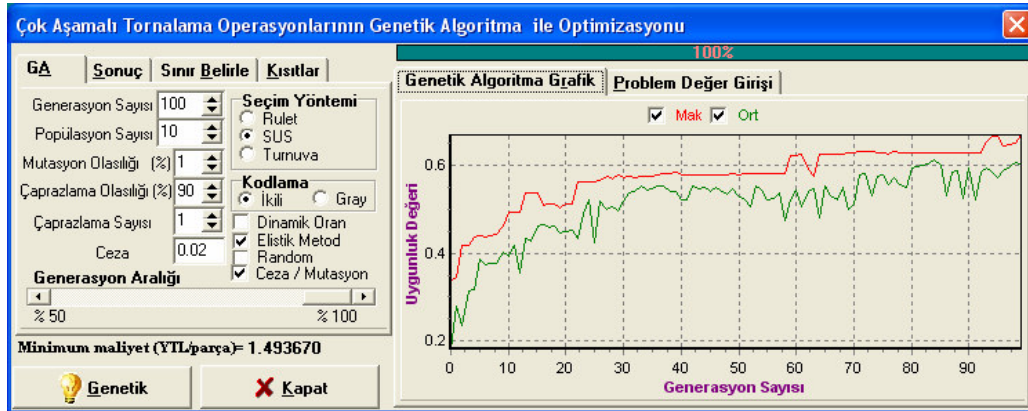
Bu bağlamda d_r talaş derinliği değerini sırasıyla 2,8 (Şekil IV.12); 3,2 (Şekil IV.13); 3,4 (Şekil IV.14); 3,6 (Şekil IV.15) ve 3,8 (Şekil IV.16) veriyorum. Parça başına elde ettiğim minimum maliyet değerleri şöyle oluyor: 1,54 YTL; 1,51 YTL; 1,38 YTL; 1,39 YTL ve 1,29 YTL. Bir önceki bölümde sınır değerlerle oynarken verilecek değerlerin seçiminde dikkat edilecek hususun sınır değerlerden çok uzaklaşmamak olduğunu söylemiştim. Ancak görülüyor ki 3,8 değerini verdiğimde maliyet oldukça düşüyor. Bu gibi durumlarda sınır değerini yaklaşık %25 oranında değiştirme riskini alabiliriz. Zira aynı riski hassas talaş derinliği değeri olan d_s için de alıyorum. d_s 'ye sırasıyla 2,8 (Şekil IV.17); 3,2 (Şekil IV.18); 3,4 (Şekil IV.19) ve 3,8 (Şekil IV.20) veriyorum. Elde ettiğim sonuçlar 1,33 YTL; 1,28 YTL; 1,23 YTL, 1,13 YTL oluyor. Yine görüldüğü gibi d_s üst sınırından uzaklaştığımda parça başına maliyet değeri oldukça düşüyor. Bu yüzden d_s 'yi 3,8 alıyorum ve f_s yani hassas ilerleme miktarının üst sınır değerinde şu değişiklikleri yapıyorum: 0,8 (Şekil IV.21); 1 (Şekil IV.22); 1,1 (Şekil IV.23); 1,2 (Şekil IV.24). Elde ettiğim sonuçlar şöyle oluyor: 1,13 YTL; 1,14 YTL, 1,11 YTL; 1,25 YTL. Burada da görüyoruz ki f_s 'ye ait en uygun sonucu 1,1 değerini girip, 1,11 YTL maliyetini bularak alıyoruz. Hassas talaş derinliğine ise 450 (Şekil IV.25) ve 550 (Şekil IV.26) değerlerini girdiğimizde elde ettiğimiz sonuç 1,14 YTL ve 1,12 YTL oluyor. Yani mevcut 1,11 YTL maliyetinden daha uygun bir maliyet bulamıyoruz. Demek ki sınır değerlerle oynayarak elimize geçen en iyi parça başı maliyet değeri 1,11 YTL oluyor. Başlangıçtaki 1,71 değerine göre 60 kuruşluk bir kar sağlanabiliyor ve bu da parça başına %35'lik bir maliyet kazancına tekabül ediyor. Programda generasyon sayısını 100'den 1000'e (Şekil.28) ve 10000'e (Şekil 10000) çıkardığımda elde ettiğim sonuçta ufak çaplı bir değişiklik olmaktadır. Generasyon sayısını arttırmakla birlikte işlem süresi de uzamaktadır.



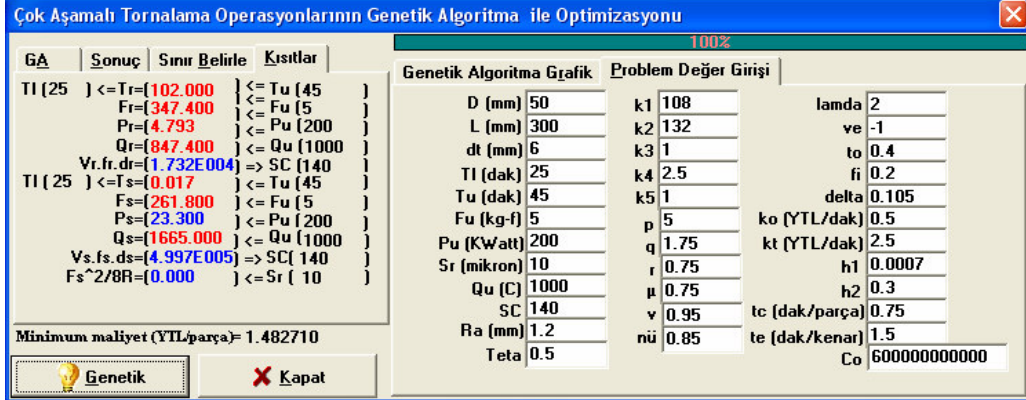
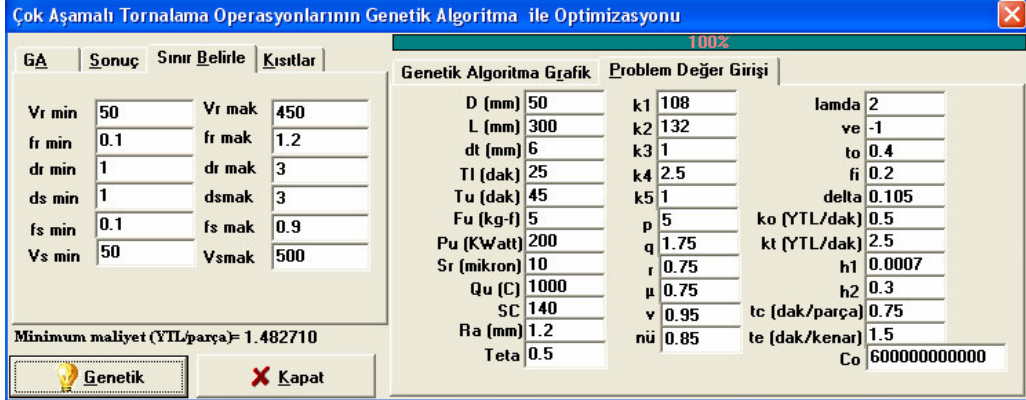
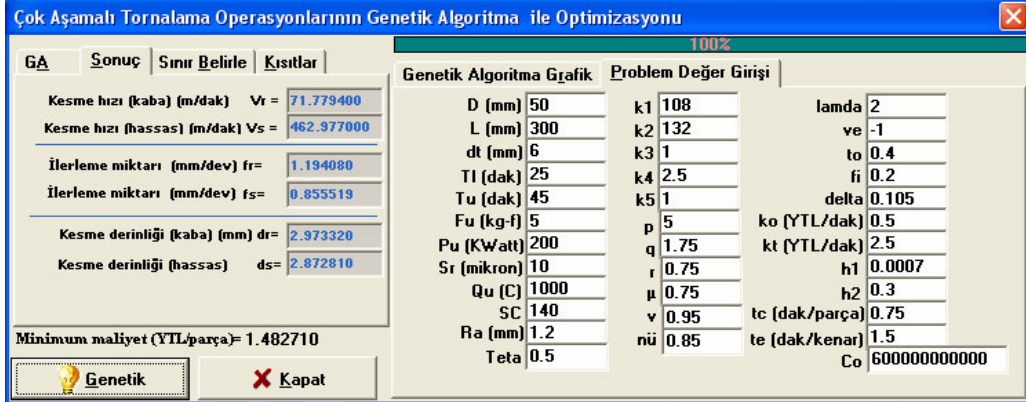
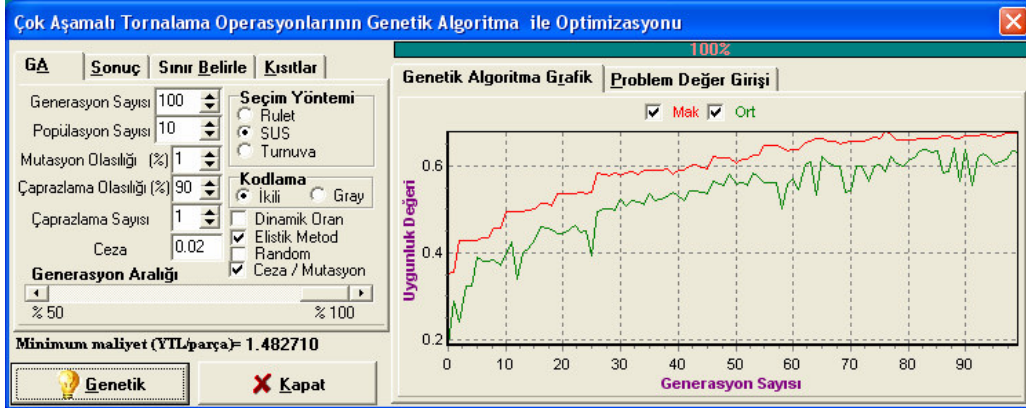
Şekil IV.8 $f_r=0,8$ için parça başına maliyet değeri hesaplaması



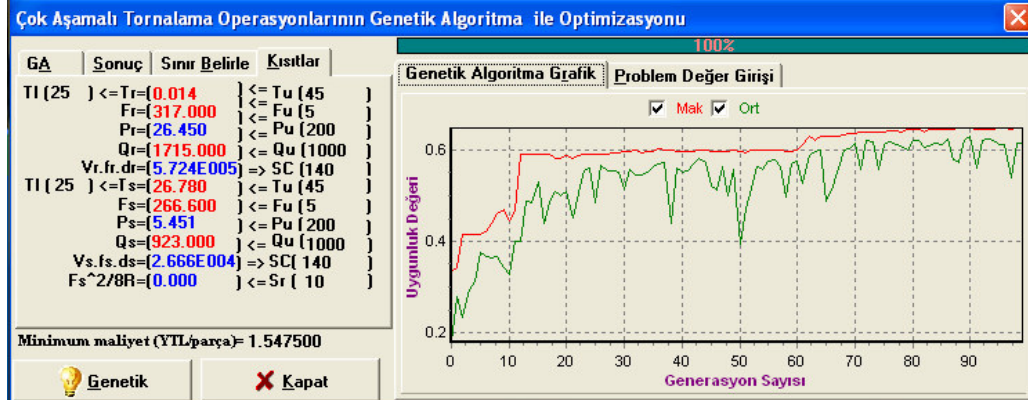
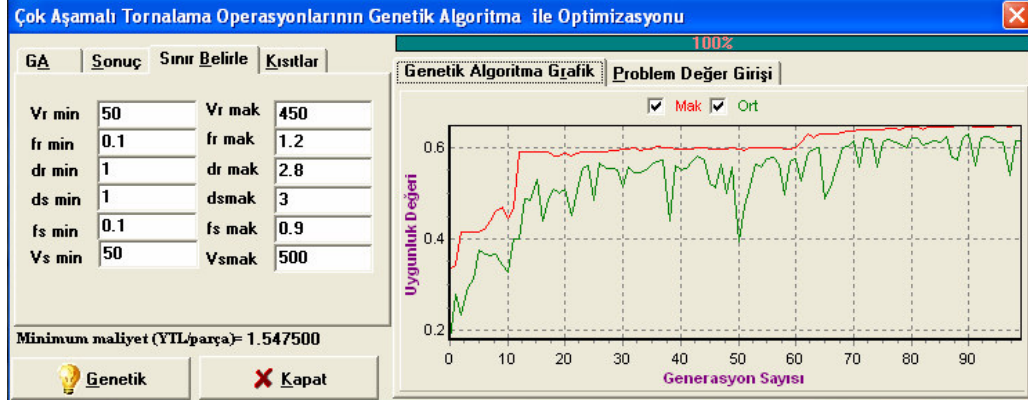
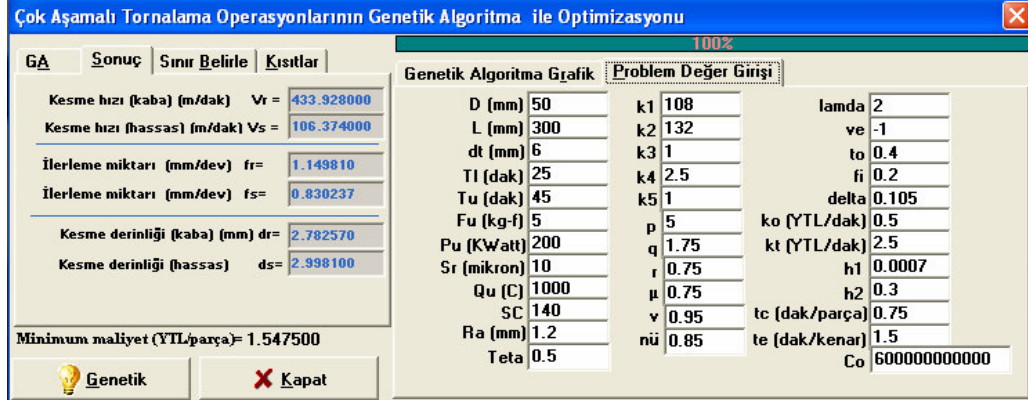
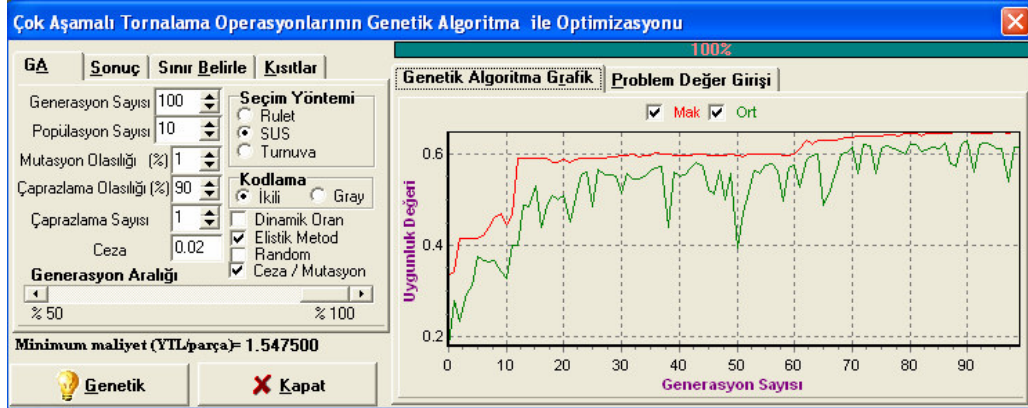
Şekil IV.9 $f_r=1,0$ için parça başına maliyet değeri hesaplaması



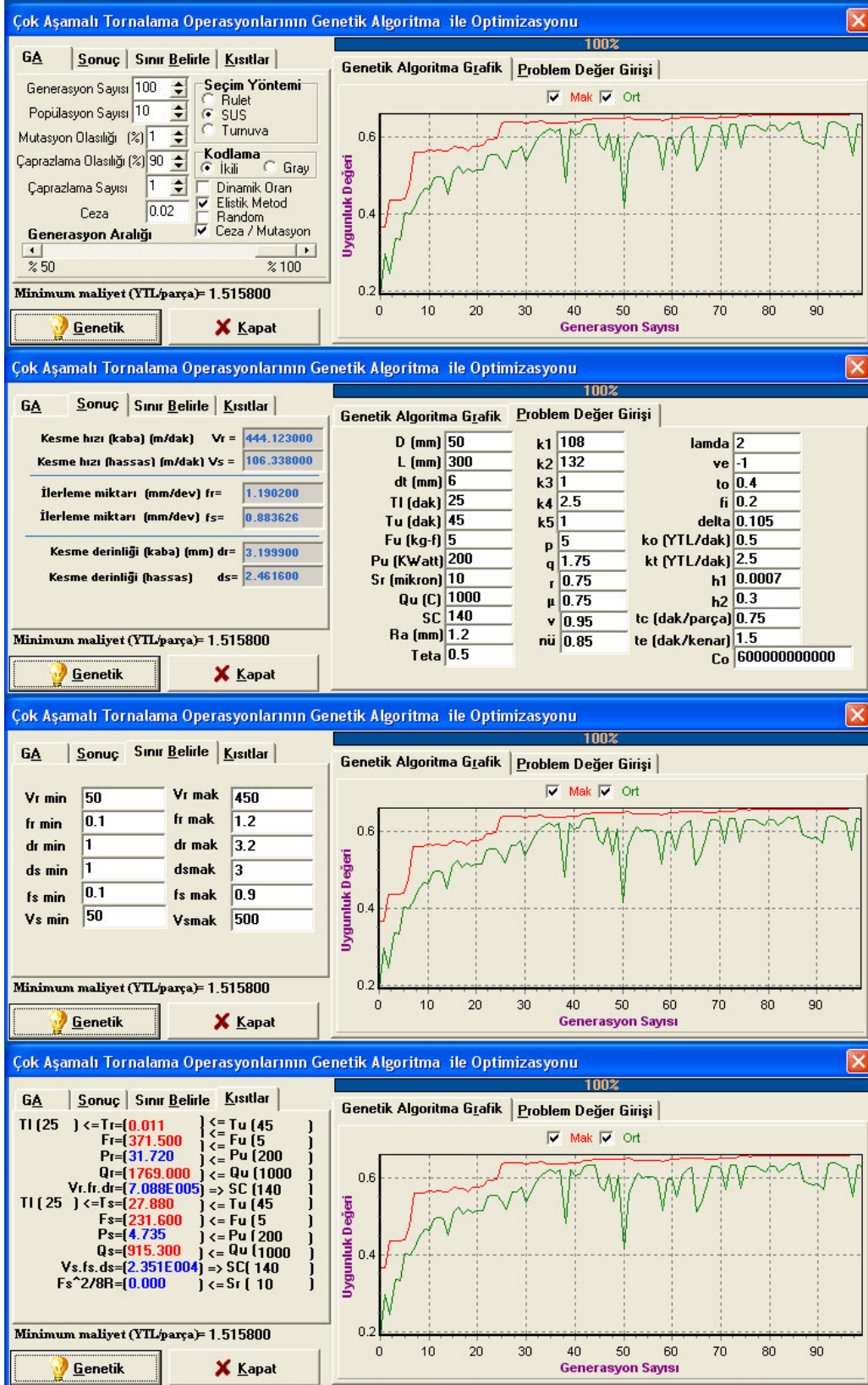
Şekil IV.10 $f_r=1,1$ için parça başına maliyet değeri hesaplaması



Şekil IV.11 $f_r=1,2$ için parça başına maliyet değeri hesaplaması



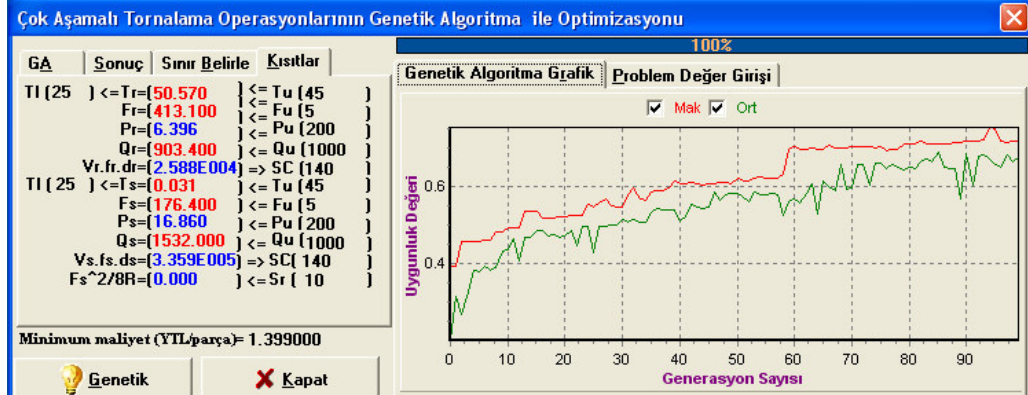
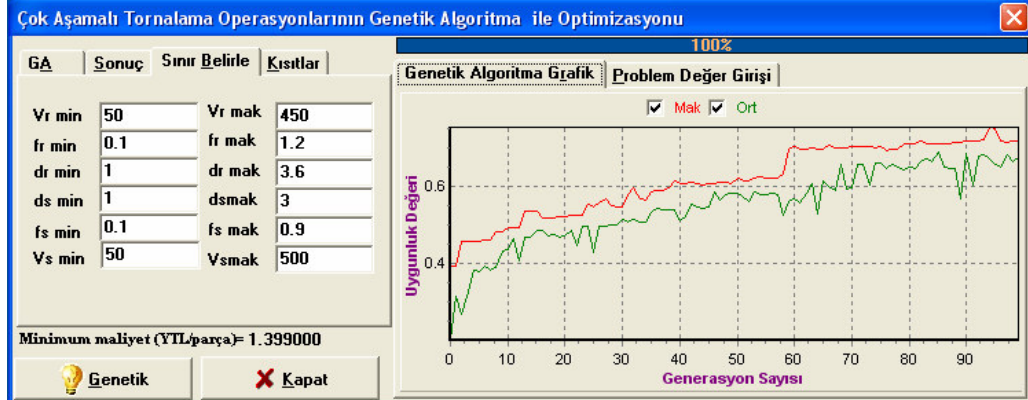
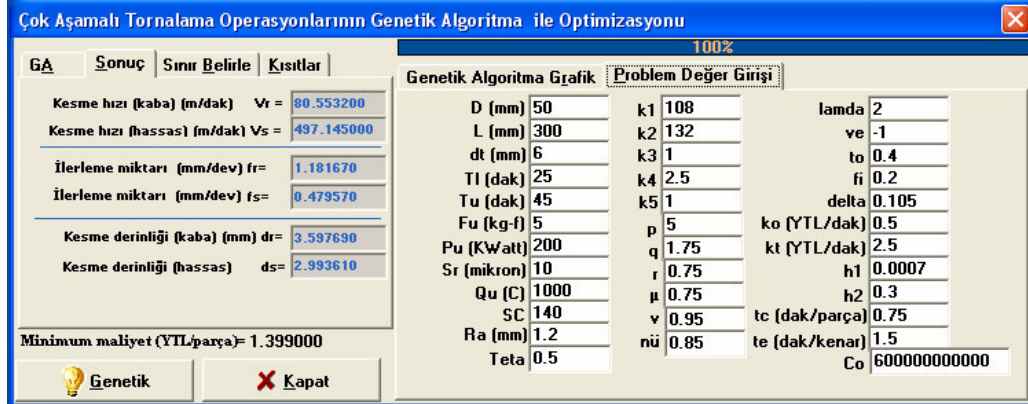
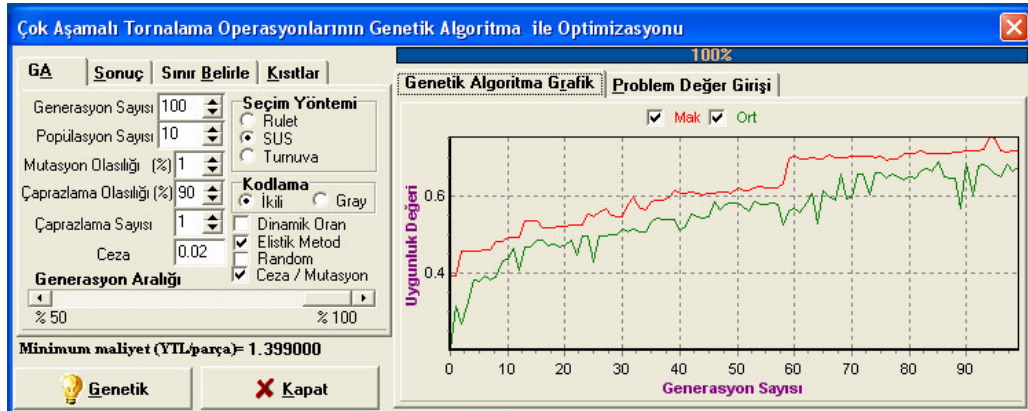
Şekil IV.12 $d_r=2,8$ için parça başına maliyet değeri hesaplaması



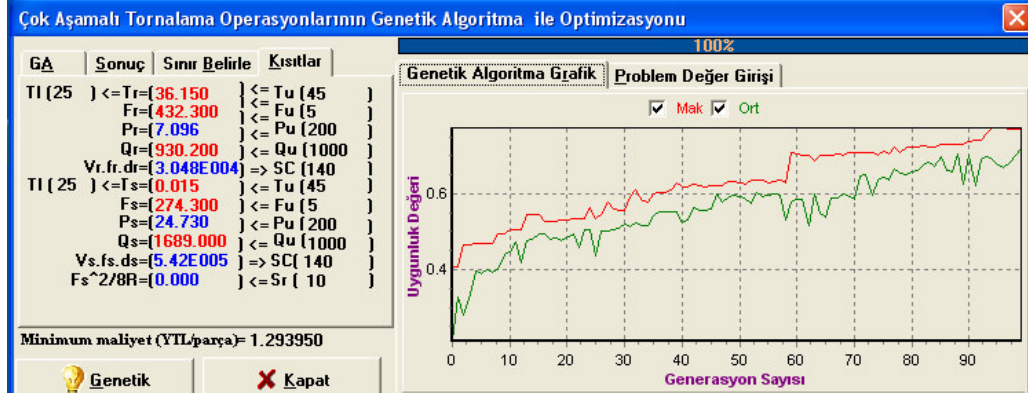
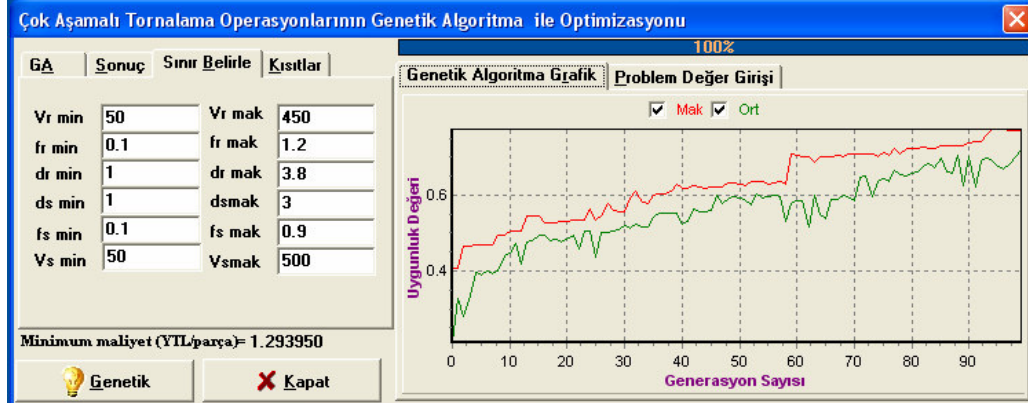
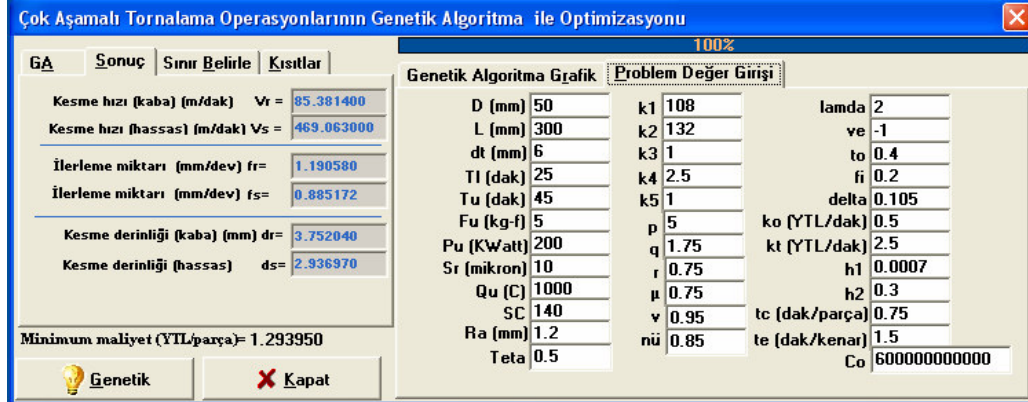
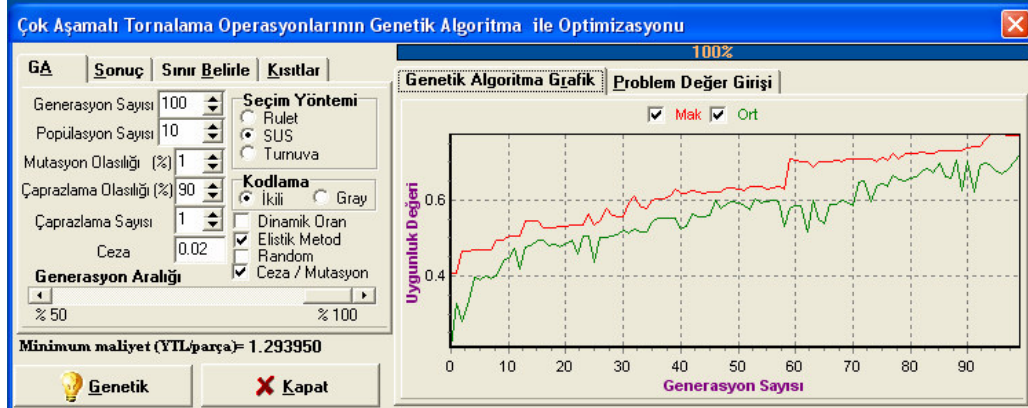
Şekil IV.13 $d_r=3,2$ için parça başına maliyet değeri hesaplaması



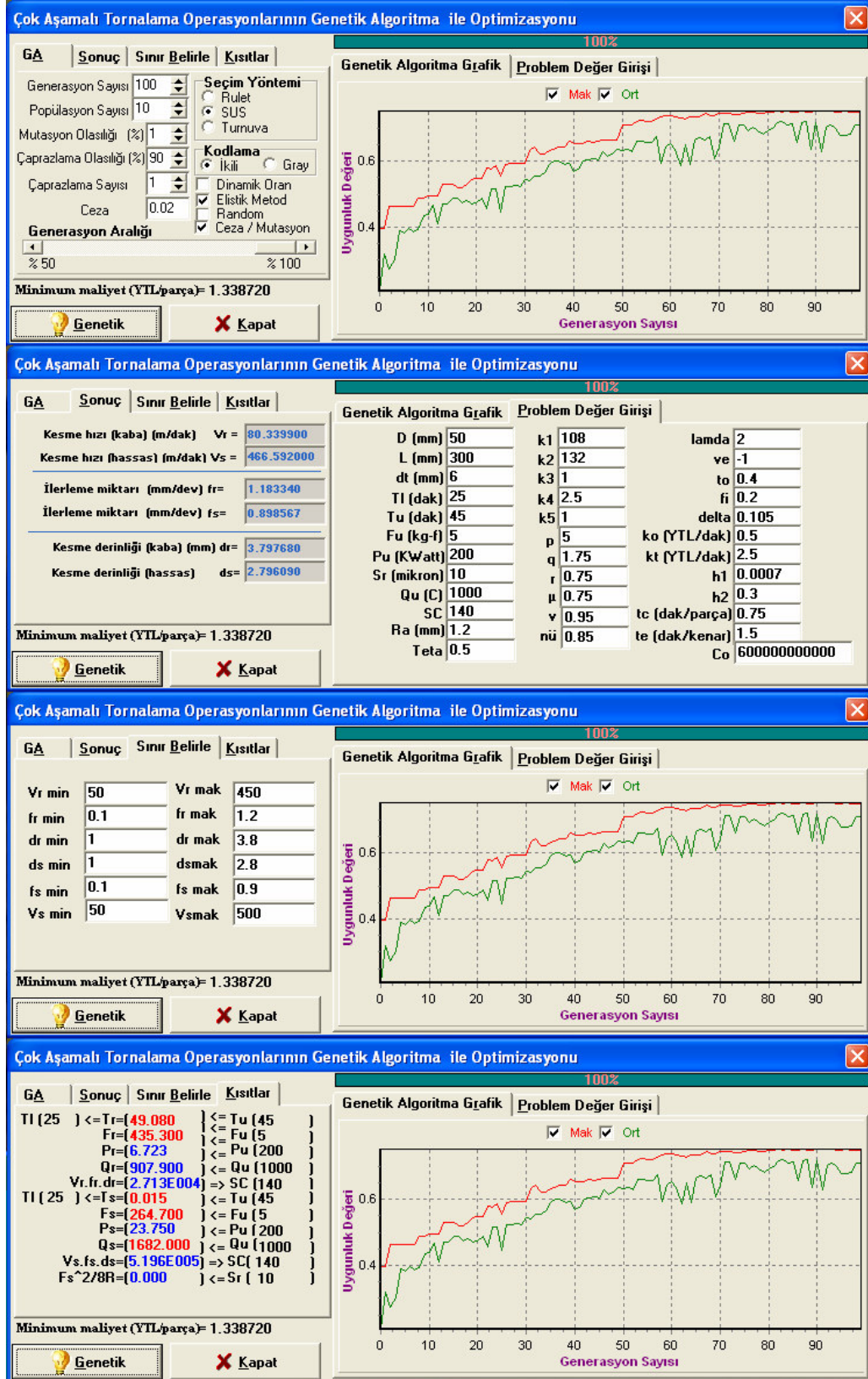
Şekil IV.14 d_r=3,4 için parça başına maliyet değeri hesaplaması



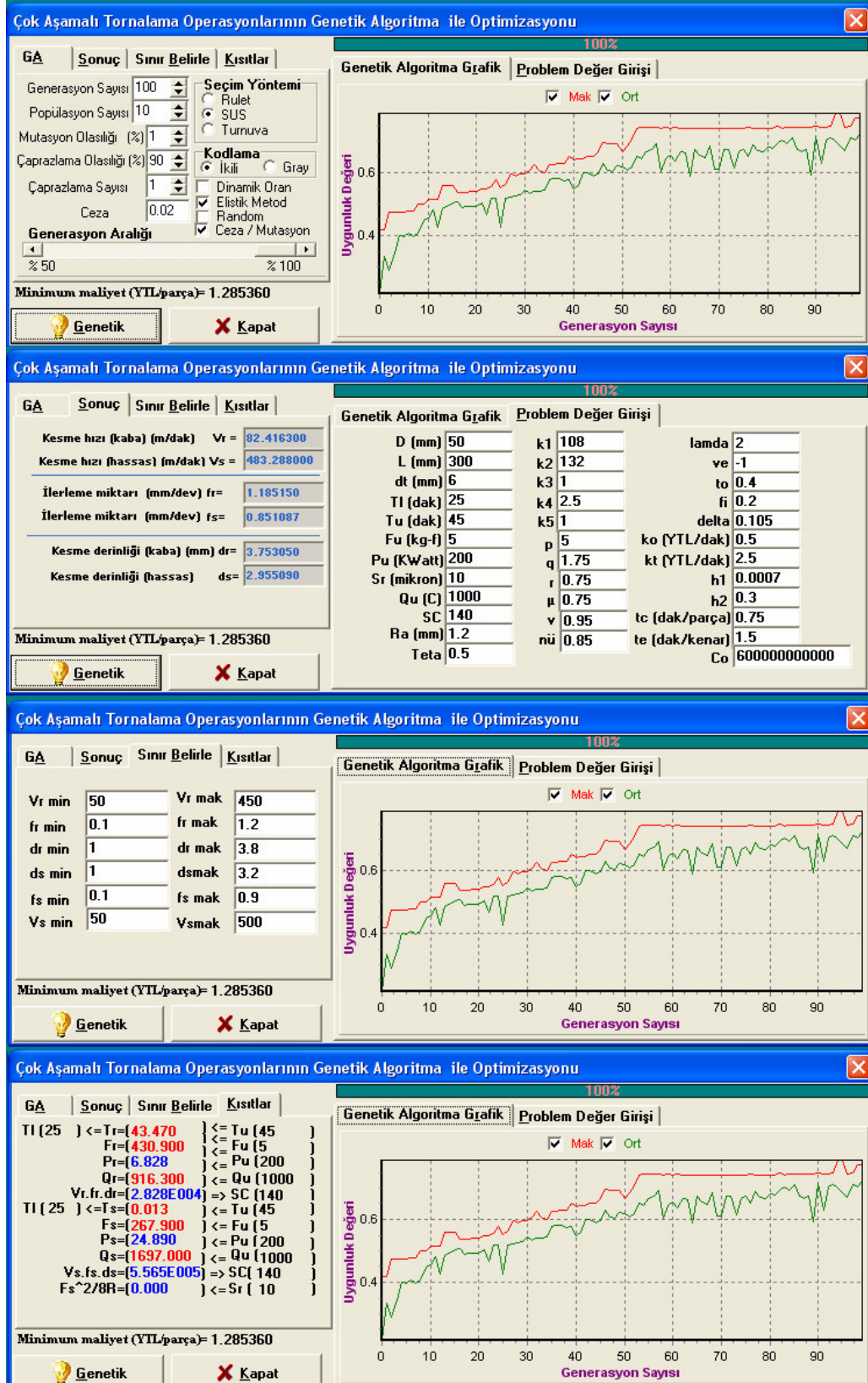
Şekil IV.15 $d_r=3,6$ için parça başına maliyet değeri hesaplaması



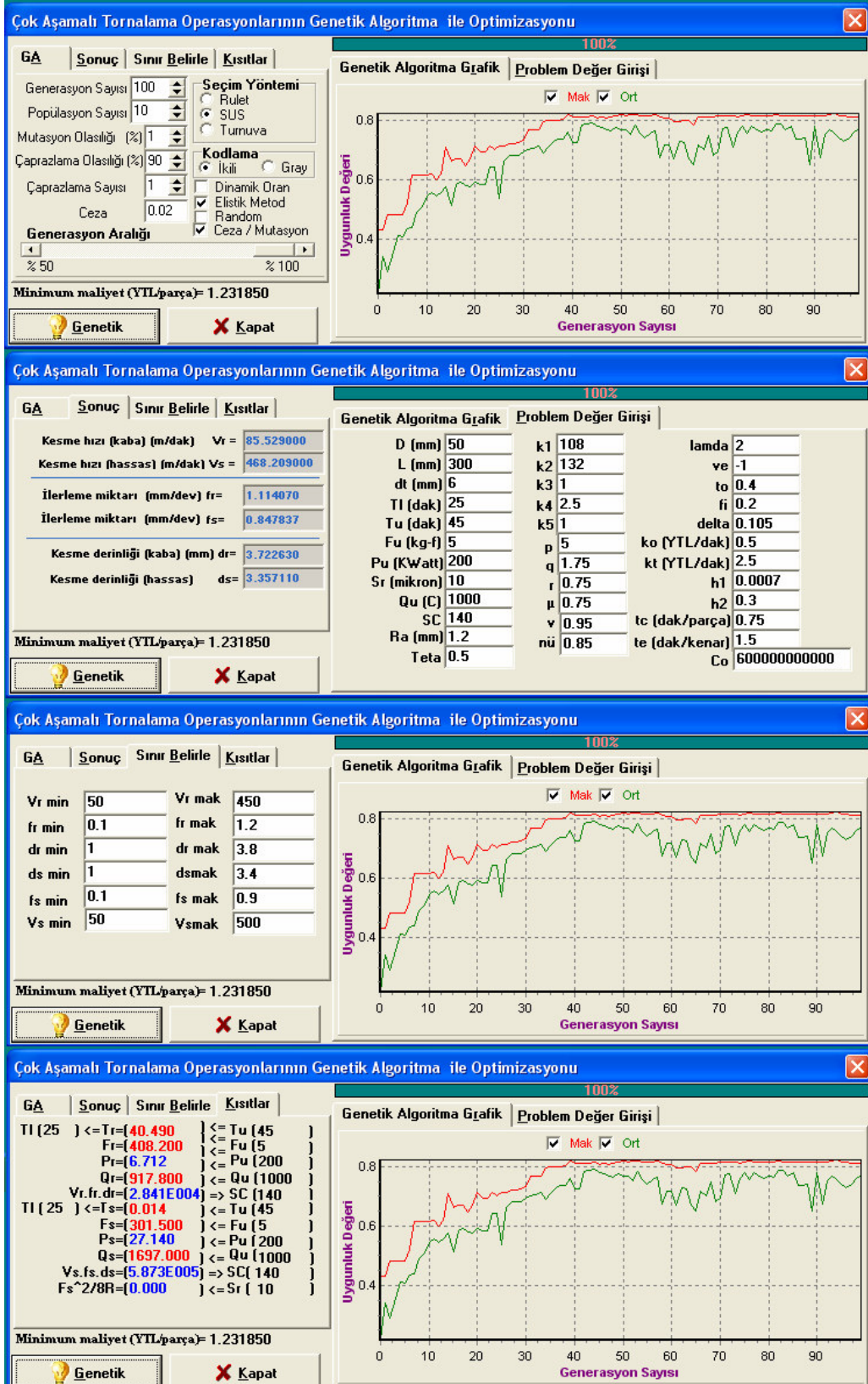
Şekil IV.16 d_r=3,8 için parça başına maliyet değeri hesaplaması



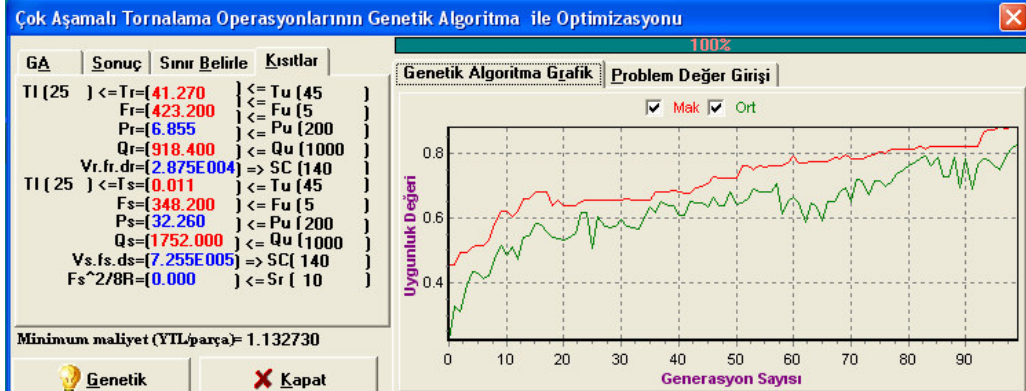
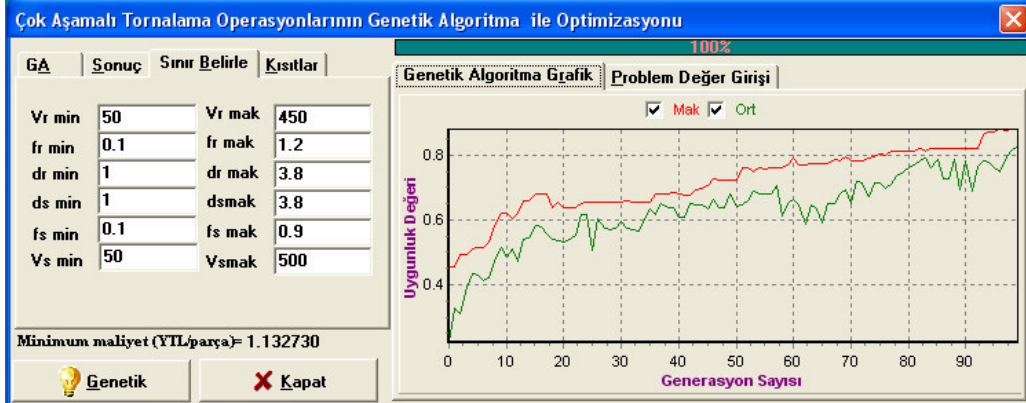
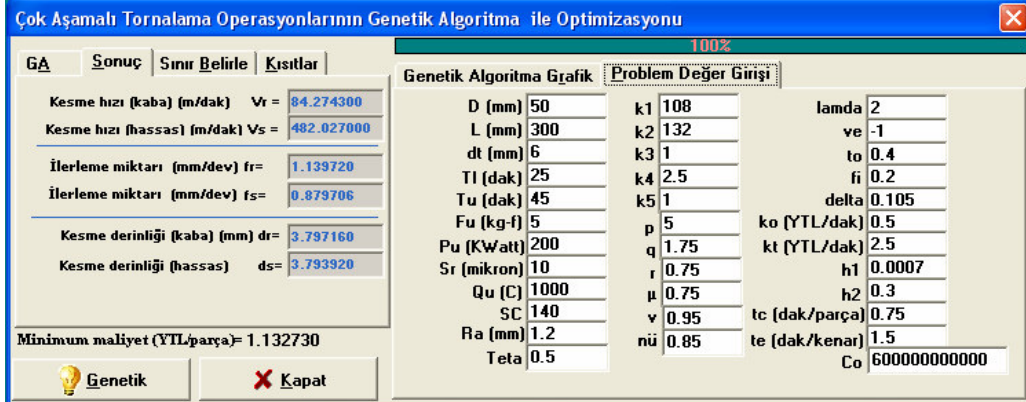
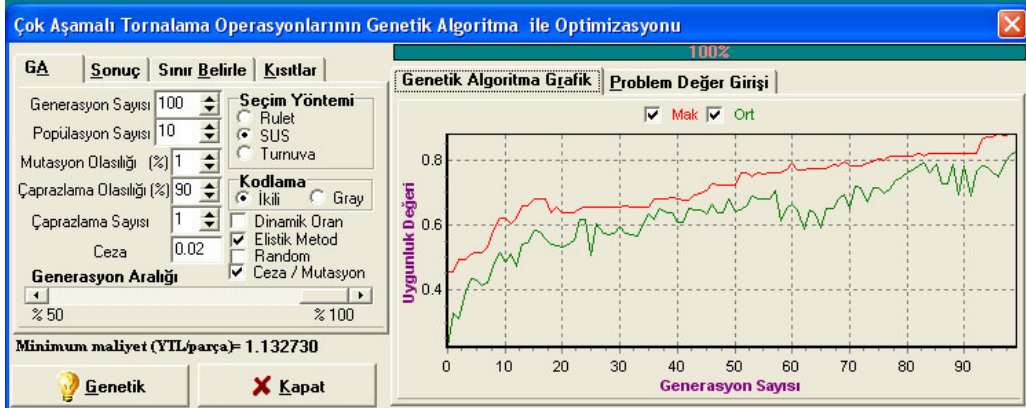
Şekil IV.17 $d_c=2,8$ için parça başına maliyet değeri hesaplaması



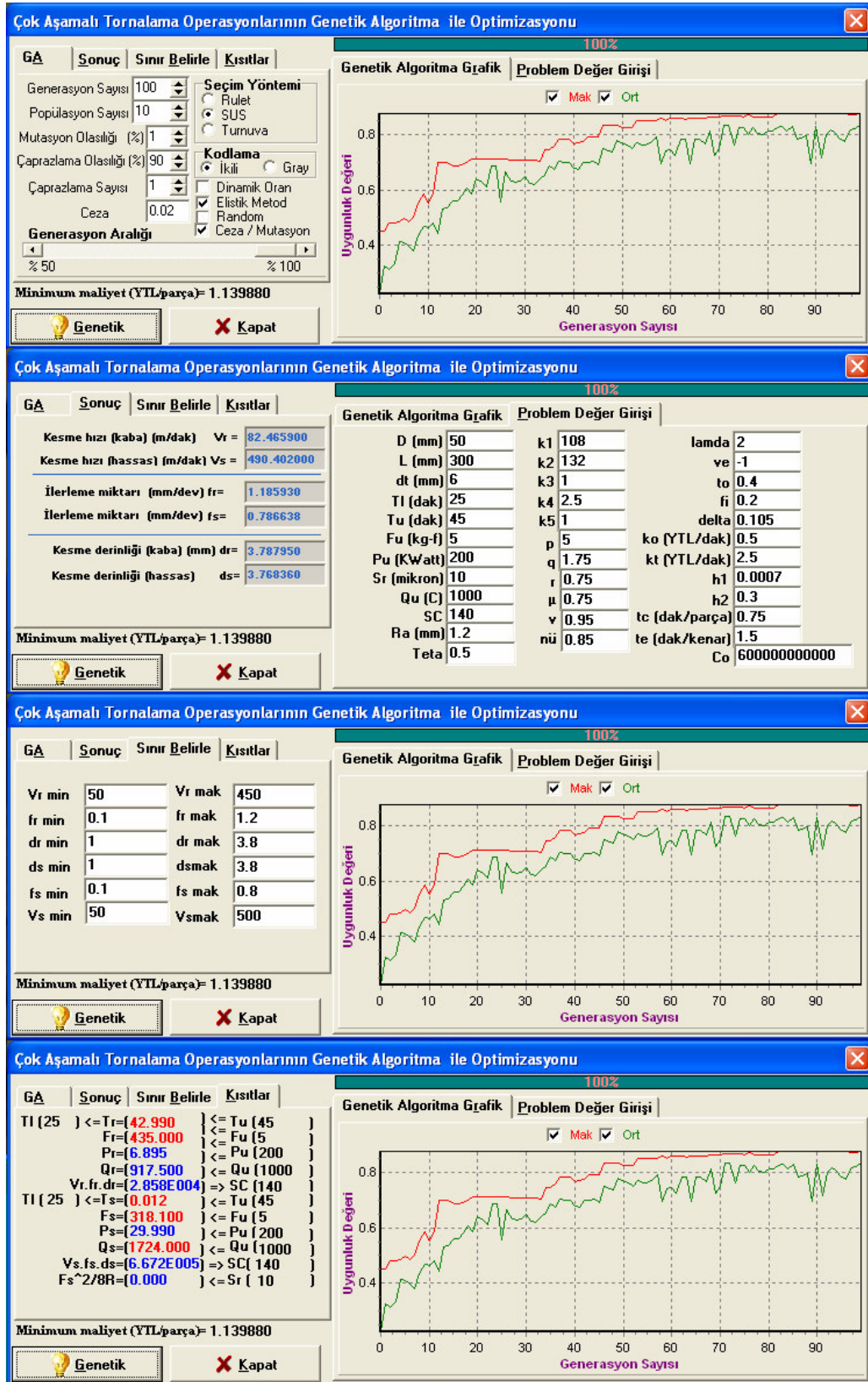
Şekil IV.18 $d_c=3,2$ için parça başına maliyet değeri hesaplaması



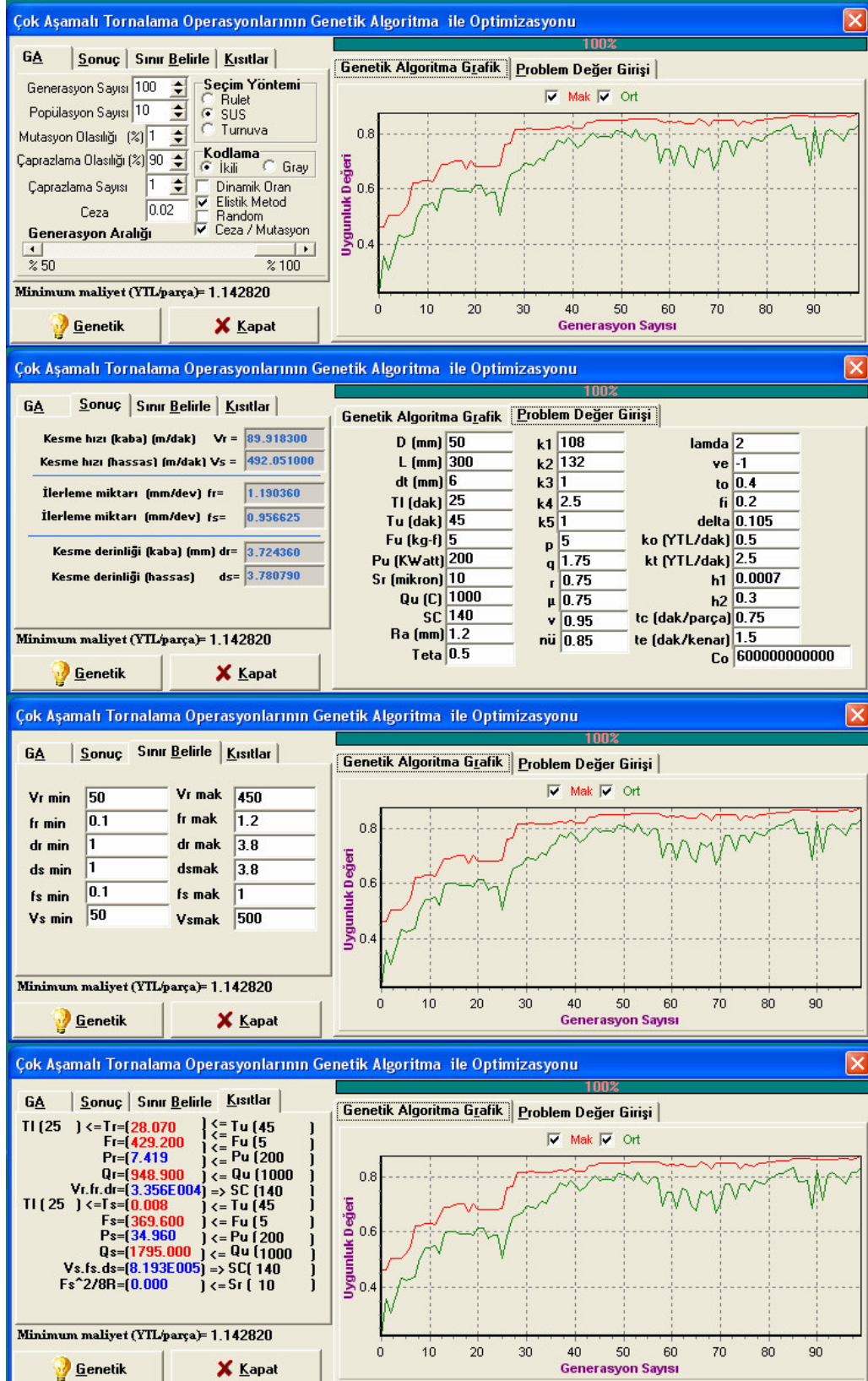
Şekil IV.19 $d_s=3,4$ için parça başına maliyet değeri hesaplaması



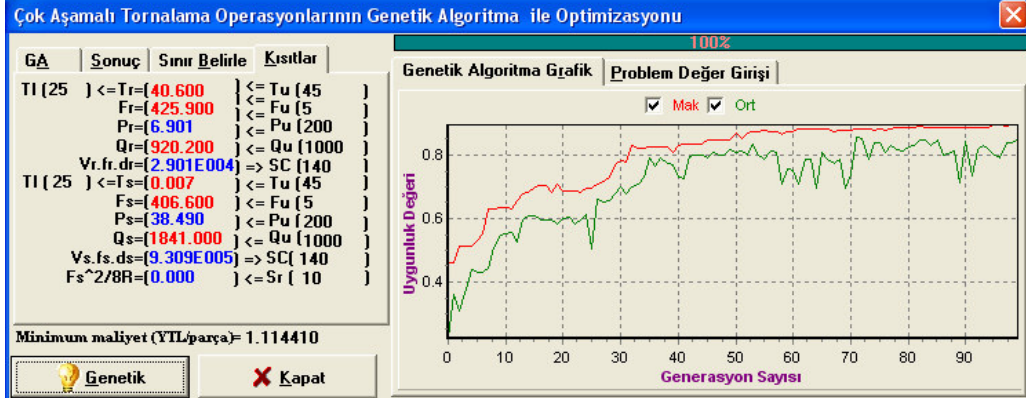
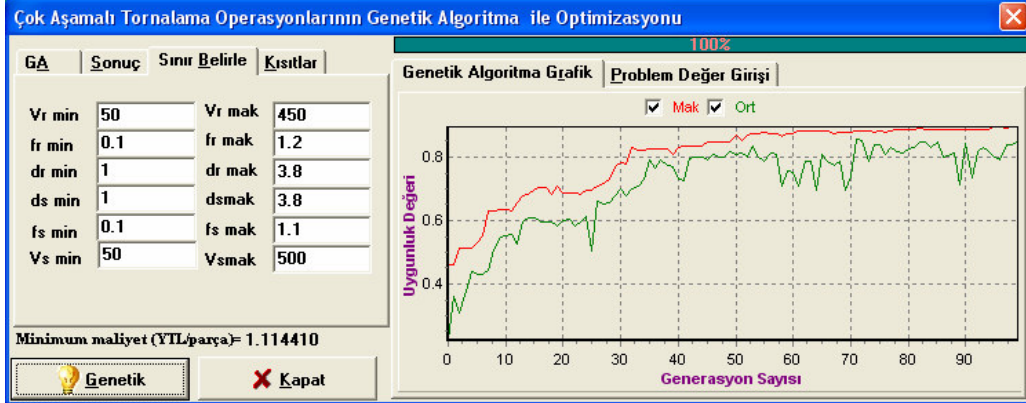
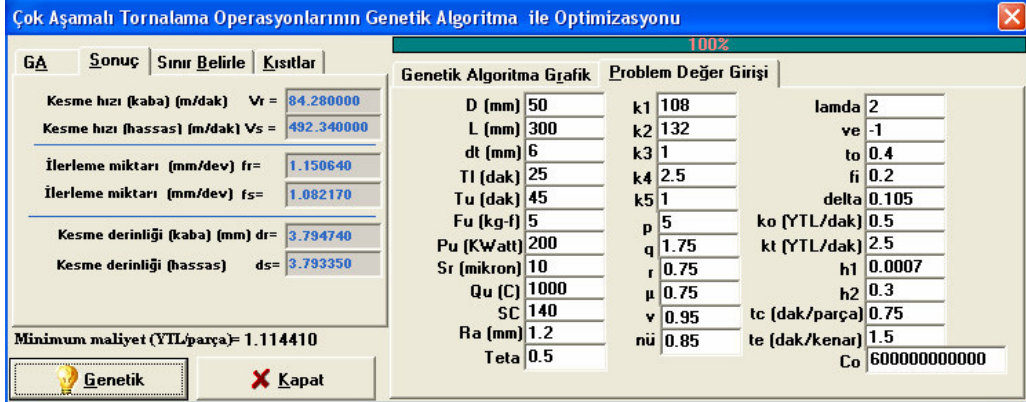
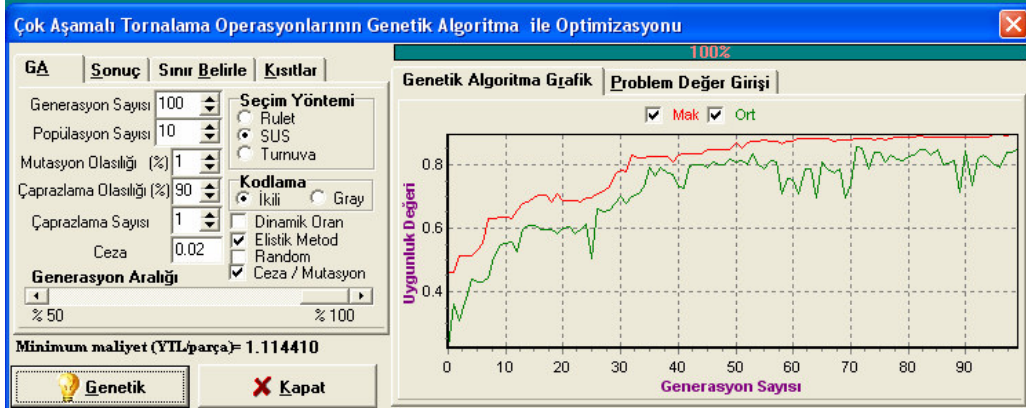
Şekil IV.20 $d_s=3,8$ için parça başına maliyet değeri hesaplaması



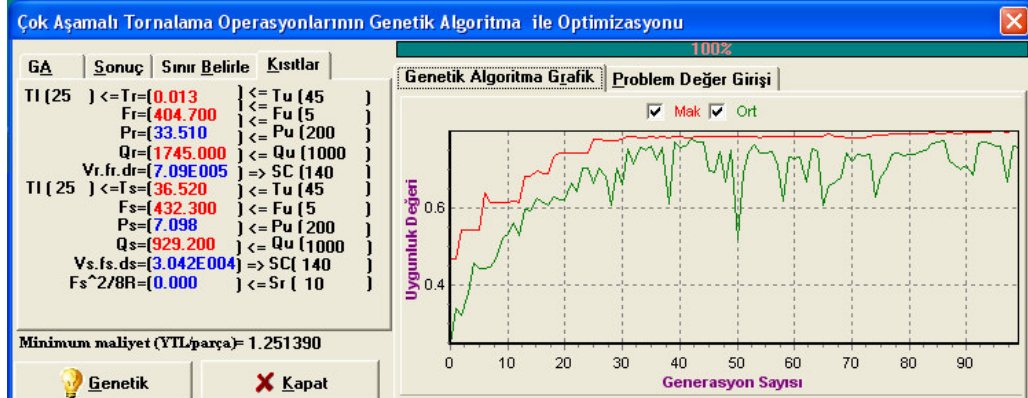
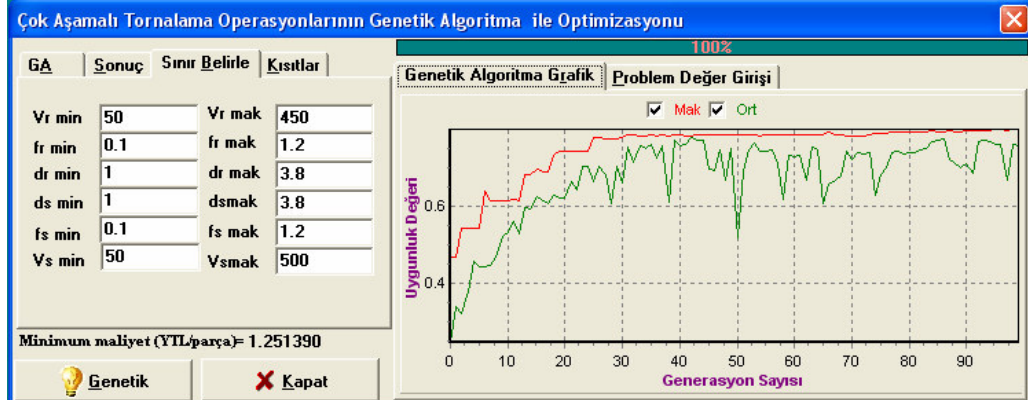
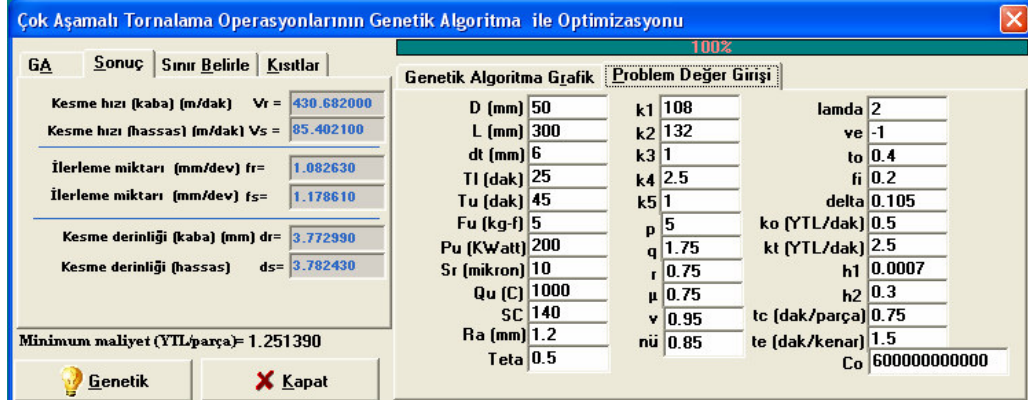
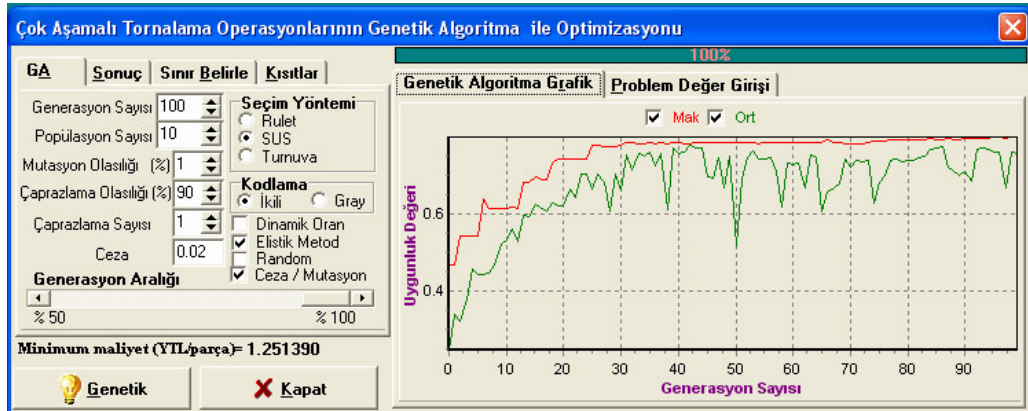
Şekil IV.21 $f_s=0,8$ için parça başına maliyet değeri hesaplaması



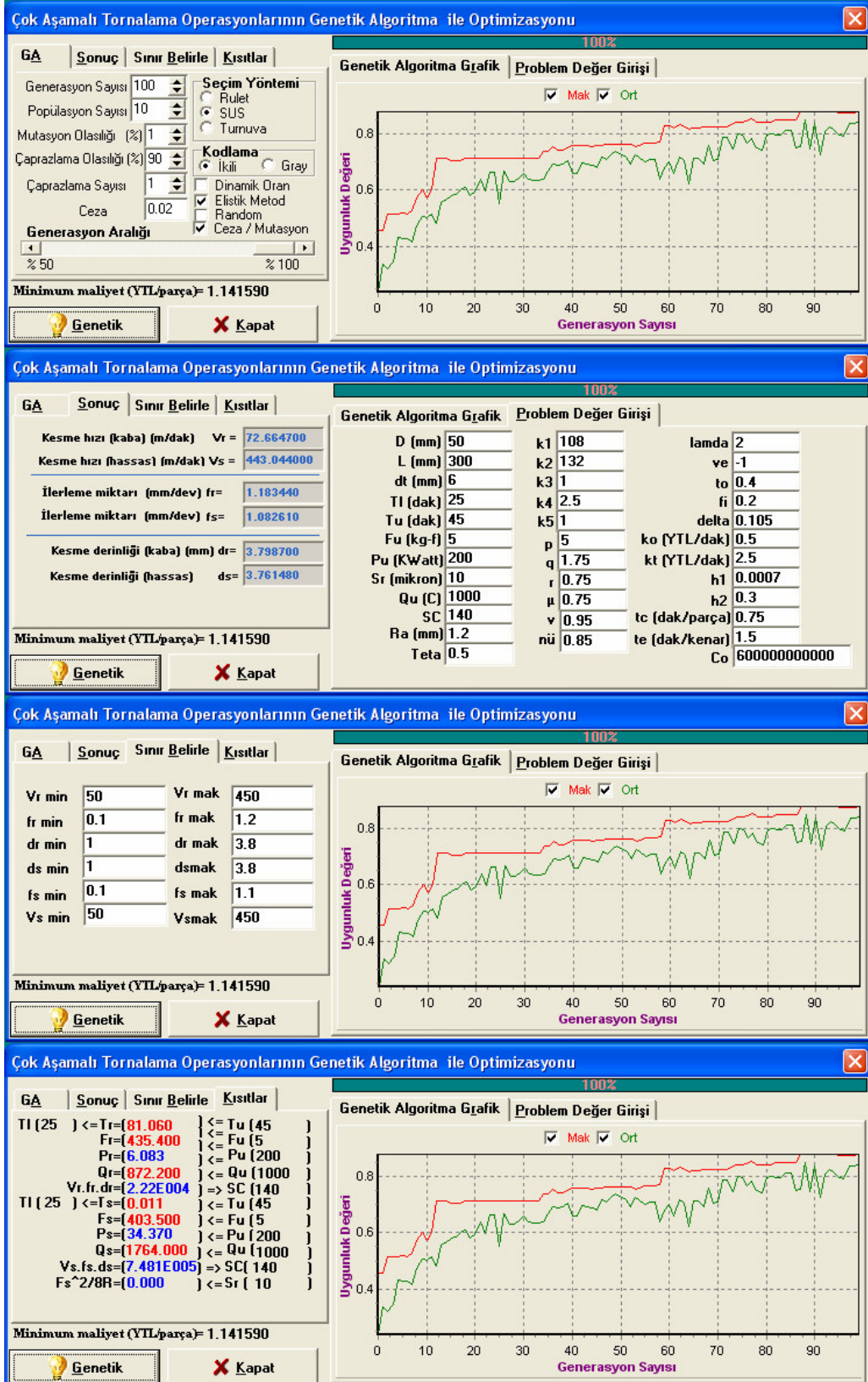
Şekil IV. 22 $f_s=1,0$ için parça başına maliyet değeri hesaplaması



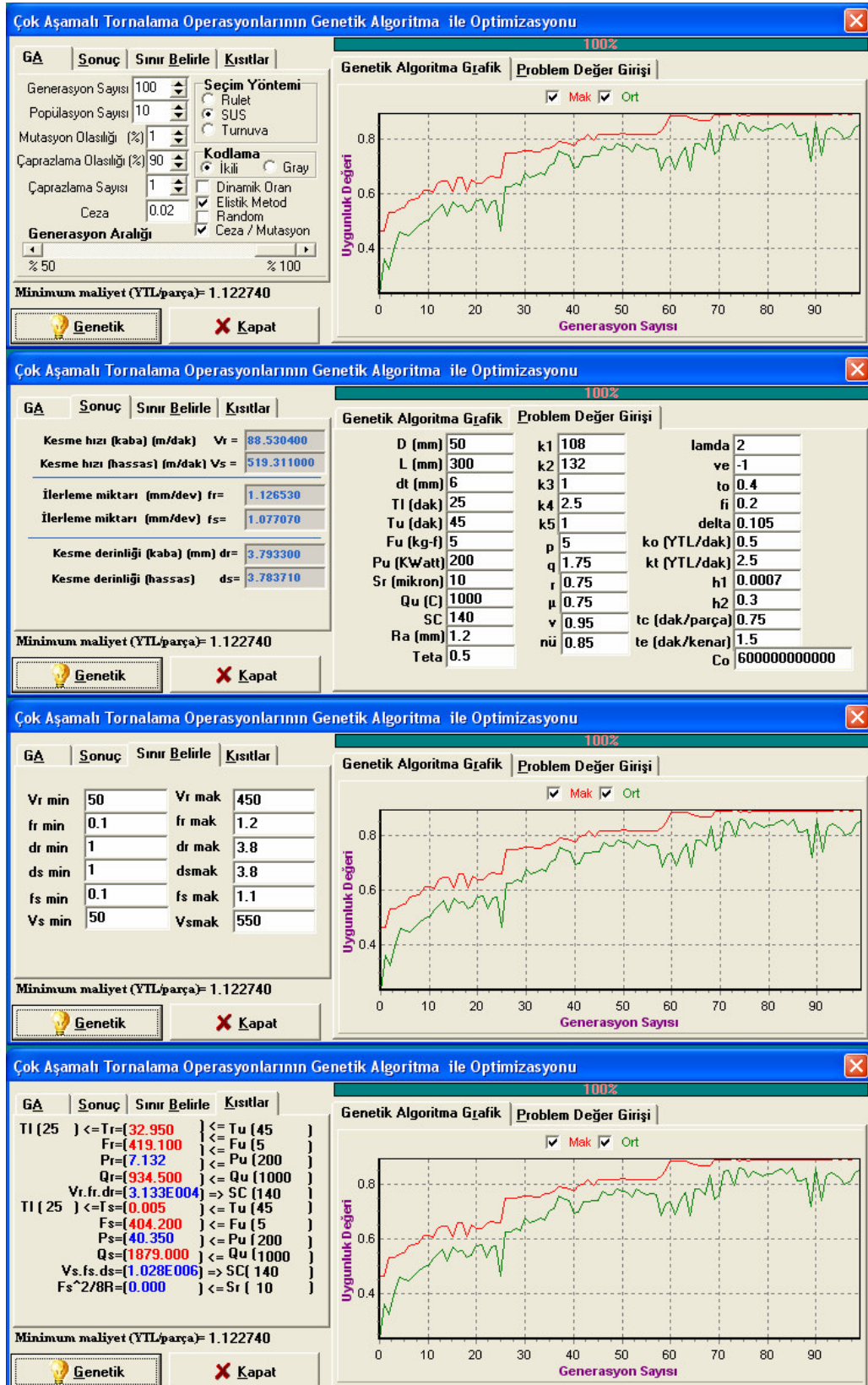
Şekil IV.23 $f_s=1,1$ için parça başına maliyet değeri hesaplaması



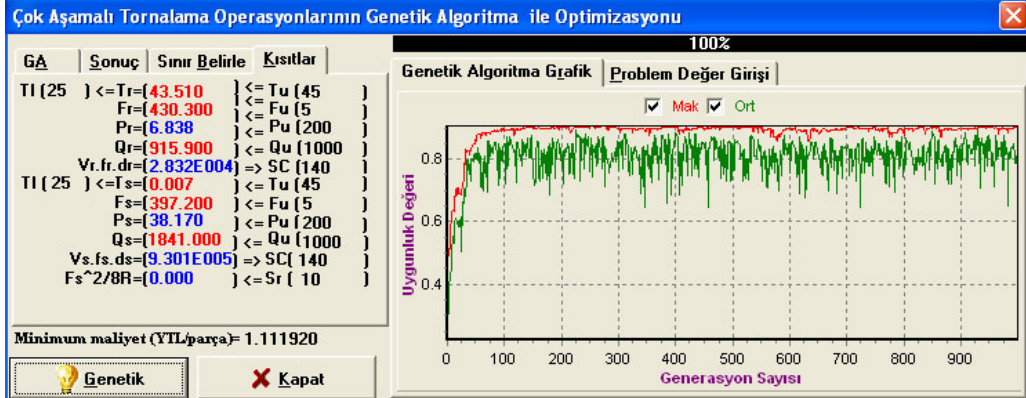
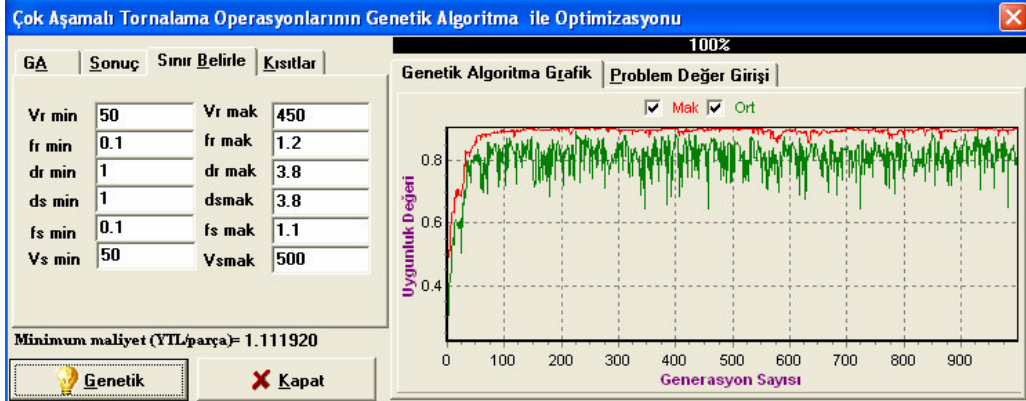
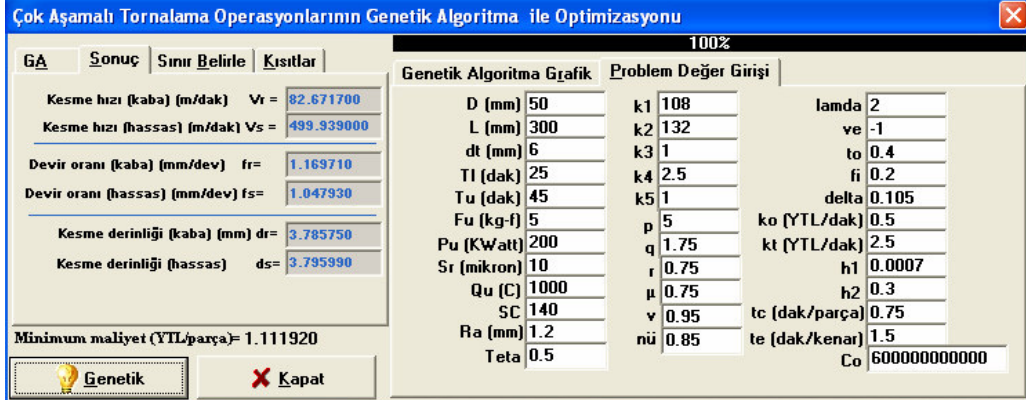
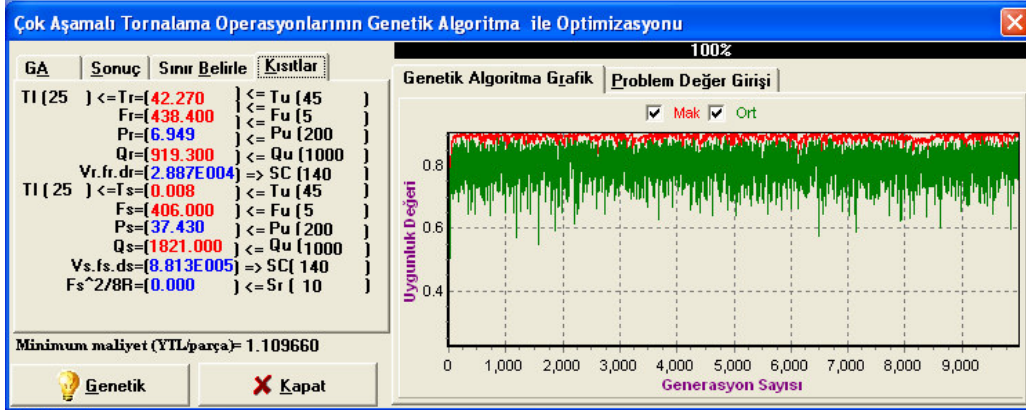
Şekil IV.24 $f_s=1,2$ için parça başına maliyet değeri hesaplaması



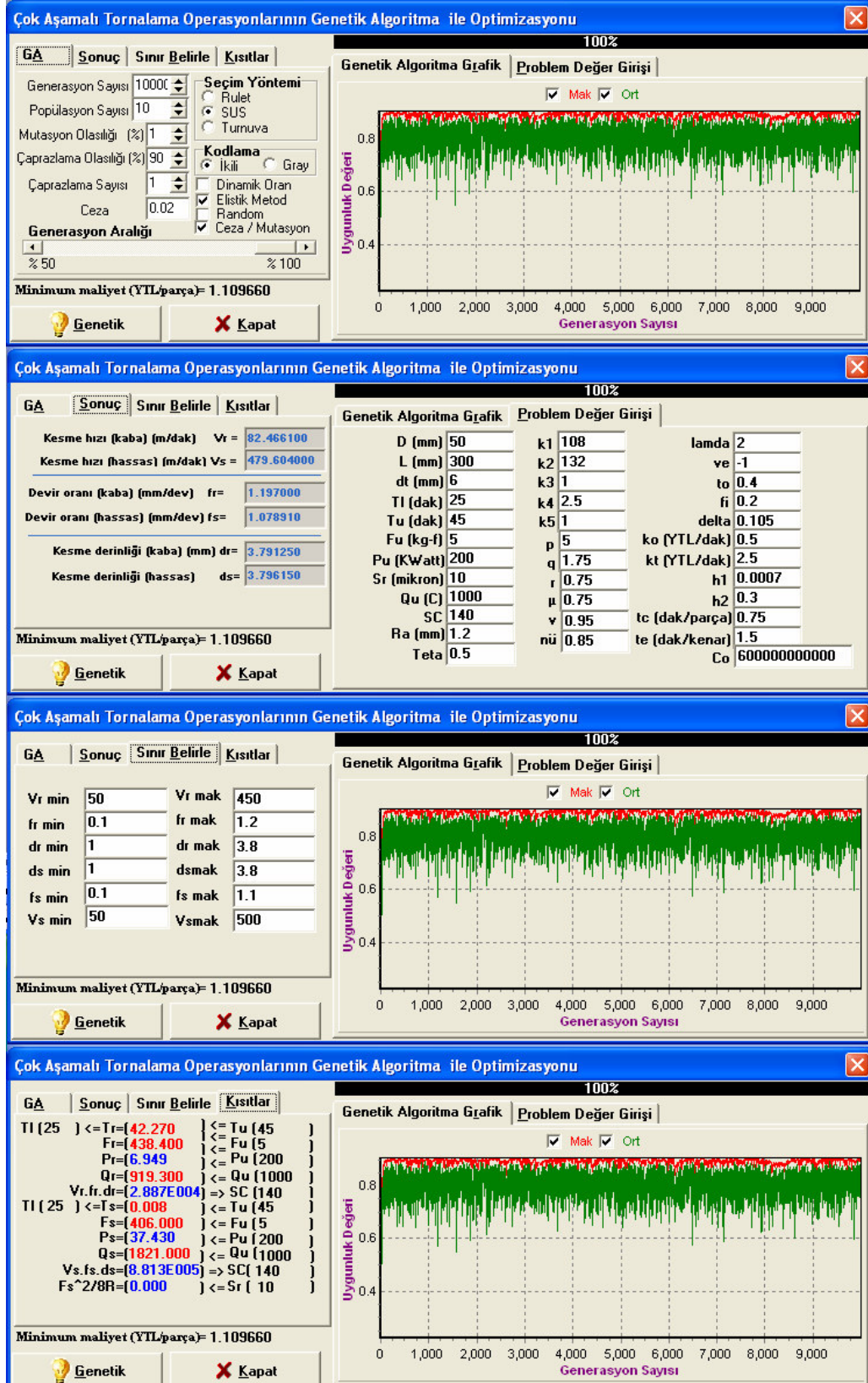
Şekil IV.25 $V_s=450$ için parça başına maliyet değeri hesaplaması



Şekil IV.26 $V_s=550$ için parça başına maliyet değeri hesaplaması



Şekil IV.27 Generasyon sayısı 1000 için parça başına maliyet değeri hesaplaması



Şekil IV.28 Generasyon sayısı 10000 için parça başına maliyet değeri hesaplaması

Problemde verilen parça ebatlarıyla ilgili değişiklikler yapıldığın da parça başına maliyet değerinde değişiklikler meydana gelmektedir. Örneğimizde parça boyu L=300 mm alınmıştır. Eğer ki parça boyunu uzatırsak maliyet artacaktır. L= 325 mm olarak düşünürsek, Şekil IV.23'te son olarak elde ettiğimiz sınır değerleri dikkate alındığında maliyet değerimiz 1,11 YTL'den 1,15 YTL'ye yükselmektedir (Şekil IV.29). Aynı şekilde, parçanın yarıçapını d=55 alırsak, parça maliyeti 1,14 YTL'ye yükselmektedir (Şekil IV.30).

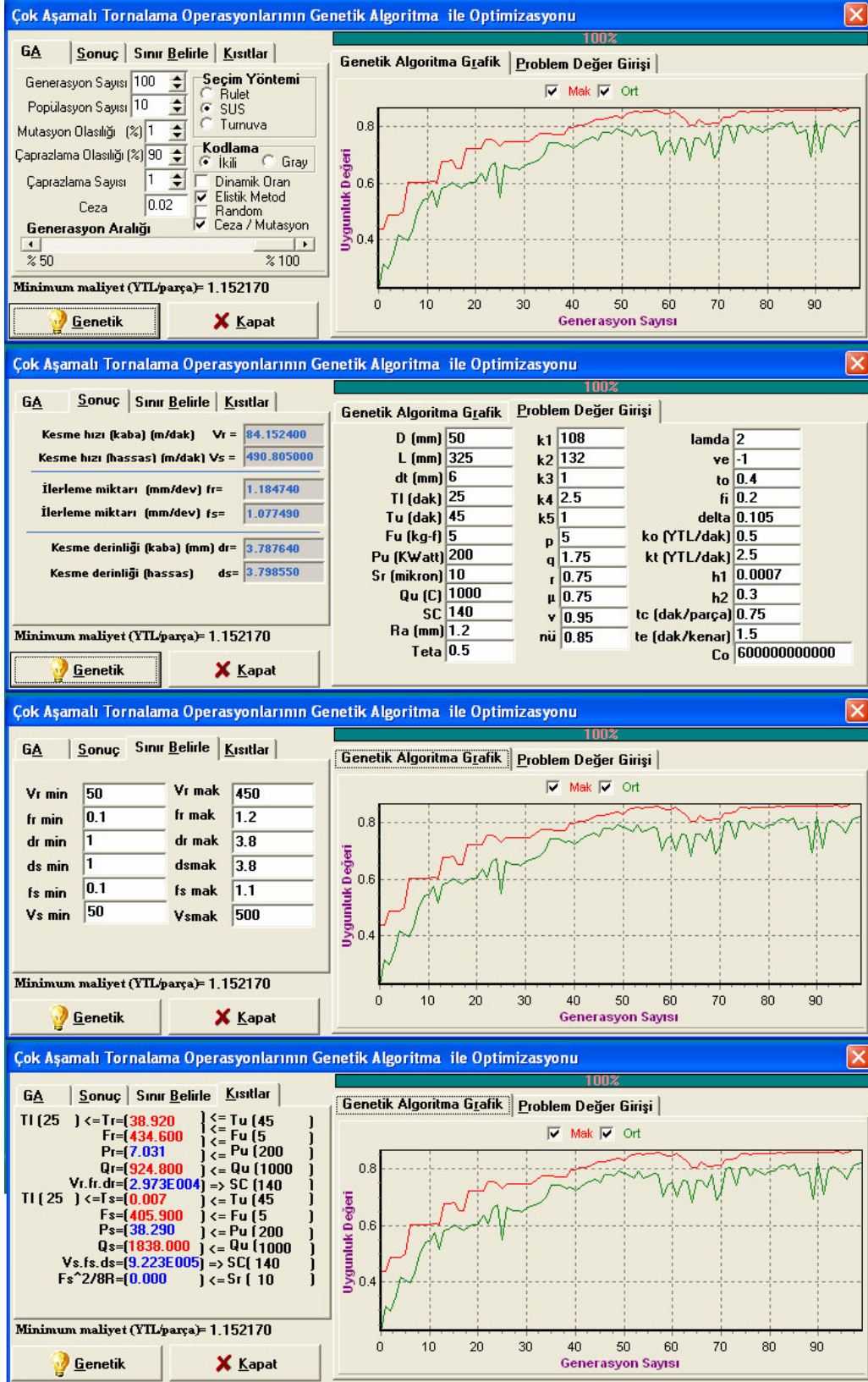
Tablo IV.2 Parça ebadına bağlı değişimlerde ortaya çıkan maliyet değerleri

Kıstas	Değer	Parça Başına Maliyet Değeri	Şekil no
L	325 mm	1,15 YTL	Şekil IV.29
D	55 mm	1,14 YTL	Şekil IV.30

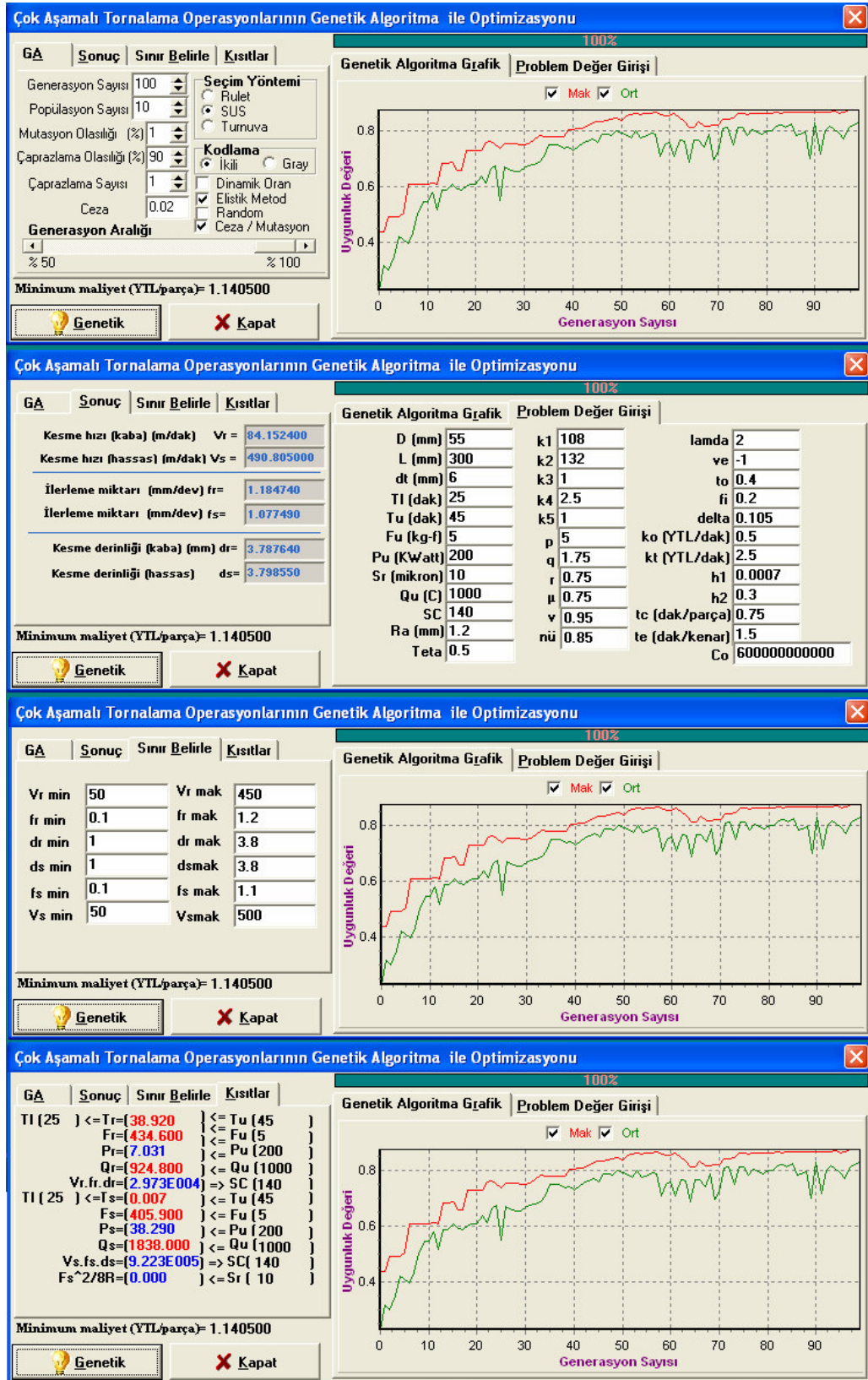
Programda uyguladığımız genetik algoritmanın seçim yöntemi ile ilgili de değişikliklerin sonuçlara etkisi vardır. Şekil IV.23'de bize en uygun maliyet değerini veren sınır değerlerini esas alarak seçim yönetimini değiştirdiğimizde karşılaştığımız sonuçlar şöyledir:

Tablo IV.3 Genetik Algoritma seçim yöntemi ve kodlamalarındaki değişimlere ait maliyet değerleri

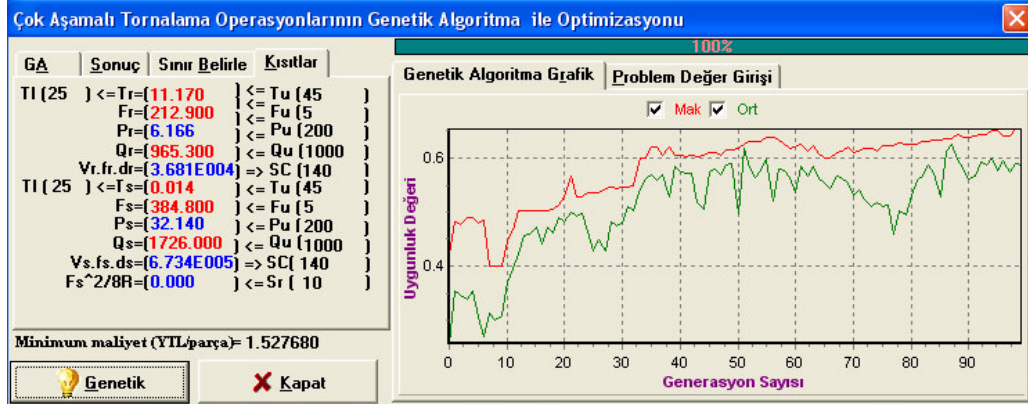
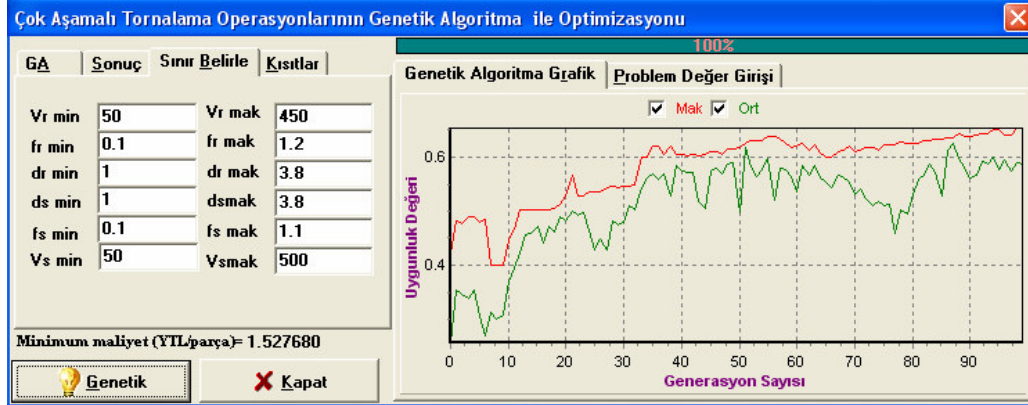
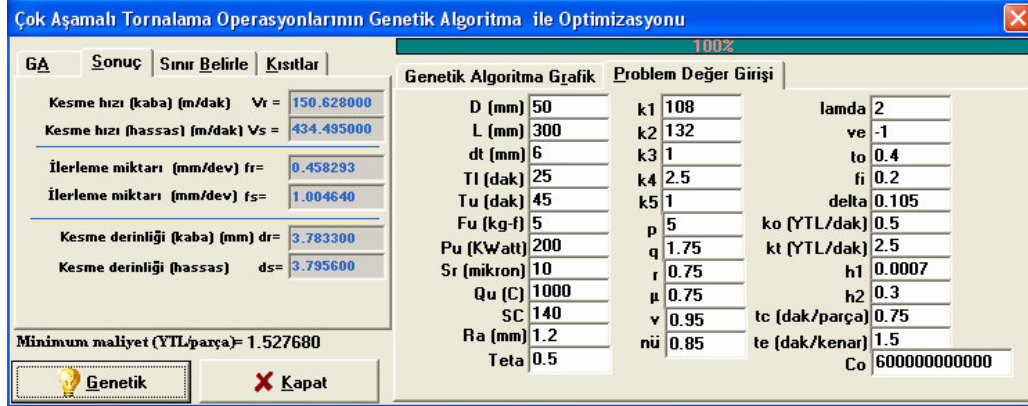
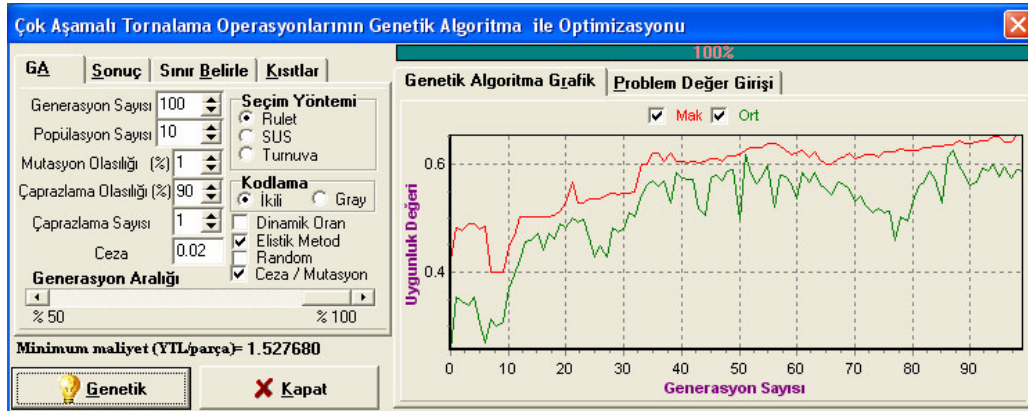
Seçim Yöntemi	Parça Başına Maliyet Değeri	Şekil no
Rulet	1,52 YTL	Şekil IV.31
SUS	1,11 YTL	Şekil IV.23
Turnuva	1,12 YTL	Şekil IV.32
İkili Alfabe	1,11 YTL	Şekil IV.23
Gray Kodlama	1,13 YTL	Şekil IV.33
Dinamik Oran	1,24 YTL	Şekil IV.34
Elastik Mod	1,11 YTL	Şekil IV.23
Random	1,35 YTL	Şekil IV.35



Şekil IV.29 L=325 mm için parça başına maliyet değeri hesaplaması



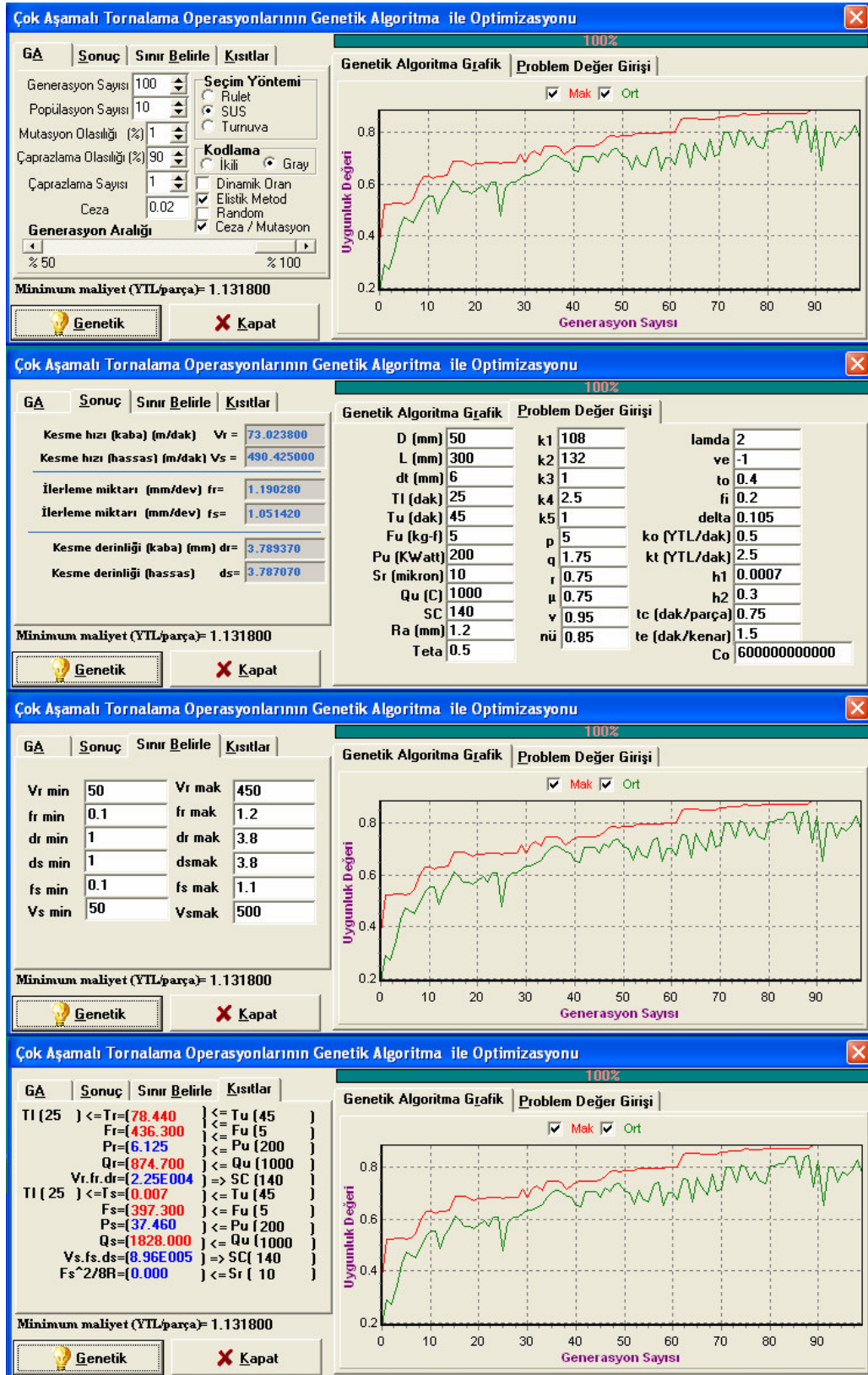
Şekil IV.30 d=55 mm için parça başına maliyet değeri hesaplaması



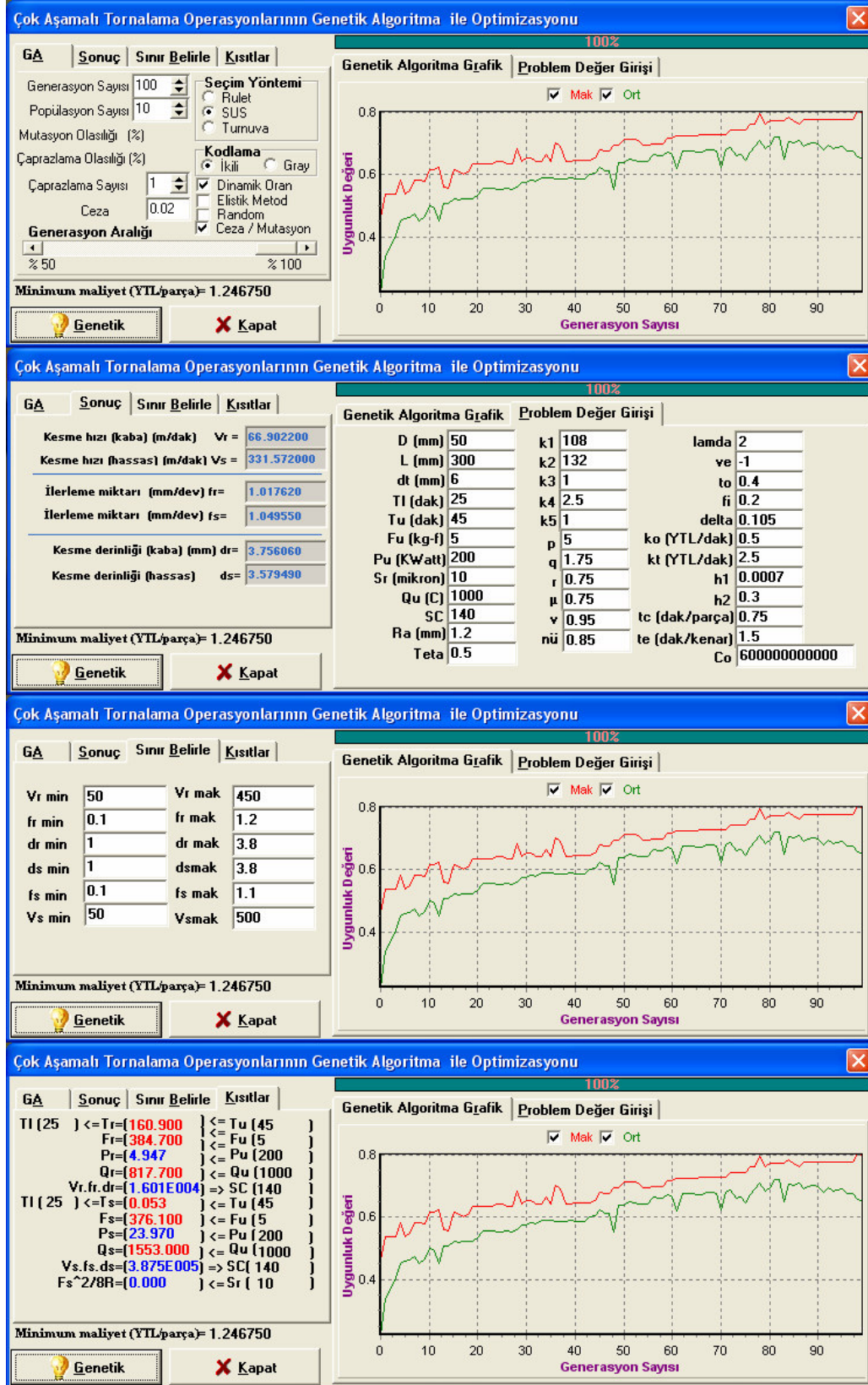
Şekil IV.31 Rulet seçim yöntemi için parça başına maliyet değeri hesaplaması



Şekil IV.32 Turnuva seçim yöntemi için parça başına maliyet değeri hesaplaması



Şekil IV.33 Gray kodlama ile parça başına maliyet değeri hesaplaması



Şekil IV.34 Dinamik Oran ile parça başına maliyet değeri hesaplaması



Şekil IV.35 Random ile parça başına maliyet değeri hesaplaması

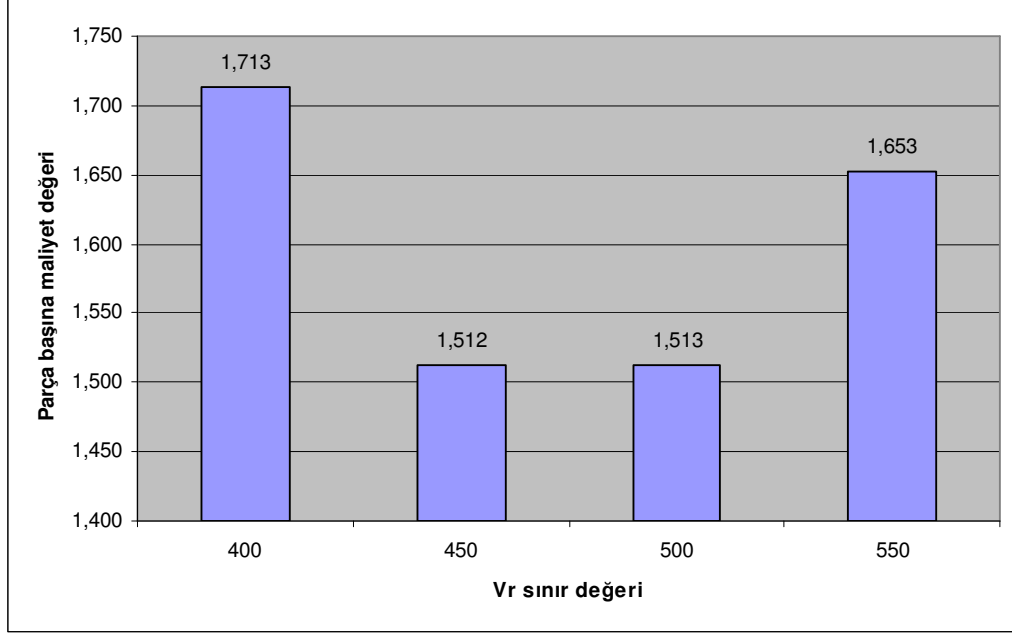
IV.3 SONUÇ

Genetik Algoritmalar heuristic, yani tahminlere dayalı sonuç veren bir optimizasyon algoritmasıdır. Yukarıda yaptığım analizlerde farklı genetik algoritma yöntemlerini deneme yanılma yoluyla seçerek, parça başına en uygun değeri bulmaya çalıştım. Bu noktada genetik algoritmanın ruhuna zıt düşmüş oldum. Eğer genetik algoritmalar tahmine dayalı bir metot ise, ben de bu tahmin yöntemini tahmin ederek yapmam. Yalnız bu noktada, farklı değerler ve seçim yöntemlerini kullanmam bana hangi tezgâhta nasıl verim alacağım konusunda fikir verdiği için, deneme yanılma yoluyla farklı değerler buldum. Yukarıda yaptığım analizlerde d_r ve d_s 'ye 3,8 değerlerini verdiğimde parça başına maliyet değeri oldukça düştü (Şekil IV.37-38), eğer ki güçlü bir tezgâhım varsa, bu talaş derinliğini risk alabilirim. Bu da demek oluyor ki, programda farklı sınır değerleri girişleri yapmam, çalışma koşullarını belirlemem de bana yardımcı olmaktadır.

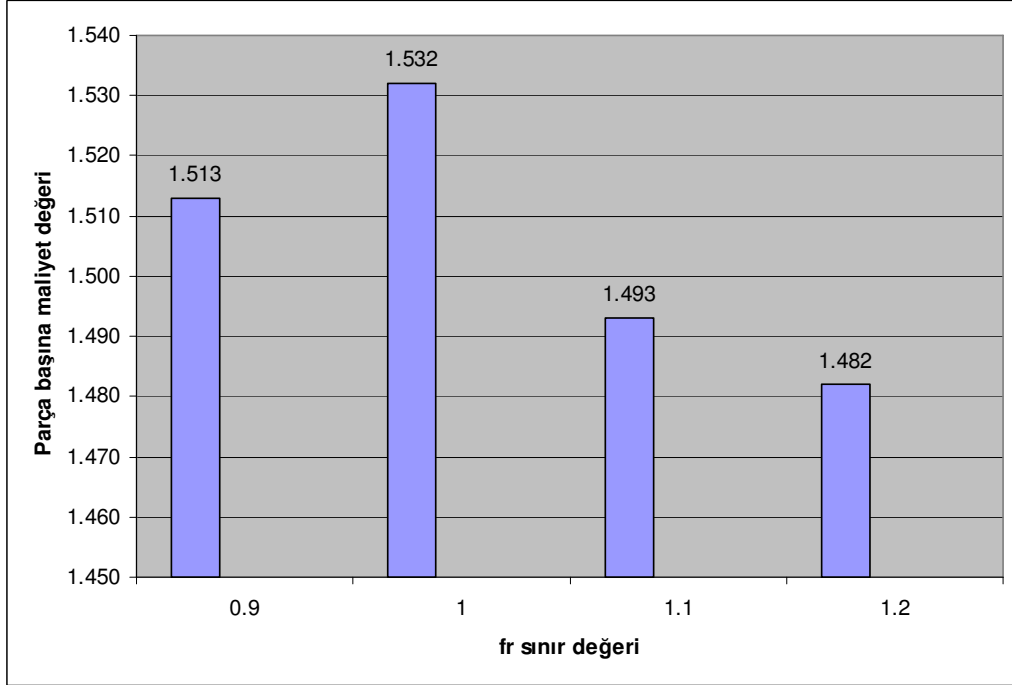
Delphi dilinde yazılmış bu program vasıtasıyla, ele aldığımız problemi genetik algoritmanın birçok opsiyonu ile birlikte çözme imkânı bulmuş olmaktadır. Ele aldığımız problemimizdeki sabit değerleri hiç değiştirmeden, sınır değerlerinde küçük oynamalar yaparak, maliyeti 1,71 YTL'den 1,11 YTL'ye kadar, yani %35 oranında indirdik. Yaklaşık 25 farklı sınır değeri denenmiştir. Öte yandan genetik algoritmalara ait diğer seçim yöntemleri, kodlama şekilleri ve metotlar da denenmiş ve en uygun yöntemin SUS seçimini, ikili alfabede, Elastik Metot ile yapılarak bulunduğu saptanmıştır. Problemin çözümünde ceza fonksiyonu kullanılmıştır.

Kodlama, sınır değerlerinde değişiklik yapmadığımızda Gray olarak seçildiğinde 1,52 YTL maliyet değerini veriyordu. Yani ikili alfabenin verdiği 1,71 YTL değerinden daha az maliyet değeri vermekteydi. Ancak, sınır değerleri değiştirdikten sonra kodlamayı Gray olarak seçtiğimizde sonuç 1,11 YTL'den 1,13 YTL'ye çıkmıştır. Yani farklı sınır değerlerine göre seçilen kodlama farklı sonuçlar verebilmektedir. Herhangi bir doğru orantı söz konusu değildir.

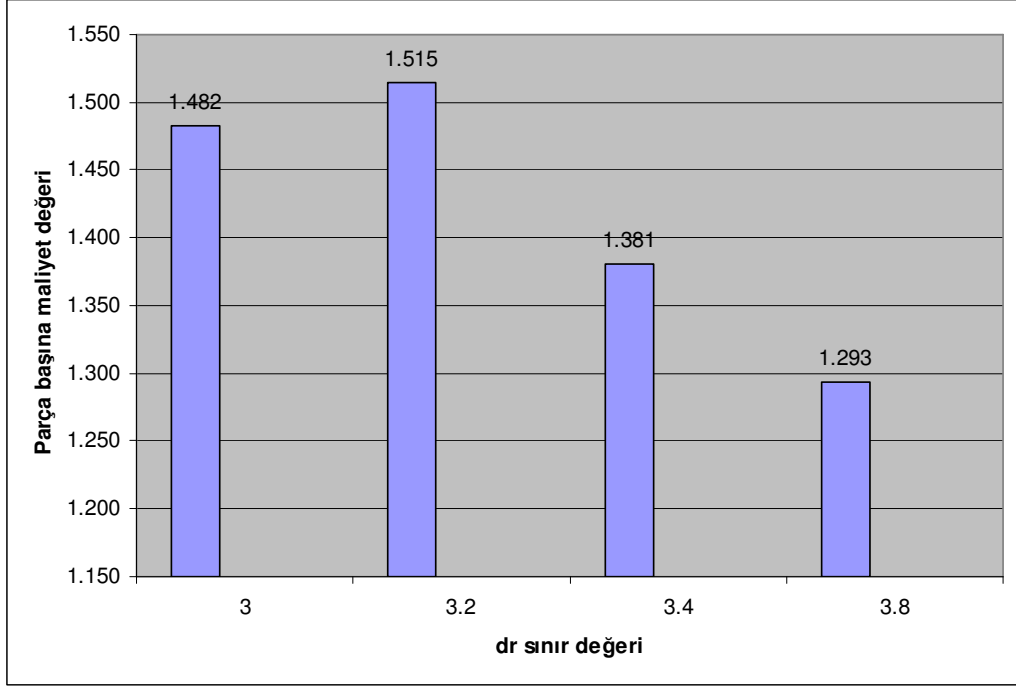
Şekil IV.36 – Şekil IV.41'de sınır değerlerde yapılan değişikliklerin grafiksel olarak değişimi görülmektedir. Y ekseninde parça başına düşen maliyet değeri (YTL/parça); X ekseninde ise ilgili sınır değerine ait veriler gösterilmektedir.



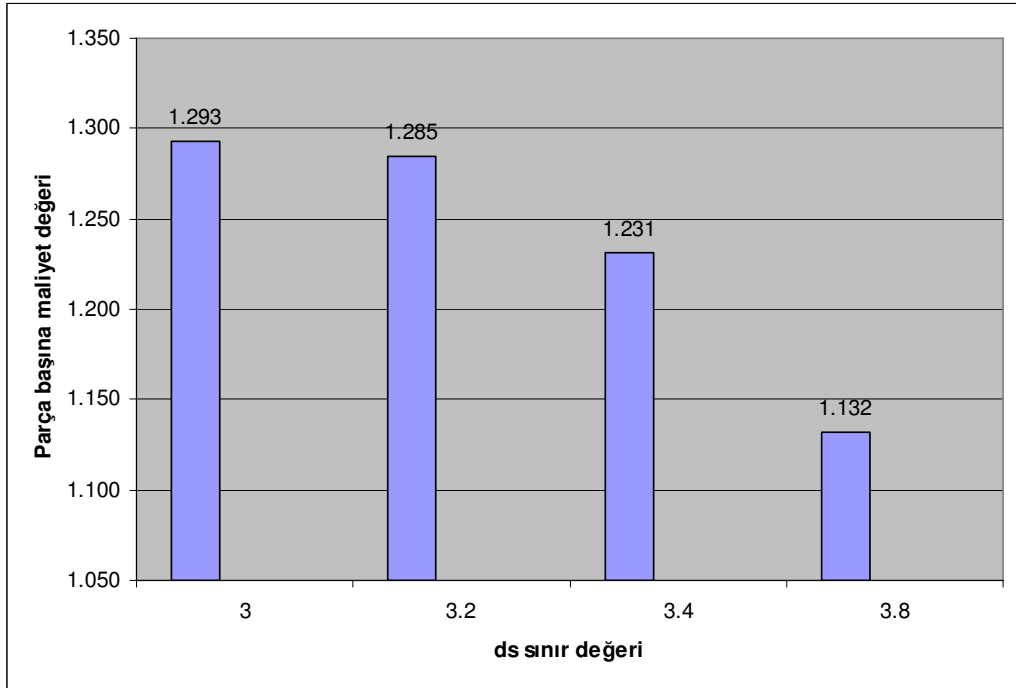
Şekil IV.36 Vr kaba tornalama kesme hızına ait girilen sınır değerlerin maliyet değeri grafiği



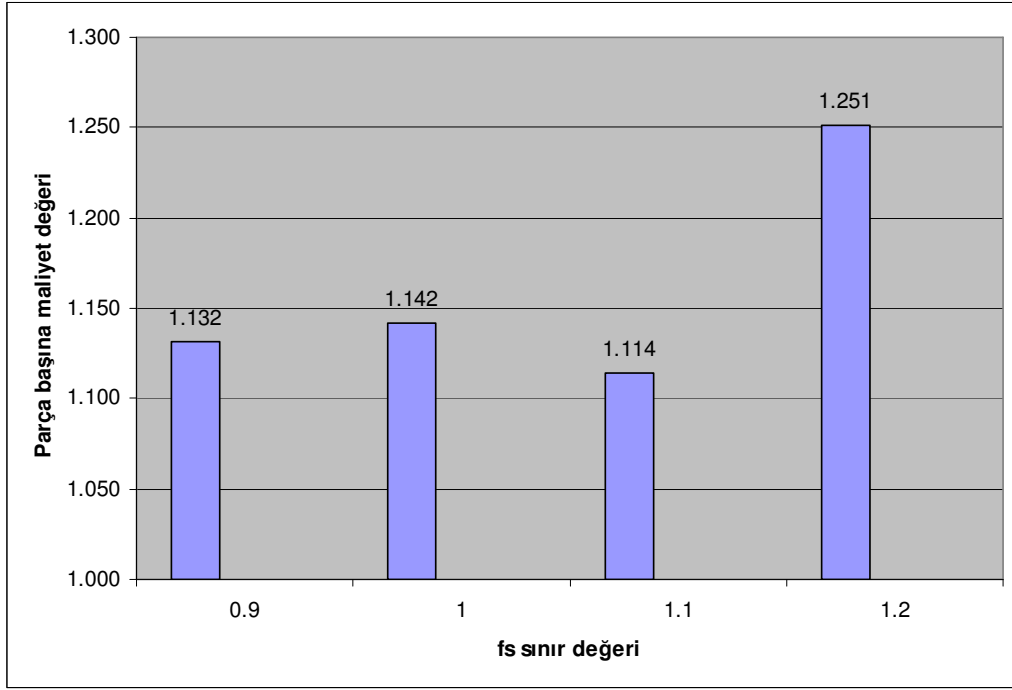
Şekil IV.37 fr kaba tornalama ilerleme miktarına ait girilen sınır değerlerin maliyet değeri grafiği



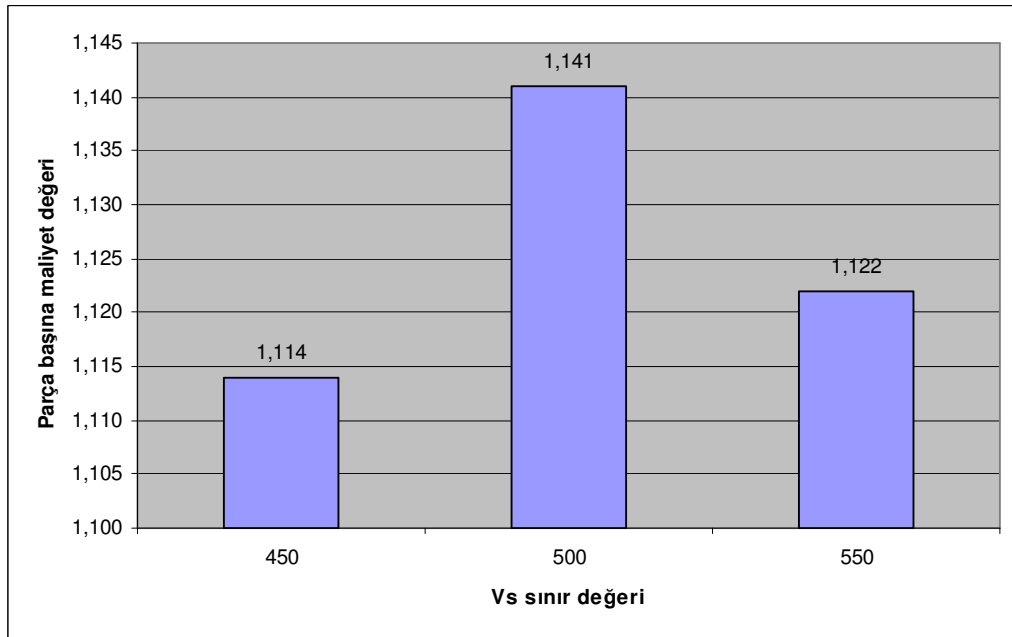
Şekil IV.38 d, kaba tornalama talaş derinliğine ait girilen sınır değerlerin maliyet değeri grafiği



Şekil IV.39 d, bitiş tornalama talaş derinliğine ait girilen sınır değerlerin maliyet değeri grafiği



Şekil IV.40 f_s bitiş tornalama ilerleme miktarına ait girilen sınır değerlerin maliyet değeri grafiği



Şekil IV.41 V_s bitiş tornalama kesme hızına ait girilen sınır değerlerin maliyet değeri grafiği

KAYNAKLAR

- [1] Güngör F.: “*Tornalamada maliyet hesaplaması*”, Marmara Üniversitesi Yüksek Lisans Tezi, İstanbul, Türkiye, (1980), 73-76
- [2] Onwubolu G.C.; Kumalo T. : “*Optimization of multipass turning operations with genetic algorithms*”, Int. J. Prod.Res., (2001), Vol. 39, No. 16, 3728
- [3] Gökay E.G.; Taşkın Ç.: “*Genetik Algoritmalar ve Uygulama Alanları*”, Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, Cilt XXI, Sayı 1, (2002), s. 129-152
- [4] Kurt M.; Semetay C.: “*Genetik Algoritma ve Uygulama Alanları*”, http://www.mmo.org.tr/muhendismakina/arsiv/2001/ekim/Genetik_Algoritma.htm, (Erişim Tarihi: Kasım (2003)
- [5] Chen M.C.; Chen K.Y.: “*Optimization of multipass turning operations with genetic algorithms: a note*”, Int. J. Prod.Res., (2003), Vol. 41, No. 14, 3385-3388
- [6] Ermer D.S; Kromodihardjo S.: “*Optimization of Multipass Turning With Constraints*”, Transactions Of The ASME, (1981), Vol.103, 462 – 468
- [7] Sardinas R.Q.; Santana M.R.; Brindis E.A.: “*Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes*”, Engineering Applications of Artificial Intelligence, (2006), v.19, Iss.2, March 2006, 127-133
- [8] http://en.wikipedia.org/wiki/Genetic_algorithms (Erişim tarihi: Haziran 2005)
- [9] Jawahir I.S.; Sadler J.P.: “*Optimization of Machining Performance in Turning Operations*”, www.elsevier.com/locate/engappai , (Erişim Tarihi: Ekim 2005)
- [10] Smith A.E.; Coit D.W.: “*Penalty Functions*”, A Joint Publication of Oxford University Press and Institute of Physics Publishing, (1996)
- [11] Meng Q.; Arsecularatne J.A.; Mathew P.: “*Calculation of optimum cutting conditions for turning operations using a machining theory*”, International Journal of Machine Tools & Manufacture, (2000), 1709-1732

- [12] Brozek M.: “*Cutting conditions optimization when turning overlays*”, Journal of Materials Processing Technology, (2005), 488-495
- [13] Bouzid W.: “*Cutting parameter optimization to minimize production time in high speed turning*”, Journal of Materials Processing Technology, (2004), 388-394
- [14] Lee B.Y.; Tarn Y.S.: “*Cutting-parameter selection for maximizing production rate or minimizing production cost in multistage turning operations*”, Journal of Materials Processing Technology, (1999), 61-66
- [15] Yang W.H.; Tarn Y.S.: “*Design optimization of cutting parameters for turning operations*” (1997), 122-129
- [16] Bhaskara Reddy S.V.; Shunmugam M.S.; Narendran T.T.: “*Optimal subdivision of the depth of cut to achieve minimum*”, Journal of Materials Processing Technology, (1998), 101-108
- [17] Çakır M.C.; Gürarda A.: “*Optimization and graphical representation of machining conditions in multi-pass turning operations*”, Journal of Materials Processing Technology, Vol.11, No.3, (1998), 157-170
- [18] Vijayakumar K.; Prabhakaran G.; Asokan P.; Saravanan R.: “*Optimization of multi-pass turning operations using ant colony*”, International Journal of Machine Tools & Manufacture, (2003), 1633-1638
- [19] Risbood K.A.; Dixit U.S.; Sahasrabudhe A.D.: “*Prediction of surface roughness and dimensional deviation by measuring cutting forces and vibrations in turning process*”, Journal of Materials Processing Technology, (2003), 203-214
- [20] Arezoo B.; Ridgway K.; Al-Ahmari A.M.A.: “*Selection of cutting tools and conditions of machining operations using an expert system*”, Computers in Industry, (2000), 43-58
- [21] Iwata K.; Murotsu Y.; Oba F.: “*Optimization of cutting conditions for multi-pass operations considering probabilistic nature in machining processes*”, Transactions of the ASME, (1977), 210-217
- [22] Gopalakrishnan b.; Al-Khayyal F.: “*Machine parameter selection for turning with constraints: an analytical approach based on geometric programming*”, Int. J. Prod.Res., (1991), Vol. 29, No. 9, 1897-1908

- [23] Shin Y.C.; Joo Y.S: “*Optimization of machining conditions with practical constraints*”, Int. J. Prod.Res., (1992), Vol. 30, No. 12, 2907-2919
- [24] Tan F.P.; Creese R.C.: “*A generalized multi-pass machining model for machining parameter selection in turning*”, Int. J. Prod.Res., (1995), Vol. 33, No. 5, 1467-1487
- [25] Gupta R.; Batra J.L; Lal G.K.: “*Determination of optimal subdivision of depth of cut in multipass turning with constraints*”, Int. J. Prod.Res., (1995), Vol. 33, No. 9, 2555-2565
- [26] Chen M.C.; Tsai D.M.: “*A simulated approach for optimization of multi-pass turning operations*”, Int. J. Prod.Res., (1996), Vol. 34, No. 10, 2803-2825
- [27] Kathryn A.: “*Genetic Algorithms – A Tool for OR?*”, Journal of the Operational Research Society (1996), 47, 550-561
- [28] Prasad K.; Rao P.N.; Rao U.R.K.: “*Optimal selection of process parameters for turning operations in a CAPP system*”, Int. J. Prod.Res., (1997), Vol. 35, No. 6, 1495-1522
- [29] Çakır M.C.; Gürarda A.: “*Optimization of machining conditions for multi-tool milling operations*”, Int. J. Prod.Res., (2000), Vol. 38, No. 15, 3537-3552
- [30] Alataş B.; Karıcı A.: “*Genetik Algoritmelerde Düzenli Popülasyon ve Düzenli Operatör*”, Afyon Kocatepe Ün. Fen Bil.Der. 1(1-2), 11-26
- [31] Cus F.; Balic J.: “*Optimization of cutting process by GA approach*”, Robotics and Computer Integrated Manufacturing, (2003), 19, 113-121
- [32] Michalewicz Z.: “*Genetic Algorithms+Data Structure=Evolution Programs*”, Third, Revised and Extended Edition, (1996)
- [33] Bodenhofer U.: “*Genetic Algorithms: Theory and Applications*”, Lecture Notes Third Edition, (2004)
- [34] Wang X.; Da Z.J.; Balaji A.K.; Jawahir I.S: “*Performance-based optimal selection of cutting conditions and cutting tools in multipass turning operations using genetic algorithms*”, Int. J. Prod.Res., (2002), Vol. 40, No.9 2053-2065

SERDAR DÜZGÜN

Petrol-iş m. Vişne s. Nahitbey Apt.

No:24/5 Kartal/İST. 34862

Ev : 0216 473 22 29

Cep : 0532.526.26.17 – 0505.395.64.17

serdarduzgun@hotmail.com

Kişisel Bilgiler

Doğum Tarihi 29.02.1980

Medeni Hali Bekar

Askerlik Ağustos 2006

Eğitim

2002 – 2006 Marmara Üniversitesi Fen Bilimleri Enstitüsü
Makine Eğitimi

1997 – 2001 Marmara Üniversitesi Teknik Eğitim Fakültesi
Makine Eğitimi Ana Bilim Dalı
Talaşlı Üretim Öğretmenliği

1993 – 1997 Haydarpaşa Teknik Lisesi
Bilgisayarlı Nümerik Kontrol (CNC) Bölümü

İş Tecrübesi

2003 – 2006 Zühtüpaşa İlköğretim Okulu
Sözleşmeli (Bilgi İşlem Operatörü)

1998 – 2000 Mar İç ve Dış Tic. Ltd. Şti
Dealer

Yabancı Dil İngilizce (TOEFL Seviyesi)

Bilgisayar ve Ofis Genel Bilgisayar (hakim), M.Office Programları, Internet,

Araçları Bilgisi AutoCad (İyi), Web tasarım (orta), Büro malzemeleri

**Katıldığı Kurs ve
Seminerler**

16.10.01	Marmara Üniversitesi İnsan İlişkileri
1999 – 2001	Marmara Üniversitesi Yabancı Diller Eğitim Programları İngilizce
1997 – 1999	Antik English İngilizce

Referans

Marmara Üniversitesi Yrd.Doç.Dr.Ferhat GÜNGÖR
0532.271.91.19
Zühtüpuşa İlköğretim Okulu
Müdür Yardımcısı Ömer ATİLLA
0216.336.01.88

İlgi Alanları

Bilgisayar, Kitaplar, Bilim/Teknoloji, Finans, İnternet, İş
Dünyası, Müzik, Sinema, Tiyatro, Spor, Seyahat Etmek