

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**OLİMPİK HAVUZ PLAN ŞEMASI TASARIMINDA GENETİK
ALGORİTMAYA DAYALI BİR MODEL**

YÜKSEK LİSANS TEZİ

Kadir ÇALIŞIR

Bilişim Anabilim Dalı

Mimari Tasarımda Bilişim Programı

Tez Danışmanı: Öğr.Gör. Dr. Hakan TONG

HAZİRAN 2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**OLİMPİK HAVUZ PLAN ŞEMASI TASARIMINDA GENETİK
ALGORİTMAYA DAYALI BİR MODEL**

YÜKSEK LİSANS TEZİ

**Kadir ÇALIŞIR
(523071011)**

Bilişim Anabilim Dalı

Mimari Tasarımda Bilişim Programı

Tez Danışmanı: Öğr. Gör. Dr. Hakan TONG

HAZİRAN 2015

İTÜ, Fen Bilimleri Enstitüsü'nün 523071011 numaralı Yüksek Lisans Öğrencisi **Kadir ÇALIŞIR**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**OLİMPİK HAVUZ PLAN ŞEMASI TASARIMINDA GENETİK ALGORİTMAYA DAYALI BİR MODEL**” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Öğr. Gör. Dr. Hakan TONG**
İstanbul Teknik Üniversitesi

Jüri Üyeleri: **Prof.Dr. Gülen ÇAĞDAŞ**
İstanbul Teknik Üniversitesi

Yrd.Doç.Dr. Togan TONG
Yıldız Teknik Üniversitesi

Teslim Tarihi : **27 Nisan 2015**
Savunma Tarihi : **25 Mayıs 2015**

Anneme,

ÖNSÖZ

Öncelikle bu çalışma sürecinde büyük bir özveri ile her konuda bana desteklerini esirgemeyen, tez danışmanlığımı yürüten çok değerli hocam, Sayın Öğr.Gör.Dr.Hakan TONG'a teşekkürlerimi sunarım.

Ayrıca yüksek lisans eğitimi boyunca derslerime giren tüm hocalarıma ve aileme teşekkürü bir borç bilirim.

Nisan 2015

Kadir ÇALIŞIR

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
ÇİZELGE LİSTESİ.....	xiii
ÖZET.....	xvii
SUMMARY	xix
1. GİRİŞ	1
2. GENETİK ALGORİTMA VE MİMARİ TASARIMDAKİ UYGULAMALARI.....	3
2.1 Evrimsel Hesaplama.....	3
2.2 Genetik Algoritma Tanımı ve Temel Kavramları	4
2.2.1 Genetik algoritma temel kavramları	5
2.3 Genetik Algoritmanın Temel Prensipleri	7
2.3.1 Rastgele arama algoritması	7
2.3.2 Genetik algoritma operatörleri	8
2.3.3 Genetik Kodlama	10
2.3.3.1 İkili Kodlama	10
2.3.3.2 Alfabetik kodlama.....	12
2.4 Mimari Tasarım Uygulamalarında Genetik Algoritmaların Kullanımı	14
2.5 Kısıt Sağlama Problemleri	19
2.5.1 Kısıt Sağlama Problemleri İçin Genel Yaklaşımlar	23
3. OLİMPİK YÜZME HAVUZLARI TASARIM KRİTERLERİ VE ÖRNEKLERİ	25
3.1 Havuzların Sınıflandırılması	25
3.2 Olimpik Yüzme Havuzlarının Mekansal Özellikleri	26
3.2.1 Olimpik yüzme havuzlarının üst mekanlarının planlanması.....	27
3.2.1.1 Islak mekanlar	27
3.2.1.2 Konum, ana giriş, fuaye ve tribünlerin planlanması	27
3.3 Olimpik Yüzme Havuzları Örnekleri	30
4. YÜZME HAVUZU PLAN ŞEMASI TASARIMI İÇİN GENETİK ALGORİTMAYA DAYALI BİR MODEL	39
4.1 Modelin Uygulanması	43
5. SONUÇLAR	53
KAYNAKLAR	55
EKLER.....	57
ÖZGEÇMİŞ.....	63

KISALTMALAR

GA : Genetik Algoritma
CSP : Constraint Satisfaction Problems

ÇİZELGE LİSTESİ

Sayfa

Çizelge 3.1 : Mevcut plan şemalarındaki her bir mekanı gösteren renk tablosu.....	30
Çizelge 3.2 : Verilen plan örneklerinin ana şema düzenleri.....	37
Çizelge 4.1 : Elde edilen plan şemaları sonuç değerleri.....	52

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : Genetik algoritma akış şeması (Holland, 1992)	9
Şekil 2.2 : İkili kodlama ile çaprazlama uygulamaları (Croce, 1994)	11
Şekil 2.3 : İkili kodlama ile mutasyon uygulamaları (Croce, 1994)	11
Şekil 2.4 : Alfabetik kodlama ile çaprazlama uygulamaları (Murata, 1996)	12
Şekil 2.5 : Alfabetik kodlama ile çaprazlama uygulamaları (Murata, 1996)	12
Şekil 2.6 : Alfabetik kodlama ile çaprazlama uygulamaları (Murata, 1996)	13
Şekil 2.7 : Alfabetik kodlama ile çaprazlama uygulamaları (Murata, 1996).	13
Şekil 2.8 : İki komşu genin karşılıklı olarak değişmesi (Murata, 1996)	13
Şekil 2.9 : Rastgele seçilen iki genin karşılıklı olarak değişmesi (Murata, 1996)	13
Şekil 2.10 : İki nesnenin üst üste gelme durumu (Bentley ve Wakefield, 1995)	15
Şekil 2.11 : Üst üste gelen nesneler (Bentley ve Wakefield, 1995).	15
Şekil 2.12 : Tek bir noktanın büyüyen nesnel biçimler (Bentley ve Wakefield, 1995)	15
Şekil 2.13 : Üst üste gelen ilkel biçimler (Bentley ve Wakefield, 1995)	16
Şekil 2.14 : Evrimsel tasarım sistemi diagramı (Bentley ve Wakefield, 1995)	17
Şekil 2.15 : Fenotip değerleri değiştirilen bir cismin evrimsel tasarımı (Bentley ve Wakefield, 1996)	17
Şekil 2.16 : Mimari plan üretimi için genel sistem tasarımı (M. Thakur ve M. Kumari)	18
Şekil 2.17 : Kullanıcı tarafından seçilecek planlar (M. Thakur ve M. Kumari)	18
Şekil 2.18 : Seçilen planın analizi ve sonuç ürünü (M. Thakur ve M. Kumari)	19
Şekil 2.19 : Avustralya kıtası bölgeleri	22
Şekil 2.20 : Avustralya kıtası bölgeleri	22
Şekil 2.21 : 4 lü N-Queens problemi	23
Şekil 2.22 : Ağaç dal metodu	24
Şekil 3.1 : Havuz Tipleri (Green guide dergisi, 2011).	26
Şekil 3.2 : Tipik bir havuz binasının ana alanları arasındaki şematik ilişkisi (swimming pools design guidance dergisi, 2011).	28
Şekil 3.3 : Oturmuş seyirciler için görüş çizgisi (Green guide dergisi, 2011).	29
Şekil 3.4 : Tribün düzenleme çeşitleri (Neufert).	29
Şekil 3.5 : Münih olimpik havuz plan ve şematik ilişkisi (Aquatic building).	31
Şekil 3.6 : Münih olimpik havuzu giriş kat planı (Aquatic building).	31
Şekil 3.7 : Hanburg olimpik havuzu plan ve şematik ilişkisi (Aquatic building).	32
Şekil 3.8 : Yakın Doğu Üni. Olimpik havuzu plan ve şematik ilişkisi (Aquatic building).	33
Şekil 3.9 : Kadir Has Olimpik havuzu plan ve şematik ilişkisi (v2.arkiv.com.tr).	34
Şekil 3.10 : Londra Olimpik Su Sporları Merkezi plan ve şematik ilişkisi (www.archdaily.com).	35
Şekil 3.11 : Beijing ulusal su sporları merkezi plan ve şematik ilişkisi.	36
Şekil 4.1 : Yüzme havuzu planı ağaç dal modeli.....	40
Şekil 4.2 : Yüzme havuzu planı akış şeması.....	41
Şekil 4.3 : Model arayüzü.....	42
Şekil 4.4 : Panel arayüzü	43
Şekil 4.5 : Maslak kampüsü yüzme havuzu alan sınırları	44
Şekil 4.6 : Maslak kampüsü havuz modeli için panel üzerinde girilen veriler.....	44
Şekil 4.7 : Maslak kampüsü olimpik havuz modeli için elde edilen sonuç.....	45
Şekil 4.8 : 10000 m ² alana uygulanacak model için panele girilen veriler.....	46

Şekil 4.9 :	10000m ² alana 500 seyircili olimpik havuz modeli için alınan sonuç	47
Şekil 4.10 :	10000m ² alana 1000 seyircili olimpik havuz modeli için alınan sonuç .	47
Şekil 4.11 :	10000m ² alana 2000 seyircili olimpik havuz modeli için alınan sonuç .	48
Şekil 4.12 :	10000m ² alana 2500 seyircili olimpik havuz modeli için alınan sonuç .	49
Şekil 4.13 :	10000m ² alana 2500 seyircili yarı olimpik model için alınan sonuç.....	50
Şekil 4.14 :	Yetersiz alanı belirten hata mesajı.....	50
Şekil 4.15 :	Oluşturulacak yüzme havuzunun alan sınırları	51
Şekil 4.16 :	1900 m ² alana uygulanacak model için panele girilen veriler.....	51

OLİMPİK HAVUZ PLAN ŞEMASI TASARIMINDA GENETİK ALGORİTMAYA DAYALI BİR MODEL

ÖZET

İnsan genetiğinin çözümlenmesiyle birlikte biyoloji ve genetik alanında oluşan gelişmeler, beraberinde farklı disiplinleri de etkilemiştir. Bu disiplinlerden biri de mimarlık alanı olmuştur. Genetik mimarlık diye adlandırılan bu yaklaşım, mimari yapıların tasarım sürecini etkileyen kararların, gerçekte yaşayan bir canlı organizmanın gelişim sürecine benzer şekilde geliştiğinin öngörülmesidir. Genetik mimarlık yaklaşımı doğadaki evrimi örnek alarak bilgisayar ortamına aktarılan genetik bilgiler ve bu bilgilere dayanarak kullanılan algoritmalar ile yeni mimari plan kurgularının üretilmesi mümkün olmaktadır. Bu tez kapsamında yapılan çalışmadaki yazılımlar Darwin'in evrim teorisinden esinlenerek oluşturulan evrimsel hesaplama tekniklerinden biri olan genetik algoritma yöntemi kullanılarak, bilgisayar ortamında genetik algoritmaya dayalı olimpik ve yarı olimpik yüzme havuzu plan şeması geliştirilecektir.

Olimpik ve yarı olimpik yüzme havuzlarının yerleşim planlarının daha etkin ve verimli kullanımı, onların tasarımına ve yerleşik düzendeki akış ilişkilerine bağlıdır. Bu ve benzeri tesislerin yerleşimindeki akış ilişkileri ve tasarım problemleri çözüm gerektiren problemlerdir. Bir arama ve optimizasyon yöntemi olan genetik algoritmaya dayalı yüzme havuzu salonlarının yerleşim planlarını oluşturmak için çalışma kapsamında öncelikle genetik algoritma kavramı hakkında bilgi verilmiştir. Genetik algoritmalarla ilgili mevcut literatür araştırması yapılmış ve çeşitli uygulama alanları incelenmiştir. Aynı zamanda genetik algoritmanın çalışma adımları, parametre seçimi ve fonksiyon optimizasyonunun mimari alandaki etkinliği, örnekler verilerek çalışma kapsamında belirtilmiştir. Bu çalışmalarda yeni bir tasarım oluşturmak ya da var olan bir tasarım iyileştirilirken seçilen parametre değerleri ile sonuca gitmek yerine genetik algoritma kullanılarak tasarımın her bir parçasını ele alıp iyileştirme sürecinin nasıl yapıldığı incelenmiştir.

Tezin üçüncü bölümünde, olimpik ve yarı olimpik yüzme havuzları plan şemaları ve mekansal özellikleri araştırılmış, mevcut yapıların planları şematik hale getirilerek elde edilen verilerle çalışma kapsamında oluşturulacak model için örnek teşkil edilmiştir.

Son bölümde önerilen modelin öncelikle akış şeması çıkarılmış, bu şemadaki adımlar takip edilerek bilgisayar ortamında modelin arayüzü oluşturulmuştur. Kullanıcı arayüzü kullanarak öncelikle yüzme havuzlarının mimari plan kurgusunun yapılacağı arazi seçimi için Google Map'ten yararlanılması öngörülmüştür. Arayüz programına Google map'in entegre edilmesinin iki ana sebebi vardır. Kullanıcının, modeli oluşturacağı alanı daha kolay belirleyebilmesi öncelikli sebeptir. Hem görsel olarak çalışma alanını görebilmekte, hem de belirlediği alanın koordinatları modele, doğru ve sağlıklı bir şekilde aktarılabilir. Diğer sebep ise, plan kurgusu oluşturulurken giriş akslarını belirlemektir. Kullanıcı uydu görüntüsü üzerinde ulaşım noktası ya da aksını daha kolay belirleyebilmektedir.

Programın arayüzü ASP .NET ortamında hazırlanmıştır. Modelin uygulanabilmesi için öncelikle harita üzerinde belirlenen sınırların ölçüleri veri tabanına aktarılmakta ve belirlenen ölçüler içerisinde program arazi ölçülerini, havuz tipini ve seyirci sayısı gibi belirlenen kısıtlarla tasarımcıya en uygun yüzme havuzu yerleşim planını ve boyutlarını sunmaktadır. Geliştirilen model ile gelecekte tasarlanacak olimpik havuzların plan şemalarının daha sağlıklı ve daha hızlı bir şekilde üretilmesi sağlanacaktır.

OLYMPIC POOL PLAN DESIGN OF A SCHEME BASED ON GENETIC ALGORITHM TO MODEL

SUMMARY

With the disintegration of human genetics, the development of biology and genetic has effected along different disciplines. Architecture has been one of these disciplines. Instead of inanimate structure called genetic architecture aproach, it prescribed of architectural structures as living organism in reality. With the aproach of genetic architecture it has been possible to produce new architectural forms by taking example of evolution in nature, transferred genetic information to computer and using algorithms based on this information. The software in the scope of work in this thesis used one of the evolutionary computation teqniques inspired by the Darwinian evolutionary genetic algorithm method. The relationship of genetic and genetic science will be explored in architecture within the scope of the natural sciences, olympic and semi-olympic swimming pool fiction architectural plans will be developed on computer.

To use more effectively and efficiently the settlement plans olympic and semi-olympic pools, depends on the flow relationship in their design and layout. The flow relationship of this and similar facilies residential and design problems are needed to be solved problems. In the work is given about the concept of genetic algorithm as priority to create the layout of the pool hall on genetic algorithm which based on a search and optimization methods. The research about the genetic algorithm has been made on the available literature and various applications of genetic algorithms were investigated. At the same time a solution example of the genetic algorithm process steps, choice of parameters and functional optimization given in the work.

At chapter two, have been utilized to provide constraint satisfaction problems under the work of thesis. Many problems in artificial intelligence can be modeled as constraint satisfaction problems. Therefore, the development of effcient solution methods for CPS's is a significant research problem. About constraint satisfaction problems, some algorithms have been reviewing to solve CPS and applied to the application problem.

Also in section two, it mentioned about the application of genetic algorithms in architectural desing.

As will be discussed extensively in the study, the genetic algorithm especially soultion space in large, transitory and in complex type of problems work according to the rules of probability simply due to the need of objective function seen successful results. Thus, much shorter in period of time they achive solution by making research. Based on the nature of evolution and genetic algorithms are used and transferred to a computer, it is possible to implement this process in the creation

of architectural form. Genetic algorithm optimization problem solution are mentioned in the work of Bentley and Wakefield in the design of solid objects using a genetic algorithm in 1995 and lastly in the M. Thakur and M. Kumari genetic algorithm using single-storey residential model application has been examined.

In the third section, olympic size swimming pool outlines the design criteria and examples that have been made will be given. Particularly planning the top venues in swimming pool, Here rather than being big or small the gym, good design and usable the space is very important for athletes and spectators. Also consideration of the first thing in the architectural planning stage of construction is to plan wet area. In this sport centers widely information are given about the criteria of location on the dressing area, toilet and wet areas which knows as shower parts. Location must be made determining before the determination of construct sports facilities. These criteria can be determined by technical and socio-cultural aspects of the assessment. The swimming pool facilities whether is a huge potential in this branch, is it a sufficient land for the audience capacity, will have international competition or not, if it is then internaiton norms should be done, criteria such as making design according to the climatic conditions in the region are important factors to determine the location of the plant construction. Fiction plans of swimming pools created with this obtained information.

Finally, in this thesis, the model proposed first issued flowchart, after that following the steps in this scheme was created in computer interface models. The user interface is primarily to benefit from using Google Map to do the architectural plans for the selection of land intended layout pool. The integration of the Google map interface program has two main reasons. Firstly, the user can select the area to be easier to create the model, both can visually see the work area, as well as determine the coordinates of the area where the model can be transferred to the right and healthy way. Another one is created layout plans to determine the input shaft. So that users satellite access point on the image or can determine the axle easier. The program prepared in the .NET programming console, primarily borders set on the map traferred to the database . The program will provide the most suitable alternatives to specified layout constrains such as the land measurements and number of spectators. The obtained results will provide better layout plans and rapid editing facilities to build a swimming pool in the future.

1. GİRİŞ

Evrimsel bilgi işleme tekniği olan yapay zeka uygulamasında, son zamanlarda genetik algoritmalara başvurduğu görülmektedir. Yapay zeka tanımını kabaca yapmak gerekirse; bir bilgisayarın veya bilgisayar denetiminde olan bir makinenin, insana özgü nitelikler olan akıl yürütme, genelleme yapmak, bir olaydan anlam çıkartmak veya geçmişte yaşanmış olaylardan deneyim kazanıp öğrenmeye yönelik yüksek zihinsel süreçlere ilişkin görevleri yerine getirme yeteneği olarak tanımlanmaktadır (Nabiyev,2010). Nispeten yeni olan bu kavramın farklı tanımları verilmektedir.

Yapay zeka üzerinde çalışma yapan bilim adamı Axe yapay zekayı şöyle tanımlamıştır; akıllı programları hedefleyen bir bilimdir. Bu programlar aşağıdakileri yapabilmelidir (Nabiyev, 2010):

1. İnsanın düşünmesini taklit ederek karmaşık problemleri çözebilmek.
2. Problemleri yorumlayıp kişiye yanıt verebilmek.
3. Öğrenebilmek ve öncesinde çözümlediği problemler ile yeni kazanmış olduğu bilgileri uyumlu bir şekilde kullanabilme ve bilgi tabanını genişletmek.

Harward üniversitesinde görev yapan öğrenme psikoloğu profesör Howard Gardner zeka kavramına farklı bir boyut getirerek çoklu zeka teorisini ortaya atmıştır. Gardner farklı zeka türlerinin olduğunu ve insanların yetenekleri ve ilgi alanlarına göre zeka türlerini 8 grupta toplamıştır. Bunlar Sözel, sosyal, görsel, mantıksal, bedensel, müzik, içsel ve doğal zeka olarak gruplamıştır (Nabiyev,2010).

Yapay zekanın başlıca çalışma yöntemlerini yapay sinir ağları, uzman sistemler, bulanık mantık ve genetik algoritmalar oluşturur. Bu tez kapsamında genetik algoritmalarından yararlanılarak mimariye yönelik plan kurgusu oluşturulacağından genetik algoritmalar hakkında geniş bir bilgi verilecektir.

Genetik algoritmalar evrime dayalı algoritmaların bir türüdür. Darwin'in evrim teorisi, günümüzde doğadaki evrimsel süreçleri baz alarak bilgisayar ortamında

uygulanan problem çözüme teknikleri her geçen gün daha da gelişmektedir. Genetik algoritmaya dayalı problem çözüme tekniğini ilk olarak Michigan Üniversitesinde psikoloji ve bilgisayar bilimi uzmanı John Holland tarafından çalışılmıştır. Holland, Darwin tarafından ortaya atılan evrim teorisinden etkilenecek canlılardaki genetik süreci bilgisayar ortamında yazılımsal (software) olarak düşünmüştür (Kurt,Semetaş, 2001).

Mimari plan kurgusu oluşturulurken buradaki problemin zorluk derecesinin bilinmesi problemin çözümü için en iyi yöntemin uygulanmasını sağlar. Mekan planlaması kurgulamasında, verilen bir plan içerisinde düzenli veya düzensiz şekil nesnelerinin yerleşimi ile alakalıdır. Günlük yaşamda bir kitaplıktaki kitapların yerleşim düzeni, araç park problemleri, bir iç mimarın iç tasarımı yaşadığı zorluklar araştırmacılar için çekici ve aynı zamanda zor bir çözümdür. Bir mimari planlama yapılırken tüm tasarım gereksinimlerini karşılamak ve tasarım tercihleri açısından tasarım kalitesini en üst düzeye çıkarmak, birbiri ile ilişkili nesneler ya da nesneler kümesi için uygun yerleri ve boyutları optimize ederek plan kurgusu oluşturulabilir. 1960'lı yılların ortalarında mimari plan problemlerinin bilgisayar ortamında çözüm üretmeye gidilmiştir. Gelişen teknoloji ile birlikte tasarım problemlerinin çözümü için algoritmik çözüm yaklaşımlarına sıklıkla başvurulmuştur.

Bu çalışma kapsamında yapılacak olan olimpik havuz plan şeması oluşturmak için öncelikle genetik algoritma kullanılmıştır. Ayrıca çalışma kapsamında amaca ulaşmak için uygulanan kısıtlamalar topolojik (yakınlık, açık ve kapalı alan) ve boyutsal (uzunluk ve genişlik) olmasının yanısıra seyirci sayısı önemli bir kısıt olarak belirlenmiştir. Tez kapsamında, Genetik algoritmaya dayalı olimpik yüzme havuzu plan şeması tasarımını bilgisayar ortamında geliştirmek için ASP.NET yazılım dili kullanılmıştır. Elde edilen yazılımın kullanıcı etkileşimli bir ara yüze sahip olup girilen veriler kapsamında yüzme havuzu plan kurgusu geliştirilmiştir.

2. GENETİK ALGORİTMA VE MİMARİ TASARIMDAKİ UYGULAMALARI

Genetik algoritmalar evrime dayalı algoritmaların bir türüdür. Bu bölümde evrimsel algoritmalar hakkında bilgi verilip ardından genetik algoritmaya dayalı olimpik ve yarı olimpik yüzme havuzlarının yerleşim planları oluşturulacağından genetik algoritma hakkında geniş bilgi verilecektir. Aynı zamanda genetik algoritmanın temel elemanları ve mimari tasarımdaki yapılan uygulamalar araştırılıp yapılmış örneklerden bahsedilecektir.

2.1 Evrimsel Hesaplama

Bir problemi çözmek için kullanılacak herhangi bir evrimsel algoritma aşağıda verilen beş maddeye ihtiyaç duymaktadır (Karaboğa, 2004).

1. Problem için çözümlerin genetik temsili (representation),
2. Çözümlerin başlangıç popülasyonunu oluşturacak bir yöntem,
3. Çözümleri uygunluk açısından değerlendirmeye tabii tutacak değerlendirme fonksiyonu (çevresel etkiler),
4. Genetik kompozisyonu değiştirecek operatörler,
5. Kontrol parametrelerinin değerleri (popülasyon büyüklüğü).

Doğada her birey belli karakteristik özelliklere sahiptir. Bu karakteristik özellikler ve yetenekler bireyin hayatta kalmasını ve çoğalmasını etkiler. Evrimsel algoritma problem çözümlemesinde tek bir bireyin karakteristik özelliğiyle değil, bireylerin popülasyonu ile ilgilenir. Her bireyin mevcut bir veri yapısı bulunur. Birey mevcut problem için muhtemel bir çözümü temsil eder. Bireylerin eşlenerek çoğalmasından sonra, yavru kromozomlar, her çiftten gelen veri yapısının baskın olan karakteristik özelliğinin bir kombinasyon sonucundan oluşur. Basit bir evrimsel algoritmanın temel adımları aşağıda şu şekilde verilmiştir (Karaboğa, 2004):

1. Başlangıç popülasyonunu oluştur.

2. Oluşturulan popülasyonu değerlendir.
3. Durdurma kriteri sağlanıncaya kadar aşağıda belirtilen işlemleri uygula
 - 3.1 . Popülasyonda oluşturulan bireylere seçme işlemi uygulayarak yeni popülasyon oluştur.
 - 3.2 Oluşturulan yeni popülasyonu değiştir.
 - 3.3 Değiştirilmiş yeni popülasyonu tekrar değerlendir.

Evrimsel hesaplamalar formülize edilişlerine göre farklı şekilde isimlendirilirler. Genetik Algoritmalar, Evrimsel Programlama, Genetik Programlama ve Evrim Stratejileri olarak adlandırılırlar.

2.2 Genetik Algoritma Tanımı ve Temel Kavramları

Genetik algoritma, doğal seçim (doğal seleksiyon), genetik aktarımı ve evrim sürecini taklit eden bir metasezgiseldir. İlk olarak John Holland tarafından 1975'te önerilmiştir ve daha sonra üzerinde çok sayıda geliştirme yapılmıştır. Genetik algoritmalar, metafiziksel fonksiyonların global optimizasyonunu hedefler (Hahnert, 1994).

GA'lar bir çözüm seti ile başlar ve sonra gelişimi için biyolojik evrimi esas alan bir süreç kullanılır (Glover, 1994).

Genetik algoritma parametreleri, biyolojideki genleri temsil ederken, parametrelerin toplu kümesinde kromozomu oluşturmaktadır. Genetik algoritmaların her bir ferdi kromozomlar (bireyler) şeklinde temsil edilen popülasyondan oluşur. Popülasyon uygunluğu, belli kurallar dahilinde maksimize ve minimize edilir. Her yeni nesil, rastgele bilgi değişimi ile oluşturulan diziler içinde hayatta kalanların birleştirilmesi ile elde edilmektedir (Angeline, 1995).

Genetik algoritmanın terminolojisinin anlaşılması için “doğal seçim” in (seleksiyonun) anlaşılması gerekir. Genetik algoritma, yönlendirilmiş rastsal araştırma algoritmalarının bir türüdür. Doğal seçim ile canlı organizmalarda bulunan genetik gelişiminin benzetişimini oluşturmaktadır. Genetik algoritmalar diğer evrimsel algoritmalarda olduğu gibi araştırma uzayında bulunan çözümler için bir başlangıç popülasyonu kullanılır. Yeryüzündeki canlı organizmaları gözlemlersek, organizmaların oluşumunda süregelen olaylarda doğal seçim göze çarpar. Birbirinden ayrı organizmalar ve bu organizmalardaki karmaşıklık, araştırma

konusudur. Organizmalardaki bu karmaşıklıklarının neden böyle olduğunu ve bu aşamaya nasıl geldiği incelenip sorgulanabilir. Başka bir açıdan bakıldığında, sanki yaşadığımız bu dünyada büyük optimizasyon algoritmaları oluşturulmuş ve binlerce iterasyon sonucunda yer yüzünde optimum çözüm elde edilmiştir (Grant, 1985). En son kuşağın en uygun bireyi, problem için optimal çözüm olmaktadır. Bu çözüm her zaman optimum çözüm olmayabilir fakat optimum çözüme yakındır.

Holland (1975), basit bit dizilerini kullanarak karmaşık yapıların kodlanabileceğini göstermiştir. Burada belirtilen yapılar, çözülmesi gereken problemler için çözümleri temsil eder. Karmaşık yapıların kodlanması ile oluşturulmuş basit bit dizileri, muhtemel tüm çözümleri içine alan araştırma uzayından alınır. Bu diziler veya çözümler genetik algoritmanın kullanacağı popülasyonu oluşturur. Verilen iterasyon sayısına bağlı olarak en iyi çözüm geliştirilebilir.

2.2.1 Genetik algoritma temel kavramları

Genetik algoritmalar, bir problemin çözümünde daha iyi sonuç elde etmek için bir yaklaşım gösteren geniş bir çözüm uzayı boyunca rastlantısal bir araştırma olanağı sağlar. Genetik algoritmanın çalışma prensibinde kullanılan temel kavramlar şu şekilde sıralanabilir.

Gen

Temel kalıtım birimidir. Kendi başına anlamlı genetik bilgi taşıyan en küçük genetik yapıdır. Genetik algoritmanın kullanıldığı programlama yapısında gen yapıları programcının tanımlamasına bağlıdır.

Kromozom

Birden fazla genin bir araya gelerek oluşturduğu diziye denir. Organizmanın her bir hücresi benzer miktarda kromozomlar içerirler. Her bir vücut hücresindeki kromozom miktarı farklıdır. Örneğin; sivrisinek de 6, kurbağada 26, insanda 46 ve alabalıkta 94 adet kromozom vardır. Genler, her biri farklı karakteristiği temsil eden iki fonksiyonel formda meydana gelir. Kromozomlar, alternatif aday çözümleri gösterirler. Kromozomlar genetik algoritma yaklaşımında üzerinde durulan en önemli birim olduğu için bilgisayar ortamında iyi ifade edilmesi gerekir (Curtis, 1975).

Popülasyon

Genetik algoritmalarla yapılan problem çözümünde çözüm uzayındaki her bir noktayı, kromozom adı verilen ikili bit dizisi ile kodlama yapılır. Kromozomlarla belirtilen bu noktaların her birinin bir uygunluk değeri bulunur. Genetik algoritmalar tek bir nokta yerine noktalar kümesini barındırır ve bu noktalar kümesine popülasyon denir. Popülasyon, alternatif çözüm kümesidir. Popülasyondaki birey sayısı (kromozom) genellikle sabit tutulur. Genetik algortmada popülasyon daki birey sayısı ile ilgili genel bir kural yoktur. Popülasyondaki kromozom sayısı arttıkça çözüme ulaşma süresi (iterasyon sayısı) azalır. Problemin özelliğine göre popülasyon sayısı belirlenir. Genetik algoritmalar, problem çözümlerinin kodlanmasını, uygunluk değerlerinin hesaplanmasını, üreme, çaprazlama ve mutasyon işlemlerinin uygulanmasını içerir (Jang, 1997).

Allel

Her bir genin bir özelliğini temsil ederek, genin alabileceği değişik değerlere denir. Her biri farklı karakteristiği temsil eden genler iki fonksiyonel formda meydana gelir. Bu formların her biri “allel” olarak bilinir (Curtis, 1975).

Lokus : Her genin kromozomda locus adı verilen belirli bir yeri vardır. Bilgisayar ortamında kullanılan genetik algoritmalarda, locus kromozom üzerindeki her bitin yerine verilen isimdir.

Genotip

Bir insan kromozomunda bulunan genlerin tümü, o insanın kalıtsal özelliğini yani genotipini oluşturur. Daha öncede belirtildiği gibi genetik algoritmalarda bireylerin oluşturduğu bir topluluk bulunmaktadır, ve bu topluluğa popülasyon adı verilmektedir. Popülasyon içerisinde, her bir bireyin kalıtsal özelliklerinin taşındığı kromozomlar bulunmaktadır. Yapay sistemlerde kromozomlar dizi (string) olarak belirtilmektedir. Yapay dünyada bir gen bir dizide özel bir karakter veya karakterler seti olarak belirtilebilir (Goldberg, 1989). Aynı zamanda kromozomlar bireyin kalıtsal özelliklerinin bilgilerini şifrelenmiş bir şekilde taşıdığı için bireyin genotipi olarak adlandırılmaktadır. Doğal sistemlerde genotip tek bir birey için birden fazla kromozomun birleşmesinden oluşabilir. Ancak yapay sistemlerde bir birey için tek bir kromozom bulunmaktadır ve bu sistemlerin genotipi kromozomun içinde taşıdığı özelliklerdir. Bunlar da yapı (structure) olarak gösterilmektedir (Goldberg, 1989).

Fenotip

Doğal yaşamda genotipi oluşturan şifrelenmiş bilgilerin, saç rengi, göz rengi, ten rengi gibi özelliklere dönüşmesiyle oluşan genetik yapıya ise fenotip adı verilmektedir. Yapay sistemlerde fenotip, parametre kümelerini oluşturur. Dolayısıyla genotip yapısı deşifrelendiğinde fenotipi oluşturacaktır (Goldberg, 1989).

2.3 Genetik Algoritmanın Temel Prensipleri

Genetik algoritma stokastik ve iteratif bir süreçtir. Algoritmanın başlangıcında çözüm uzayı rasgele veya sezgisel belirlenmektedir. GA'nın en önemli özelliği olan rasgelelik başlangıçtan genetik operatörlerin kullanılmasına kadar algoritma içerisinde sürdürülür. GA tekrarlanan bir süreçtir. Bu özellik uygun çözümün bulunmasına kadar algoritmanın işletilmesinde görülmektedir.

2.3.1 Rastgele arama algoritması

Rastgele Arama, belki de en basit arama işlemidir. Bir başlangıç arama noktasından veya başlangıç noktalarının kümesinden başlayan arama işlemi, arama uzayında rasgele noktaları araştırır ve kabul edilebilir bir çözüme ulaşıncaya veya maksimum iterasyon sayısı ulaşıncaya kadar devam eder. Rasgele aramayı gerçekleştirmek son derece basit olmakla beraber, verimsiz olabilir. Uygun çözüm elde edinceye kadar geçen zaman çok uzun olabilir. Rasgele araştırma için bir algoritma çalışması ve işleyişi aşağıda sunulmuştur.

Adım 1. N başlangıç arama noktaları kümesini seç. $C_g = \{ C_{g,n} \mid n=1,2,\dots,N \}$. Burada, $C_{g,n}$ I değişkenlerinin vektörü ve $g=0$ dır. Her bir $C_{g,n}$ elemanı, $U(\min, \max)$ değişken değerlerin sınırı olmak üzere, verilen aralıklarda üretilir.

Adım 2. Her bir $C_{g,n}$ vektörünün (“uygunluk”) $F(C_{g,n})$ doğruluğunu değerlendir.

Adım 3. Her bir $C_{g,n}$ vektörünün (“uygunluk”) $F(C_{g,n})$ doğruluğunu değerlendir.

Adım 4. if $C_g, best < C_{best}$ then $C_{best} = C_g, best$ C_{best} tümünün en iyi çözümü.

Adım 5. if C_{best} kabul edilebilir bir çözüm ise veya maksimum iterasyon sayısı aşılmış ise o zaman “dur” ve çözüm olarak C_{best} “dön”

Adım 6. Her bir $C_{g,n}$, $\Delta C_{g,n}$ ile karıştırılır. Burada, $\Delta C_{g,n} \approx N(0, \sigma^2)$ ve σ^2 li küçük bir değişimi ifade eder.

Adım 7. $g = g + 1$ artır ve adım 2 ye git.

2.3.2 Genetik algoritma operatörleri

Genetik algoritma diğer yapay zekanın optimizasyon yöntemlerinde olduğu gibi doğal olayları modelleyen bir tekniktir. Özellikle Darwin'in doğal seçme ve en iyi uyum sağlayanın hayatta kalması prensiplerinden ortaya çıkmıştır (Holland,1992). Bu yüzden ki özellikle temel prensipte yer alan çaprazlama (crossover), mutasyon (mutation) ve yeniden üreme (reproduction) genetik algoritmanın önemli operatörlerini oluşturmaktadır. Genetik algoritma operatörleri ile yapılan uygulamada izlenen ilk adım, ilk popülasyon oluşturularak, uyum değerleri hesaplanır. Daha sonraki adımda mevcut nesile temel genetik algoritma operatörleri olan çaprazlama, mutasyon ve yeniden üreme uygulanır. Elde edilen her nesil için uyum değeri hesaplanır. Bu durum şekil 2.1 deki akış şemasında görüldüğü gibi en iyi sonuç kriterleri sağlanana kadar devam eder.

Çaprazlama (Crossover)

İki adet yeni nesil elde etmek için kromozomların bulunduğu eşleme havuzundan iki adet kromozom seçilir. Eşleme sürecinde, seçilen kromozomlardan bir ve birden fazla yeni nesil oluşturma olayına “çaprazlama” denir. En yaygın olarak kullanılan iki kromozomdan iki tane yeni nesil elde edilmesidir.

Mutasyon (Mutation)

Kromozom yapısı içinde değişiklikler yapma işlemidir. Mutasyon işlemi ile yeni uygun çözümler elde edilmeye çalışılır. kromozomdaki yapının küçük bir yüzdesini değiştirir.

Üreme (Reproduction)

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 (Fırlıklı, 2002).

2.3.3 Genetik Kodlama

Bir problemin genetik algoritmaya dayalı çözümü için, problemin kendisine özgü bilgilerin genetik algoritmanın kullanabileceği yapıya çevrilerek çözüm kodlamasına dönüştürülmüş olur. Her problem kendi çözüm kümesine özgü farklı kodlamalara gereksinim duyabilir. Tüm problem çözümü için genel bir kodlama tekniği bulunmamaktadır. Genetik kodlama aşağıda belirtilen kodlama türleri ile incelenebilir.

2.3.3.1 İkili Kodlama

Problemdeki parametreleri temsil eden, kromozom yapısı içindeki genlerin dizilişinin gösterimi, daha çok ikili sistem (0-1 sistemi) veya diğer rakamların da kullanılmasıyla gerçekleştirilmektedir. Rakamların ikili sistem dışında kullanımında, problemin parametreleri gereği, sıralı veya sırasız rakamların birer defa kullanımının yanı sıra, aynı rakamların birden çok kullanımına da rastlanmaktadır.

Genetik algoritmalar, ikili (binary) dizi üzerinde temellenmiş ve adaptasyon işlemini modellemek üzere düzenlenmiştir. Mutasyon gibi temel operatörler yardımıyla bir rekombinasyon (yeni oluşum) operatörü kullanmaktadır. Evrimsel hesaplama stratejileri, özellikle parametre optimizasyon problemlerini çözmek için oluşturulmuşlardır. Bu yaklaşımdan yola çıkarak popülasyondaki her bir birey iki tane vektörden oluşmaktadır. ($m = x + y$). x vektörü, araştırma uzayındaki bir noktayı, y vektörü ise standart sapmalara karşılık gelmektedir. Evrimsel programlama metodları bireylerin kromozom yapılarını kullanır ve yeni birey oluşturmak için önceki bireylerin rastgele mutasyona uğraması ile oluşturdukları bireyleri ele alır.

Çaprazlama uygulamaları

Çapraz değişim ile ilgili uygulama şeklinde, önce kromozom yapı üzerinde rastgele bir ayırım noktası belirlenmektedir. Kromozomlar, çapraz değişim işleminde bu ayırım noktası öncesindeki gen yapısını aynen korurlarken, ayırım noktası sonrasındaki gen yapısını ise karşılıklı olarak değiştirmektedirler (Thomas.1995). Aşağıdaki örnekte de görüleceği gibi, ayırım sonrası genleri karşılıklı olarak yer değiştirmiştir.

Mesela, iki kromozom x_6 ve x_2 ikinci genden sonra çaprazlanabilir. Her biri iki yavru üretir. Aşağıdaki gibi gösterilir:

X_6	<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr></table>	1	0	0	1	X_6'	<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr></table>	1	0	0	0
1	0	0	1								
1	0	0	0								
X_2	<table border="1"><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>	0	1	0	0	X_2'	<table border="1"><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>	0	1	0	1
0	1	0	0								
0	1	0	1								
Ebeveynler		Çocuklar									

Şekil 2.2 : İkili kodlama ile çaprazlama uygulamaları (Croce, 1994)

Mutasyon uygulamaları

Kromozom yapısı üzerindeki dahili değişim, genellikle rastgele bir şekilde belirlenen bir gen'in değiştirilmesiyle gerçekleştirilmektedir. Bu uygunlukta önemli bir gelişmeye neden olabilir. Fakat çoğunlukla, oldukça faydalı sonuçları vardır. Mutasyonun rolü, arama algoritmasının bir local optimuma takılmamasının garantisini sağlamaktır. Seçim sırası ve çaprazlama operatörleri herhangi bir homojen çözüm kümesinde durgunlaşabilir. Böyle şartlar altında, tüm kromozomlar özdeşir ve bu yüzden popülasyonun ortalama uygunluğu geliştirilemeyebilir. Çözüm sadece optimal (veya local olarak oldukça optimal) olmak görünebilir. Çünkü arama algoritması daha fazla ilerlemeyebilir. Mutasyon rasgele bir aramaya eşdeğerdir ve genetik farklılıkların kaybının korunmasında bize yardım eder.

Mutasyon operatörü bir kromozomda rasgele bir şekilde seçilen geni çevirir. Mesela, aşağıda gösterildiği gibi, x_1 ' ikinci geninde ve kromozom x_2 de üçüncü geninde mutasyona uğratılmış olabilir.

X_1	<table border="1"><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>	1	1	1	0	X_1'	<table border="1"><tr><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	1	1	0	0
1	1	1	0								
1	1	0	0								
X_2	<table border="1"><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>	0	1	0	0	X_2'	<table border="1"><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr></table>	0	1	1	0
0	1	0	0								
0	1	1	0								
Ebeveynler		Çocuklar									

Şekil 2.3 : İkili kodlama ile mutasyon uygulamaları (Croce, 1994)

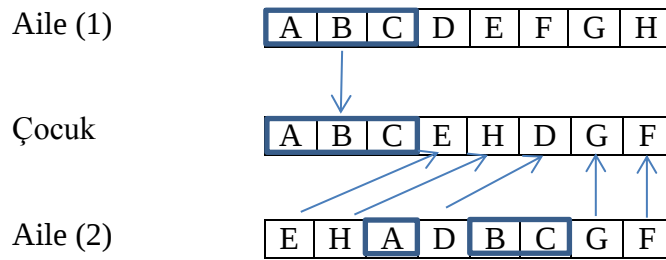
Mutasyon, bazı olasılıklı kromozomda, herhangi bir gende meydana gelebilir. Mutasyon olasılığı doğada oldukça küçüktür ve GA larda oldukça küçük tutulur. Tipik olarak 0.001 ile 0.01 aralığında (Thomas, 1995).

2.3.3.2 Alfabetik kodlama

Problemdeki Parametreleri temsil eden, kromozom yapısı içindeki genlerin dizilişinin gösterimi, alfabetik sembollerin kullanılmasıyla da gerçekleştirilmektedir. Bu gösterim şeklinde, rakamlarla gösterim şeklinde de olguyu gibi, sıralı veya sırasız harflerin birer defa ya da birden çok kullanımına rastlanmaktadır.

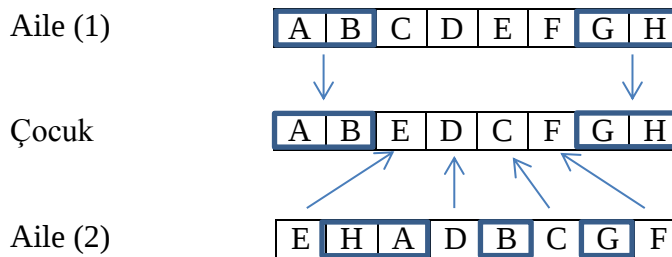
Alfabetik kodlamada çaprazlama uygulamaları

Çapraz değişim ile ilgili uygulamalar incelendiğinde, en çok rastlanılanların, aşağıdaki verilen dört örnekteki gibi olduğu görülecektir (Murata, 1996) Takip eden örnekte, tek ayırım noktasının söz konusu olması durumundaki çapraz değişim gösterilmektedir. Görüldüğü gibi, Aile (1)'den alınan (ABC) genleri, çocukta aynı sırada ve aynı pozisyonda yer alırken; (EHDGF) genleri ise, Aile(2)'den alınarak yine aynı sıra ile sıralanmaktadır.



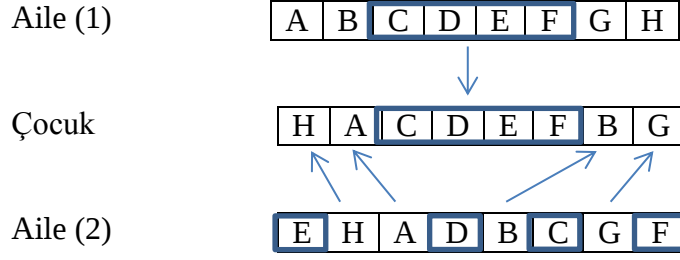
Şekil 2.4 : Alfabetik kodlama ile çaprazlama uygulamaları (Murata, 1996)

İkinci örneğimizde, iki ayırım noktasının olması durumunda, şekil 2.4' de görüldüğü gibi, Aile (1)'den alınan (AB) ve (GH) genleri aynı pozisyonda çocukta da yer alırken, Aile(2)'den alınan diğer genler (EDCF) de aynı sıra içerisinde çocukta da yer almaktadır.



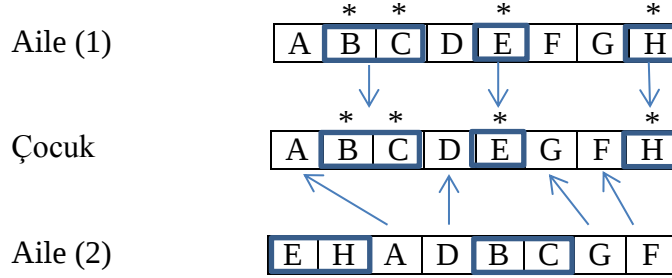
Şekil 2.5 : Alfabetik kodlama ile çaprazlama uygulamaları (Murata, 1996)

Üçüncü örneğimizde, yine iki ayırım noktasının olması durumunda, ancak bu defa Aile (1)'den alınan (CDEF) genleri aynı pozisyonda olmak üzere Çocuk'ta yer alırken, Aile (2)'den alınan diğer genler de aynı sırada Çocuk'ta yer almaktadır.



Şekil 2.6 : Alfabetik kodlama ile çaprazlama uygulamaları (Murata, 1996)

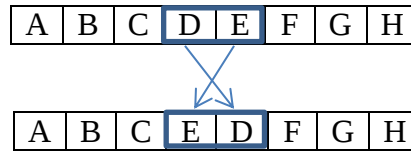
Son örneğimizde ise, Aile (1)’den rastgele seçilen “*” işaretli genler, Çocuk’ta kalıtsal olarak aynı pozisyonlarda yer alırken, Aile (2)’den alınan diğer genler (ADGF) de, kalan pozisyonları yine aynı sırada doldurmuştur.



Şekil 2.7 : Alfabetik kodlama ile çaprazlama uygulamaları (Murata, 1996).

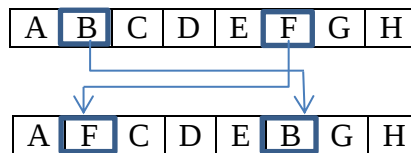
Alfabetik kodlamada mutasyon uygulamaları

Alfabetik kodlama ile yapılan mutasyon uygulamalarında, genellikle aşağıda verilen dört örnekte görüldüğü gibidir (Murata, 1996). Verilen ilk örnekte rastgele seçilen iki komşu genin, karşılıklı olarak değiştirilmesiyle mutasyon gerçekleştirilmektedir.



Şekil 2.8 : İki komşu genin karşılıklı olarak değişmesi (Murata, 1996)

Şekil 2.10 da verilen ikinci örneğimizde de görüldüğü gibi rastgele seçilen iki gen, karşılıklı olarak da değiştirilebilmektedir.



Şekil 2.9 : Rastgele seçilen iki genin karşılıklı olarak değişmesi (Murata, 1996)

Yeniden üreme (reproduction), çapraz değişim uygulaması (crossover), ve mutasyon (mutation) aşamalarının her tamamlanışında, sonuçların öncelikle karşılaştırılması gerekmektedir. Buradaki amaç en iyi kromozomu elde etmek; başka bir ifadeyle, en

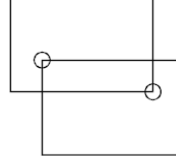
iyi çözüme ulaşmaktır. Bu nedenle, her çevrimde elde edilen yeni kromozomların uygunluk dereceleri belirlenerek, sıralamaya dahil edilirler. En iyiler 'Aile' görevi ile yeni nesil üretimine devam ederek, daha iyi nitelikli 'çocuk' elde edilmesine çalışırlar. Bu süreç, amaç gerçekleşinceye kadar devam eder. Bir eşleme havuzuna kopyalama işleminden ve havuzda bireyleri çiftler halinde gruplara ayırma işleminden oluşur (Fırlı, 2002).

2.4 Mimari Tasarım Uygulamalarında Genetik Algoritmaların Kullanımı

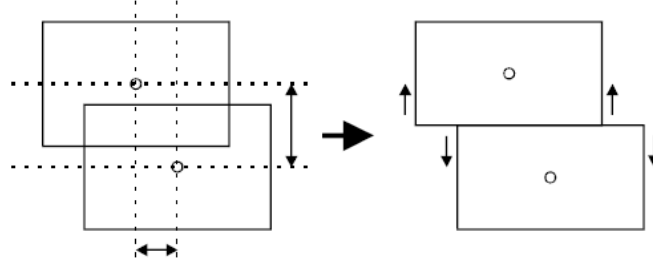
Darwin'ın evrim teorisinden esinlenerek oluşturulan ve evrimsel hesaplama tekniklerinden biri olan genetik algoritma yöntemi karmaşık problemleri hızlı ve optime yakın olarak çözebilen ve aynı zamanda karmaşık problem tiplerinde uygulanabilmektedir. Genetik algoritmalar; fonksiyon optimizasyonu, çizelgeleme, mekanik öğrenme, tasarım, hücresel üretim gibi alanlarda yapılmış uygulamalar görülmektedir. Genetik algoritma özellikle çözüm uzayının geniş, süreksiz ve karmaşık olan problem tiplerinde olasılık kurallarına göre çalışıp sadece amaç fonksiyonuna ihtiyaç duyarak başarılı sonuçlar verirler. Böylece, etkin bir arama yaparak çok daha kısa bir sürede çözüme ulaşırlar (Goldberg, 1989).

Var olan bir tasarım iyileştirilirken, sadece seçilen parametre değerlerinin değiştirilmesine gerek duyulur fakat, genetik algoritma ile yapılan tasarımlarda genetik algoritmaların tasarımın her bir parçasını değiştirebilmelidir (Bentley ve Wakefield, 1995). Yeni tasarımların oluşması için kullanılan genetik algoritmalar en uygun tasarım seçeneklerini çözüm uzayında otomatik olarak oluştururlar. Bu durum tasarımcıya oldukça faydalı ve kolaylaştırıcı seçenekler sunar. Örnek olarak, çok katlı bina bloklarının oluşumunda genetik algoritma yöntemi kullanılarak blok tasarımında yeni kavramsal tasarımlar oluşturabilir.

Bentley ve Wakefield, (1995) aykırı tasarımların düzeltilmesi için çeşitli alternatif yöntemler uygulamışlardır. Şekil 2.10 da verilen iki nesneden birinin köşesi diğer nesnenin üzerinde yer almasıyla kesişen bu iki nesnenin pozisyonları birbirine çok fazla yakındır diyebiliriz. Bu nesnelerin görünüşlerindeki fark x doğrultusunda ise, nesnelerin genişlikleri yani x değerleri azaltılır, eğer fark y doğrultusunda ise bu defa yükseklikleri azaltılır, z doğrultusunda fark olduğunda derinlik azaltılır



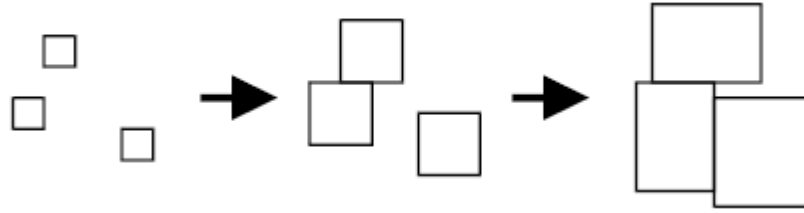
Şekil 2.10 : İki nesnenin üst üste gelme durumu (Bentley ve Wakefield, 1995)



Şekil 2.11 : Üst üste gelen nesneler (Bentley ve Wakefield, 1995).

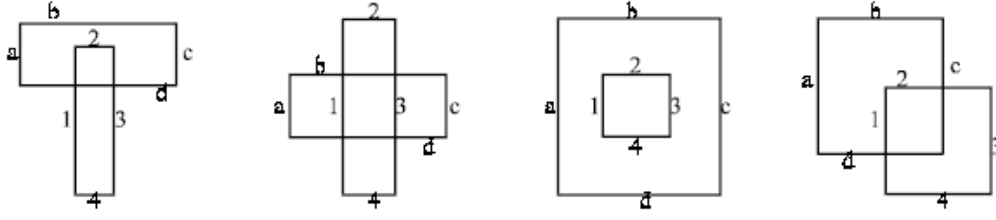
Şekil 2.11 de görüldüğü gibi üst üste gelmiş nesneler arası y doğrultusu fazla ve yükseklik değiştirilerek tasarım düzeltilir.

Bir diğer çözüm ise merkez nokta pozisyonları genotipte belirtilen biçimlerin tek bir nokta olarak hayata başlamasıdır. Şekil 2.12 de verilen bu noktalar küçük oranlarla büyütülüp kenarların birbirine temas etmesiyle çözüm uzayındaki büyüme durdurulur. Büyüme sürecinde genotipler fenotip olarak yola devam ederler.



Şekil 2.12 : Tek bir noktanın büyüyen nesnel biçimler (Bentley ve Wakefield, 1995)

Son olarak üst üste gelen nesnelerin sıkıştırılması yöntemi ise daha önce verilen iki yöntemin birleşimi ile elde edilen bir yöntemdir diyebiliriz. Nesnelerin üst üste geldikleri tespit edildiğinde 1.yöntem uygulanır, bu yöntem uygulandıktan sonra hala düzelme görülüyorsa 2.yöntem kullanılır.



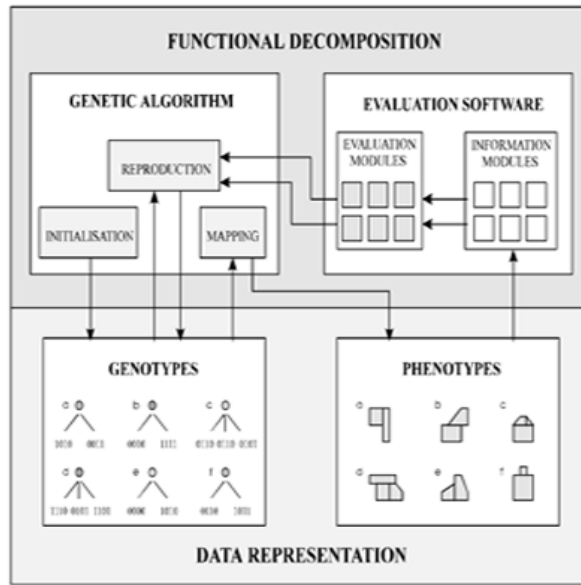
Şekil 2.13 : Üst üste gelen ilkel biçimler (Bentley ve Wakefield, 1995)

Şekil 2.13 de görüldüğü gibi eğer 1 ve 3 kenarları a ve c arasında yer alırsa, ya da eğer a ya da c kenarı 1 ve 3 arasındaysa ve eğer 2 yada 4 ,b ve d arasındaysa ya da b ya da d 2 ile 4 arasındaysa, bu iki ilkel biçim üst üste biniyor demektir (Bentley ve Wakefield, 1995).

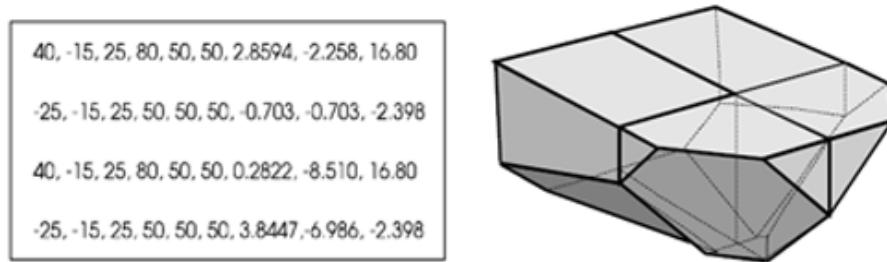
Bir diğer örneğimizde üç boyutlu katı cisimlerin genetik algoritma kullanılarak katı cismin nasıl deformasyona uğradığı gösterilecektir. Bu konuda yapılan tasarım çalışmalarında, problem çözümlerinde genetik algoritma kullanılırken 4 ana unsur dikkate alınmıştır. Öncelikle uygulama yapılacak katı cisim için algoritmadaki fenotipler belirlenmiştir. İkinci olarak, problem çözümünde kullanılacak olan genotipler tanımlanır, daha sonraki aşamada, problemde uygulanacak en uygun algoritma belirlenir ve son olarak problemin çözüm uzayında potansiyel çözümlerin değerlendirilmesi için uygunluk fonksiyonları oluşturulur (Bentley, 1996).

Genetik algoritma hiyerarşik genotipleri (kodlu çözümleri) yönetir ve genetik algoritmada belirlenen genotipler düşük parametrelili uzaysal-bölümleme gösterimi ile tanımlanan fenotiplerle eşleştirilir. Şekil 2.14 de görüldüğü gibi belirtilen 4 ana unsurun kombinasyonu ile farklı katı cisimlerin evrimsel tasarım diyagramı verilmektedir.

Bentley ve Wakefield'in şekil 2.15 te katı cisimler üzerinde yaptığı çalışmalarda küp ve kare bloklarındaki genişlik derinlik ve yükseklikler için değişken atanmakta aynı zamanda her blok için 9 fenotip tanımlanmaktadır. Tanımlanan bu fenotipler belli değerlerde evrim geçirerek farklı bir görünüş sergilemektedirler.



Şekil 2.14 : Evrimsel tasarım sistemi diagramı (Bentley ve Wakefield, 1995)



Şekil 2.15 : Fenotip değerleri değiştirilen bir cismin evrimsel tasarımı (Bentley ve Wakefield, 1996)

Şekil 2.15 de görüldüğü gibi katı bir cismin fenotip değerleri mutasyona uğrayarak farklı bir görünüme girmektedir.

Tez kapsamında yapılacak olan yüzme havuzu plan kurgulamasında bize yardımcı olacak bir diğer örnek M. Thakur ve M. Kumari'nin genetik algoritma kullanarak tek katlı konut modeli uygulamasıdır. Bu çalışmada, belirlenen alan içerisinde yatak odası, mutfak, balkon, yaşam odası, ve yemek odasının yer alması düşünülmüştür. Model içerisinde topolojik (Yakınlık, açık ve kapalı alan) ve boyutsal (uzunluk ve genişlik) olmak üzere iki seviyede kısıtlama kullanılmıştır. Genetik algoritmalar kurgu içerisinde alanın topolojik düzenlemesini üretmek için kullanılmış ayrıca gerektiğinde boyutsal analizinin uygunluk değerlerinin de kontrolü öngörülmüştür.. Önerilen modelde genetik algoritma yöntemi kullanılarak çok katlı bina içerisinde yer alan bir dairenin mimari planı oluşturulmaya çalışılmıştır. Şekil 2.16 da verilen mekan planlamasının adımları incelendiğinde ilk planların topolojik kısıtlamalar

kullanılarak oluşturulduğu, ardından planın kullanıcı tarafından seçildiği ve kullanıcının her talebi için plan uygunluğunun boyutsal analizinin yapılması önerilmiştir. Verilen doluluk, kesişen yerler, açık ve kapalı mekan kısıtlamaları mimari plan kurgusunun topolojik üretimi boyunca kullanılmaktadır.

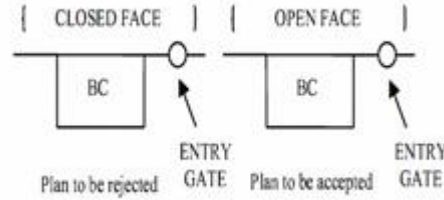
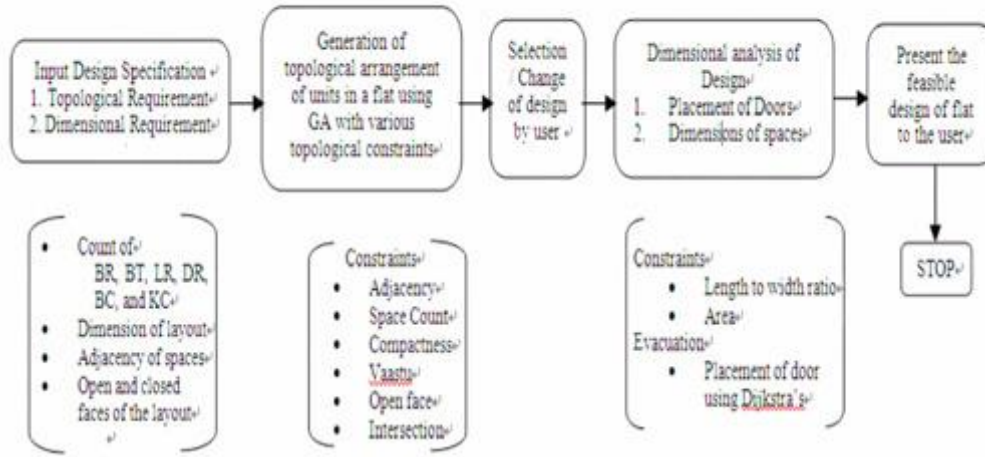
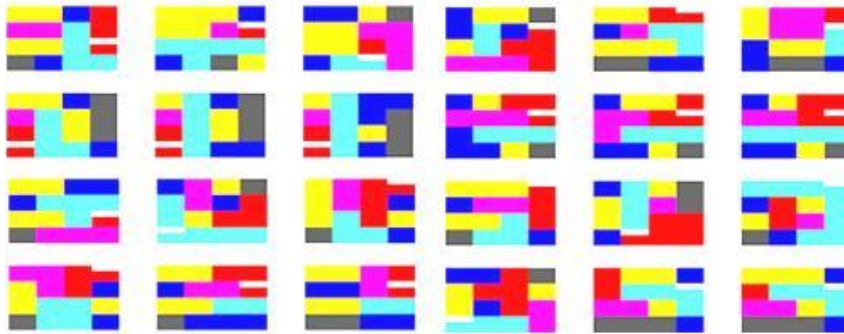


Figure 1. Selection or rejection of plan based upon open face constraint.



Şekil 2.16 : Mimari plan üretimi için genel sistem tasarımı (M. Thakur ve M. Kumari)

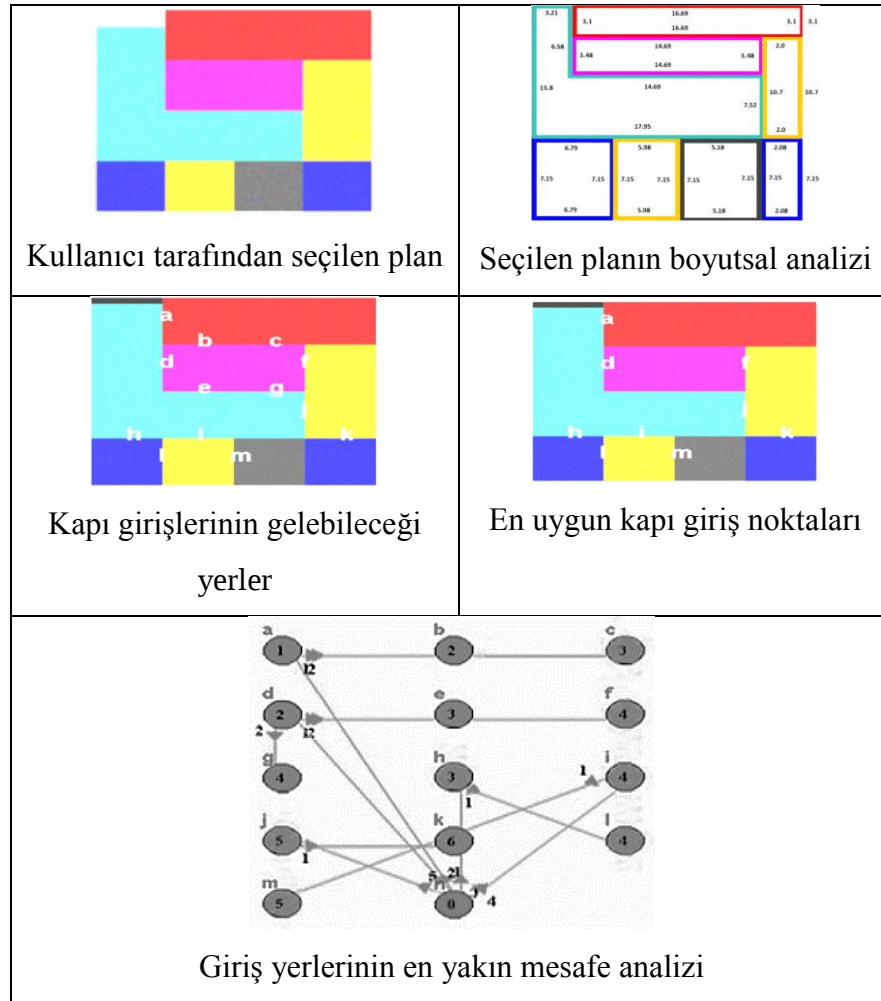
Önerilen model, 4 X 4 matrix, her bir mekan için belirtilen uzunluk kısıtlamaları, tek nokta çaprazlama kullanılarak üretilmiştir.



Şekil 2.17 : Kullanıcı tarafından seçilecek planlar (M. Thakur ve M. Kumari)

Kullanıcı bilgisayar ortamında üretilen planlardan herhangi birini seçtiğinde, modelin boyutsal analizi yapılmakta ve genetik algoritma uygulaması ile alternatif planlar

üretilmektedir. Daha sonra üretilen plan şeması üzerinde giriş için en uygun nokta program tarafından çözümlenmektedir.



Şekil 2.18 : Seçilen planın analizi ve sonuç ürünü (M. Thakur ve M. Kumari)

Şekil 2.18 de görüldüğü gibi seçilen model boyutsal olarak analiz edilip belirlenen alan üzerinde düzgün nesneler en uygun bir şekilde yerleştirilmeye çalışılmış ve her bir mekanın ana giriş kapısına olan mesafe uzaklıkları hesaplanıp en uygun kapı girişleri belirlenmiştir.

2.5 Kısıt Sağlama Problemleri

Çalışma kapsamında yapılacak olan yüzme havuzlarının plan kurgulamasında tek bir çözüm üretmek yerine genetik algoritmaların farklı çözüm kümeleri üretmesinden yararlanılacaktır. Üretilen bu çözüm kümelerinin plan kurgulamasını oluşturabilmesi için aynı zamanda kısıt sağlama problemlerinden (Constraint Satisfaction Problems) yararlanılmıştır. Böylece, çözüm uzayında aynı anda birçok nokta değerlendirilmekte

ve verilen kısıtlamalarla elde edilecek sonuç bütünsel bir plan kurgusu çözümüne ulaşma olasılığını yükseltecektir.

Yapay zeka içerisindeki bir çok problem kısıt sağlama problemleri (Constraint satisfaction problems) olarak modellenebilmektedir. Bu yüzden CSP'ler için etkin çözüm tekniklerinin geliştirilmesi önemli bir araştırma problemidir. CSP uygulamaları, doğal işlemleri ve veri tabanında optimizasyon sağlama gibi birçok alanda görülebilmektedir. Teknolojinin çok hızlı değiştiği, her alanda rekabetin arttığı ve karmaşık hale gelen sistemlerin oluşturduğu problemlerin günümüzde klasik yöntemlerle (analitik veya sayısal) çözümlerin güçleşmesinden dolayı optimizasyon kavramının her zaman güncel kalmasının en önemli sebeplerinden biridir. Optimizasyonun en basit tanımına bakacak olursak, belirli kısıtlamaları sağlayarak, bilinmeyen parametre değerlerinin bulunması için herhangi bir problem, optimizasyon problemi olarak adlandırılabilir. Optimizasyon, verilen bir problemin çözümünü bulmak için en iyi değerini veren kısıtlardaki değişkenlerin değerini bulmaktır. Bir başka deyişle optimizasyon en iyi sonuçları içeren işlemler topluluğudur. Bir işin yapılmış olması demek o işin en iyi şekilde yapılmış olduğu anlamına gelmez. Aslında buradaki en iyi tanımlaması izafi bir kavramdır. Optimizasyon ile elde edilen optimal çözümler, bir bakıma problemi formülize eden kişiye bağlıdır. İnsanların eğitim düzeyleri, ideolojileri, bilimde geline nokta, sosyal durumlar “en iyi” tanımlamasında etkili faktörlerdir.

Genel bir optimizasyon probleminin çözümü altı adımda gerçekleştirilir.

1. Yapılacak işlem analiz edilir ve işlem değişkenlerinin listesi çıkarılır. Örneğin bir kapalı spor salonu tasarımı planlanırken seyirci kapasitesi, servis alanları, park yerleri gibi kriterler belirlenmelidir.
2. Optimizasyon için amaç fonksiyonunu tanımlayacak kriter belirlenir. Optimizasyonun tek parametrelimi yoksa çok parametrelimi olduğu belirlenir boyut sayısı arttıkça optimizasyonun zorluk dereceside artar.
3. Problemin çözümü için yapılacak işlemler matematiksel ifadeler kullanılarak çözüm bulunabilir.
4. Problem çok büyük ve karmaşık ise;
 - a) Model basite indirilmeye çalışılır.
 - b) Amaç fonksiyonu tekniği matematiksel ifadelerle uygulanmaya çalışılır.
5. Problem çözümü için uygun optimizasyon tekniği matematiksel ifadeye çevrilir.
6. Sonuçlar kontrol edilir.

Bir problemin birden fazla çözümü olabilir. Örneğin bir binanın tasarımında çok sayıda model üretilebilir. En iyi tasarım sonucunu belirleyebilmek için alternatif modelleri karşılaştıracak bir veya birkaç kriter olmalıdır. Bu tür kriterlere hedef fonksiyonu denir. Hedef fonksiyonu isteğe bağlı olarak ya maksimum değeri alınır veya minimize edilir.

Bir CSP genel olarak sınırlı sayıda değer ve kısıtlamalarla dolu bir set içeren sınırlı sayıda veri içermektedir. Her kısıtlama esas bir veri kümesinin alt kümesi üzerinde belirlenmiştir. Belirlenen bu verilerin birbiri ardına alabileceği değerleri kısıtlamaktadır. Görev eşleşmelerin tüm kısıtlamaları karşıladığı verileri değerler ile eşleştirmektedir. Bazı problemlerde hedef bu gibi tüm eşleşmeleri bulmaktır. Dünyadaki gerçek sorunların büyük bir kısmı CPS olarak formüle edilebilir. Buna örnek olarak, ülke planlaması yapılırken düşünülen kaynak ayırımı için örnek ülke planlamasının yapılması veya yapılacak olan sınavların belirlenmiş zaman dilimleri içerisinde takvimlendirilebilir. Sınırlı sayıda ders sınıfları varken her sınav bir sınıf gerektirmektedir. Farklı sınıfların farklı kapasitesi vardır ve bir sınavın gerçekleştirilmesinin tek yolu bu sınavı alacak olan öğrenciler için yeterli sayıda koltuğu olan sınıflarda yapmaktır. Bazı öğrenciler birden fazla sınava katılabilmekte ve bu sınavlar aynı zaman dilimi içinde takvimlendirilememektedir. Bu sorunu modellemek için her sınavı bir veri, olası zaman dilimleri ile sınıfları onun etkinlik alanı ve kısıtlamalar olarak ta belirli sınavların aynı zamanda gerçekleştirilememesini yapabiliriz. Kaynak ayırmanın daha karışık ama daha gerçekçi bir örneği havaalanı kapı ayırmasıdır. Genellikle hem fiziksel kısıtlamalar (örneğin belirli körükler yalnızca belirli türdeki uçak karşılayabilir) ve kullanıcı tercihlerinin (örneğin farklı hava yollarının hangi apron bölümlerini tercih etmeleri) dikkate alınması gerekir. Bu konuyu aşağıda verilen uygulamalı örneklerle problemin kısıt sağlama ile nasıl çözümlendiğini anlayabiliriz.

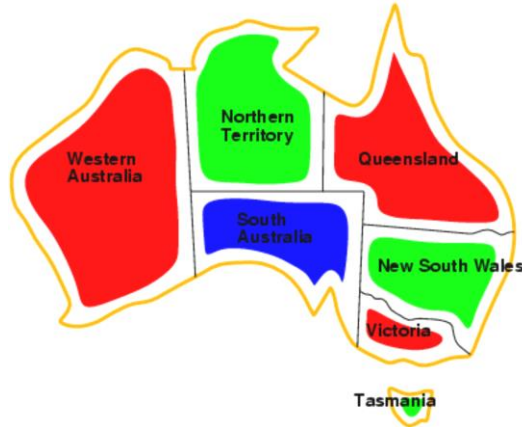
İlk olarak harita renklendirmesi örneğini verecek olursak, aşağıda verilen haritada birbirine komşu olan altı farklı bölge ve bu altı farklı bölgeden bağımsız olan bir bölge mevcuttur. Sınır çizgileri ile belirtilen her bölge bir değişken olarak atanıyor. Verilen kısıtlara uyacak çözüm yollarını bulmak için $X_1, X_2, X_3, \dots, X_n$ değişkenler kümesi ve $C_1, C_2, C_3, \dots, C_m$ kısıtlamalar kümesi oluşturulur. Her bir X_i için oluşturulan değerler alanı, belirtilen D_i değerleri değişkenlere atanır. Her X_i değişkeni D_i etki alanında değerleri olan X_i değişkenleri ile tanımlanır. Her

kısıt değişkenlerin bir alt kümesini içerir ve bu alt grup için izin verilen kombinasyonları belirtir.



Şekil 2.19 : Avustralya kıtası bölgeleri

Yukarıda verilen haritada sınırları belirtilmiş her bir bölge için bir değişken tanımlanır: *WA, NT, Q, NSW, V, SA, T*. Her bir değişken kümesinin etki alanında renkler kümesi oluşturulur : *red, green, blue*. Verilen kısıtlamalarla her bölge komşu olduğu bölge ile farklı renkte olmasını gerektirir. Örnek olarak, $WA \neq NT$. *WA* ve *NT* bölgelerine atanan değişkenler şu şekilde çiftli kombinasyon oluşturabilir $\{(red, green), (red, blue), (green, red), (green, blue), (blue, red), (blue, green)\}$.



Şekil 2.20 : Avustralya kıtası bölgeleri

Elde edilen çözümler $WA = kırmızı$, $NT = yeşil$, $Q = kırmızı$, $NSW = yeşil$, $V = kırmızı$, $SA = mavi$, $T = yeşil$ olarak istenilen bir sonuç elde edilmiştir.

	A	B	C	D
0			♔	
1	♔			
2				♔
3		♔		

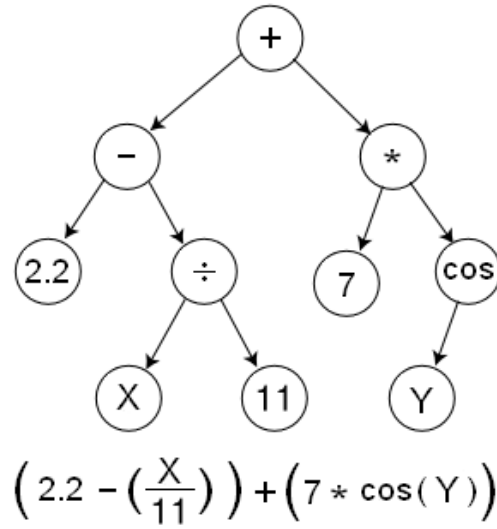
Şekil 2.21 : 4 lü N-Queens problemi

Şekil 2.21 deki başka bir örnek ile konuyu pekiştirecek olursak burada N-Queens problemini örnek olarak verebiliriz. N-Queens problemi kısıt sağlama problemi olarak modellenebilir. Verilen N tamsayısı $N * N$ santraç tahtası üzerinde N farklı karelerde santrac taşlarını yerleştirmektir. Bu durumda kısıtlamayı sağlayan her bir santrac taşı bir diğerini tehdit etmeyecek şekilde yerleştirilir. Aynı zamanda santrac taşlarının aynı satır, sütun ve çapraz durumda yan yana gelmesi durumunda bir birini tehdit ediyor anlamına gelir.

2.5.1 Kısıt Sağlama Problemleri İçin Genel Yaklaşımlar

Kısıt sağlama problem çözümünde, yaklaşım metodlarının bir çoğu “generate - and - test “ metodudur. Değişkenlere atanan olası her bir değer sistematik olarak oluşturulur. Daha sonra bütün kısıtlamaların yeterli olup olmadığını görmek için test edilir. Bütün kısıt sağlama problemlerini bulmaya çalıştığımızda düşünülen atama sayısı, bütün değişken alanlarının kartezyen çarpımının boyutundadır. Böylece, bu yaklaşımın zaman karmaşıklığı değişken sayıları içerisinde üst değerdedir.

Genel olarak kısıt sağlama problem çözümleri için üç standart yaklaşım vardır bunlar, ağaç arama, kısıt yayılımı ve geriye dönük aramadır Ağaç arama algoritma metodu kısıt sağlama problemi için standart bir tekniktir. Buradaki ağaç arama metodunun temel algoritma yapısı basit bir geri izleme (backtracking) metodudur. Ağaç arama algoritması kısıt sağlama problem çözümü içerisinde bir çok algoritma için temel bir metod olarak görülür.



Şekil 2.22 : Ağaç dal metodu

Tez kapsamında yapılacak olan yüzme havuzu plan kurgusu oluşturulurken, model içerisinde belirlenen mekanlar ağaç dal modeli kullanılarak, mekanlar arası ilişkiler oluşturulacaktır. Böylece bilgisayar ortamında yapılacak olan modelin arayüzü hazırlanırken, ağaç dal modelindeki adımlar takip edilerek kod yazımında büyük bir kolaylık sağlayıp belirlenen alan içerisindeki oluşturulacak havuz modelinin mekan yerleşimi doğru bir şekilde planlanması yapılacaktır.

3. OLİMPİK YÜZME HAVUZLARI TASARIM KRİTERLERİ VE ÖRNEKLERİ

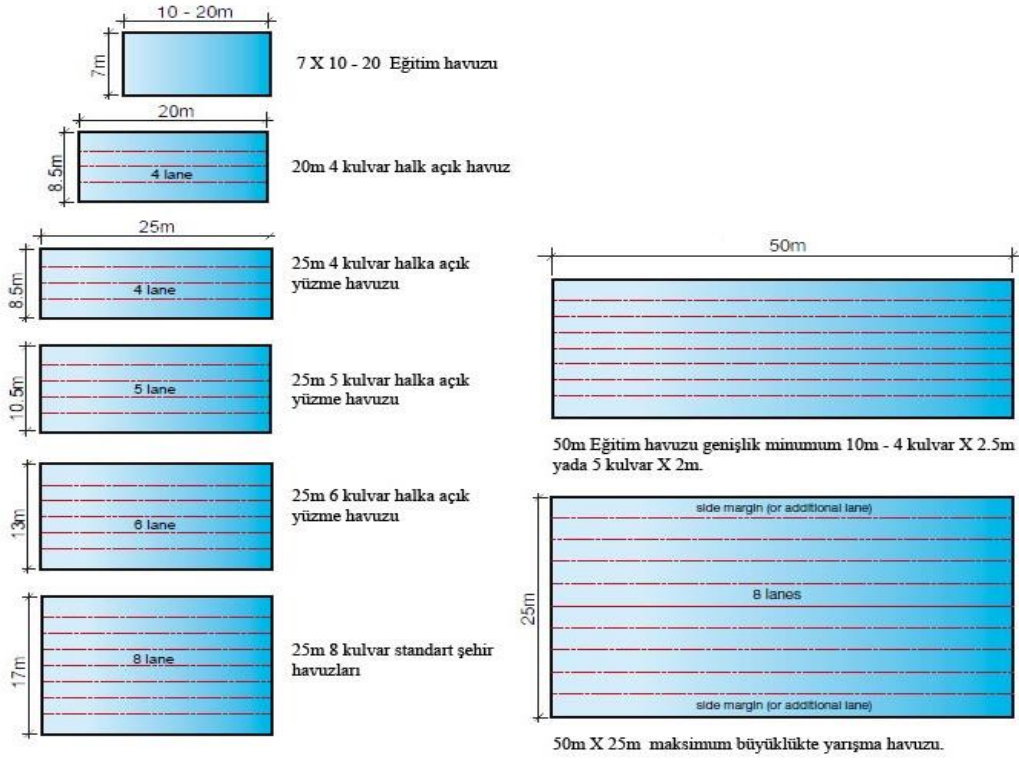
Spor amaçlı kullanımının yanı sıra, insanların eğlenme ve serinleme ihtiyaçlarını da karşılayan FINA'nın (Federation Internationale de Natation Amateur) belirttiği kurallar çerçevesinde inşa edilen yönetmenliklere uygun bir şekilde suyunun sürekli devir daim akışı olan ve halkın kullanımına açık olan yapılara yüzme havuzu denir (Yüzme Havuzlarının Tabi Olacağı Esas ve Şartlar Hakkında Yönetmelik,2011).

Olimpik yüzme havuzu : Uzunluğu 50 m olan bu havuz türleri 25 m genişliğe sahiptir. Bu tür havuzların derinliği ise asgari 2 m olup, her biri 2,5 m genişliğe sahip toplam 10 kulvardan oluşmaktadır. Yarışma esnasında kenarlardaki dalgalanmanın yüzücüleri olumsuz yönde etkilememesi için yüzme müsabakalarında sadece 8 kulvar kullanılmaktadır.

Yarı olimpik yüzme havuzu : Uzunluğu 25 metre ve genişliği 12,5 metre olan bu havuz türleri asgari 2 metre derinliğe sahiptir. Yarı olimpik yüzme havuzlarında her biri 2,5m genişliğinde, 5 kulvardan oluşmaktadır.

3.1 Havuzların Sınıflandırılması

Şekil 4.1 deki farklı boyutlarda ve özellikte verilen havuz tipleri kullanım amaçlarına göre farklılık göstermektedir. Bunlar; yapılarına göre havuz türleri, kullanım amacına göre yüzme havuzları, kullanılan suyun cinsine göre yüzme havuzları olarak sınıflandırılabilir ve yapılan bu sınıflandırmalar havuz boyutlarını da etkiler. Bu tez kapsamında olimpik ve yarı olimpik yüzme havuzu plan kurgulaması yapılacağından yapılarına ve kullanım amacına göre yüzme havuzları incelenecektir.



Şekil 3.1 : Havuz Tipleri (Green guide dergisi, 2011).

3.2 Olimpik Yüzme Havuzlarının Mekansal Özellikleri

1896'da modern olimpiyat oyunlarının tekrar başlatılması ile düzenlenen ilk olimpiyatlarda yüzme yarışlarına da yer verildi. Bütün dünyada örgütlü bir spor olarak yaygınlık kazanması ve olimpiyat programına alınması ile birlikte, bu spor dalı için uluslararası bir federasyon kurulması gerekliliği ortaya çıktı. Bu gereksinimden dolayı 1909'da Londra'da Uluslar arası Amatör Yüzme Federasyonu FINA (Federation Internationale de Natation Amateur) kuruldu. FINA'nın kurulmasından önce olimpiyatlarda yer alan yüzme yarışları günümüzdeki olimpiyat yarışmalardan çok uzaktı. 200 m engelli yüzme yarışları, bir direğe tırmanmayı ve bir dizi kayığın üstünden geçtikten sonra, bu kayıkların altlarından yüzerek geçmeyi içeriyordu. Diğer yarışlar ise, su altında en uzun mesafe yüzme, 4000 m yüzme gibi yarışlardı. FINA'nın kurulmasıyla FINA yönetmeliği esas alındı. Bununla birlikte esas alınan kurallara uygun olimpik yüzme havuzlarında mekansal özellikleri oluşturuldu (www.yuzmeegitimi.com).

3.2.1 Olimpik yüzme havuzlarının üst mekanlarının planlanması

3.2.1.1 Islak mekanlar

Bir yapının mimari planlama aşamasında ilk düşünülen şeylerden biri de ıslak hacim alanlarının planlanmasıdır. Spor salonlarında soyunma alanı, WC ve duş bölümleri olarak bilinen ıslak mekanların planlanması aşağıda geniş bir şekilde verilmiştir.

Soyunma alanı : yüzme havuzu spor salonlarında soyunma alanlarının büyüklüğü sporcu kapasitesine bağlı olarak değişebilmektedir. Soyunma alanları su yüzeyine paralel olarak sporcu kapasitesine göre belirlenir. 1000m² su yüzeyi kapasitesinde en az 10 soyunma kabini bulunmalıdır (Kurt, 2012).

Tuvaletler : ANSI standartlarının belirlediği tuvalet sayıları su yüzeyine paralel olarak sporcu ve seyirci sayısına göre belirlenmiştir. 1000m² su yüzeyi olan bir yüzme havuzu salonunda erkek tuvaleti için 2 adet WC ve 4 pisuar gerekirken, bayanlar için 4 adet WC bulunması gerekir. Ayrıca ANSI standartlarında ilk 100 erkek kullanıcı için 1 WC, 1 pisuar ve 1 lavabo olmalı. Eklenecek her 100 erkek kullanıcı için, 1 WC, 1 pisuar ve 1 lavabo daha eklenmelidir. Bayanlarda ise her 100 bayan için, 2 WC, 2 lavabo olmalıdır (swimming pools design guidance dergisi, 2011).

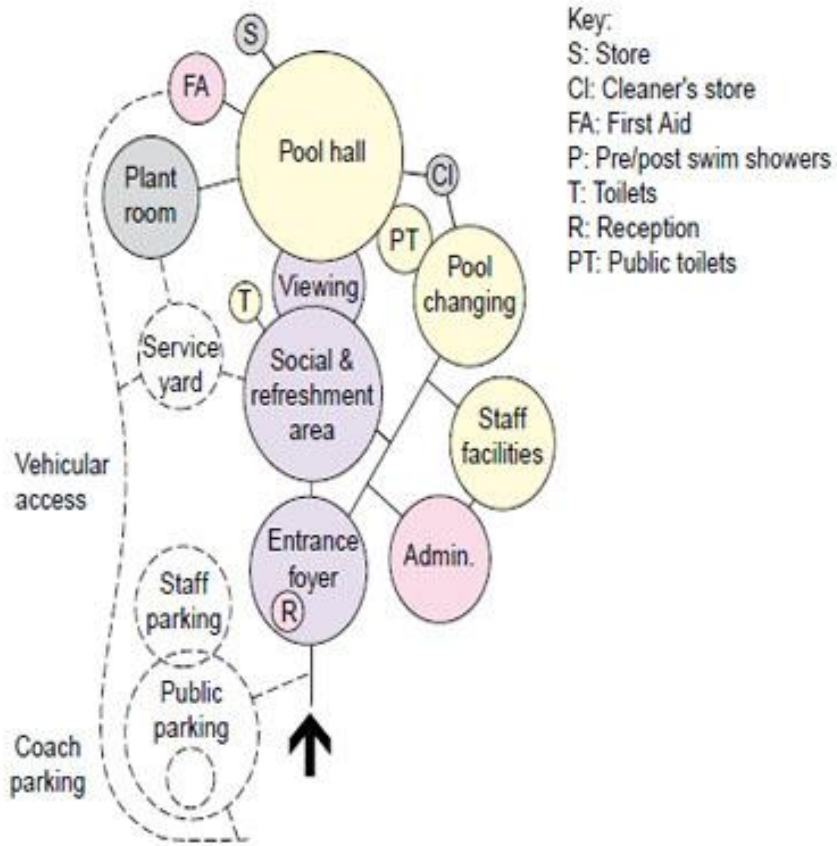
Duş Bölümü : Her 10 kişi için 1 adet duş yeri ve ön mahallerde el yıkama lavaboları olmalıdır (swimming pools design guidance dergisi, 2011). Duş bölümleri genellikle ayırma duvarı olan ve ayırma duvarı olmayan olmak üzere 2 tipten oluşmaktadır. Duş alanındaki duş kabinleri karşılıklı olarak tasarlanmışsa duşlar arasındaki koridor genişliği 1,10m, tuvalet kapısının içeriye doğru açılır olması uygun bir kurgu yapısı olarak yapılmış olur. Duş alanları cinsiyete göre ayrılmış olmalı ve de mümkün olduğunca duş alanlarının soyunma bölümüne yakın olmalıdır (Kurt, 2012).

3.2.1.2 Konum, ana giriş, fuaye ve tribünlerin planlanması

Konum belirlenmesi: Bir spor tesisinin yapım aşamasından önce nerede yapılması gerektiğine ilişkin bazı kriterler belirlenmelidir. Bu kriterler belirlenirken teknik açıdan ve de sosyo-kültürel yönden değerlendirme yapılabilir. Yapılacak olan yüzme havuzu spor tesisinin bu dalda büyük bir potansiyelin olup olmadığına, belirlenen seyirci kapasitesi için yeterli arazinin olmasına, uluslararası yarışmaların yapıp yapılmayacağına, eğer yapılacaksa uluslararası normların oluşturulması, bölgedeki

iklim şartlarına göre tasarım yapılması gibi kriterler tesis yapımının konumunu belirlemede önemli etkenlerdir.

Ana giriş: Spor tesisinin bulunduğu konuma göre ana giriş belirlenir. Bu kriterler belirlenirken izleyici ve sporcuların daha rahat giriş çıkışlarının olması en önemli etkenlerden biridir. Aynı zamanda seyirci kapasitesi göz önünde bulundurularak merdiven genişliği ve turnikelerin direk seyirci akımını çıkışlara yönlendirecek şekilde konumu oluşturulur (Kurt, 2012).

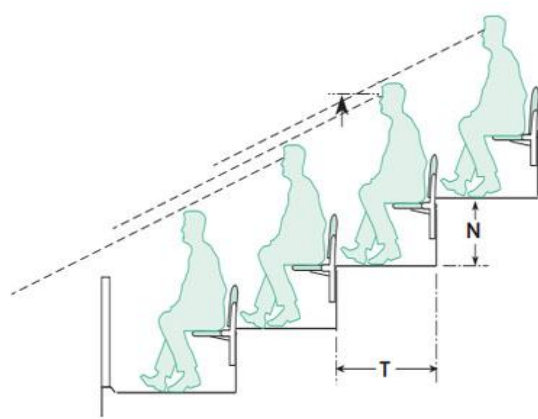


Şekil 3.2 : Tipik bir havuz binasının ana alanları arasındaki şematik ilişkisi (swimming pools design guidance dergisi, 2011).

Fuaye: Bilet gişesinin içinde olduğu giriş alanı ile birlikte temizlik odası, ilk yardım odası, acil servis odası, kafeterya ve idari bölüm bulunur. 20000 ve üzeri seyirci sayısı için ilk yardım odalarına ihtiyaç duyulmaktadır. Bu ilk yardım odaları 15m² lik tedavi odası 2m² alana sahip acil yardım deposu ve 2 tuvalet bulunmalıdır. 30000 ve üstü seyirci kapasiteli kapalı yüzme havuzlarında 15m² alana sahip acil servis odalarına ihtiyaç vardır (Kurt, 2012).

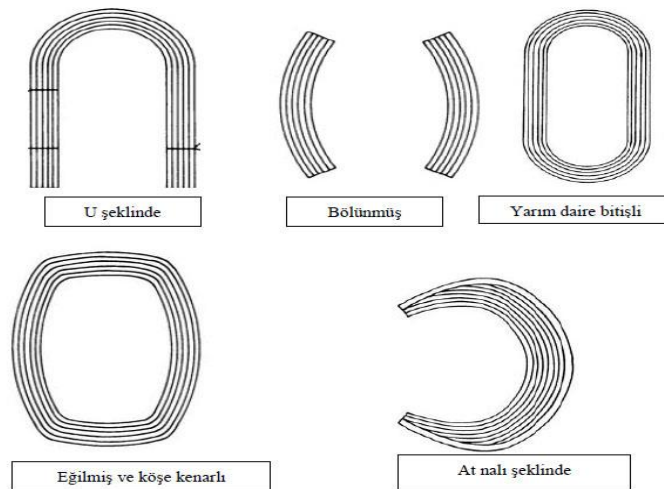
Tribünler : spor müsabakalarının yapıldığı alanlarda seyircilerin yarışmaları izlemek için oturma ve ayakta durma alanlarının olduğu kısımdır. Tribünler bloklar halinde olmalı ve her blok arasında 1m genişliğinde çıkışa giden yollar olmalıdır.

Seyircilerin yarışmaları daha iyi görebilmeleri için izleme standartlarının oluşturulması önemli bir faktördür. Tribünlerdeki seyircilerin göz hizası iki sıra önündeki seyircinin başının üst kısmından alanı görebilecek yükseklikte olmalıdır.



Şekil 3.3 : Oturmuş seyirciler için görüş çizgisi (Green guide dergisi, 2011).

Tribünler tasarlanırken kapasiteye göre planlanır. Seyirci sayısı 1000 kişiden az ise tribün görüşünün en yakın olması için uzun konumlanır. Eğer seyirci sayısı 1000 kişiden fazla ise bütün havuz alanını çevreleyerek konumlandırılır. Tribünler genel olarak aşağıdaki 5 şekilde tasarlanabilir (Neufert).

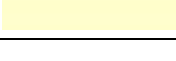


Şekil 3.4 : Tribün düzenleme çeşitleri (Neufert).

3.3 Olimpik Yüzme Havuzları Örnekleri

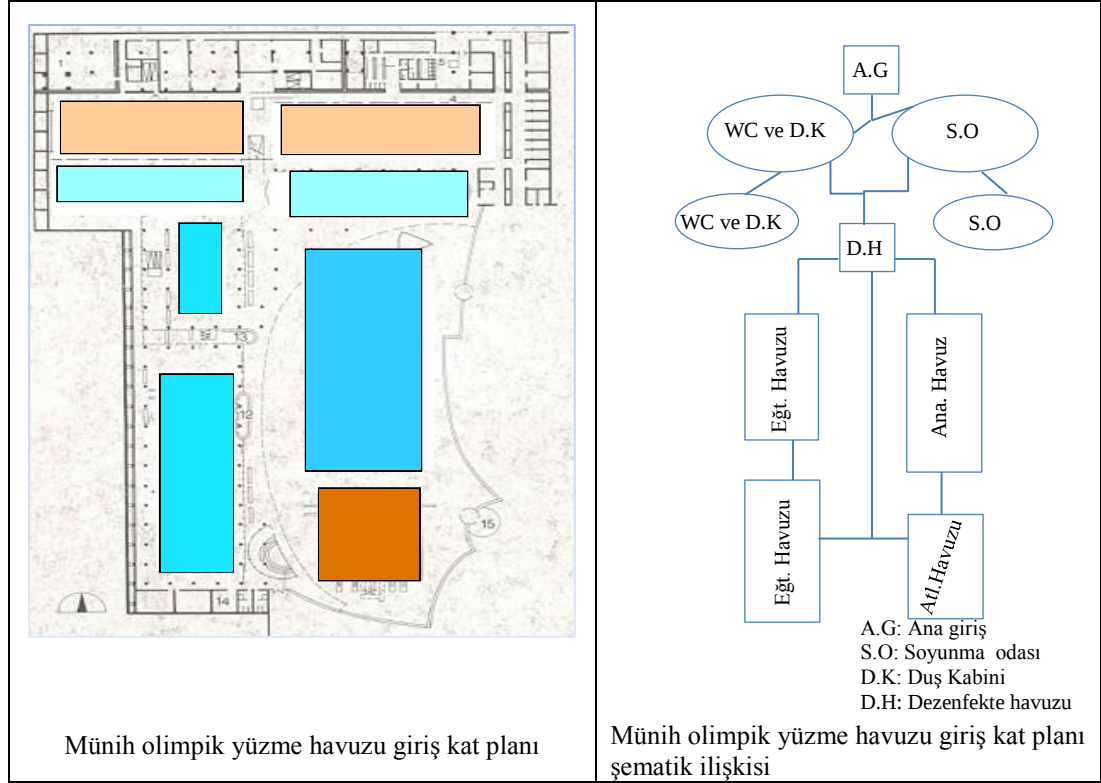
Bu bölümde faaliyette olan olimpik ve yarı olimpik yüzme havuzlarının plan analizleri yapılacak ve yapılan bu analizler doğrultusunda, mekânsal ilişkiler şematik hale getirilecektir. Şematik hale getirilen bu veriler bilgisayar ortamında oluşturulacak olimpik yüzme havuzu plan kurgulanmasında kullanılacaktır. Ayrıca verilen plan örneklerinin daha iyi okunabilmesi açısından her bir mekanı temsil eden renkler, aşağıda tablo halinde verilmiştir;

Çizelge 3.1 : Mevcut plan şemalarındaki her bir mekanı gösteren renk tablosu

	Soyunma odası
	Dezenfekte havuzu
	Islak hacim
	Havuz
	Atlama havuzu
	Eğitim havuzu
	Tribün
	Personel odası
	Servis Alanı
	Teras
	Fuaye
	Sporcu girişi

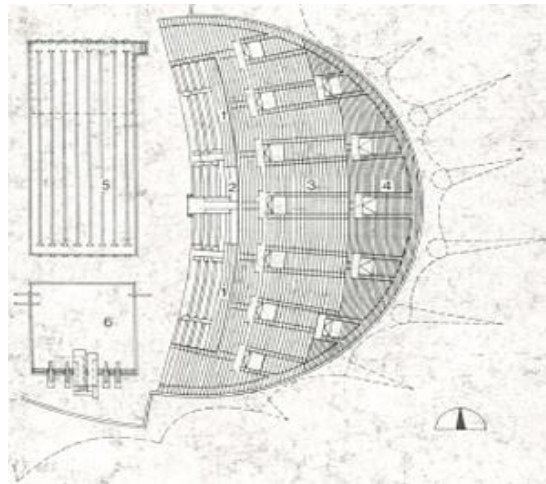
Münih Olimpik Yüzme Havuzu: 1972 yılında inşa edilen Münih olimpik yüzme havuzu 8250m² lik bir alana sahiptir. Münih olimpik yüzme salonu dışarıda güneşlenme alanları ile birlikte atlama tesisleri, yüzme yarış kulvarları ve tribünleriyle bir spor merkezi olarak tasarlanmıştır. Yaklaşık 9500 seyirci kapasiteli

tribüne sahiptir. Ayrıca sporcular için 108 soyunma kabini, 96 duş kabini ve toplam 42 WC bulunmaktadır.



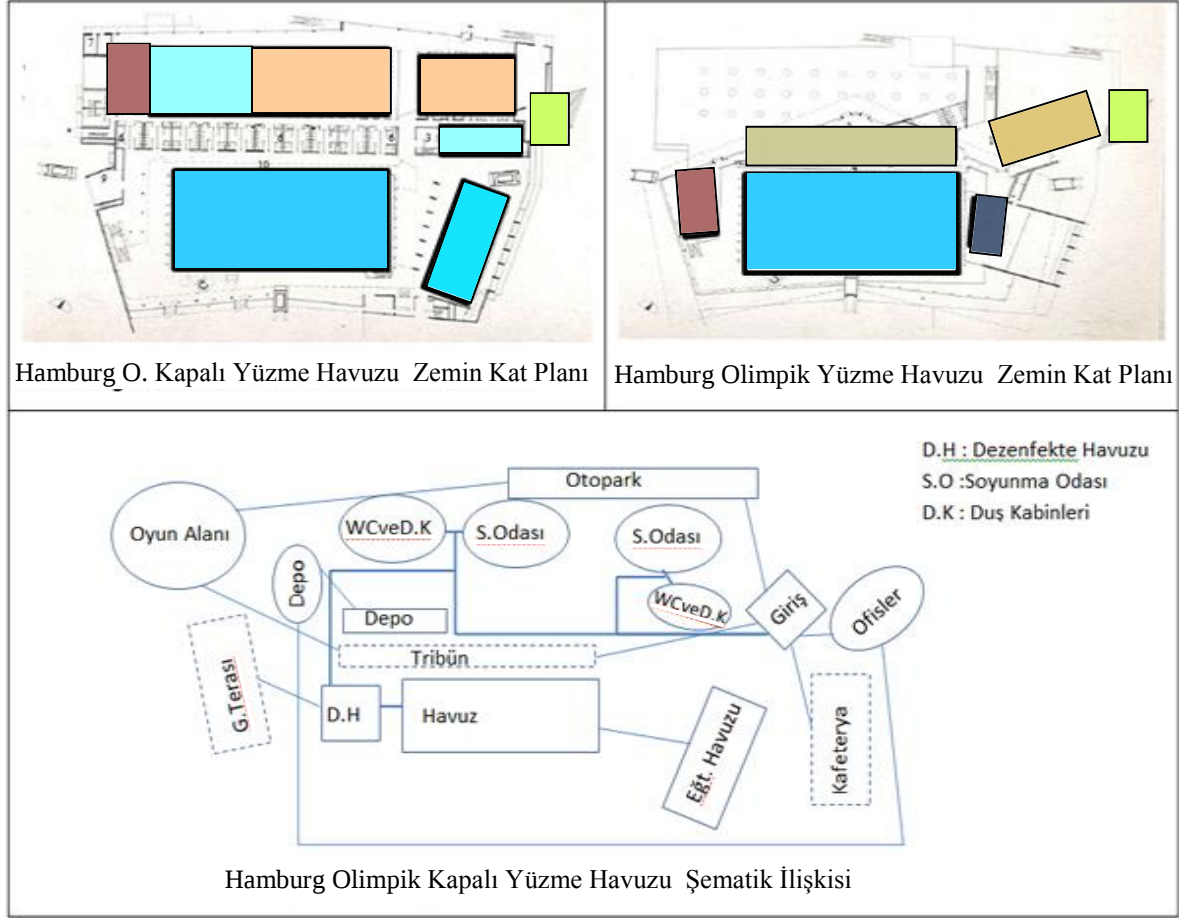
Şekil 3.5 : Münih olimpik havuz plan ve şematik ilişkisi (Aquatic building).

Münih olimpik yüzme salonunun kat planları analiz edildiğinde soyunma kabinlerinin duş kabinlerine yakın konumlandığı, sporcuların daha rahat bir yüzme eğitimi veya antrenman yapabilmeleri için ayrı bir havuzun yer aldığı görülmektedir. Tribünler seyirci kapasitesinin daha fazla olabilmesi için şekil 5.9 da görüldüğü gibi yarım bir hilal şeklinde tasarlanmıştır.



Şekil 3.6 : Münih olimpik havuzu giriş kat planı (Aquatic building).

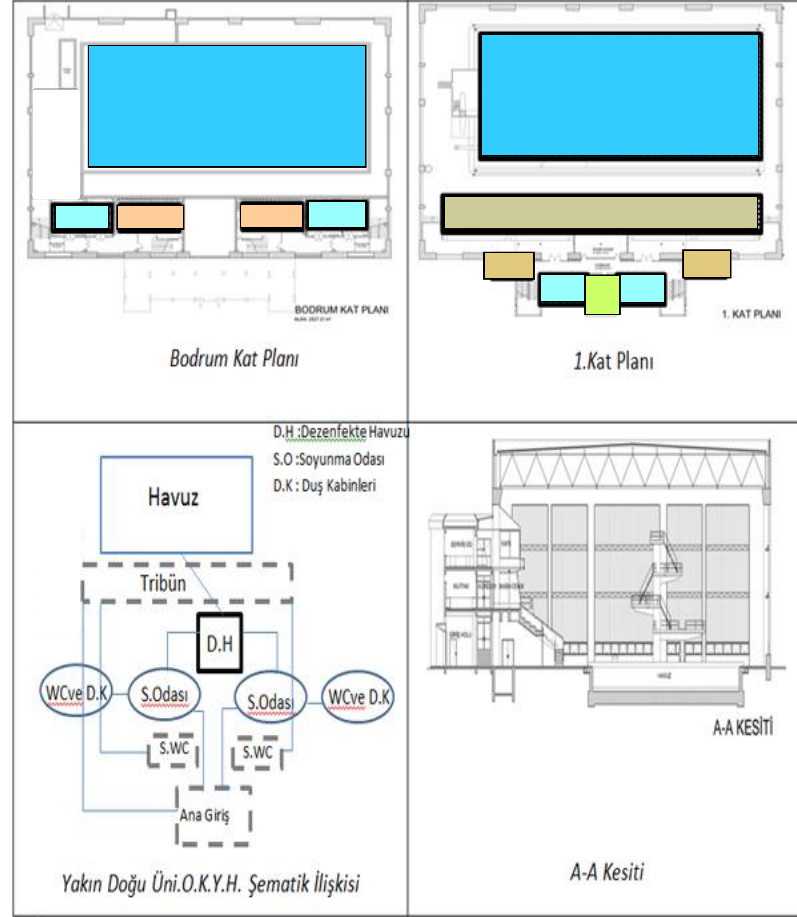
Hamburg Olimpiyat Kapalı Yüzme Havuzu: 1972 yılında Almanya'nın Hamburg şehrinde inşa edilen kapalı olimpik yüzme havuzu 16250 m2 alan üzerinde kurulmuş olup yaklaşık 680 seyirci kapasitesine sahip bir yüzme spor merkezidir. Ayrıca bu kapalı yüzme havuzunda tam olimpik yüzme havuzu ve eğitim havuzu olmak üzere 2 adet havuz, çocuk ve yetişkin sporcular için ayrı toplam 126 soyunma kabini, 42 duş kabini ve toplam 24 WC bulunmaktadır. Seyirciler ve sporcular için araç park alanı, güneşlenme çim alanı ve aynı zamanda jimnastik odası mevcuttur.



Şekil 3.7 : Hanburg olimpik havuzu plan ve şematik ilişkisi (Aquatic building).

Hamburg olimpik kapalı yüzme havuzu kat analizleri yapıldığında; Soyunma odaları, görevli personel odası, jimnastik odası ve satış noktalarının zemin katta olduğu soyunma odaları ile havuza giriş kapısının doğrudan olmadığı ve sporcuların kafeterya ya gitmek için tekrardan salona giriş kısmına yönelmeleri gerektiği görülmektedir. Bununla birlikte fuaye, ofisler, restoran, tribün ve güneşlenme terasına giden çıkış kapıları giriş katta konumlandırılmıştır.

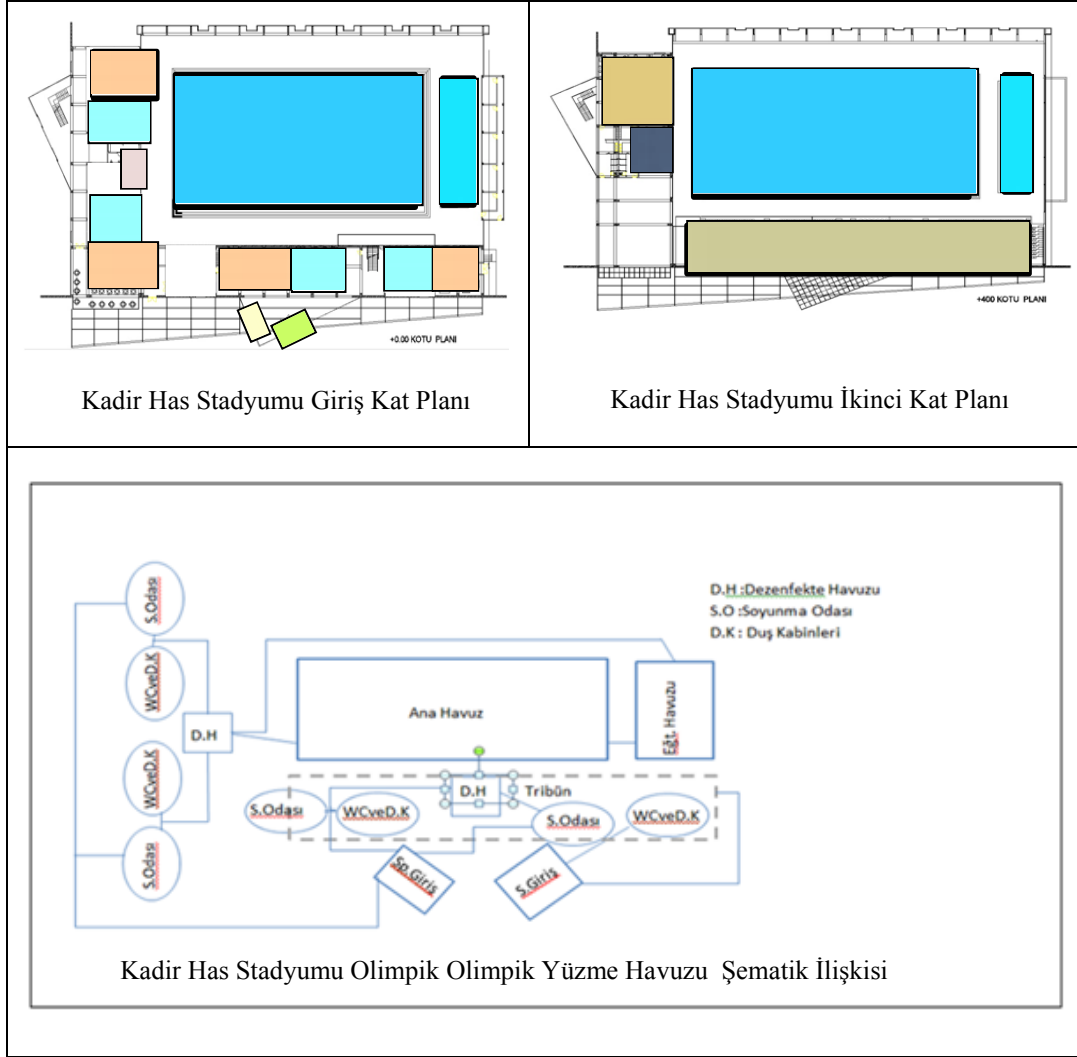
Yakın Doğu Üniversitesi Olimpik Kapalı Yüzme Havuzu: 1997 yılında inşa edilen yakın doğu üniversitesi kapalı yüzme havuzu, bodrum katı ile birlikte toplam 4 kattan oluşmakta ve yaklaşık 5300 m² alana kurulu bir yüzme spor merkezidir . 1000 kişilik seyirci kapasitesine sahip olan yüzme havuzu olimpik ölçülerde 9 kulvarı ve 3m, 5m, 7.5m, 10m gibi farklı atlama seviyelerine sahip atlama kulesi bulunmaktadır.



Şekil 3.8 : Yakın Doğu Üni. Olimpik havuzu plan ve şematik ilişkisi (Aquatic building).

Soyunma odalarından havuz bölgesine girişlerde dezenfekte havuzları bulunmaktadır. Havuz mekanı yaklaşık 1865 m² alana sahiptir. Bodrum kat ise yaklaşık olarak 2550 m² olup, içerisinde mekanik oda, bay ve bayan soyunma odaları, duş kabinleri ve WC bulunmaktadır. Bay ve bayan bölümlerinden iki ayrı çıkış merdivenlerinden zemin kat ve havuz katına ulaşılmaktadır. Zemin kat ise yaklaşık 1750 m² lik kapalı bir alana sahip olup, giriş alanında danışma kısmı, kantin, ilk yardım odası bulunmaktadır.

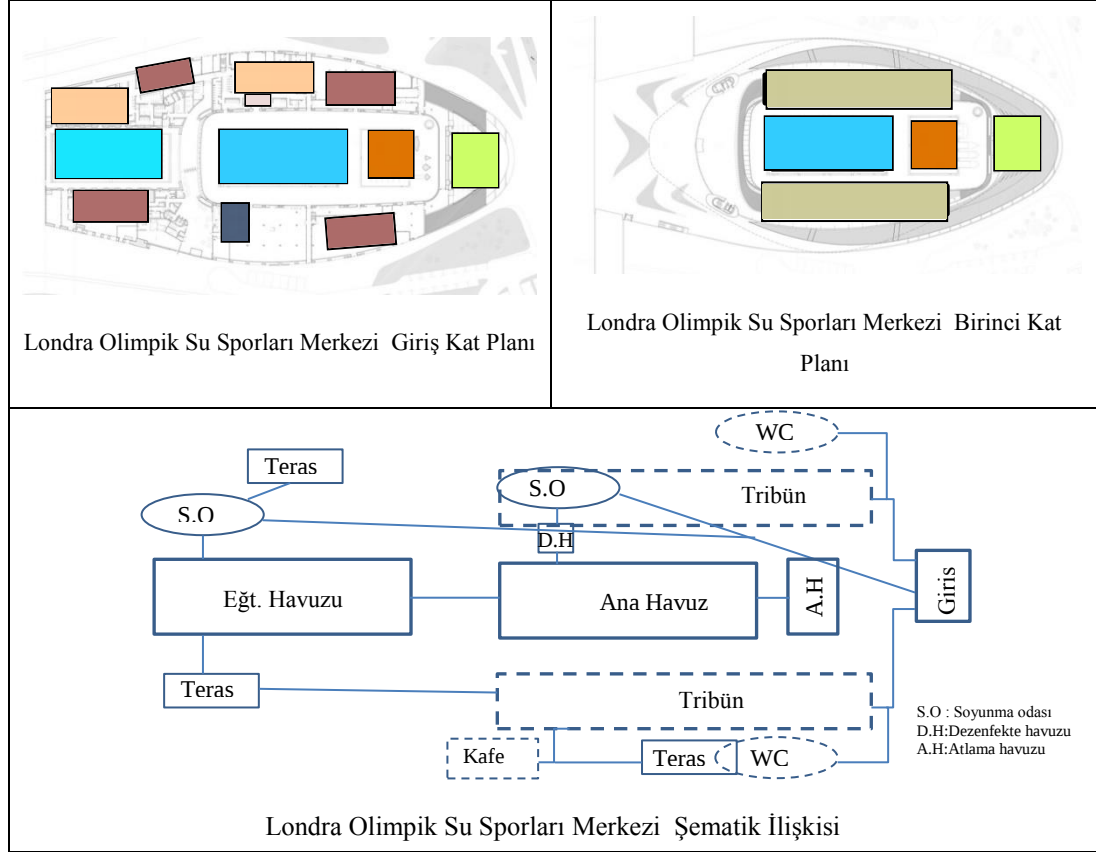
Kadir Has Stadyumu Olimpik Yüzme Havuzu: Kayseri il sınırları içerisinde bulunan Kadir Has olimpik yüzme havuzu 2007 yılında inşa edilmiştir. Toplam 20.000 m² arsa alanına sahip olan bu spor kompleksinin kapalı alanı ise 10.000 m² olup toplam 1500 seyirci kapasiteli tribünlere sahiptir. Ayrıca bu kapalı yüzme havuzunda tam olimpik yüzme havuzu ve çırpınma havuzu olmak üzere 2 adet havuz, 4 adet soyunma odası, kafeterya ve otopark bulunmaktadır (v2.arkiv.com.tr).



Şekil 3.9 : Kadir Has Olimpik havuzu plan ve şematik ilişkisi (v2.arkiv.com.tr).

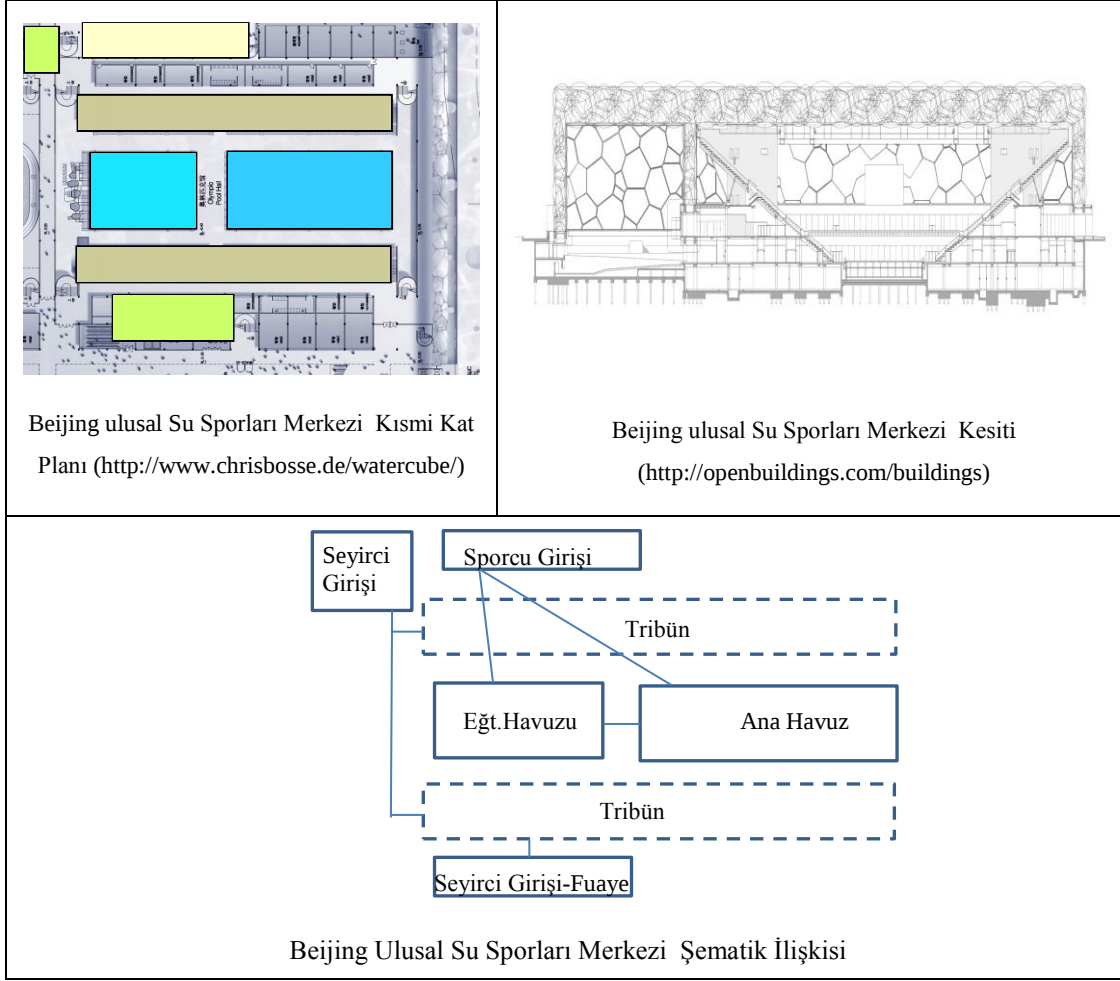
Londra Olimpik Su Sporları Merkezi : Yapımına 2011 yılında başlanan üç kattan oluşan Londra olimpik su sporları merkezi yaklaşık 21.500 m² alan üzerine kurulmuştur. Toplamda 17.500 seyirci kapasitesine sahip olan Londra olimpik su sporları merkezi'nin mimari konseptinde suyun hareketi içerisindeki akış geometrilerinden ilham alınmıştır. Londra olimpik su sporları merkezinin plan kurgusu incelendiğinde olimpik havuz, eğitim havuzu, yüksek atlama havuzu ve karşılıklı iki tribün bulunmaktadır. Planda iki farklı soyunma odasının bulunduğunu

görmekteyiz. Bunlardan biri olimpik havuza diğeri ise eğitim havuzuna daha yakın mesafede konumlandırılmış. Ayrıca olimpik havuzuna yakın bulunan soyunma odası ile olimpik havuz arasında dezenfekte havuzu bulunurken, eğitim havuzu ile eğitim havuzuna bağlı bulunan soyunma odası arasında dezenfekte havuzu bulunmamaktadır. (<http://www.archdaily.com>).



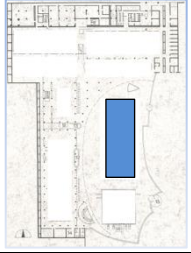
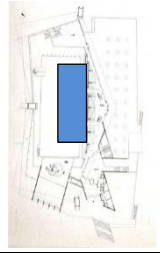
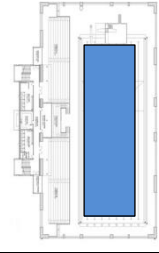
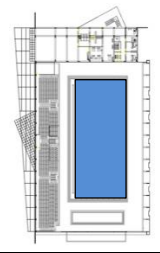
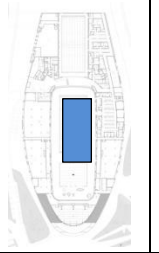
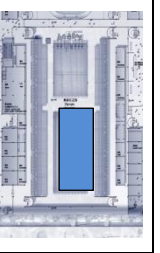
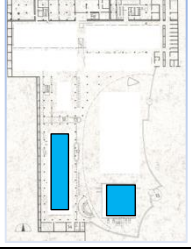
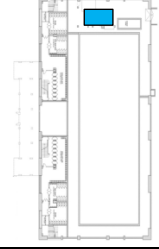
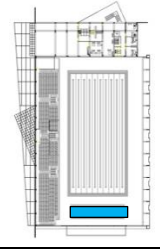
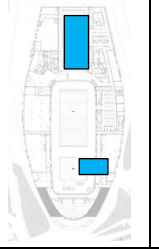
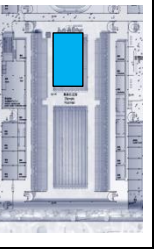
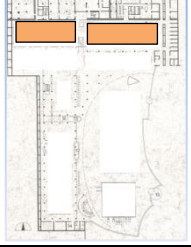
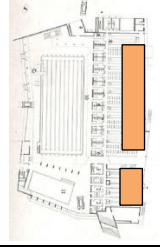
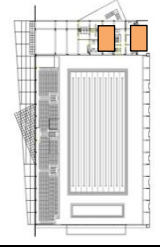
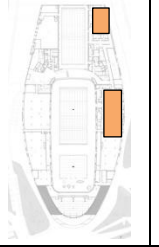
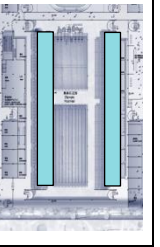
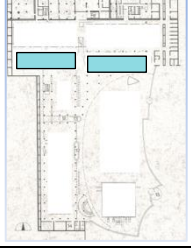
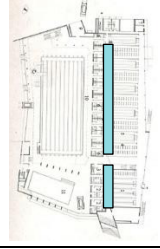
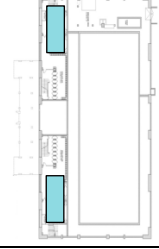
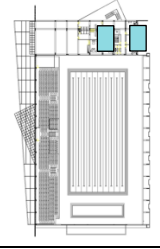
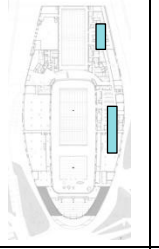
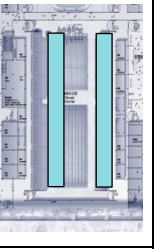
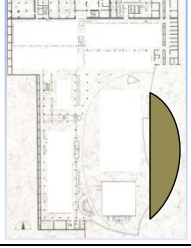
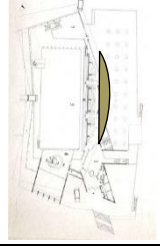
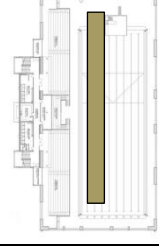
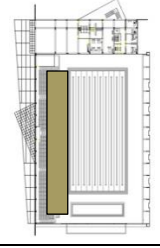
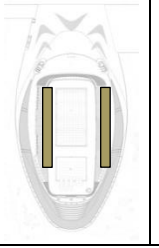
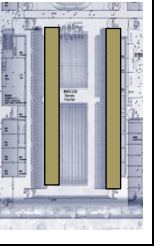
Şekil 3.10 : Londra Olimpik Su Sporları Merkezi plan ve şematik ilişkisi (www.archdaily.com).

Beijing Ulusal Su Sporları Merkezi : Çin'in Beijing şehrinde yapılmış olan ulusal yüzme merkezi veya diğer adı su küpü (water Cube) olarak bilinen Beijing ulusal su merkezi 2008 yılında faaliyete girmiştir. Yaklaşık 8120 m² alan üzerinde konumlanmış olan bu yapı karşılıklı iki tribün düzenine sahip olup yaklaşık 17.000 seyirci kapasitesine sahiptir. İki yüzme havuzuna sahip ve aynı zamanda 25 m X 30 metre bir dalış havuzu bulunmaktadır. 2008 Pekin olimpiyat oyunlarından sonra olimpik yüzme havuzunun yanısıra su parkıda eklenmiştir (www.chrisbosse.de/watercube/).



Şekil 3.11 : Beijing ulusal su sporları merkezi plan ve şematik ilişkisi.

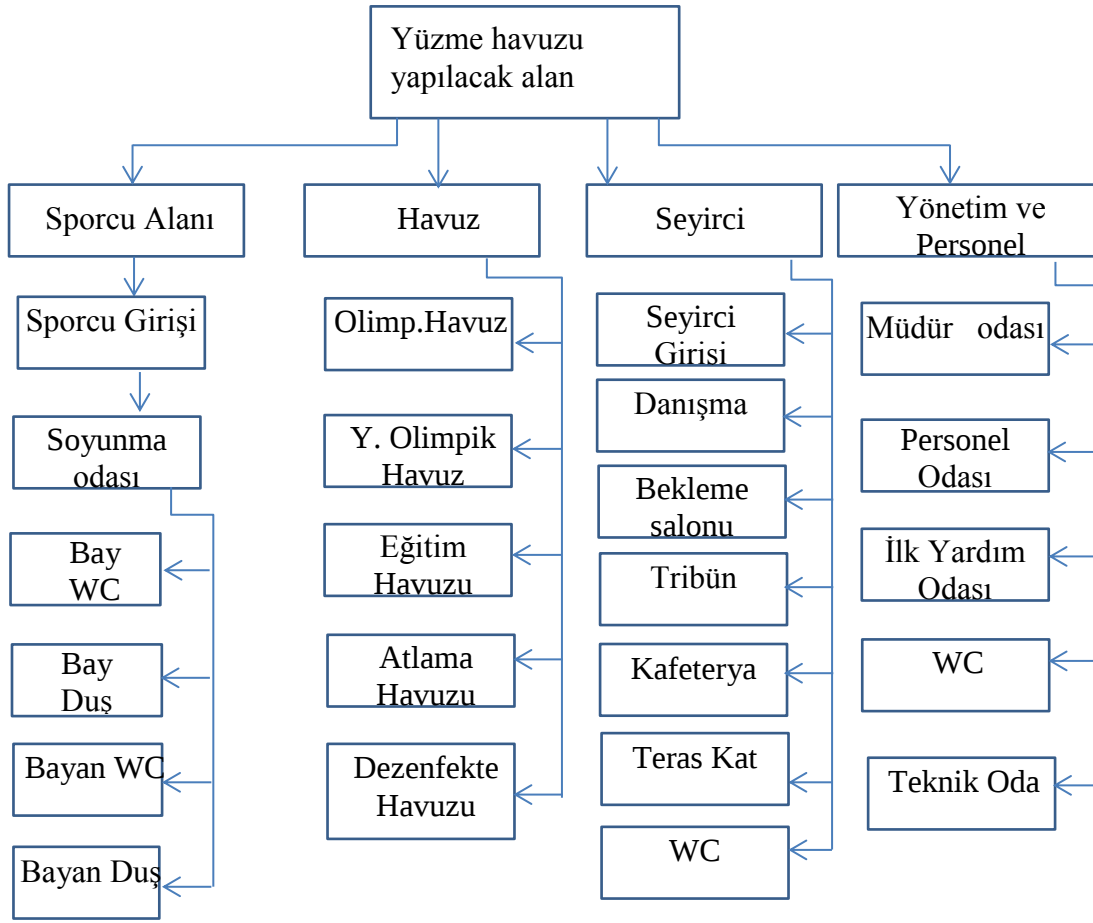
Çizelge 3.2 : Verilen plan örneklerinin ana şema düzenleri

	Münih Olmp. Y.H	Hamburg Olmp. Y.H	Yakın Doğu Üni. Olmp. Y.H	Kadir Has Std. Olmp. Y.H	Londra Olmp. Y.H	Beijing Y.H
Olimpik Havuz						
Eğitim Havuzu						
Soyunma Odası						
Islak Alan						
Tribün						

4. YÜZME HAVUZU PLAN ŞEMASI TASARIMI İÇİN GENETİK ALGORİTMAYA DAYALI BİR MODEL

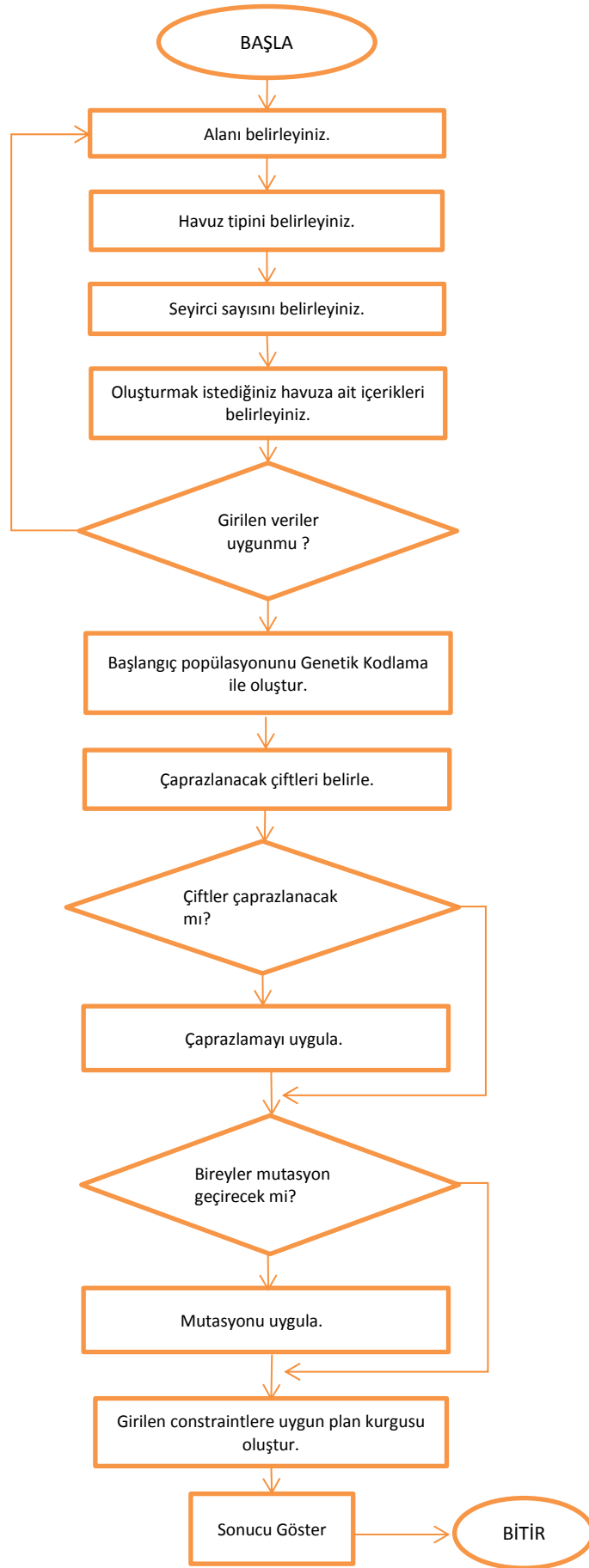
Bilgisayar ortamında biyolojik dinamiklerin evrimsel süreçleri incelendiğinde, sayısal ortamda sanal bitki veya hayvanlar tasarlanarak onların sanal ortamdaki genetik yapılarının nasıl değişim geçirdikleri takip edilebilir. Burada sanal ortamdaki genlerin türettiği sanal bedensel karakterler arasındaki ilişkinin tanımlanmasıdır. Ebeveynler arasındaki çiftleşmeden hangi genin hangi genle eşleşmesi gerektiği, yeni türeyen bireylere değerlerin atanması, bir genin birkaç nesil sonrasında nasıl yayıldığı gibi olaylar bilgisayar ortamında genetik algoritma kullanılarak bu süreç izlenebilir. Bu tip yazılımlar formülize edilip çalışma özellikleri incelendiği zaman söz konusu biyolojik araştırmalardan ayrılarak başlı başına bir araştırma alanı içerisine girmektedir. Bu tez çalışmasında genetik algoritma kullanılarak mimari plan kurgusuna nasıl yardımcı olabileceği ve de bu plan kurgusunun gelişimine odaklanılacaktır.

Yapılacak olan modelin bilgisayar ortamında kod kısmı yazılmadan önce şekil 4.1 deki ağaç dal model oluşturularak mekanlar arasındaki ilişkiler belirlenip, yazılım (software) kısmında plan şemasını oluşturan mekanlar bloklar halinde işlem görecektir. Oluşturulan her bir mekan plan şemasında yerleştirilirken mekanlar arasındaki ilişki ve belirlenen kısıtlara göre plan kurgusu oluşacaktır. Ayrıca yazılım kısmında mekanların bloklar halinde işlem görmesi, plan kurgulanmasında kısa sürede sonuca ulaşılabilme ve hata oranını minimuma indirilmesi hedeflenmiştir. Yapılacak olan yüzme havuzu plan şemasının ağaç dal modelini oluşturmak için, öncelikle yüzme havuzu salonunu dört ana bölüme ayırıp her bölüm kendi içerisinde belirtilen kısıtlar ile bir kombinasyon oluşturularak ağaç dal yapısı oluşturuldu. Oluşturulan ağaç dal yapısı ile mekanlar arasındaki ilişkiler belirlenip bu ilişkiler doğrultusunda mekan kurguları oluşturulmaya başlandı.



Şekil 4.1 : Yüzme havuzu planı ağaç dal modeli

Bilgisayar ortamında yüzme havuzu plan şeması oluşturulabilmesi için elimizde bazı verilerin olması gereklidir örneğin, kaç metre kare alan üzerinde kurulacağı, seyirci sayısının kaç kişi olacağı ve hangi havuz türlerinin yer alacağı (olimpik, yarı olimpik, eğitim havuzu ve atlama havuzu gibi) belirtilmelidir. Bu çalışma kapsamında hazırlanan program 3 ana kısımdan oluşmaktadır. Programı ilk açtığımızda şekil 4.3 de verilen bir pencere açılır. Bu kısımda kullanıcı adı ve şifre sorulmaktadır. Çalışma kapsamında şimdilik kullanıcı adı “mimarsinan” ve şifre olarak ise “calisir” olarak belirlenmiştir. Gerekli bilgiler girilip giriş butonuna tıklayarak programda uygulama yapılacak yeni bir pencere açılacaktır. Açılan bu pencerede programa entegre edilmiş Google Map ve havuz plan şemasının oluşturulabilmesi için gerekli verilerin girileceği bir panel mevcuttur. Öncelikle havuzu oluşturacağımız alanı belirlemeden önce açılan penceremizin sol üst köşesinde bulunan Map ve Satellite butonlarından herhangi birini seçerek çalışmak istediğimiz görüntü seçeneklerini belirleyebiliriz. Plan şemasını oluşturacağımız



Şekil 4.2 : Yüzme havuzu planı akış şeması

alanı belirlemek için ekrandaki draw a shape  kare içerisine alınmış seçim aracına mouse ile tıklayarak aktif hale getirilip harita üzerinde alan sınırlarımızı belirleriz. Kare içerisine alınmış şeklin sol tarafındaki stop drawing butonu aktifleştirdiğimizde çizimi sonlandırır aynı zamanda ekrandaki görüntüyü sağa-sola ve aşağı-yukarı yönlerinde hareket ettirmemizi sağlar sağ taraftaki draw a line butonu uzunluk ölçmemizi sağlar, bu çalışmada uzunluk ölçüsü kullanılmayacağından bu araç butonu pasif duruma getirilmiştir.



Şekil 4.3 : Model arayüzü

Panel kısmında öncelikle alanını hesaplamak istediğiniz bölgeye ait koordinatları belirleyin yazısının altında bulunan Alan kısmı, draw a shape aracını aktif hale getirip havuz planının yapılacağı yeri belirleyip panelin alan kısmında seçim yapılan yerin kaç metre kare alan belirlediğimiz bilgisi görülecektir. Daha sonraki aşamada oluşturmak istediğimiz havuz tipini seçmemiz gerekmektedir. Bu kısımda olimpik ve yarı olimpik seçenekleri bulunmaktadır. Seçeceğimiz havuz tipine göre program havuzu oluşturacaktır. Bir diğer girilmesi gereken veri tipi seyirci sayısı ile alakalı kısımdır. Seyirci sayısı bu çalışma içerisinde maksimum 2500 kişi ile sınırlandırılmıştır. Panelde bulunan oluşturmak istediğimiz havuz içerisinde eğitim havuzunun ve yüksek atlama kulesinin olup olmaması seçenekleri mevcuttur. Bu seçenekler zorunlu olmayıp kullanıcının isteğine bırakılmıştır. Panelin en alt kısmında Hesapla ve Generate butonları bulunmaktadır. Gerekli veriler girildikten sonra Hesapla butonuna tıklayarak oluşturulacak modelimizin ana mekanları, her bir mekanın kaç metre kare alan üzerinde kurulacağı, yeni bir pencere açılarak plan şeması oluşturulacaktır.

Alanını hesaplamak istediğiniz bölgeye ait koordinatları belirleyin.

Alan metre kare

Oluşturmak istediğiniz havuzu tanımlayınız.

Oluşturmak istediğiniz havuzu tipi;

☐ olimpik

☐ yarı-olimpik

Belirtilen alan için ;

seyirci sayısı kişi

Oluşturmak istediğiniz havuz;

☐ eğitim havuzu içerir

☐ yüksek atlama kulesi içerir

Belirttiğiniz kriterlere uygun planı görmek için tıklayınız.

Şekil 4.4 : Panel arayüzü

Generate butonuna tıkladığımız zaman ise her bir ana mekanın kendi içerisinde bulundurduğu mekanları modelleyecektir. Örneğin, sporcu alanı için belirlenen mekan içerisinde soyunma alanının, duş kabinlerinin veya WC lerin nerde oluşması gerektiğini ve her bir mekanın ne kadar alanı kapsayacağını, fuaye ve sporcu giriş kısmındaki mekan plan şemalarını alternatifler sunarak model oluşturacaktır.

4.1 Modelin Uygulanması

İlk olarak İstanbul Teknik Üniversitesi Maslak kampüsünde bulunan olimpik yüzme havuzu mevcut yapısı üzerinde örnek model oluşturacağız. Maslak kampüsündeki olimpik yüzme havuzu yapılan etüt çalışmalarında elde edinilen veriler, yapının iki kattan oluştuğu, olimpik bir havuz ve eğitim havuzuna sahip olduğu, fuaye kısmının olduğu, sporcu soyunma odalarının tribün ile üst üste gelecek şekilde tasarlandığı ve de tek tribünden oluşan yapının yaklaşık 700 kişilik bir seyirci kapasitesine sahip olduğu belirlenmiştir. Çalışma kapsamında yapılacak örnek modelde aynı alanı seçip belirlenen veriler girilerek programın benzer bir plan şemasını verip vermeyeceğini test edeceğiz. Öncelikle modeli oluşturacağımız alanı google map üzerinde draw a

shape butonunu tıklayarak aktif hale getirip mevcut yapı üzerinde alanımızı belirleriz. Alan sınırları belirlendikten sonra panel üzerinde oluşturmak istediğimiz havuz tipini tanımlarız. Maslak kampüsündeki havuz tipi olimpik olduğundan, havuz tipi olarak olimpik seçimini işaretleriz. Bir sonraki aşamamızda elimizde bulunan verilerden yararlanarak belirtilen alan için seyirci sayısı 700 olarak girilir. Mevcut yapı içerisinde eğitim havuzu bulunduğundan, panel kısmında oluşturmak istediğiniz havuz içerisinde eğitim havuzunda seçeriz. Bu veriler panel kısmında belirtildikten sonra hesapla butonuna tıklayarak program bize şekil 4.7 de görüldüğü gibi bir model oluşturmaktadır.



Şekil 4.5 : Maslak kampüsü yüzme havuzu alan sınırları

Alanını hesaplamak istediğiniz bölgeye ait koordinatları belirleyin.

Alan metre kare

Oluşturmak istediğiniz havuzu tanımlayınız.

Oluşturmak istediğiniz havuzu tipi;

☒ olimpik

☐ yarı-olimpik

Belirtilen alan için ;

seyirci sayısı kişi

Oluşturmak istediğiniz havuz;

☒ eğitim havuzu içerir

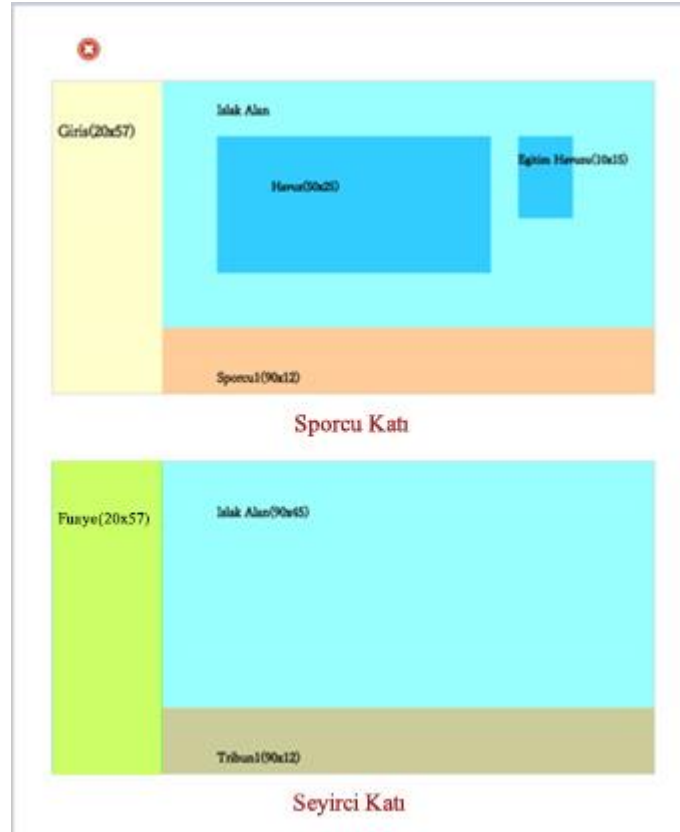
☐ yüksek atlama kulesi içerir

Belirttiğiniz kriterlere uygun planı görmek için tıklayınız.

Şekil 4.6 : Maslak kampüsü havuz modeli için panel üzerinde girilen veriler

Üretilen modelde görüldüğü gibi belirlenen alan içerisinde $90 \times 45 \text{ m}^2$ ıslak mekan alanı belirlenip bu ıslak mekan üzerinde olimpik yüzme havuzu ve $10 \times 15 \text{ m}^2$ lik bir eğitim havuzu alanı hesaplanmıştır. Sporcu katı planında ıslak mekanın yanısıra $90 \times 12 \text{ m}^2$ lik bir alan içerisinde sporculara ait bir alan sınırı oluşturulduğu ve aynı kat içerisinde $20 \times 57 \text{ m}^2$ lik bir alanı giriş kısmı olarak program belirlemiştir. Seyirci katı planı incelendiğinde, $20 \times 57 \text{ m}^2$ alan üzerinde fuaye kısmını yerleştirdiğini, $90 \times 12 \text{ m}^2$ alan üzerinde ıslak mekan alanına eşit aynı paralel doğrultuda tek tribünlü olarak model oluşturulmuştur.

Görüldüğü gibi Maslak kampüsü olimpik yüzme havuzunun mevcut yapısında bulunan tek tribün ile üretilen model karşılaştırıldığında, mevcut yapıya yakın bir model oluşturulduğu gözlenmektedir.



Şekil 4.7 : Maslak kampüsü olimpik havuz modeli için elde edilen sonuç

Bir başka örnek için yaklaşık 10000 m^2 bir alan belirleyip seyirci sayısını ilk etapta 500 kişi olarak girilecek ve daha sonra seyirci sayılarını yükselterek plan şemalarında nasıl değişiklikler olduğu analiz edilecektir. Belirlenen alan içerisinde oluşturulacak model için, panel kısmımızda oluşturmak istediğiniz havuz tipi olimpik

olarak seçim yapılmış seyirci sayısı 500 olarak belirlenip, son olarak oluşturulmak istenen havuz içerisinde eğitim havuzu ve yüksek atlama kulesi bulunacaktır.

Alanını hesaplamak istediğiniz bölgeye ait koordinatları belirleyin.

Alan metre kare

Oluşturmak istediğiniz havuzu tanımlayınız.

Oluşturmak istediğiniz havuzu tipi:

☐ olimpik

☐ yarı-olimpik

Belirtilen alan için :

seyirci sayısı kişi

Oluşturmak istediğiniz havuz:

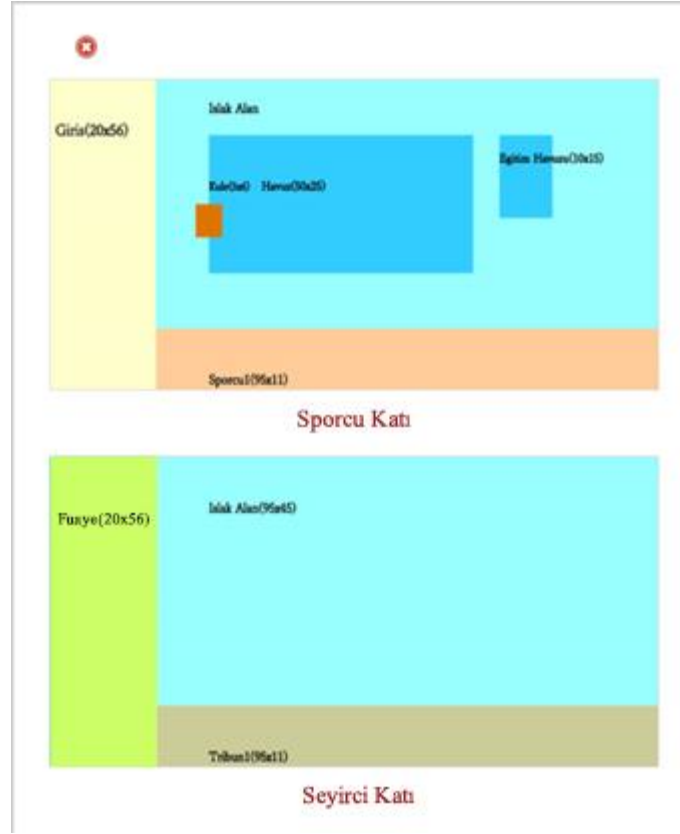
☒ eğitim havuzu içerir

☒ yüksek atlama kulesi içerir

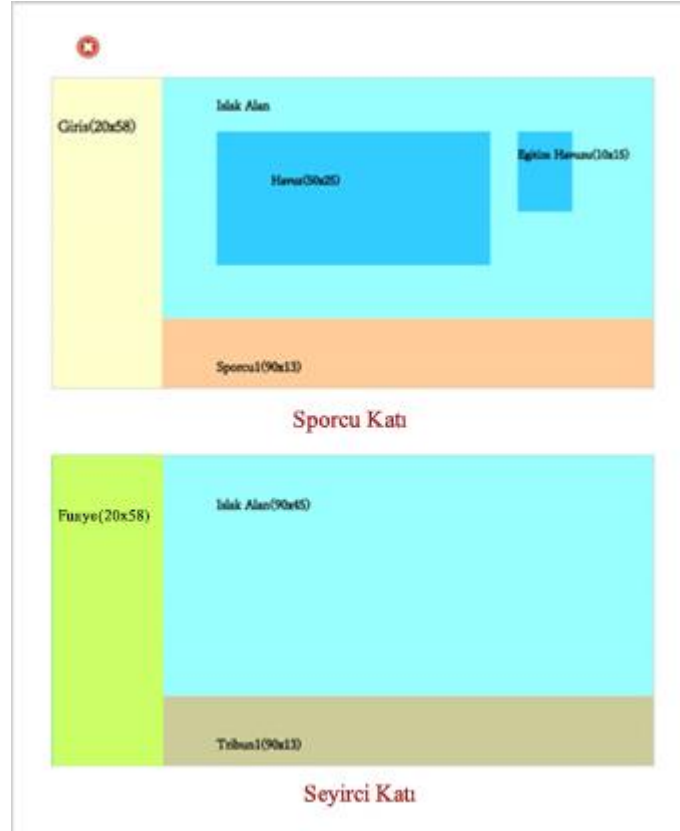
Belirttiğiniz kriterlere uygun planı görmek için tıklayınız.

Şekil 4.8 : 10000 m² alana uygulanacak model için panele girilen veriler

Panel kısmında bulunan hesapla butonuna basarak elde edinilen plan şeması analiz edildiğinde, belirlenen alan içerisinde 95 * 45 m² ıslak mekan alanı belirlenip bu ıslak mekan üzerinde olimpik yüzme havuzu ve 10 * 15 m² lik bir eğitim havuzu, 5 * 6 m² atlama kulesi alanı hesaplanmıştır. Sporcu katı planında ıslak mekanın yanısıra 95 * 11 m² lik bir alan içerisinde sporculara ait bir alan sınırı oluşturulduğu ve aynı kat içerisinde 20 * 56 m² lik bir alanı giriş kısmı olarak program belirlemiştir. Seyirci katı planı incelendiğinde, 20 * 56 m² alan üzerinde fuaye kısmını yerleştirdiğini, 95 * 11 m² alan üzerinde ıslak mekan alan uzunluğuna paralel bir uzantıda tek tribünlü olarak model oluşturulmuştur. 500 kişilik seyirci kapasitesine yönelik üretilen modelde elde edilen sonuç analizleri yapıldıktan sonra, panel kısmında seyirci sayısını 1000'e yükseltip atlama kulesi olmadan elde edilecek plan şeması analiz edilecektir. Bir önceki mevcut veriler girilip seyirci sayısını yükselterek hesapla butonuna tıklayıp program tarafından oluşturulan model şekil 4.10 da görüldüğü gibi, 95 * 45 m² ıslak mekan alanı belirlenip bu ıslak mekan üzerinde olimpik yüzme havuzu ve 10 * 15 m² lik bir eğitim havuzu, 5 * 6 m² atlama kulesi, 90 * 13 m² sporcu alanı ve 20 * 58 m² giriş alanı oluşturulmuştur. Bir üst katta ise tribün boyutu 90 * 13 m² ve 20 * 58 m² lik bir fuaye alanı belirlemiştir.

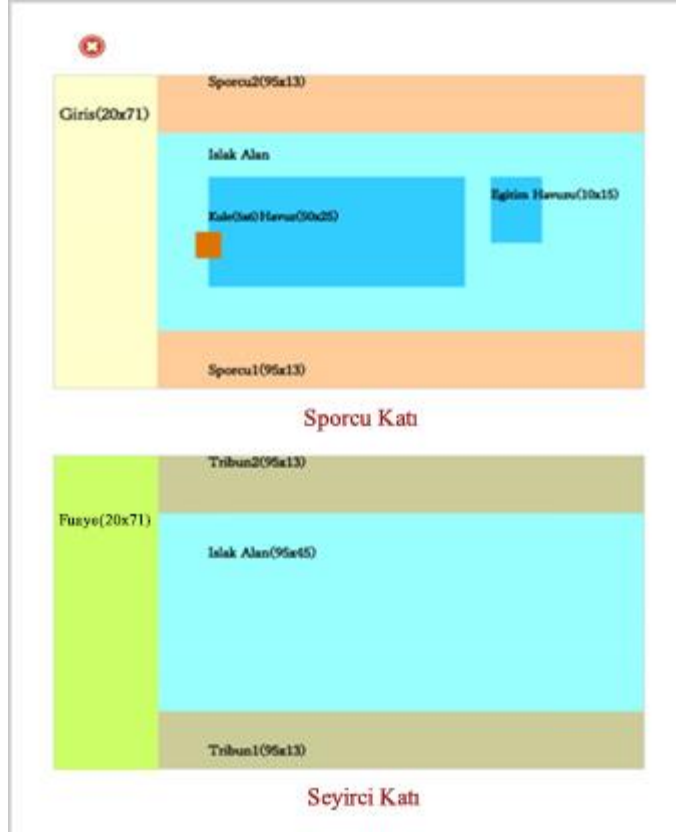


Şekil 4.9 : 10000m² alana 500 seyircili olimpik havuz modeli için alınan sonuç



Şekil 4.10 : 10000m² alana 1000 seyircili olimpik havuz modeli için alınan sonuç

Bir diğer örnek için seyirci sayımızı 2000 yapıp şekil 4.11 da elde edilen plan şemasında, karşılıklı iki tribün ve iki sporcu alanının plan şemasının oluştuğunu, Aynı zamanda seyirci sayısına paralel olarak giriş ve fuaye kısmınında alan olarak genişlediğini görmekteyiz.

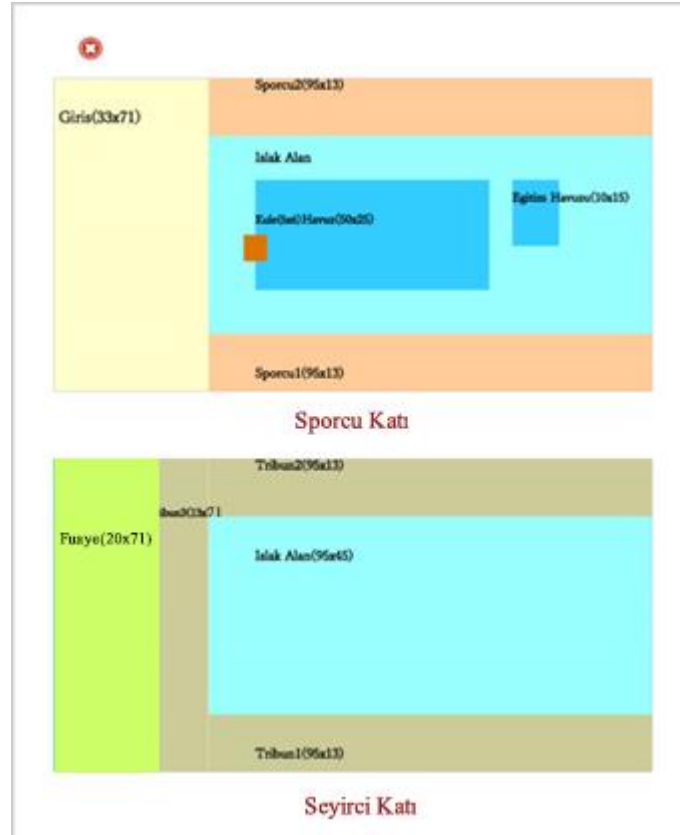


Şekil 4.11 : 10000m² alana 2000 seyircili olimpik havuz modeli için alınan sonuç

Son olarak seyirci sayımızı 2500'e çıkardığımızda elde edilen modelde, 95 * 45 m² ıslak alan, 95 * 13 m² karşılıklı iki sporcu alanı, 33 * 71 m² giriş, ıslak alan içerisinde eğitim havuzu ve atlama kulesi yer almaktadır. Bir üst katta ise ıslak alanın uzun kenarına paralel 95 * 13 m² lik karşılıklı iki tribün ve tribün3 ise ıslak mekanın kısa kenarına paralel olacak şekilde 71 * 13 m² olarak plan şemasında yer almıştır. Son olarak 71 * 20 m² fuaye kısmı elde edilmiştir.

Bir diğer örnekte aynı alan sınırları içerisinde 2500 seyirci kapasiteli ve eğitim havuzu olacak şekilde yarı olimpik yüzme havuzu plan şeması oluşturulduğunda elde edilen model analiz edildiğinde, 65 * 32 m² ıslak hacim, olimpik yüzme havuzu plan şemasında olduğu gibi ıslak alanın uzun kenarına paralel karşılıklı 65 * 14 m² lik iki tribün ve iki sporcu alanı ve diğer üçüncü tribün kısa kenar doğrultusunda 14 * 61

m² lik bir alan kapladığı görülmektedir. 34 * 61 m² giriş kısmı yer alırken, 20 * 61 m² alan üzerinde fuaye yer almıştır.

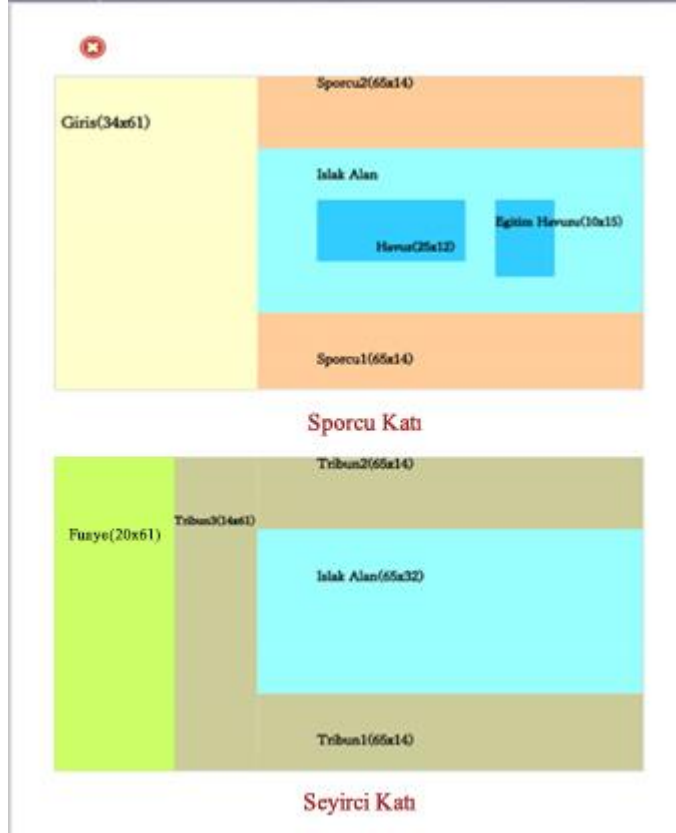


Şekil 4.12 : 10000m² alana 2500 seyircili olimpik havuz modeli için alınan sonuç

Bir önceki örnekte olimpik yüzme havuzu modelini oluşturmak için panel kısmına girilen değerleri Yarı olimpik yüzme havuzu oluşturup bu iki modeli karşılaştırdığımızda, olimpik yüzme havuzunda ıslak alanın uzun kenarına paralel karşılıklı tribünlerin uzun kenarları 95 m iken yarı olimpikte bu kenarlar 65 metreye düşmektedir. Aynı zamanda ıslak alanın kısa kenar konumunda yer alan üçüncü tribününde uzun kenarı 71 metreden 61 metreye düştüğü görülmüştür. Bu da program içerisinde belirlenen mekanları yerleştirirken, seyirci ile havuz arasındaki görüş mesafesini dikkate aldığı görülmektedir.

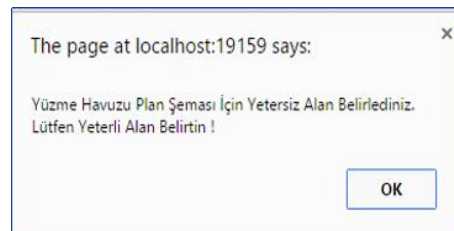
Verilen örneklerde olimpik ve yarı olimpik yüzme havuzu plan şeması oluştururken, google map üzerinden alan seçilip bu alan üzerinde oluşturulacak plan şemaları hesaplanıp plan oluşturulmaktadır. Program öncelikle panel kısmındaki kullanıcı tarafından girilen verileri ele alıp, istenilen plan şemasının kaç m² bir alana ihtiyaç duyduğunu hesaplayıp elde edilen bu veri ile seçilen alan kısmındaki veriyi

karşılaştırmaktadır. Eğer oluşturulmak istenen plan şeması için yeterli alan yoksa program kullanıcıya yeterli alan seçmesi için bir uyarı mesajı verecektir.



Şekil 4.13 : 10000m² alana 2500 seyircili yarı olimpik model için alınan sonuç

Örnek bir uygulama ile, şekil 4.15 te görüldüğü gibi bir alan seçiyoruz. Seçilen alanımızın yaklaşık 1900 m² olduğu görülmektedir. Bir önceki örneklerde 10000 m² bir alan belirleyip, 500 kişilik seyirci kapasiteli olimpik yüzme havuzu plan şeması için program bu alanın yaklaşık 6000 m²’sini kullandığı görülmüştü. Şuan uygulayacağımız 1900 m² alan üzerinde 500 kişilik seyirci kapasiteli olimpik yüzme havuzu plan şeması için, panel kısmındaki hesapla butonuna tıkladığımızda program tarafından “Yüzme Havuzu Plan Şeması İçin Yetersiz Alan Belirlediniz. Lütfen Yeterli Alan Belirtin” diye bir uyarı mesajı vermektedir.



Şekil 4.14 : Yetersiz alanı belirten hata mesajı



Şekil 4.15 : Oluşturulacak yüzme havuzunun alan sınırları

**Alanını hesaplamak
istediğiniz bölgeye alt
koordinatları belirleyin.**

Alan metre kare

**Oluşturmak istediğiniz
havuzu tanımlayınız.**

Oluşturmak istediğiniz havuzu tipi:

☒ olimpik
☐ yarı-olimpik

Belirtilen alan için :

seyirci sayısı
kişi

Oluşturmak istediğiniz havuz:

☐ eğitim havuzu içerir
☐ yüksek atlama kulesi içerir

**Belirttiğiniz kriterlere uygun
planı görmek için tıklayınız.**

Şekil 4.16 : 1900 m² alana uygulanacak model için panele girilen veriler

Uyarı mesajını aldıktan sonra OK butonuna basıp, yeni bir alan ölçüsü belirleyip tekrar işlemi başlatıyoruz. Sonuç olarak, program kod kısmında verilen kısıtlamalar içerisinde ıslak mekan alanı, sporculara ayrılan mekan, fuaye-giriş kısmı ve tribün için gerekli alanları oluşturup elde edilen alan ölçüsü ile kullanıcının belirlediği alan sınırları ölçüsünü karşılaştırıp, minumum değerde yeterli alan olmadığı zaman kullanıcıya uyarı mesajı göndermektedir.

Çizelge 4.1 : Elde edilen plan şemaları sonuç değerleri

Havuz Tipi	Seçilen Alan (m ²)	Seyirci Sayısı	Elde Edilen Tribün (m ²)	Elde Edilen Islak Alan (m ²)	Fuaye (m ²)	Giriş (m ²)	Sporcu Alanı (m ²)	Yüksek Atlama Kulesi	Eğitim Havuzu	Elde Edilen Plan Şeması			
Olimpik	6700	700	90 x 12	90 x 45	20 x 57	20 x 57	90 x 12	Yok	Var				
Olimpik	10000	500	95 x 11	95 x 45	20 x 56	20 x 56	95 x 11	Var	Var				
Olimpik	10000	1000	90 x 13	95 x 45	20 x 58	20 x 58	90 x 13	yok	Var				
Olimpik	10000	2000	Tribün1 (95 x 13)	95 x 45	20 x 71	20 x 71	Sporcu1 (95 x 13)	Var	Var				
			Tribün2 (95 x 13)				Sporcu2 (95 x 13)						
Olimpik	10000	2500	Tribün1 (95 x 13)	95 x 45	20 x 71	33 x 71	Sporcu1 (95 x 13)	Var	Var				
			Tribün2 (95 x 13)				Sporcu2 (95 x 13)						
			Tribün3 (13 x 71)										
Y.Olimpik	10000	2500	Tribün1 (65 x 14)	65 x 32	20 x 61	34 x 61	Sporcu1 (65 x 14)	yok	Var				
			Tribün2 (65 x 14)				Sporcu2 (65 x 14)						
			Tribün3 (14 x 61)										

5. SONUÇLAR

Evrimleşen mimari yapılar biyolojide olduğu kadar verimli kombinasyonlar üreterek somut mimari yapılar oluşturabilir. Genetik mimarlık yaklaşımı doğadaki evrimi örnek alarak bilgisayar ortamına aktarılan genetik bilgiler ve bu bilgilere dayanarak kullanılan algoritmalar ile yeni mimari formların üretilmesi mümkün olmaktadır. Genetik algoritmaların problemlere yaklaşımı evrimsel süreç izlenilerek çözümler üretilebilmektedir. Son yıllarda Darwin'in evrim kuramı ile genetik algoritmaya dayalı problem çözümleri bilişim dünyasında yeni yaklaşımlar ortaya çıkarmıştır. Bu tez kapsamı içerisinde yapılan çalışmadaki yazılım, Darwin'in evrim teorisinden esinlenerek oluşturulan evrimsel hesaplama tekniklerinden biri olan genetik algoritma yöntemi kullanılarak, olimpiik ve yarı olimpiik yüzme havuzlarının plan şemalarını oluşturulmayı hedeflemektedir.

Olimpiik ve yarı olimpiik yüzme havuzlarını sporcular ve seyirciler tarafından daha etkin ve kullanışlı kılan şey, onların tasarımına ve yerleşik düzendeki akış ilişkilerine bağlıdır. Bu ve benzeri tesislerin yerleşimindeki akış ilişkileri ve tasarım problemleri çözüm gerektiren problemlerdir. Bu çalışmada mimari plan kurgularının genetik algoritma kullanılarak bilgisayar ortamında bu problemlere çözüm üretilebileceği görülmüştür. Bu amaçla tez kapsamında ASP .NET programlama dili kullanılarak web kullanıcı etkileşimli bir program oluşturulmuştur.

Bu tez çalışmasında yüzme havuzlarının mimari plan kurgusunun yapılacağı arazi seçimi, çalışma kapsamında oluşturduğumuz yazılıma entegre edilen Google Map aracılığı ile yapılmaktadır. OPSO (Olympic pool space organizer) adını verdiğimiz programın panel bölümünün alan kısmında öncelikle harita üzerinde belirlenen sınırların ölçüleri metre kare cinsinde verilmektedir. Bu ölçüler veri tabanına aktarılmakta ve oluşturmak istediğimiz havuz tipi, seyirci sayıları gibi veriler girilerek hesaplama butonuna bastığımızda, program kullanıcıya en uygun havuz plan şemasını oluşturmaktadır. Elde edilen plan sonuçları ile gelecekte inşa etmek istediğimiz yüzme havuzlarının yerleşim planlarını daha iyi, daha hızlı bir şekilde kurgulama olanağı sunacaktır. Ayrıca çalışma kapsamında panel kısmındaki Hesaplama butonunu aktif duruma getirerek yüzme havuzu plan şemasının ana mekanlarını elde

edip, aktif duruma getirilecek Generate butonu ile mekanlar içerisindeki bölümleri oluşturmak hedeflenmişti. Süre kısıtı nedeni ile mekanların içerisindeki bölümler oluşturulamamıştır. İleriki çalışmada bu model daha da geliştirilip kullanıcılara üç boyutlu alternatifler sunması, bunun yanısıra programın makine öğrenme kısmını geliştirip otomatik olarak internet üzerinden yayınlanmış havuz salonlarının planlarını kendi veri tabanına alıp açık kaynaklı bir program haline dönüşebilmesi amaçlanmaktadır.

KAYNAKLAR

- Angeline, P. J.** (1995). Evolution revolution: An introduction to the special track on genetic and evolutionary programming, IEEE Expert Intelligent Systems and their Applications 10, June pp.6-10
- Bentley, P. J., & Wakefield, J. P.** (1997). Conceptual evolutionary design by a genetic algorithm. *Engineering design and automation*, 3, 119-132.
- Curtis, H.** (1975). Biology, 2nd Ed., New York: Worth publisher.
- Caldas, L. G., & Norford, L. K.** (2003). Genetic algorithms for optimization of building envelopes and the design and control of HVAC systems. *Journal of solar energy engineering*, 125(3), 343-351.
- Emel, G. G., & Taşkın, Ç.** (2002). Genetik Algoritmalar ve Uygulama Alanları. *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 21(1), 129-152.
- Goldberg, D. E., & Holland, J. H.** (1988). Genetic algorithms and machine learning. *Machine learning*, 3(2), 95-99.
- Holland, J. H.** (1975). Adaptation in Natural and Artificial Systems, Ann Arbor: The University of Michigan Press.
- Holland, J. H.** (1992). Genetic algorithms, Sci. Am. Pp. 66-72.
- Haupt, R. L. ve Haupt, S. E.** (1998). Practical Genetic Algorithms, A Willey-Interscience Publication, USA.
- Kumar, V.** (1992). Algorithms for constraint-satisfaction problems: A survey. *AI magazine*, 13(1), 32.
- Kurt, S.** (2012) *Spor Salonu Tasarımı: Prof. Dr. Ahmet Taner Kısıralı Kapalı Spor Salonu*.
- Liu, Z.** (1998). *Algorithms for Constraint Satisfaction Problems (CSPs)* (Doctoral dissertation, University of Waterloo).
- Lynn, G.** (2006). Embriyolojik ev modeli. Adres: <http://www.mimarizm.com/isDisi/Detay.aspx?id=2740&BultenID=96&Page=3>
- Michalek, J., Choudhary, R., & Papalambros, P.** (2002). Architectural layout design optimization. *Engineering optimization*, 34(5), 461-484.
- Nabiyev, V. V.** (2012). *Yapay zeka: insan-bilgisayar etkileşimi*. Seçkin Yayıncılık.
- Neufert, E., ve Neufert, P.** (2008). *Indoor Swimming Pools*, s.53.
- Thakur, M. K., Kumari, M., & Das, M.** (2010, July). Architectural layout planning using genetic algorithms. In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on* (Vol. 4, pp. 5-11). IEEE.

Yürekli, D. (2012). Olimpik yüzme havuzu binalarının yapısal performanslarının incelenmesi Duygu Yürekli.

Url-1 <<http://v2.arkiv.com.tr/p7957-kadir-has-stadyumu-olimpik-havuzu.html>>, alındığı tarih: 5.10.2014.

Url-2 <<http://www.neu.edu.tr/en/node/4149>>, alındığı tarih: 07.10.2014.

Url-3 <<http://klignoncamp.wordpress.com/2012/05/31/evrimsel-mimari-uzerine-incelemeler/>>, alındığı tarih: 07.10.2014.

Url-4 <<http://web.deu.edu.tr/fmd/s25/25-07.pdf>>, alındığı tarih: 11.10.2014.

Url-5 <<http://www.chrisbosse.de/watercube/>>, alındığı tarih: 18.04.2015.

EKLER

EK A: Kodlar

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Microsoft.SolverFoundation.Services;
using System.Globalization;
using System.Security.Cryptography;
using System.Text;
using System.Collections;
using Mapack;

public partial class beyin : System.Web.UI.Page
{
    public String placeWithGeneticAlg(String input,String domain,int iteration)
    {
        String output = "";
        String[] vals = input.Split('/');
        Hashtable M = new Hashtable();
        List<String> geneticMaterials=new List<String>();
        foreach (String val in vals)
        {
            M.Add(val.Split('=')[0], val.Split('=')[1]);
            geneticMaterials.Add(val.Split('=')[0] );
        }
        //genetic algorithm
        Hashtable p = randomPopulation(geneticMaterials);
        for (int i = 0; i < iteration; i++)
        {
            p = evaluateFitnesses(p, domain, M);
            p = chooseWell(p);
            p = crossMatch(p, geneticMaterials);
            p = mutationEffect(p);
            if (i == iteration - 1)
            {
                p = evaluateFitnesses(p, domain, M);
                output = getBest(p);
            }
        }
        return output;
    }
    public Hashtable randomPopulation(List<String> geneticMetarials)
    {
        //This func creates a random population using genetic metarials
        //The created metarial represented with a string whose individuals are sperated with a specific char
        //Pupulation size:4
        int populationSize = 4;
        String individual = "";
        Hashtable population = new Hashtable();
        Random r = new Random();
```

```

int x=0;
List<String> geneList=new List<String>();
for (int i = 0; i < populationSize; i++)
{
    geneList = geneticMetarials.ToList<String>();
    individual = "";
    while (geneList.Count != 0)
    {
        x = r.Next(0, geneList.Count);
        individual += geneList[x];
        if (geneList.Count > 1)
            individual += ",";
        geneList.RemoveAt(x);
    }
    try
    {
        //Response.Write(individual);
        population.Add(individual, 0);
    }
    catch (Exception ex)
    {
        i--;
        continue;
    }
}
return population;
}
public Hashtable evaluateFitnesses(Hashtable population, String domain, Hashtable M)
{
    int en = Convert.ToInt32(domain.Split('x')[0].ToString());
    int boy = Convert.ToInt32(domain.Split('x')[1].ToString());
    Hashtable scoresOfIndividuals =new Hashtable();
    foreach(String individual in population.Keys)
    {
        //This part of the code can be specialized according to the purpose
        //In this code
        //The first component of individual is placed top left of the domain
        //Then the second is placed to the right side of the previous one if there is enough area for it, else it is
        placed in the under the previous one
        Matrix m = new Matrix(en, boy);
        // Response.Write("<br>" + individual + "yeni " + m.ToString() + "<br>");
        foreach(String c in individual.Split(','))
        {
            m = setMap(m, c, M);
        }
        scoresOfIndividuals.Add(individual, countZero(m));
        //Response.Write("<br>" + m.ToString() + "<br>" + countZero(m));
    }
    return scoresOfIndividuals;
}
public Matrix setMap(Matrix m, String c, Hashtable M)
{
    int e1 = Convert.ToInt32(M[c].ToString().Split('x')[0]);
    int b1 = Convert.ToInt32(M[c].ToString().Split('x')[1]);
    for (int i = 0; i < m.Rows; i++)
        for (int j = 0; j < m.Columns; j++)
            if (m[i, j] == 0)
            {
                try
                {
                    for (int k = 0; k < b1; k++)
                    {
                        for (int h = 0; h < e1; h++)
                            if (m[i + k, j + h] != 1)
                            {
                                m[i + k, j + h] = 1;
                            }
                    }
                }
            }
}

```

```

        else
            break;
        }
        return m;
    }
    catch (Exception e)
    {
        break;
    }
}
return m;
}
public int countZero(Matrix m)
{
    int n=0;
    for(int i=0;i<m.Rows-1;i++)
        for(int j=0;j<m.Columns-1;j++)
            if (m[i, j] == 0 && m[i + 1, j] == 0 && m[i, j + 1] == 0 && m[i+1, j+1] == 0)
                n++;
    return n;
}
public Hashtable chooseWell(Hashtable scoresOfIndividuals)
{
    ArrayList V=new ArrayList(scoresOfIndividuals.Values);
    int [] arr=V.ToArray(typeof(int)) as int [];
    Array.Sort(arr);
    Hashtable wells=new Hashtable();
    for (int i = 0; i < arr.Count();i++ )
    {
        foreach (String k in scoresOfIndividuals.Keys)
        {
            if (arr[i] == (int)scoresOfIndividuals[k])
                wells.Add(k, arr[i]);
            if (wells.Count == (int)scoresOfIndividuals.Count / 2)
                return wells;
        }
    }
    return wells;
}
public String getBest(Hashtable scoresOfIndividuals)
{
    String output = "";
    ArrayList V = new ArrayList(scoresOfIndividuals.Values);
    int[] arr = V.ToArray(typeof(int)) as int[];
    Array.Sort(arr);
    Hashtable wells = new Hashtable();
    foreach (String k in scoresOfIndividuals.Keys)
        if (arr[0] == (int)scoresOfIndividuals[k])
            output = k;
    return output;
}
public Hashtable crossMatch(Hashtable wells,List<String> geneticMetarials)
{
    int geneCount = geneticMetarials.Count();
    Hashtable newPopulation = new Hashtable();
    String[] part1 = new String[geneCount];
    String[] part2 = new String[geneCount];
    Random r = new Random();
    int k=0,x=0;
    foreach (String individual in wells.Keys)
    {
        //Response.Write("<-->" + individual);
        newPopulation.Add(individual, 0);
        if (k == 0)
        {

```

```

        x = r.Next(0, geneCount - 1);
        for (int i = 0; i < geneCount; i++)
        {
            if (i < x)
                part1[i] = individual.Split(',')[i];
            else
                part2[i] = individual.Split(',')[i];
            k = 1;
        }
    }
    else
        for (int i = 0; i < geneCount; i++)
        {
            if (i < x)
                part2[i] = individual.Split(',')[i];
            else
                part1[i] = individual.Split(',')[i];
            k = 11;
        }
    if (k == 11)
    {
        try
        {
            newPopulation.Add(normalizeIt(string.Join(",", part1), geneticMetarials), 0);
        }
        catch (Exception e)
        {
            continue;
        }
        try
        {
            newPopulation.Add(normalizeIt(string.Join(",", part2), geneticMetarials), 0);
        }
        catch (Exception e)
        {
            continue;
        }
        //Response.Write(string.Join(",", part1));
        k = 0;
    }
}
return newPopulation;
}
public String normalizeIt(String individual, List<String> geneticMetarials)
{
    String[] genes = individual.Split(',');
    List<String> genesList = new List<String>(geneticMetarials);
    for (int i = 0; i < geneticMetarials.Count(); i++)
    {
        if (!genesList.Contains(genes[i]))
        {
            genes[i] = genesList[0];
        }
        genesList.Remove(genes[i]);
    }
    individual = string.Join(",", genes);
    return individual;
}
public Hashtable mutationEffect(Hashtable population)
{
    Random r = new Random();
    int x = 0, y = 0;
    String tmp = "";
    List<String> keys = population.Keys.OfType<String>().ToList();
    foreach (String individual in keys)
    {
        if (r.Next(0, 100) > 70)

```

```

    {
        try
        {
            population.Remove(individual);
            String[] G = individual.Split(',');
            x = r.Next(0, G.Count() - 1);
            y = r.Next(0, G.Count() - 1);
            tmp = G[y];
            G[y] = G[x];
            G[x] = tmp;
            population.Add(string.Join(",", G), 0);
        }
        catch
        {
            continue;
        }
    }
    return population;
}

```


ÖZGEÇMİŞ

Ad Soyad : Kadir ÇALIŞIR

Doğum Yeri ve Tarihi : 06.01.1978

E-Posta : kcalisir@itu.edu.tr

ÖĞRENİM DURUMU:

- **Lisans** : 2006, Girne Amerikan Üni., Yönetim Bilişim Sis.