Yıldız Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü

DERS

BLM4510 –YAPAY ZEKA

KONU

Paralel Tepe Tırmanma Ve Genetik Algoritma İle Sudoku Çözücü

DERSİN YÜRÜTÜCÜSÜ

Yrd. Doç. Dr. Fatih AMASYALI

ÖĞRENCİLER

NUMARA: 13011704
AD: Abdullah
SOYAD: KAYAHAN

NUMARA: 13011705
AD: Uhud
SOYAD: POYRAZ

2015

PROBLEM

Problem Tanımı: Paralel tepe tırmanma ve genetik algoritma kullanarak verilen bir sudoku tablosunu çözmek.

ÇÖZÜM

Yapmış olduğumuz projede sadece genetik algoritma kullanılarak sonuca gidilmeye çalışılmıştır. Çözüm sürecinde;

1- Kullanıcıdan başlangıç değerlerinin alınması
2- Alınan değerlere göre N* tane yeni tahta oluşturulacak.
3- Üretilen her tahta için uygunluk fonksiyonu hesaplanacak
4- Hesaplanan uygunluk fonksiyonuna göre çaprazlama ve mutasyon işlemleri yapılacak
5- Oluşturulan N tane tahtadan yeni K** tane tahta üretilecek
6- 3. Ve 4. Adımlar 5. Adımda üretilen tahtalar içinde geçerli olacak
7- En yüksek uygunluğa sahip olan tahta ekrana yazdırılacak

* N: Başlangıç Popülasyonu , ** K: Maksimum Popülasyon Limiti

OLUSTUR.CS

Hesaplama Fonksiyonu

1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1
9	8	7	6	5	4	3	2	1
9	8	7	6	5	4	3	2	1
4	5	6	7	8	9	1	2	3
4	5	6	7	8	9	1	2	3
4	5	6	7	8	9	1	2	3

Hesaplama Fonksiyonu Kendi içinde 3 kısma ayrılır. Ve tüm tablo için bir uygunluk değeri üretir.

1- Her Bir Satırı(Yeşil)
2- Her Bir Sütunu(Sarı)
3- Her Bir 3X3 Kareyi(Mavi-Beyaz)

Bu uygunluk değeri hesaplanırken Hashtable'lar kullanılır. Burada hashtable kullanmamızın amacı, key-value mekanizması sayesinde satır, sütun ve karelerdeki değerlerimizi key olarak atarak benzersiz olup olmadığını ilgili hashtable boyutu sayesinde geri döndürüyor.

Bir satır için ele alacak olursak; Dönen değer 9 ise satır uygunluğu 1/9 olacaktır. Her bir satır için bu işlem tekrarlanınca oluşacak olan değerlerin toplamı bize toplam satır uygunluğunu verecektir.

Bu işlem sütunlar ve kareler içinde aynı şekilde işler.

En sonunda oluşan değerler çarpılarak toplam uygunluk fonksiyonu oluşturulur.

Çıkan sonuç Sıfır ile Bir [0,1] arasındadır. 0 en kötü 1 ise en iyi sonuçtur. Yani 1'e yaklaştıkça o tablo için uygunluk değeri yükseliyor diyebiliriz.

Çaprazlama Fonksiyonu

Bu fonksiyonda 2 adet child tablo oluşturuluyor. Ve bu oluşturulan tablolar bizim kullandığımız ana tablo üzerinde rastgele seçilen rastgele bir satır ve sütun değerine göre değişim yapar.

1	2	3	4	5	6	7	8	9
2	3	4	5	6	7	8	9	1
4	5	6	7	8	9	1	2	3
3	2	1	6	5	4	9	7	8
9	1	8	7	2	5	2	3	4
1	2	3	4	5	6	7	8	9
2	3	4	5	6	7	8	9	1
1	4	5	3	2	8	9	1	6
8	7	6	5	4	3	9	1	2

ANA TABLO

9	8	7	6	5	4	3	2	1
1	2	3	9	8	7	6	5	4
6	5	4	3	2	1	9	8	7
3	1	2	4	6	5	7	9	8
1	2	3	4	5	6	7	8	9
5	6	7	8	9	1	2	3	4
9	1	2	3	4	8	6	5	7
1	2	9	8	7	5	4	3	6
6	1	5	7	7	8	9	4	5

CHILD 1

1	5	4	5	6	8	4	2	5
4	5	6	2	1	4	7	8	9
5	5	4	9	8	6	7	1	2
3	3	4	5	6	7	9	5	6
7	8	3	2	1	4	2	4	6
9	8	3	4	5	5	2	1	4
6	7	9	2	1	1	7	9	2
1	2	3	4	5	6	7	8	9
9	8	7	3	2	1	4	5	6

CHILD2

Burada seçim işlemi rastgele yapılıyor tek bir random değer satır mı yoksa sütun mu değişecek onu seçiyor. Diğer bir random değer hangi satırın veya sütunun seçileceğini belirlemek için oluşturuluyor.

Varsayım olarak; 1. Random tek bir değer üretti ve satır değiştireceğiz 2. Random ise 5 değeri üretti yukarıdaki tablolar aşağıdaki şekle döndü. Bu işlemler bittikten sonra bir random değerde oluşan yeni child'lardan hangisinin geri gönderileceğini seçiyor.

1	2	3	4	5	6	7	8	9
2	3	4	5	6	7	8	9	1
4	5	6	7	8	9	1	2	3
3	2	1	6	5	4	9	7	8
9	1	8	7	2	5	2	3	4
1	2	3	4	5	6	7	8	9
2	3	4	5	6	7	8	9	1
1	4	5	3	2	8	9	1	6
8	7	6	5	4	3	9	1	2

ANA TABLO

9	8	7	6	5	4	3	2	1
1	2	3	9	8	7	6	5	4
6	5	4	3	2	1	9	8	7
3	1	2	4	6	5	7	9	8
9	1	8	7	2	5	2	3	4
5	6	7	8	9	1	2	3	4
9	1	2	3	4	8	6	5	7
1	2	9	8	7	5	4	3	6
6	1	5	7	7	8	9	4	5

CHILD 1

1	5	4	5	6	8	4	2	5
4	5	6	2	1	4	7	8	9
5	5	4	9	8	6	7	1	2
3	3	4	5	6	7	9	5	6
9	1	8	7	2	5	2	3	4
9	8	3	4	5	5	2	1	4
6	7	9	2	1	1	7	9	2
1	2	3	4	5	6	7	8	9
9	8	7	3	2	1	4	5	6

CHILD2

Mutasyon

Bu fonksiyon ise tablodaki random yerdeki değerleri değiştiren fonksiyon. Bu işi 3 farklı şekilde yapıyor.

1. Her hangi 3 sayı üretiyor. Ürettiği ilk iki sayı matrisin indisi diğer sayı ise 0 indise eklenecek olan değer.
2. Matrisin[1. Üretilen ,2.üretilen]⇔Matrisin[3.üretilen ,2.üretilen] bu iki değeri birbiri ile değiştiriyor.
3. Matrisin[2. Üretilen ,1.üretilen]⇔Matrisin[2.üretilen ,3.üretilen] bu iki değeri birbiri ile değiştiriyor.

Bu değişimler yapılırken tempMatrisi'nde ilgili yerin 0 olduğundan emin olunması gerekiyor. Yani kullanıcının verdiği değerleri değiştirmemesi lazım.

Yukarıdaki 3 işlemden birini seçmek için çeşitli random değerlerle şans faktörü ile değişim yapılmıştır.

POPULASYON.CS

Yeni Jenerasyon Üretme

Kullanıcının girdiği maksimum jenerasyon sayısı kadar yeni tahtalar üretecek. Üretilen tahtaların her biri için uygunluk değerini hesaplayıp çıkan sonuca göre oluşan yeni tahtayı siliyor veya o tahtayı kullanıyor.

Çaprazlama Fonksiyonu

Üretilen yeni tahtalardan kullanılacak olan tahtalar bu fonksiyona girer. Fonksiyon içerisinde yeni 2 adet tahta dizisi oluşturulur. Bunlar bizim parent tahtalarımızın tutulacağı yerlerdir. Bu dizilere random yolla oluşturulan tahtalar eklenir. Ekleme işlemi bittikten sonra iki dizininde boyutu eşit olmalıdır. Bunun sebebi çaprazlama yaparken hiçbir tahta açıkta kalmayacak şekilde olmalıdır. Eşitleme işlemi bittikten sonra oluşan dizinin boyutu kadar sürecek olan bir döngüye girer. Bu döngü içinde her adımda 2 adet child oluşturulur. Oluşturulan child'lar parent'lerin birbirleri ile çaprazlanmış halidir.

Bu işlemden sonra bütün oluşan tahtlar bir diziye yazılarak sıralanır. Ve en yüksek olan 2 tanesi seçilir. Ve bir sonuç dizisine eklenir.

Ana programda bu sonuç dizisine eklenen tahtalardan en yüksek olanı seçilecektir.

FORM1.CS

Form1 ana programımızın olduğu yerdir. Öncelikle form üzerindeki textBox'lar oluşturulur ve renklendirilir. Ayrıca bu textBox'lar sadece 1'den 9'a kadar nümerik değerler girilebilecek şekilde ayarlanır.

ilkDegerAl Fonksiyonu

Bu fonksiyon başlangıç sütununda bulundan textBox grubu içinde çalışır. Ve içerisinde yazan değerleri bir matrise alır. Boş ise ilgili matris değerine 0 yazar.

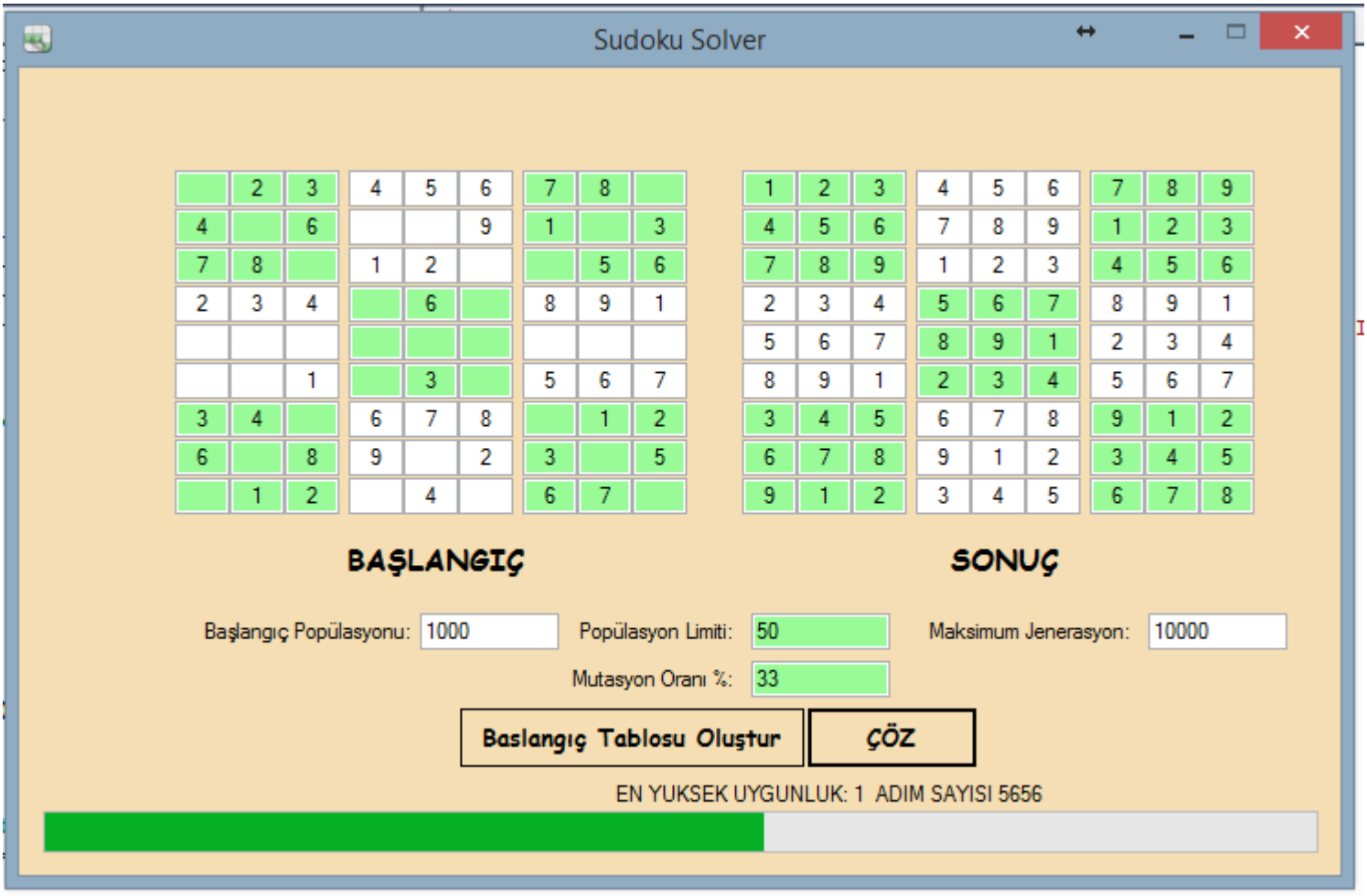
Solver Fonksiyonu

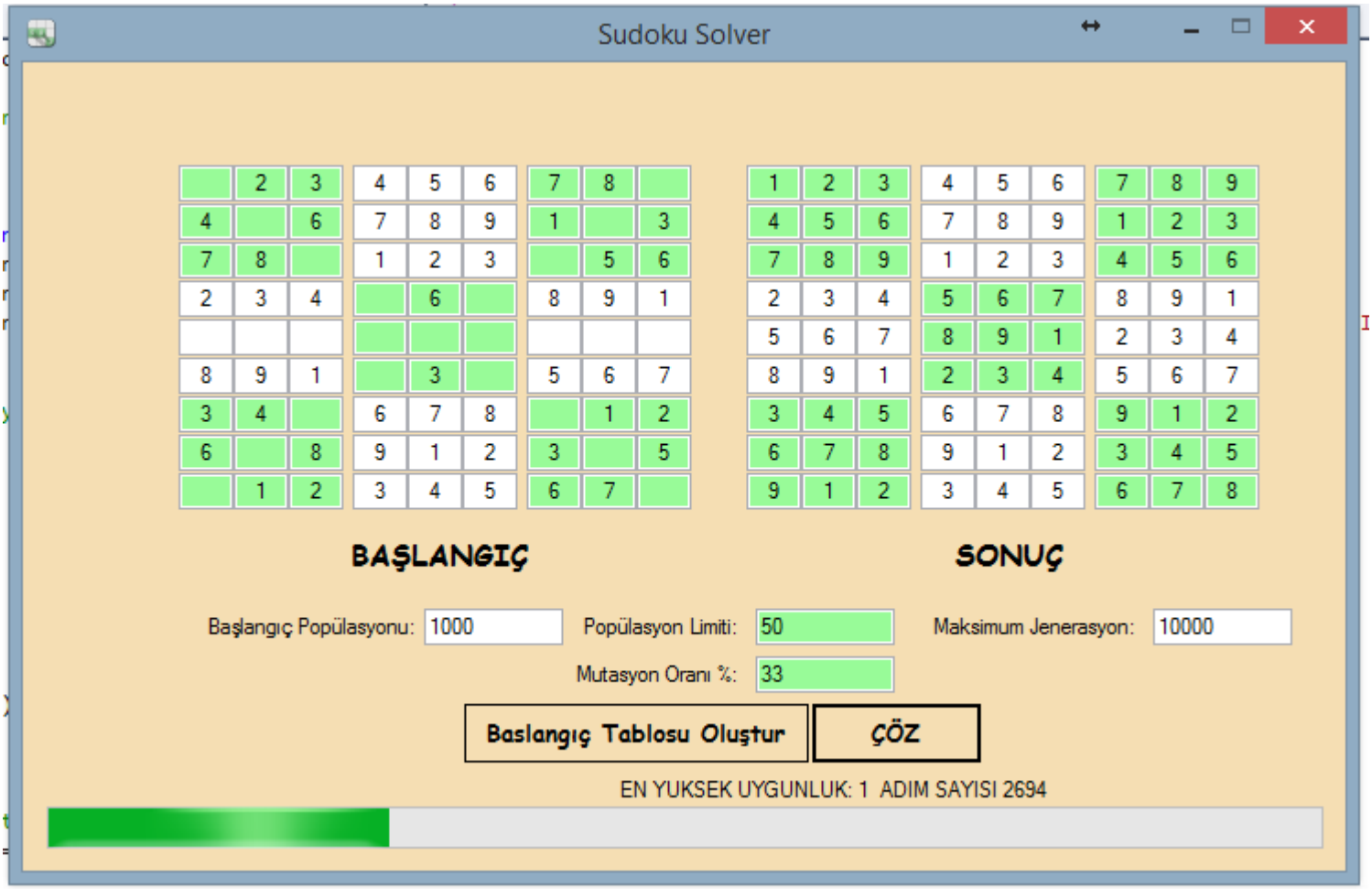
Tüm işi yapacak fonksiyondur. Öncelikle bir popülasyon oluşturur. Ardından maksimum jenerasyon sayısı kadar jenerasyon üretmeye başlar. Üretilen yeni jenerasyondaki en yüksek uygunluğa sahip tahtayı getirir. Bir sonraki tahtanın uygunluğu daha yüksek ise o tahtayı getirir. Eğer uygunluk 0,9999 dan büyükse sonuç direk olarak o tahtadır diyebiliriz. Ve döngüyü kırarak çıkarız. Döngüden çıktıktan sonra elimizde olan tahta en yüksek uygunluğu olan tahtadır.

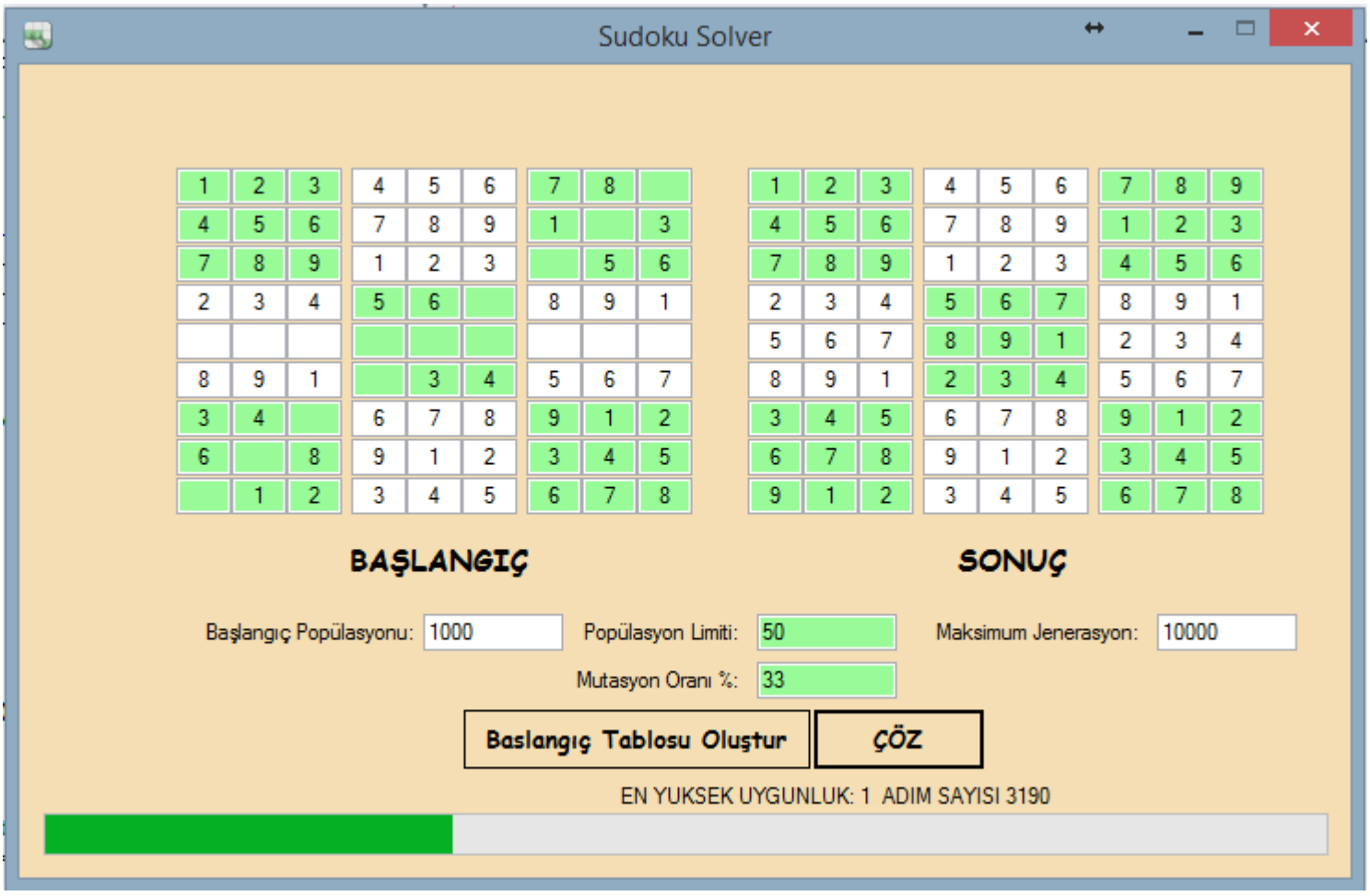
PROBLEM ÇÖZERKEN KARŞILAŞILAN SORUNLAR VE ÇÖZÜMÜ

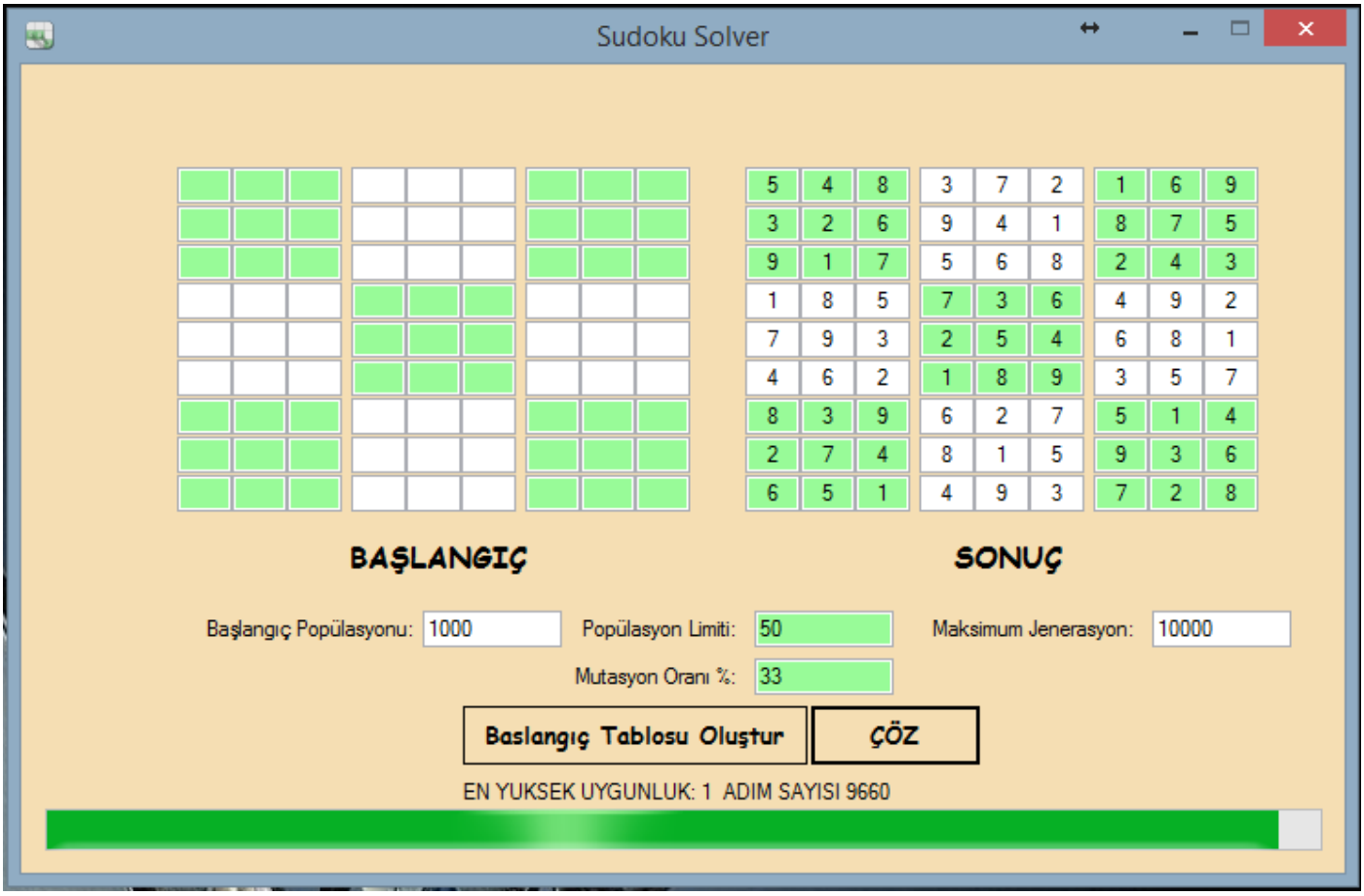
1- Öncelikle problemi çözmek için tüm tahtayı değil her satırı ve sütunu çözmeyi düşünmüştük. Ancak araştırmalarımız sonucu tüm tahtanın değiştirilmesinin daha mantıklı olduğunu gördük.
2- Uygunluk değeri hesaplamada öncelikle her satır, sütun ve kareler için her bir değeri toplayarak 45 sayısına yakınlığını kontrol etmeyi düşündük. Fakat farklı olmayan değerler için bile 45 değerini verdiğini gördük. Bu nedenle hashTable yöntemini kullanarak hashTable boyutunu baz alarak işlem yaptık.
3- Problem çözümünün çok yavaş çalışması: Bu durum yaşanırken" ilkdeğeral "fonksiyonu oluştur sınıfı içerisindeydi. Yaptığımızda testlerde yavaşlama sebep olan şey her seferinde formu çağırıp üzerindeki textBox'ları okuması olduğunu gördük. Bu testlerden sonra "ilkdeğeral" fonksiyonunu oluştur sınıfından form sınıfına aldık ve bir int matris ile bu değerleri tuttuk. Oluştur sınıfında setTempMatris adlı bir fonksiyon ile formda oluşturulan matrisi oluştur sınıfına gönderdik.
4- En yüksek değerli fonksiyonun sonuç grubu textbox'lara yazılmaması. Program çalıştığı sırada yüksek değerli tahtalar buluyor ancak sonuç textBox'larına en son bulduğu değeri yazıyordu. Sorunu çözmek için Solver fonksiyonunda bir if ile önceki ve yeni değerli karşılaştırdık hangisi büyük ise sonuç olarak onu yazdıracak şekilde problemi çözdük.
5- Programın çalışıp çalışmadığını belli olmaması sorunu. Program çalıştığı anda görev yöneticisinde yanıt vermiyor olarak görünmektedir. Kullanıcının programının donmadığını anlaması bir sorun teşkil etmekteydi. Bunun çözümü için tasarıma bir adet progressBar ekledik. Bu bar sayesinde işlemin ne kadar kaldığı şu anki en yüksek uygunluk gibi değerleri kullanıcının anlık olarak görmesini sağladık.

EKRAN GÖRÜNTÜLERİ




Sudoku Solver
BAŞLANGIÇ
SONUÇ
Başlangıç Popülasyonu: 1000
Popülasyon Limiti: 50
Maksimum Jenerasyon: 10000
Mutasyon Oranı %: 33
Baslangıç Tablosu Oluştur
ÇÖZ
EN YUKSEK UYGUNLUK: 1 ADIM SAYISI 2694


Sudoku Solver
BAŞLANGIÇ
SONUÇ
Başlangıç Popülasyonu: 1000
Popülasyon Limiti: 50
Maksimum Jenerasyon: 10000
Mutasyon Oranı %: 33
Baslangıç Tablosu Oluştur
ÇÖZ
EN YUKSEK UYGUNLUK: 1 ADIM SAYISI 3190



KAYNAKÇA

1- https://www.youtube.com/watch?v=ShyWEYe2Mfo
2- http://www.bikod.com/c/c-dinamik-olarak-progressbara-metin-yazi-ekleme.html