

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**SÜRDÜRÜLEBİLİR TOPLU KONUT YERLEŞMESİ
TASARIMI İÇİN PARETO GENETİK ALGORİTMAYA DAYALI
BİR MODEL ÖNERİSİ: SSPM**



DOKTORA TEZİ

Yazgı AKSOY

Bilişim Anabilim Dalı

Mimari Tasarımda Bilişim Programı

MAYIS 2016

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**SÜRDÜRÜLEBİLİR TOPLU KONUT YERLEŞMESİ
TASARIMI İÇİN PARETO GENETİK ALGORİTMAYA DAYALI
BİR MODEL ÖNERİSİ: SSPM**

DOKTORA TEZİ

**Yazgı AKSOY
(523082006)**

Bilişim Anabilim Dalı

Mimari Tasarımda Bilişim Programı

Tez Danışmanı: Prof. Dr. Gülen ÇAĞDAŞ

MAYIS 2016

İTÜ, Fen Bilimleri Enstitüsü'nün 523082006 numaralı Doktora Öğrencisi Yazgı AKSOY, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “SÜRDÜRÜLEBİLİR TOPLU KONUT YERLEŞMESİ TASARIMI İÇİN PARETO GENETİK ALGORTİMAYA DAYALI BİR MODEL ÖNERİSİ: SSPM” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. Gülen ÇAĞDAŞ**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Mine ÖZKAR**
İstanbul Teknik Üniversitesi

Prof. Dr. Deniz E.ÖNDER
Yıldız Teknik Üniversitesi

Prof. Dr. Salih Ofluğlu
Mimar Sinan Güzel Sanatlar Üniversitesi

Yrd.Doç. Dr. Sema Alaçam
İstanbul Teknik Üniversitesi

Teslim Tarihi : **7 Nisan 2016**
Savunma Tarihi : **6 Mayıs 2016**





Kızıma,



ÖNSÖZ

Sevgili Prof. Dr. Gülen ÇAĞDAŞ'a, tüm üniversite öğrenim sürecimde bana yol gösterdiği, kritikleriyle ilham verdiği, teşvikleriyle beni bu yolculuğa yönelterek ufkumu açtığı ve adım adım her aşamayı kontrol altında tutarak bu noktaya gelmemi sağladığı için; Prof.Dr. Mine ÖZKAR ve Prof. Dr. Deniz ERİNSEL ÖNDER'e çalışmamın gelişmesine jüri üyeleri olarak yaptıkları değerli katkıları için; Özgün BALABAN'a, modelin kurgusunda ve yazılımında yardımcı olarak bu çalışmanın önünü açtığı için sonsuz teşekkürler.

Bu tezi, “Sürdürülebilir Toplu Konut Yerleşmesi Tasarımı için Pareto Genetik Algoritmaya Dayalı Bir Model Önerisi: SSPM ” isimli araştırma projesi ile destekleyen İTÜ Bilimsel Araştırma Projeleri Birimi (BAP)'ne destekleri nedeniyle teşekkür ederim.

Son olarak çalışmamın birçok aşamasında tüm ilgi ve sabırlarıyla her türlü desteği veren eşim Koray AKSOY'a, her anlamda beni destekleyen, sevgili annem Nursen BADEM'e, canım babam Güray BADEM'e, çok değerli kayınvalidem Gülten AKSOY ve kayınpederim Enver AKSOY'a, çevirileriyle işimi kolaylaştıran biricik kardeşim Ezgi BADEM'e yardımlarından ötürü minnettarım. Varlığıyla bizi mutlu eden dünyalar tatlısı kızım Selin'e de tezin her sayfasında parmağı olduğu için en içten teşekkürlerimi sunarım.

Mayıs 2016

Yazgı Aksoy
(Y.Mimar)



İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET.....	xvii
SUMMARY	xxi
1. GİRİŞ.....	1
1.1 Problemin Belirlenmesi ve Tanımı	2
1.2 Tezin Amacı	6
1.3 Tezde İzlenen Yöntem	8
2. EVRİMSEL ALGORİTMALAR	11
2.1 Genetik Algoritmalar.....	12
2.1.1 Çok amaçlı genetik algoritmalar	15
2.1.1.1 Pareto optimal kavramı	16
2.1.1.2 Pareto baskınlık ilişkisi, pareto optimal küme ve pareto öncüsü	17
2.1.2 Çok amaçlı genetik algoritma örnek modelleri.....	19
2.2 NSGA-II Evrimsel Algoritması (Bastırılmamış Sınıflandırılmalı Genetik Algoritma)	23
3. TASARIMDA SÜRDÜRÜLEBİLİRLİK	25
3.1 Sürdürülebilirlik Kavramı ve İlkeleri	25
3.2 Sürdürülebilir Arazi Yerleşim Kriterleri	26
3.2.1 İmar planları ve yerel yönetmelikler	27
3.2.2 Bina puanlama sistemleri	28
3.2.2.1 Leed (Leadership in energy and environmental design)	28
3.2.2.2 Breeam (Building research establishment environmental assessment methodology)	30
3.2.3 Fiziksel çevre kontrolü kriterleri	32
4. SÜRDÜRÜLEBİLİR TASARIMDA KULLANILAN BENZETİM MODELLERİ VE MEVCUT EVRİMSEL MODELLER.....	35
4.1 Sürdürülebilir Bina Tasarım Sürecinde Kullanılan Benzetim Modelleri	35
4.2 Sürdürülebilir Bina Tasarım Sürecinde Kullanılan Evrimsel Modeller	40
4.2.1 Yeşil bina tasarımı için plan eniyileme modeli.....	40
4.2.2 SMOLA-Mekansal eniyileme yöntemi ile sürdürülebilir arazi kullanım modeli.....	42
5. SÜRDÜRÜLEBİLİRLİK BAĞLAMINDA TOPLU KONUT YERLEŞİMİ TASARLAYAN EVRİMSEL BİR MODEL: SSPM.....	47
5.1 Sürdürülebilir Yerleşim Planı Modeli: SSPM.....	47
5.2 Arazinin Tanım Tekniği	50
5.3 Modelin Kullanıcı Arayüzü.....	53

5.4 Modelin Değişkenleri	54
5.5 Modelin Üretim ve Uygunluk Fonksiyonları Bağlamında Değerlendirme Sistemi	56
5.5.1 Modelin değerlendirme kriterleri	57
5.5.2 Modelin uygunluk puanları	62
5.5.3 Modelin veri girişi, üretim ve seçim sistemi	64
5.5.4 Modelin veri girişi, üretim ve seçim algoritması	66
5.6 Modelin Uygulanması	69
6. SONUÇLAR	85
KAYNAKLAR.....	91
EKLER.....	95
ÖZGEÇMİŞ.....	125



KISALTMALAR

BRE	: Building Research Establishment
BREEAM	: Building Research Establishment Environmental Assessment Methodology
ÇEDBİK	: Çevre Dostu Yeşil Binalar Derneği
EA	: Evolutionary Algorithms
EP	: Evolutionary Programming
ES	: Evolutionary Strategies
FAR	: Floor Area Ratio
FSR	: Floor Space Ratio
GA	: Genetic Algorithms
GP	: Genetic Programming
LCA	: Life-cycle assessment
LEED	: Leadership in Energy and Environmental Design
LEED – NC	: LEED for new Construction
LEED – EB	: LEED for Existing Buildings
LEED – CI	: LEED for Commercial Interiors
LEED – CS	: LEED for Core and Shell
LEED – H	: LEED for Homes
LEED – ND	: LEED for Neighbourhood Development
MOGA	: Multi-Objective Genetic Algorithms
MOOP	: Multi-Objective Optimization
NSGA	: Nondominated Sorting Genetic Algorithm
PESA	: Pareto Envelope Based Selection Algorithm
SMOLA	: Sustainable Multi Objective Land Use Allocation Model
SOOP	: Single-Objective Optimization Problem
SPEA	: Strength Pareto Evolutionary Algorithm
SSPM	: Sustainable Site Planning Model
TAKS	: Taban Alanı Katsayısı
UIA	: International Union of Architects
UNESCO	: United Nations Educational, Scientific and Cultural Organization
USGBC	: The U.S. Green Building Council
VEGA	: Vector Evaluated Genetic Algorithm



ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 : Çok amaçlı evrimsel algoritma örnek modeller	21
Çizelge 4.1 : Bina benzetim modelleri	37





ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : SSPM Modeli Kavramsal Şeması	9
Şekil 2.1 : Genetik Algoritma Akış Diyagramı.	13
Şekil 2.2 : Örnek amaç pareto grafiği (f1, f2).....	17
Şekil 2.3 : Örnek Pareto Eğrisi(minimum uzaklık-minimum maliyet).	18
Şekil 4.1 : Binanın oryantasyonunun tanımlanması.	41
Şekil 4.2 : 20x20 hücreden oluşan bir arazi parçası.	43
Şekil 4.3 : Arazi kullanım biçimleri	44
Şekil 4.4 : Modelle üretilen arazi kullanım örüntülerinin karşılaştırılması.	45
Şekil 5.1 : Modelin akış diyagramı.....	49
Şekil 5.2 : Örnek arazinin Excel'de matrisle tanımlanması ...	52
Şekil 5.3 : Excel'de tanımlanmış 3m-6m kot farklı örnek arazi parçası matrisi	52
Şekil 5.4 : Verilerin girildiği kullanıcı ara yüzü ...	54
Şekil 5.5 : Sürdürülebilir tasarım modeli ana kavramları ...	57
Şekil 5.6 : Binanın arkadan ve yandan yaklaşma mesafelerine karşılık gelen bahçe, parsel ve bina boyutu tanımlamaları ...	59
Şekil 5.7 : Rüzgâr puanının gösterimi ...	63
Şekil 5.8 : Manzara puanının gösterimi.....	64
Şekil 5.9 : Modelde verilen çaprazlama noktası.....	65
Şekil 5.10 : Bina yönlenmesine göre bina yaklaşma mesafelerine karşılık gelen yan bahçe, arka bahçe konumları	66
Şekil 5.11: Modelin test edildiği Kağıthane arazisi ...	70
Şekil 5.12 : Seçilen arazinin görsel tanımı	70
Şekil 5.13 : Seçilen arazinin Excel'de hazırlanan matrisi	71
Şekil 5.14 : B Arazinin Excel'de hazırlanan matrisinden bir bölüm (A).....	72
Şekil 5.15 : Arazinin Excel'de hazırlanan matrisinden bir bölüm (B) ...	73
Şekil 5.16 : Arazinin Excel'de hazırlanan matrisinden bir bölüm (C).....	74
Şekil 5.17 : Kağıthane arazisi için kullanıcı arayüzünden girilen değerler	75
Şekil 5.18 : Rastgele üretilen bina tipleri için arazi yerleşim planı üretimleri 1.Nesil.....	77
Şekil 5.19 : Üç tip bina için arazi yerleşim planı üretimleri 1. Nesil	78
Şekil 5.20 : Üç tip bloktan oluşan üretimin 10. nesil bireyleri ve pareto eğrileri	80
Şekil 5.21 : Üç tip bloktan oluşan üretimin 15. nesil bireyleri ve pareto eğrileri	81
Şekil 5.22 : 15.nesildeki en yüksek puanlı bireyin 3 boyutlu gösterimi.....	82



SÜRDÜRÜLEBİLİR TOPLU KONUT YERLEŞMESİ TASARIMI İÇİN PARETO GENETİK ALGORİTMAYA DAYALI BİR MODEL ÖNERİSİ: SSPM

ÖZET

Sürdürülebilir yerleşme ve bina tasarımı geleneksel bina tasarımına göre daha karmaşık bir sürece sahiptir. Bunun nedeni birlikte alınması gereken fakat birbiriyle çelişen pek çok tasarım kararının erken tasarım evresinde alınması gerekliliğidir. Sürdürülebilir tasarımın erken evresinde tasarımcıya yardımcı bir karar destek sistemine ihtiyaç kaçınılmazdır; fakat tasarım endüstrisinde, bilgisayar tabanlı araçlar tasarım sürecinin önemli bir parçasını oluşturmalarına rağmen, erken tasarım evresi bilgisayar desteğinin en az kullanıldığı evre olmuştur. Bu çalışmada, pareto temelli bastırılmamış sınıflandırmalı genetik algoritma (NSGA-II) yöntemi kullanılarak pek çok tasarım kriterini göz önünde bulundurması gereken tasarımcıya karar verme sürecinde yardımcı olacak evrimsel bir model geliştirilmesi amaçlanmıştır. Sürdürülebilir Arazi Yerleşme Planlama Modeli (SSPM) olarak adlandırılan model, evrimsel algoritmaların sürdürülebilir arazi yerleşiminde kullanılmasıyla, erken tasarım evresinde toplu konutların araziye yerleşiminde mimarı yönlendirebilecek bir karar destek sistemi olarak önerilmiştir.

Evrimsel algoritmalarda en iyi çözümler, evrimsel süreçlere maruz bırakıldıktan sonra, olası çözümler kümesinden elde edilir. Eğer eniyilenecek tek bir amaç varsa, problem “tek amaçlı eniyileme problemi”; birden fazla amaç olması durumunda ise “çok amaçlı eniyileme problemi” olarak tanımlanır. Çok amaçlı eniyileme problemlerinde, birbiriyle çelişen amaçlar nedeniyle problemin çözümleri birden fazladır. Bu çözümlere etkin çözümler denir. Pareto analizi değişik sayıdaki önemli etkenleri, daha az önemde olan etkenlerden ayırmak için kullanılan bir tekniktir. Pareto-optimal kavramının evrimsel algoritmalara uyarlanmasıyla, SSPM modelinde kullanılan pareto temelli çok amaçlı evrimsel eniyileme algoritmaları geliştirilmiştir. Buna göre, popülasyondaki en iyi çözüm, amaçların herhangi biri içinde en kötü olmayan ve en azından bir amaç içinde diğerlerinden daha iyi olan çözüm anlamındadır. Tasarımın ilk aşamasında eldeki tasarım problemine uygun tüm sınırlandırmaları göz önünde bulundurarak toplu konut yerleşim önerileri sunan evrimsel modelin tanımlanması ve modelin alan çalışması yapılarak değerlendirilmesi bu çalışmanın konusunu oluşturacaktır.

Sürdürülebilir tasarım sürecinde, sezgisel ve uygulama tecrübesine dayanan geleneksel tasarım yöntemi yerine, hesaplama yöntemleriyle problem alanına daha geniş açıdan bakılarak tatminkâr çözüm alternatifleri elde edilebilir. Tasarımın ilk evresinde sayısal ortamda planlanmış ve tasarlanmış bir çevre, tasarımcının yaratıcılığının doğru yönde gelişmesini sağlar. Bu sayede üretilen oldukça fazla sayıda alternatif objektif olarak değerlendirilebilir.

Tez kapsamında, mimari tasarımda öncelikle çok amaçlı eniyilme modelleriyle sürdürülebilir tasarım konuları incelenmiştir. Tezin 4. bölümünde bu anlamda

geliştirilen iki örnek model tanıtılmıştır. İlk olarak Weimin Wang, Hugues Rivard ve Radu Zimeureanu tarafından geliştirilen, yeşil bina tasarımı için planlar üreten eniyileme modeli ele alınmıştır. Bu çalışmada bina biçimlerinin genetik algoritma kullanılarak eniyilenmesi amaçlanmıştır. Çok amaçlı genetik algoritma, üretilen plan alternatiflerini duvar malzemeleri ile birlikte değerlendirerek, maksimum doğal ışık ve minimum enerji tüketimine göre eleyerek, pareto optimal alternatifi kullanıcıya sunmaktadır. Bina biçimi yanında malzeme ve maliyeti göz önünde bulundurması açısından işlevsel bir model olarak tanımlanabilir, fakat erken tasarım evresinde birden fazla binadan oluşan kompleks sistemlerin araziye yerleşiminde yetersiz kalmaktadır. Zielinska, Church ve Jankowski tarafından geliştirilen SMOLA (Sustainable Multi- Objective Land Use Allocation Model) ise çok amaçlı mekânsal eniyilemeyi hedeflemektedir. SMOLA çelişen iki kriter olan maksimum açık alan ve maksimum kalkınma, komşu yerleşimlere uygunluk ve var olan yerleşim alanlarına yakınlık şartlarını göz önünde bulunduran kentsel ölçekli çok amaçlı mekânsal eniyileme modelidir. Model kentsel yerleşim problemleri için kısıtlayıcı olarak düşünülebilir, fakat mekânsal eniyileme modellerinin gelişen şehirlerin kentsel formunu yönlendirmede büyük rolü olduğu düşünülmektedir.

Bu tez kapsamında önerilen sürdürülebilir bina modeli sadece enerji etkin bina modeli olarak değil, eskiz aşamasından başlayarak bina ölçeğinde sürdürülebilir arazi kullanımına önem veren bütünsel bir modeldir. Bu konuda yapılmış diğer modellerden farklı olarak, Amerika kökenli LEED (Leadership in Energy and Environmental Design) ve İngiltere kökenli BREEAM (Building Research Establishment Environmental Assessment Method) arazi yerleşim kriterlerinin yanında yerel yönetmelikleri ve yerel iklim şartlarını da göz önünde bulunduran modelin, üretilen alternatifleri uygunluk puanlarına göre birbirleriyle kıyaslayarak pareto optimal sonuçlar üretmesi, tasarımcıyı sürdürülebilir arazi kullanım çözümlerine götürmesi hedeflenmiştir.

Toplu konut yerleşmesinin tasarlanacağı arazi, matrisle tanım tekniği kullanılarak temsil edilmiş ve matrisler Excel'de oluşturulmuştur. Arazinin tanımlanmasında Excel programı, veri girişinin pratik ve kolay tanımlanabilir olması ve matrisle tanım tekniğine olanak sağlaması nedeniyle tercih edilmiştir. Matrisin satırlarını ve sütunlarını oluşturan her bir elemanı (hücresi), arazinin alanı 1 m² olan bir birimini temsil etmektedir. Arazi köşe noktalarının ve arazi üzerindeki mevcut elemanların x,y koordinatlarına göre matris oluşturulmaktadır. Arazinin topografik yapısını ve doğal oluşumları göz ardı etmeden, çözümün gerçekçi sonuçlar verebilmesi için, arazi verilerinin de gerçeğe yakın modellenmesi gerekmektedir. Bunu sağlamak için sayısal araziye eldeki eğim değerlerine göre üç boyutlu karakter kazandırılmıştır. Yükselti farklarının çözümünde, araziden gelen eğim değerleri kullanıcı isteğine göre farklı kotlardaki parçalara bölünerek modele tanıtılmaktadır. Bu sayede yüksek kot farkına sahip kademeli arazilerde teraslama yapılarak, topografya ile uyumlu arazi yerleşim çözümlerine gidilebilecektir.

Sürdürülebilir bina tasarımı amaçlayarak geliştirilen pek çok evrimsel modelden farklı olarak, arazideki kot farklarını ve üretilen bina yüksekliklerini 3. boyutta kullanıcıya gösteren SSPM modeli, gerçek arazi verilerini sayısal olarak kullanarak, arazideki korunacak ağaçları, su öğelerini ve yeşil alanları üretimlerine dâhil etmektedir. Böylece güncel veri grubu üzerinden üretimlerini yaparak gerçekçi sonuçlar vermektedir.

Modelin sınanması için Kağıthane'de yer alan 4500 m²' büyüklüğünde bir arazi seçilmiştir. Hipotetik bir arazi yerine, daha önceden bir bölümünün kullanılmış

olması, sınırları içerisinde korunacak ağaç ve su öğelerini içermesi sebebiyle bu arazi tercih edilmiştir. Arazi İstanbul ilinde olması sebebiyle uygunluk fonksiyonlarının bir kısmını İstanbul İli İmar Yönetmeliği kuralları oluşturmuştur. Araziye ait TAKS, KAKS, hmax, manzara, rüzgar ve iklim verilerine göre üretilmiş ilk popülasyon, sonrasında NSGA-II algoritması döngüsü içerisinde sokularak pareto optimal arazi yerleşim bireyleri elde edilmiştir. Her nesilde üretilen birey sayısı 100 ile sınırlandırılmıştır, kullanılan bilgisayarın kapasitesi nedeniyle daha fazla birey üretimi modelin çalışma hızını oldukça düşürmektedir. Yerleşim, manzara ve rüzgâr açıları kuzey-güney-doğu ve batı olmak üzere doğrusaldır, açılı yerleşimler modelde üretim dışında tutulmuştur. Arazi, bünyesindeki kot farklarına göre kullanıcı tarafından teraslanarak modele tanıtılmıştır.

Hem yerleşilen arazinin eğim verilerini okuyabilen, hem de yerleştirilen blokların yüksekliğini göz önünde bulundurarak yaklaşma mesafelerini hesaplayan model konut bloklarının yerleşimi için alternatif sonuçlar üretmiştir. Çalışmada farklı nesillerde, oluşan pareto eğrileri kıyaslanarak modelin üretimleri değerlendirilmiştir. İlk popülasyon, sonrasında NSGA-II algoritması döngüsü içerisinde sokularak pareto optimal arazi yerleşim bireyleri elde edilmiştir.

Uygunluk puanları verilen bireylerin toplam skorları 0 ile 1 arasında normalize edilmiştir; 1 en başarılı bireylerdir, 0 en başarısız bireylerdir; sonrasında turnuva seçilimi uygulanır. Turnuva seçiminde skoru yüksek olan bireylerin üreme şansı daha fazladır. Bireyler kendi skorlarına göre oluşturulan matrizen seçilirler.

İlk üretilen bireylere bakıldığında arazideki korunması gerekli ağaçlar dikkate alınarak blok yerleşimleri yapılmış, Kağıthane Deresi ve 30'ar metre çevresi de koruma altına alınarak bloklar bu bölgeden uzağa yerleştirilmiştir. Bloklar manzaraya yönelmiştir. Birbirinin manzarasını kesmeyen fazla puanlı alternatiflerin üretim şansı da fazladır. Elde edilen ilk neslin pareto eğrisi eniyileme tamamlanmadığı için doğrusal olarak elde edilmiştir. Nesil ilerledikçe pareto eğrisi sıcak iklime sahip olan Kağıthane arazisi için hem manzara hem de rüzgar fonksiyonlarını maksimize etmek için konkavlaşmıştır. İleriki nesillerin pareto grafiğinde yeşille gösterilen pareto optimal bireyler hem manzara hem de rüzgarı maksimize etmiş ve üretilmiş diğer hiç bir birey tarafından bastırılmamış en yüksek puanlı bireyler olarak karşımıza çıkmıştır.

Elde edilen pareto optimal arazi yerleşim bireyleri, kütle yerleştirme problemlerinde genetik algoritmanın iyi ve tutarlı sonuçlar verdiğini göstermektedir. Özellikle yoğun yapılaşmaya izin veren araziler üzerinde birbiriyle çakışan arazi yerleşim kriterlerini göz önünde bulundurarak blok yerleşimlerini yapmak geleneksel yöntemle oldukça zaman almaktadır. Modelin ürettiği pareto optimal arazi bireylerinin çok amaçlı genetik algoritmanın yardımıyla birkaç dakika içerisinde elde edildiği düşünülürse, bu modelin erken tasarım evresindeki önemli bir kullanım potansiyeline sahip olduğu söylenebilir. Sayısal ortamda planlanmış ve tasarlanmış bir çevre, tasarımcının yaratıcılığını doğru yönde ilerletmesine olanak sağlayarak oldukça fazla sayıda alternatifi değerlendirmede yardımcı olacaktır. İki farklı disiplin olan evrimsel algoritmalar ve sürdürülebilir tasarım ara kesitinde yer alan bu çalışma ile gerçekleştirilecek sürdürülebilir yerleşme tasarımı modelinde amaç, uzman bir tasarımcının yerini almak değil; tasarımcıya tasarım kararlarının alınması sürecinde yardımcı olmasını sağlamaktır. Sürdürülebilir tasarım sürecinde, sezgisel ve uygulama tecrübesine dayanan geleneksel tasarım yöntem ve teknikleri yanında;

hesaplamalı yöntemlerle problem alanına daha geniş açıdan bakarak çözüm alternatifleri üretmek, tatminkâr çözümlerin elde edilmesini sağlayacaktır.



A PARETO BASED GENETIC ALGORITHM MODEL FOR SUSTAINABLE SITE LAYOUT DESIGN OF HOUSING: SSPM

SUMMARY

Nowadays as the aim to reduce the environmental impact of buildings becomes more apparent, a new architectural design approach is gaining momentum called sustainable architectural design. Sustainable architectural design process includes some regulations itself, which requires calculations, comparisons and consists of several possible conflicting objectives that need to be considered together. In design industry, advanced computer aided design tools have an important impact on design process, but still early design stage and sustainable design are problematic issues, and need to be solved. Sustainable building design refers to a process that begins with selecting the site and optimizing economic and environmental performance throughout a building's life cycle. In order to achieve a successful sustainable building, particular attention needs to be paid to the conceptual design stage when the most important decisions are taken; nevertheless it is the stage with least computer support.

In architectural design process, generally optimization algorithms have been used to automate the generation of design layouts. Hence there is a need for design tools that can help designers better manage collaborative design development. It is becoming difficult to improve the performance of building design based only on improvements in individual disciplines. For this reason, better, system-orientated, holistic, multidisciplinary approaches to building design are needed.

A successful green building design can be performed by the creation of alternative designs generated according to all the sustainability parameters and local regulations. Green building rating systems and certification programs are accepted as one of sustainability parameters. In this thesis, LEED (Leadership in Energy and Environmental Design) and BREEAM (Building Research Establishment Environmental Assessment Method) certification systems are going to be considered, as being the most representative building environment assessment schemes that are in use. Although these certification systems are used all over the world, the parameters are prepared according to America's and Britain's geographical, economic and cultural conditions, though other countries are experiencing difficulties during sustainable design process. As a result of this, green buildings should be designed also according to the climate of the region and local building construction regulations. To summarize, today some tools developed to assist designers during sustainable building design process, address more detailed design stages when important design decisions already have been taken in conceptual stage. As there are conflicting criteria's according green building rating systems sustainable site parameters, local regulations and local climate conditions, an efficient decision support system can be developed by the help of Pareto based non-dominated genetic algorithm (NSGA-II) which is used for several possibly conflicting objectives that

need to be considered together. Genetic algorithm is a population-based search technique inspired from the biological principles of natural selection and genetic recombination. Genetic algorithm is a suitable method for multi-objective optimization problems because it can generate multiple Pareto optimal solutions in a single simulation run. NSGA is a very effective algorithm but it has computational complexity, lack of elitism. NSGAII was developed as a modification, which has a better sorting algorithm, incorporates elitism and no sharing parameter needs to be chosen a priori. The NSGA-II procedure has three features to find multiple Pareto-optimal solutions in a multi-objective optimization problem: It uses an elitist principle and an explicit diversity preserving mechanism, and also it emphasizes non-dominated solutions NSGA-II-based optimization process is used to develop the Sustainable Site Planning Model (SSPM).

There are simultaneous optimization of several possibly conflicting objectives in multi-objective optimization problems that result in a set of non-dominated solutions. There does not exist a single solution that simultaneously optimizes each objective. In that case, there exist an infinite number of non-dominated solutions which are also known as Pareto optimal solutions. In sustainable design, maximum energy conservation and utilization of natural light can be given as an example of two conflicting objectives, so during sustainable design process Pareto genetic algorithm will be successful to generate design alternatives according to conflicting criteria.

Genetic Algorithms (GAs) are adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. As such they represent an intelligent exploitation of a random search used to solve optimization problems. GAs simulate the survival of the fittest among individuals over consecutive generation for solving a problem. Each individual represents a point in a search space and a possible solution. The individuals in the population are then made to go through a process of evolution. Although randomised, GAs are by no means random, instead they exploit historical information to direct the search into the region of better performance within the search space.

The presence of multiple objectives in a problem, in principle, gives rise to a set of optimal solutions known as Pareto-optimal solutions, instead of a single optimal solution. In the absence of any further information, one of these Pareto-optimal solutions cannot be said to be better than the other. This demands a user to find as many Pareto-optimal solutions as possible. Classical optimization methods suggest converting the multiobjective optimization problem to a single-objective optimization problem by emphasizing one particular Pareto-optimal solution at a time. When such a method is to be used for finding multiple solutions, it has to be applied many times, hopefully finding a different solution at each simulation run. Over the past decade, a number of multiobjective evolutionary algorithm. The primary reason for this is their ability to find multiple Pareto-optimal solutions in one single simulation run. Since evolutionary algorithms (EAs) work with a population of solutions, a simple EA can be extended to maintain a diverse set of solutions. With an emphasis for moving toward the true Pareto-optimal region, an EA can be used to find multiple Pareto-optimal solutions in one single simulation run. The nondominated sorting genetic algorithm (NSGA) is one of the first such EAs. It is a popular non-domination based genetic algorithm for multi-objective optimization.

Especially in green building design process many efforts have been made to integrate multi objective optimization models and sustainable design. Wang, Zmeureanu and

Rivard optimized the building envelope using multi-objective genetic algorithm. They concentrate on building envelope because of its importance in environmental and economic performance of buildings. As a case study the design of a single-story office building located in Montreal, Canada is determined, only heating and cooling energy consumption are considered. The multi-objective optimization model they used for building envelope design, can be used to locate optimum or near optimum green building designs for given conditions (Wang, Zmeureanu and Rivard, 2005).

In sustainable design land use is another subject that designers concentrate on. Zelinska, Church and Jankowski (2008) present a new multi objective spatial optimization model (SMOLA) which minimizes the conflicting objectives of open space development, infill and redevelopment, land use neighborhood compatibility and cost distance to already urbanized areas. These developed models are chosen for using evolutionary algorithms to solve different scaled design problems. The plan optimization model for green building design developed by Wang, Rivard and Zmeureanu is functional for considering material and cost information with building form, but it is insufficient for early stages of design. Also it is mostly concentrated on energy conservation; site planning for building complexes is disregarded. SMOLA model developed by Zelinska, Church and Jankowski generates land use patterns according to the building functions. For city scale the model is useful and has visual readability, but it is not convenient for small-scale problems.

The SSPM model presented in this thesis is an integrated model that concentrates on wide sustainable design criteria; such as green building certification systems and local building codes for.

The model presented in this thesis will generate site-planning alternatives for social housing according to sustainable design objectives. Multi-objective genetic algorithm is going to be used as a computational approach to generate design alternatives. The Sustainable Site Planning Model (SSPM) is written in Processing which is a Java based programming language. The first step of the model is to define of the site to the computer. Matrix definition technique in Excel is used to define the site with numbers. The defined digital site is used as a base to generate site planning alternatives by SSPM. Excel program is chosen for providing quick and easy data input for matrix definition. The rows and columns of the matrix represent 1 unit = 1 m² of the site. The x, y values of the site corner points and the center or corner points of existing elements are used to define the matrix. Each of the Excel cell is considered as a point in a coordinate system. R1C1 reference style in Excel is used to represent x and y axes to make data input apparent. The rectangular area which uses the values of maximum x and y coordinates of site corner points as its dimensions, is the boundary of the matrix. The cells inside the defined boundary are coloured according to their functions so that visual presentation is provided. In addition, user chooses climate type, the direction of sun, wind and view. SSPM generates site-planning alternatives on defined cells. According to fitness functions and crossover and mutation operators are applied to the population to find the pareto optimal site planning solution.

The selected site must be digitalized closed to the real terrain data without disregarding the topography and the natural formations, so that the site is presented in 3D grid with its slope data. User will divide the site into different zones according to its slope data. As a result of this, terracing high-leveled sites will be possible and

environmentally sensitive solutions will be able to generate. The SSPM model will generate social housing cells according to this terraced site-zones.

The total fitness scores of the site individuals are normalized between 0 and 1; 1 is for most successful individuals, 0 is for unsuccessful individuals. At this stage tournament selection is done. Tournament selection works by selecting a number of individuals from the population at random, a tournament, and then selecting only the best of those individuals. The "tournament" isn't much of a tournament at all, it just involves generating a random value between zero and one and comparing it to a pre-determined selection probability. If the random value is less than or equal to the selection probability, the fitter candidate is selected, otherwise the weaker candidate is chosen. Chosen individuals are copied in direct proportion with their fitness scores, then they divided into two groups as parents. Next, crossover operator is applied between parents. Crossover selects genes from parent chromosomes and creates a new offspring.

The SSPM model was tested on a site in Kağıthane in İstanbul. The reason for choosing Kağıthane was twofold. First, urban regeneration in the residential areas of Kağıthane has recently been a central issue of consideration in Turkey. Second, the selected study area is a realistic example that has reserved water supplies, green areas and polluted areas to test sustainable housing units. The focus of the case study has been the adaptation of the building blocks to local conditions, as well as the sustainable site usage parameters. The site layout alternatives that were generated by the SSPM model show a good adaptation to the site specific constraints and parameters, and good variation, within the limits that are imposed by the view, wind, floor area ratio (FAR) and floor space ratio (FSR) values.

When we examine the first produced individuals, existing trees are protected and the building units are placed 30 meters far from Kağıthane River. Each housing units are directed to the view. The pareto curve of the first produced individuals is linear because the optimization of site plan is not finished. As the generation progresses, the pareto curve for Kağıthane where has hot climate, becomes concave to maximize view and wind functions. For the next generations green individuals are pareto optimal individuals which have maximum view and wind scores and are not dominated by any other individuals.

Pareto analysis and genetic algorithms are two built-in evaluation techniques guiding the creative output of optioneering tools. It can be argued that these evaluation techniques increase the creativity capacity of designer generating a large number of informed guesses at desirable designs. Pareto analysis focuses the set of design alternatives to the most optimal solutions, while stochastic methods like genetic algorithms introduce randomness to expand the space. As the environmental impact of buildings becomes more apparent, new architectural design approaches are arising. In this thesis, it is proposed to combine cellular structures with a multi-objective genetic algorithm for using its search ability to find Pareto-optimal sustainable site planning solutions for social housing complexes. This approach would introduce an effective computational design tool for early design stage of sustainable design, which does not currently achieved by current technologies.

1. GİRİŞ

Sayısal teknolojinin gelişimiyle birlikte, hesaplamalı tasarım (computational design) kavramı 1970’li yılların ortalarında yaygın şekilde kullanılmaya başlansa da, mimaride kullanımı daha da önceye dayanmaktadır. 1960’larda mimarlıkta bilimsel yaklaşımlara eğilim gözlenmiş ve tasarım araştırmalarına odaklanarak, tasarım yöntemlerinin geliştirilip tartışıldığı bir ortam oluşmuştur. Bilgisayarın tasarım araştırmalarında sayısal hesaplayıcı olarak mimarlar tarafından kullanılmaya başlanması da bu dönemde mimarlığın gündemine girmektedir (Mitchell,1977). Alexander’ın (1964) örüntü kavramını ortaya koymasıyla son ürünün tasarımına yönelim yerini tasarım sürecine ve süreçte kullanılan yöntemlere bırakmıştır. Sürdürülebilir tasarım kavramı da o yıllarda, Alexander ve Chermayeff’in (1963) insanla doğanın sağlıklı ilişkisi için gerekli koşulların çağdaş teknolojilerle bozulmaya başladığını belirtmeleriyle kendini göstermiştir. Alexander ve Chermayeff’in (1963) izlenimlerine göre yeni teknoloji insan eli değmemiş gibi kesin, katı ürünler vermektedir ve bütün bu yeni güçler insan yaşantısında mahremiyeti yok etmektedir. Alexander ve Chermayeff’e (1963) göre hem mahremiyete saygılı, hem de toplumsal yaşantının kolaylıklarını sağlayan bir insan-çevre ilişkisi kurulması gerekmektedir. Alexander (1964) değişen çevre şartlarından dolayı tasarıma yeni bir bakış açısı ve bilimsel bir temel gerektiğini belirtmiştir. Bu temelin de matematik üzerine kurulması gerekmektedir. Alexander (1964), düşünce aktarımının içgüdüsel olması nedeniyle açıkça kontrol edilemediğini savunmuştur. Bu nedenle düşüncedeki “çevrenin” matematiğin açık ve kontrol edilebilir düzenine konulmasını, biçime dönüşümün bu düzen içinde olmasını önermiştir. Bu doğrultuda, doğadaki biçimlerin ve oluşum süreçlerinin bilgisayar temelli kurgular içinde yeniden değerlendirilmesi sistemlerin anlaşılmasında ve yapılandırılmasında mimarlığa önemli bir araştırma alanı sunmaktadır (Dritsas, 2012). Tasarım sürecinde, doğal ve yapay süreçlerin araştırılmasıyla, bilgisayar temelli araçlarla hesaplamalı kuramların kullanılması, mimarlığın biyoloji, genetik, bilişim teknolojileri, matematik ve daha birçok disiplinle bilgi alışverişini güçlendirmiştir

(Erdoğan ve Sorgu, 2011). Evrimsel hesaplama yaklaşımı, bu bilgi alışverişinin sonucunda ortaya çıkan konulardan biridir. Evrimsel hesaplamalar formülize edilişlerine göre değişik isimlerle anılırlar. Bunlardan biri olan genetik algoritmalar eniyileme ve öğrenme gibi problem alanlarında biyolojik popülasyon genetiği kurallarına dayanarak çalışmaktadır. Hesaplamalı tasarımının yanında günümüz mimarlığını şekillendiren önemli bir diğer konu da ilk adımlarını Alexander ve Chermayeff'in (1963) attığı sürdürülebilir tasarım olmuştur. Sürdürülebilir tasarım, çevreye duyarlı olarak mevcut ve gelecekteki insani ihtiyaçların karşılanması için, tasarımın stratejik kullanımını içermektedir. Sürdürülebilir bina olarak adlandırılan doğayla uyumlu yapılar, yapının arazi seçiminden başlayarak değerlendirilen, iklim verilerine ve çevre koşullarına uygun, yenilenebilir enerji kaynaklarını kullanmayı amaçlayan ve atık üretmeyen yapılar olarak karşımıza çıkmaktadır. Sürdürülebilir bina tasarımı geleneksel bina tasarımına göre daha karmaşık bir süreçtir. Bunun nedeni birlikte alınması gereken fakat birbiriyle çelişen pek çok tasarım kararının fayda zarar ilişkisinin göz önünde bulundurularak değerlendirilmesi gerekliliğidir. Günümüzde, sürdürülebilir mimarlığı teşvik etmek amacıyla tasarım kararlarını yönlendirmeyi hedefleyen, her tür proje ve binaya uygulanabilen yeşil bina sertifika sistemleri geliştirilmiştir. Fakat yeşil bina tasarım kriterleri olarak sadece sertifika sistemlerini ele almak yerine, tasarım arazisinin bulunduğu bölgenin yerel yapı yönetmeliklerini ve yerel iklimini de göz önünde bulundurarak sürdürülebilir mimari tasarımının gerçekleştirilmesi daha doğru olacaktır.

Bu tez kapsamında, pareto analizinin evrimsel algoritmalarla uyarlanmasıyla elde edilen pareto genetik algoritmayla geliştirilen, LEED ve BREEAM yeşil bina sertifika sistemleri yanında yerel yapı yönetmeliklerini ve iklimsel verileri uygunluk fonksiyonu olarak kabul eden, bu kriterlere göre toplu konut yerleşim alternatifleri üreten "Sürdürülebilir Arazi Yerleşme Planlama Modeli" (Sustainable Site Planning Model-SSPM) olarak adlandırılan evrimsel bir model önerilmiştir.

1.1 Problemin Belirlenmesi ve Tanımı

Günümüzde donanım ve yazılım teknolojisinin gelişimiyle birlikte bilgisayar, mimari tasarım sürecinde aktif biçimde rol almaya başlayarak, mimari tasarımı yaratıcı ve hızlı problem çözümleriyle desteklemektedir. Disiplinler arası çalışmaların sonucu olarak evrimsel hesaplama (evolutionary computation), mimari tasarım alanında

1980'li yılların başından itibaren yer almaya başlamıştır (Bentley,1999). Yıllardır tasarım endüstrisinde, bilgisayar tabanlı araçlar tasarım sürecinin önemli bir parçasını oluşturmalarına rağmen; erken tasarım evresi bilgisayar desteğinin en az kullanıldığı evre olmuştur (Rivard, 2006). Oysa bu evre tasarım sürecinin ilerleyen evrelerine göre daha karmaşıktır. Tasarımcı çok sayıda tasarım ilişkisini bu evrede değerlendirmek zorundadır. Bundan kurtulmak için, Alexander (1964) düşüncedeki "çevrenin" matematiğin açık ve kontrol edilebilir düzenine konmasını önermiştir. Böylece, aktarma işlemi hem içgüdüsellikten kurtulacak, hem de istendiği zaman kontrol edilebilecektir.

Özellikle sürdürülebilir bina tasarımında, kararların çoğu zaten oldukça karmaşık ve bulanık olan erken tasarım evresinde alınmalıdır. Sürdürülebilir tasarımda enerji korunumunun ve doğal ışıktan faydalanmanın maksimum olması, manzaraya yönelerek tüm bu şartların sağlanmaya çalışılması birbiriyle çelişen amaçlara örnek olarak verilebilir.

Bu bağlamda, tez kapsamında üzerinde durulacak sürdürülebilir tasarımda pareto genetik algoritmanın tercih edilmesinin nedeni daha iyi anlaşılacaktır. Pareto analizi değişik sayıdaki önemli etkenleri, daha az önemde olan etkenlerden ayırmak için kullanılan bir tekniktir. Pareto-optimal kavramının evrimsel algoritmalara uyarlanmasıyla, çok amaçlı evrimsel eniyileme algoritmaları geliştirilmiştir. Buna göre, popülasyondaki bir çözüm amaç değerlerine göre, en iyi, en kötü ve diğer çözümlerden farksız olabilir. En iyi çözüm, amaçların herhangi biri içinde en kötü olmayan ve en azından bir amaç için de diğerlerinden daha iyi olan çözüm anlamındadır. Sürdürülebilir tasarım sürecinde, sezgisel ve uygulama tecrübesine dayanan geleneksel tasarım yöntem ve teknikleri yanında; bilgisayar tabanlı araçlarla problem alanına daha geniş açıdan bakılarak tatminkâr çözüm alternatifleri elde edilebilir. Bu nedenle sürdürülebilir tasarım sürecinin kendine özgü yapısı ve değişkenliği, bilgisayar tabanlı tasarım araçlarının bu yapı ve değişkenliğe göre kurgulanmasını gerektirmektedir. Tasarımın ilk evresinde sayısal ortamda planlanmış ve tasarlanmış bir çevre, tasarımcının yaratıcılığının doğru yönde gelişmesini sağlar. Bu sayede üretilen oldukça fazla sayıda alternatif objektif olarak değerlendirilebilir.

Tasarımcıların tasarımlarını çevresel verileri kullanarak çeşitli benzetimlerle test etmelerini sağlayan pek çok program geliştirilmiştir; bina enerji benzetim programları, bina kabuk analizi programları, maliyet hesabı programları, iç hava

kalitesi analiz programları, hesaplamalı akışkanlar dinamiği programları ve son olarak geliştirilen enerji ve gaz emisyonlarını hesaplayan yaşam döngüsü analizleri (LCA) örnek olarak verilebilir (Baouendi, Rivard, Zmeureanu, 2004). Bu farklı yapıdaki araçları birlikte kullanmak oldukça zordur; çünkü bütünleşik bir sistem içerisinde çalışmamaktadırlar ve birbirleriyle çelişen sonuçlar üretebilmektedirler. Ayrıca çoğu benzetim aracı, daha detaylandırılmış tasarım aşamalarında kullanılabilir; oysa sürdürülebilir bina tasarımında en önemli kararlar tasarımın ilk evresi olan kavramsal aşamada alınmaktadır ve değişiklikler ileri aşamalarda oldukça zor olmaktadır. Diğer yandan, günümüzde kullanılan sürdürülebilir tasarımda yardımcı araçların veri girişleri çok zaman almaktadır ve öğrenilip kullanılması oldukça karmaşıktır (Rivard, 2006). Bu nedenle mimari tasarım sürecinde mimarlar dışında konuda uzman mühendislik ekibi ile değerlendirmeyi gerektiren sonuçlar üretmektedirler.

Günümüzde sürdürülebilir binaların yaygın olarak kullanılmasında yeşil bina değerlendirme sistemleri ve sertifika programları etkin rol oynamaktadır. Belirlenen kriterler doğrultusunda puanlama esasına göre çalışan bu sistemler pek çok ülke tarafından yeşil bina tasarımında yol gösterici olarak kullanılmaktadırlar. Bu değerlendirme sistemleri arasında en gelişmiş kabul edilenleri, 1990 yılından itibaren kullanılmakta olan ve şu ana dek toplam 116.000 binayı değerlendirmek için kullanılmış, kayıtlı olarak bekleyen yarım milyon binada da kullanılması planlanan İngiltere kökenli BREEAM (Building Research Establishment Environmental Assessment Method) ile 1998 yılında kullanılmaya başlanan (Url-1), Amerika kökenli ve şu ana dek toplam 52.490 binayı değerlendirmek için kullanılmış LEED (Leadership in Energy and Environmental Design)'dir (Url-2). Başlangıçta amaç her ülkenin kendi yerel standartlarını, inşaat yöntemlerini, iklimsel verilerini ve kültürlerini gözeterik kendine özgü bir sistem oluşturmasydı. Ancak son yıllarda LEED ve BREEAM yaptıkları ataklarla uluslararası bir kimlik kazandılar ve Türkiye gibi kendine özgü yeşil bina değerlendirme sistemleri bulunmayan ülkelerde kabul görmeye başladılar. Bu çerçevede, 2007 yılında kurulan ÇEDBİK tarafından, Türkiye'de ulusal koşullara uygun bir değerlendirme sistemi oluşturma çalışmalarına başlanmıştır (Url-3). Sektördeki açığın bir an önce kapatılabilmesi için, mevcuttaki uluslararası sertifika sistemlerinden Türkiye'ye en yakın olacağı düşünülen BREEAM'in uyarlanması ile ilgili çalışmalara başlanabileceği kararlaştırılmıştır.

Bununla ilgili olarak da, öncelikli dil sorununu ortadan kaldırmak üzere, BREEAM Avrupa Ticari Binalar Kullanım Kılavuzu, ÇEDBİK Sertifika Komitesi Üyelerinin desteğiyle Türkçeye çevrilmiştir. Gerek uyarlama, gerekse ulusal bir yeşil bina performans değerlendirme sisteminin oluşturulması konusunda çalışmalar devam etmekte olup, henüz neticelendirilmiş bir sistem bulunmamaktadır (Yılmaz, 2012). Bu nedenle tez kapsamında en gelişmiş iki sertifika sistemi olarak kabul edilen ve yaygın olarak kullanılan LEED ve BREEAM sistemleri ele alınacaktır. Fakat BREEAM İngiliz yasalarına ve standartlarına yakınlığı nedeniyle global pazarda LEED'in gerisinde kalmaktadır Bu nedenle çok uluslu şirketler LEED sertifikasını tercih etmektedirler. BREEAM Avrupa standartlarını ve kriterlerini kullanması ve Türkiye'ye daha yakın olması nedeniyle tez kapsamında LEED ile birlikte ele alınacaktır. Bu sertifika sistemleri her ne kadar tüm dünyada kullanılmakta olsalar da, Amerika ve İngiltere'nin coğrafi, ekonomik ve kültürel koşulları ile o ülkelere özgü şartnameler doğrultusunda hazırlandıkları için diğer ülkelerde kullanılmasında zorluklar yaşanmaktadır.

Sonuç olarak, yukarıda verilen bilgiler ışığında sürdürülebilir bina tasarım problemi üç şekilde karşımıza çıkmaktadır:

- Sürdürülebilir bina tasarımı geleneksel bina tasarımına göre daha karmaşık bir sürece sahiptir. Birbiriyle çelişen pek çok tasarım kararı erken tasarım evresinde bir arada ele alınmalıdır, fakat günümüzde erken tasarım evresi bilgisayar desteğinin en az kullanıldığı evredir. Sürdürülebilir bina tasarım problemini çözmeye yardımcı bir karar destek sistemine gereksinim vardır.
- Günümüzde sürdürülebilir bina tasarımına yardımcı olmayı hedefleyen benzetim modelleri daha detaylandırılmış ileri tasarım aşamalarında kullanılmaktadır; oysa sürdürülebilir bina tasarımında önemli kararların çoğu, tasarımın ilk evresi olan kavramsal aşamada alınmaktadır. Ayrıca benzetim modellerinin veri girişleri çok zaman almaktadır, karmaşık arayüzlerinin öğrenilmesi ve kullanılması da oldukça zordur.
- Sürdürülebilir mimarlığı teşvik etmek amacıyla pek çok ülke tarafından geliştirilen sertifika sistemleri mevcuttur. Türkiye'de henüz bir sertifika sistemi oluşturulmamıştır. Var olan sistemler coğrafi, ekonomik ve kültürel özellikleri ile hazırlandıkları ülkelere özgü oldukları için zorluklar yaşanmaktadır.

LEED ve BREEAM arazi yerleşim kriterlerine, buna ek olarak yerel yönetmelik ve iklim şartları gibi birbiriyle çelişebilen pek çok ölçüte göre değerlendirme yapılmasını gerektiren sürdürülebilir yerleşme tasarımı problemini çözmeye yardımcı bir karar destek sistemi, çelişen ölçütlere uygun çözüm önerisi geliştirmeye yönelik bir yaklaşım olan pareto genetik algoritma ile geliştirilebilir.

1.2 Tezin Amacı

Günümüzde donanım ve yazılım teknolojisinin Tezin amacı; öncelikle Türkiye için, yeşil bina tasarım kriterleri olarak sertifika sistemlerinin yanında, tasarımın yapılacağı arazinin bulunduğu bölgenin yerel yapı yönetmeliklerine ve iklimine göre değişen kriterler de göz önünde bulundurularak, sürdürülebilir arazi yerleşim planı üreten, genetik algoritmaya dayalı bir modelin oluşturulmasıdır.

Tasarımın ilk aşamasında eldeki tasarım problemine uygun tüm sınırlandırmaları göz önünde bulundurarak toplu konut yerleşim önerileri sunan evrimsel modelin tanımlanması ve modelin alan çalışması yapılarak değerlendirilmesi çalışmanın konusunu oluşturacaktır. Bu noktada toplu konut kavramına değinmek gerekirse; belli ölçülere göre bir komşuluk birimi ya da böyle bir birimin her gereksinimini kapsayabilecek büyüklükte bir alan üzerine yerleşen, insanların barınma ihtiyaçlarını karşılayan binaların tümü ve bunlarla ilgili sosyal tesislerdir. Bu bağlamda toplu konut şehir içinde veya şehre yakın olan, konutlar arası ilişkilerin kurulduğu, alt yapı problemlerinin çözüldüğü, insanların ihtiyaçlarını karşılayabilecekleri, iletişim kurabilecekleri sosyal tesis ve yeşil alanlara sahip, gelecekteki büyümenin de göz önüne alınarak planlandığı konut toplulukları olarak tanımlanabilirler. İçerisinde farklı fonksiyonlara sahip birimleri bulundurması, yaya-taşıt yolu, konut ve yeşil alanlarıyla sürdürülebilir tasarım kriterlerinin çoğunun ele alınarak arazi yerleşimine olanak sağlaması açısından, toplu konut yerleşimi modelde tercih edilmiştir.

Bu konuda yapılmış diğer modellerden farklı olarak, LEED ve BREAM arazi yerleşim kriterlerinin yanında yerel yönetmelikleri ve yerel iklim şartlarını da göz önünde bulunduran modelin, üretilen alternatifleri uygunluk puanlarına göre birbirleriyle kıyaslayarak pareto optimal sonuçlar üreterek, tasarımcıyı sürdürülebilir arazi kullanım çözümlerine götürmesi hedeflenmiştir. Arazinin topografik yapısını ve doğal oluşumları göz ardı etmeden, çözümün gerçekçi sonuçlar verebilmesi için, arazi verilerinin de gerçeğe yakın modellenmesi gerekmektedir. Arazinin

tanımlanmasında Excel programı, veri girişinin pratik ve kolay tanımlanabilir olması ve matrisle tanım tekniğine olanak sağlaması nedeniyle tercih edilmiştir. Matrisin satırlarını ve sütunlarını oluşturan her bir elemanı (hücre), arazinin alanı 1 m² olan bir birimini temsil etmektedir. Arazi köşe noktalarının ve arazi üzerindeki mevcut elemanların x,y koordinatlarına göre matris oluşturulmaktadır. Böylece sadece dörtgen araziler değil, her farklı şekildeki araziler tüm girinti ve çıkıntılarıyla modele tanıtılabilmektedir. Arazinin topografik yapısına uygun çözümler üretebilmek için sayısal araziye eldeki eğim değerlerine göre 3 boyutlu karakter kazandırılmıştır. Yükselti farklarının çözümünde, araziden gelen eğim değerleri kullanıcı isteğine göre farklı kotlardaki parçalara bölünerek modele tanıtılmaktadır. Bu sayede yüksek kot farkına sahip kademeli arazilerde teraslama yapılarak, topografya ile uyumlu arazi yerleşim çözümlerine gidilebilecektir.

Sürdürülebilir tasarım, topografyaya uyumlu olmasının yanında, bulunduğu çevre ile de bir bütün olmalıdır. Yapının arazi üzerine doğru şekilde konumlandırılması ve yönlendirilmesi arazi özelliklerinin incelenip analiz edilmesiyle mümkündür. Arazi içerisinde ekolojik değere bağlı değerlendirmeler yapıldıktan sonra bina ya da binaların yerleri ve birbirlerine göre konumları belirlenirken, binaların yapılacağı bölgedeki iklim tipi ve binaların çevrelerinde oluşturacakları gölge, rüzgar ve gürültü etkilerinin yanı sıra imar yönetmeliğinden gelen kurallar da dikkate alınmalıdır. SSPM modeli bu amaç doğrultusunda, tasarım arazisinin iklimine göre rüzgâr ve manzaraya yönelen, topografya ile uyumlu ve yerel imar planlarını da göz önünde bulunduran konut yerleşim alternatifleri üreterek sürdürülebilir toplu konut yerleşmesi tasarımı yapan bir model olarak kurgulanmıştır.

Bu tez, karmaşık problemlerin çözümünde tercih edilen evrimsel algoritmaların, sürdürülebilir arazi yerleşiminde kullanılmasıyla, hesaplamalı tasarım ve sürdürülebilir mimarlık kavramlarının arakesitinde yer alacaktır. Sürdürülebilir Arazi Yerleşme Planlama Modeli (SSPM), her iki disiplinin olanaklarını bir arada kullanarak, erken tasarım evresinde toplu konutların araziye yerleşiminde mimarı yönlendirebilecek bir karar destek sistemi olarak tasarım sürecinde önemli rol oynayacaktır.

1.3 Tezde İzlenen Yöntem

Araştırmada izlenen yöntem üç adımda açıklanabilir:

Analiz Çalışması

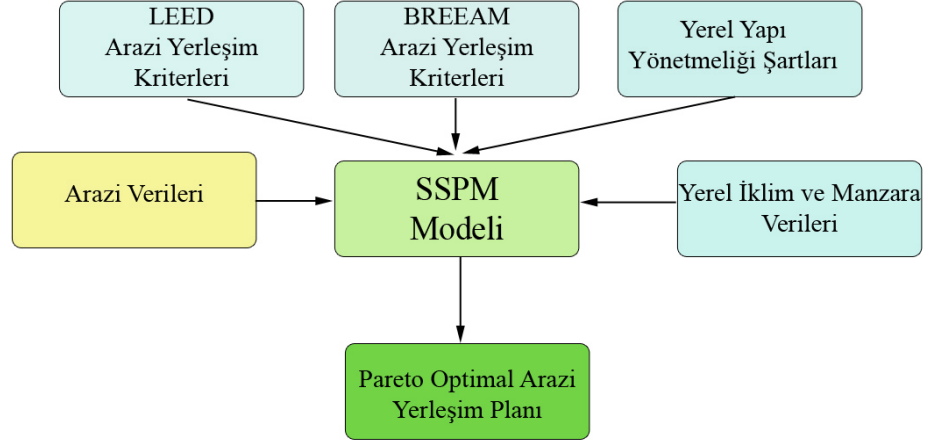
İlk olarak problemin çözümüne yönelik olarak literatürde ve uygulamada mevcut sistem, model ve yaklaşımlar incelenmiştir. Tezin ilk bölümünde kullanılacak olan evrimsel algoritmalar ve bir alt başlığı olan pareto genetik algoritmalar ile ilgili literatür çalışmasına yer verilmiştir. Genetik algoritma terminolojisi ele alınarak, çok amaçlı genetik algoritmaların çalışma prensipleri, amaçları ve gerçekleştirilen en başarılı algoritmaların çalışma prensipleri incelenmiştir. Tasarımda sürdürülebilirlik kavramı incelenmiş ve günümüzde kullanılan iki önemli bina sertifika sistemi olan LEED ve BREEAM sistemleri arazi yerleşim kriterleri göz önünde bulundurularak incelenmiştir. Tez kapsamında uygunluk fonksiyonu olarak kullanılacak diğer konular olan imar yönetmeliği ve fiziksel çevre kontrolü kriterleri açıklanmıştır. Sürdürülebilir mimarlık alanında literatür çalışması yapılarak bu konuda bilgisayar destekli olarak gerçekleştirilmiş modeller üzerinde durulmuştur. Bu modeller iki ayrı başlık olan benzetim modelleri ve evrimsel modeller olarak sınıflandırılarak, yaygın olarak kullanılan benzetim modellerinin analizi yapılmıştır. Birbirleriyle kıyaslanarak avantaj ve dezavantajları açıklanmıştır. Evrimsel hesaplama yöntemi ile daha önceden geliştirilen ve yöntem olarak bu çalışmaya yakın duran iki örnek model ayrıntılı olarak incelenmiştir.

Tez kapsamında geliştirilen SSPM modelinde araç olarak pareto temelli bastırılmamış sınıflandırılmalı genetik algoritma (NSGA-II) kullanılarak; pek çok tasarım kriterini göz önünde bulundurması gereken tasarımcıya, tasarımın ilk aşamasındaki karar verme sürecinde yardımcı olacak bir model geliştirilmesi amaçlanmıştır. Genel olarak NSGA-II'nin, tüm test problemlerinde diğer algoritmalarla göre pareto yüzeyi bulma başarısının daha iyi olduğu söylenebilir (Sağ ve Cunkaş, 2009) . NSGA-II algoritmasının incelenen algoritmalar arasında en başarılı sonuçları ürettiği düşünülerek modelde araç olarak tercih edilmiştir.

Modelin Oluşturulması

Sürdürülebilir tasarım alanında tanımlanan problemler ve yapılan çalışmalar ışığında modelin uygunluk fonksiyonları oluşturulmuştur. “Sürdürülebilir Arazi Yerleşme Planlama Modeli” (Sustainable Site Planning Model-SSPM) olarak adlandırılan

modelin uygunluk fonksiyonları, LEED ve BREEAM yeşil bina sertifika sistemlerinin arazi yerleşim kriterleri, iklim koşullarına göre belirlenen yerleşim kriterleri ve yerel yapı yönetmeliği kuralları olarak belirlenmiştir. Şekil 1.1'de SSPM modelinin kavramsal şeması gösterilmiştir.



Şekil 1.1 : SSPM Modeli Kavramsal Şeması.

Kullanılacak genetik algoritmanın uygunluk fonksiyonları olarak kabul edilen değerlendirme kriterleri, arazi kullanımı, bina yerleşimi ve yönlenmesi ve bina formu olmak üzere üç ana başlık altında sınıflandırılarak tanıtılmıştır. Modelin değişkenleri tanımlanmış ve uygunluk fonksiyonları bağlamında değerlendirme sistemini oluşturan formüller kurgulanmıştır. Formüllerin oluşturulmasında sertifika sistemlerinden ve yerel yapı yönetmeliklerinden gelen oranlar ve sınırlandırmalar kullanılmıştır. Bunlara ek olarak rüzgâr ve manzara yönünün sayısal olarak modele tanıtılıp üretilen alternatiflerin puanlanması için konutları oluşturan 1m2'lik hücrelerin birbirleriyle ve rüzgâr-manzara yönleriyle ilişkileri formülize edilmiştir.

Modelin Uygulanması

Modelin sınanması için Kağıthane'de yer alan 4500 m2' büyüklüğünde bir arazi seçilmiştir. Hipotetik bir arazi yerine, daha önceden bir bölümünün kullanılmış olması, yapısında korunacak ağaç ve su öğelerini içermesi sebebiyle bu arazi tercih edilmiştir. Arazi İstanbul ilinde olması sebebiyle uygunluk fonksiyonlarının bir kısmını İstanbul İli İmar Yönetmeliği kuralları oluşturmuştur. Araziye ait TAKS, KAKS, hmax, manzara, rüzgar ve iklim verilerine göre üretilmiş ilk populasyon, sonrasında NSGA-II algortiması döngüsü içerisine sokularak pareto optimal arazi yerleşim bireyleri elde edilmiştir. Her nesilde üretilen birey sayısı 100 ile

sınırlandırılmıştır; daha fazla birey üretimi, kullanılan bilgisayarın özellikleri nedeniyle, modelin çalışma hızını oldukça düşürmektedir. Yerleşim, manzara ve rüzgâr açıları kuzey-güney-doğu ve batı olmak üzere doğrusaldır, açılı yerleşimler modelde üretim dışında tutulmuştur. Arazinin topoğrafyası 1 metrelik kot farklarıyla teraslanarak modele tanıtılmıştır. Farklı nesillerde oluşan pareto eğrileri kıyaslanarak modelin üretimleri değerlendirilmiştir.



2. EVRİMSEL ALGORİTMALAR

Eniyileme (optimizasyon), yapılmış veya yapılmakta olan işin en iyi çözümünü ortaya koymak anlamındadır. Eniyileme, bir problemin çözümünde matematiksel modeller geliştirerek, yönetsel karar üretmeye yardım eden disiplindir (Deb, 2001). Eniyileme problemlerinin çözümünde öncelikle doğrusal modellemeler kullanılmış, ancak bu yöntemin her problemin çözümü için yeterli olmadığı görülmüştür. Son zamanlarda ise eniyileme problemlerinin çözümünde evrimsel algoritmalar yaygın olarak kullanılmaya başlanmıştır. Evrim kuramı bilişim dünyasında oldukça yeni yaklaşımların doğmasını sağlamıştır. Bu kuram göz önünde bulundurularak biçimlenen evrimsel bilgi işleme yöntemi, geleneksel yöntemlerin çözmekte zorlandığı karmaşık problemlerin çözümünde kullanılmaktadır.

Evrimsel Algoritmalar (EA), doğadaki evrimsel süreçleri model olarak kullanan bilgisayara dayalı problem çözme teknikleridir. Geleneksel programlama teknikleriyle çözülmesi güç olan karmaşık problemler özellikle sınıflandırma ve çok boyutlu eniyileme problemleri, evrimsel algoritmalar yardımıyla daha kolay ve hızlı olarak çözülebilmektedir (Mitchell,1996).

Evrimsel hesaplama, doğal seçimden yola çıkarak geliştirilmiş bir arama ve eniyileme algoritmasıdır. Doğal seçim işlemi, uygunluk değeri en yüksek, diğer bir deyişle çevreye adaptasyonu en yüksek olan bireyin seçilmesini sağlar. Evrimsel algoritma, verilen bir probleme en uygun çözüm için olasılıklara göre bir arama gerçekleştirir. Evrimsel hesaplama algoritmalarının amacı, hayatta kalacak ebeveynlerin daha iyi özelliklere sahip bireyler meydana getirmelerini sağlamaktır (Bentley,1999).

Evrimsel algoritmalar formülize edilişlerine göre; genetik algoritmalar (GA), genetik programlama (GP), evrimsel programlama (EP) ve evrimsel strateji (ES) gibi alt başlıklara ayrılmıştır (Bentley, 1999). Tez kapsamında genetik algoritmalar başlığı üzerinde durulacaktır.

2.1 Genetik Algoritmalar

Tez kapsamında incelenecek olan genetik algoritmaların tanımı için doğada var olan seçilimin anlaşılması gerekir. Dünyadaki doğal seçimde amaç fonksiyon yaşam mücadelesini temsil eder ve insanoğlu bunu en yüksek seviyeye getirmek ister. Doğal yaşam şartlarına uyum sağlayabilme derecesi dünyada uzun süre var olabilmenin göstergesidir. Genetik algoritmalar biyolojik süreci modelleyerek fonksiyonları optimize eden evrim algoritmalarıdır. Michigan Üniversitesinde psikoloji ve bilgisayar bilimi uzmanı olan John Holland 1960 yılında bu konuda ilk çalışmaları yapan kişidir. Makine öğrenmesi (machine learning) konusunda çalışan Holland, Darwin'in evrim kuramından etkilenerek canlılarda yaşanan genetik süreci bilgisayar ortamında gerçekleştirmeyi düşünmüştür. Genetik Algoritma (GA) böylece John Holland tarafından bu düşüncenin bir sonucu olarak bulunmuştur (Holland,1975).

GA parametreleri, biyolojideki genleri temsil ederken, parametrelerin toplu kümesi de kromozomu oluşturmaktadır. GA'ların her bir elemanı, bir kromozomla temsil edilir. Bu aday çözümler kümesi de popülasyon olarak adlandırılır. Popülasyonun uygunluğu, belirli kurallar dâhilinde maksimize veya minimize edilir. Her yeni nesil, rastgele bilgi değişimi ile oluşturulan diziler içinde hayatta kalanların birleştirilmesi ile elde edilmektedir (Angeline, 1995).

Genetik algoritma aşamaları (Goldberg, 1989);

a. Başlangıç: N adet kromozom içeren popülasyonun oluşturulması,

b. Uyumluluk: Her x kromozomu için uyumluluğun $f(x)$ değerlendirilmesi:

Yüksek uygunluğa sahip bireyler yeniden üretilirken, düşük uygunluklu bireyler popülasyondan çıkarılır. Burada amaç, arama uzayının özel parçaları üzerinde aramaya odaklanmak ve popülasyonun ortalama uygunluk değerini yükseltmektir. Eniyileme işlemine göre bir bireyin uygunluğu, sayısal bir değer ile temsil edilir.

c. Yeni popülasyon: Yeni popülasyon oluşuncaya kadar aşağıdaki adımların tekrar edilmesi:

1. Seçim: İki ebeveyn kromozomun uyumluluğuna göre seçimi (daha iyi uyum seçilme şansını artırır),

2. Çaprazlama: Yeni bir birey oluşturmak için ebeveynlerin bir çaprazlama olasılığına göre çaprazlanmasıdır. Bu tip çaprazlamada, öncelikle kromozomun ilk biti ile son biti arasında çaprazlama noktası rastgele seçilir. Birinci kromozomun çaprazlama noktasının solundaki bitlerin hepsi ikinci kromozoma geçerken, benzer tarzda ikinci kromozomun çaprazlama noktasının solundaki bitler de birinci kromozoma geçerler. Meydana gelen yeni kromozomlar, her iki kromozomun karakterlerini içermektedirler. Eğer çaprazlama yapılmazsa yeni birey anne veya babanın kopyası olacaktır.

3. Mutasyon: Yeni bireyin mutasyon olasılığına göre kromozom içindeki konumu (lokus) değiştirilir. Kromozomdaki bitlerin küçük bir yüzdesini değiştirir. Mutasyon ile kromozomdaki bitler “1” ise “0”, “0” ise “1” yapılır. Bir nüfus matrisinde, mutasyon noktaları rastgele seçilir. Mutasyonun % değeri artırılırsa, algoritma daha geniş parametre uzayını tarar.

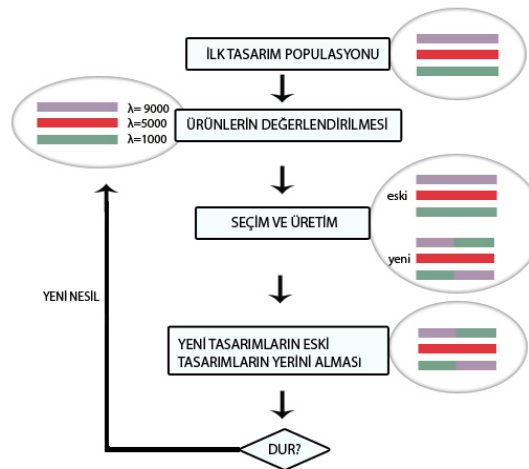
4. Ekleme: Yeni bireyin yeni popülasyona eklenmesi,

5. Değiştirme: Algoritmanın yeniden çalıştırılmasında oluşan yeni popülasyonun kullanılması,

6. Test: Eğer sonuç tatmin ediyorsa algoritmanın sona erdirilmesi ve son popülasyonun çözüm olarak sunulması,

7. Döngü: 2. adıma geri dönülmesi,

şeklinde. Tüm bu işlemler sonsuz döngü içerisinde yapılır. Eğer bir durdurma kriteri belirlenmez ise bu süreç sonsuza kadar devam eder.(Şekil 2.1)



Şekil 2.1 : Genetik Algoritma Akış Diyagramı (Url-4).

Sağ ve Cunkaş'ın (2009) çalışmasında gösterdiği gibi temel algoritma aşağıdaki şekilde ilerlemektedir.

Girdi : N: Popülasyon boyutu

T: nesil sayısı

Pc: çaprazlama olasılığı

Pm: mutasyon olasılığı

Çıktı: A: Bastırılmamış küme

Adım 1- Başla: $P_0 = \emptyset$ ve $t = 0$ For $i=1, \dots, N$ do

a) Bazı olasılık dağılımlarına göre bir $i \in I$ seç.

b) $P_0 = P_0 + i$

Adım 2- Uygunluk ataması: Her bir $i \in P_t$ bireyi için hem kodlanan karar vektörü $x=m(i)$ hem de $y = f(x)$ amaç vektörünü tanımla ve sayısal uygunluk değeri F yi hesapla,

Adım 3- Seçim: $P' = \emptyset$. For $i=1, \dots, N$ do

a) Verilen tasarıma ve uygunluk değeri F' 'ye bağlı olarak bir $i \in P_t$ bireyi seç.

b) $P' = P' + i$

Geçici popülasyon P' eşleşme havuzu olarak adlandırılır.

Adım 4- Çaprazlama: $P'' = \emptyset$. For $i=1, \dots, N/2$ do

a) $i, j \in P'$ olan iki birey seç ve P' den çıkar.

b) i ve j bireylerini çaprazla. Sonuçta oluşan çocuklar $k, l \in I$ dır.

c) k, l yi pc olasılığı ile P'' ye ekle. Aksi halde i, j bireylerini P'' ye ekle

Adım 5- Mutasyon: $P''' = \emptyset$. For her bir $i \in P''$ bireyi için do

a) P_m mutasyon oranı ile i bireyine mutasyon uygulanır. Oluşan birey $j \in I$ dır.

b) $P''' = P''' + i$

Adım 6- Bitir: $P_{t+1} = P'''$ ve $t=t+1$. Eğer $t \geq T$ veya herhangi bir sonlanma kriteri sağlanmışsa $A = p(n(P_1))$, aksi halde Adım 2'ye git.

Görüldüğü üzere genetik algoritmanın yapısı oldukça geneldir ve herhangi bir probleme uygulanabilir. Kromozomların tanımlanması genellikle ikili düzendeki sayılarla yapılır. Çaprazlama işlemi için kullanılan bireyler iyi bireylerden seçilir.

Genetik algoritmalar otomatik programlama, öğrenme kabiliyetli makineler, ekonomi, çevrebilim, planlama, üretim hattı yerleşimi gibi alanlarda kullanılmaktadır. Bu problemlerin hemen hemen hepsi çok geniş bir çözüm havzasının taranmasını gerektirmektedir. Bu çözüm havzasının geleneksel yöntemlerle taranması çok uzun sürmekte, genetik algoritmayla ise kısa bir sürede kabul edilebilir bir sonuç alınabilmektedir.

Tez kapsamında hesaplamalı tasarım bağlamında en çok kullanılan evrimsel algoritmalarından biri olan çok amaçlı genetik algoritmalar incelenerek kavramlar açıklanacaktır.

2.1.1 Çok amaçlı genetik algoritmalar

Hesaplamalı mimari tasarımda amaç, bilgisayarın mimari tasarım sürecinde mimari tasarımı alternatif problem çözümleriyle destekleyerek en iyi çözümün bulunmasını sağlamaktır. Genetik algoritmalarda en iyi çözüm(ler), evrimsel süreçlere maruz bırakıldıktan sonra, olası çözüm(ler) kümesinden elde edilir. Eğer eniyilenecek tek bir amaç varsa, problem “tek amaçlı eniyileme problemi”; birden fazla amaç olması durumunda ise “çok amaçlı eniyileme problemi” olarak tanımlanır. (Sağ ve Cunkaş, 2009)

Tek amaçlı eniyileme probleminin (single-objective optimization problem - SOOP) aksine çok amaçlı eniyileme probleminin (multi-objective optimization problem- MOOP) hedefi, aynı anda birden fazla amacın eniyilenmesidir ve genellikle hedeflenen amaçlar birbiriyle çelişkilidir. Örneğin, tasarlanan bir yazılımın amacı daha az maliyet, daha iyi performans ve daha az enerji tüketimi olabilir (Chen, 2001).

Çok amaçlı eniyileme problemlerinde, birbiriyle çelişen amaçlar nedeniyle problemin çözümleri birden fazladır. Bu çözümlere pareto optimal çözümler (etkin çözümler) denir.

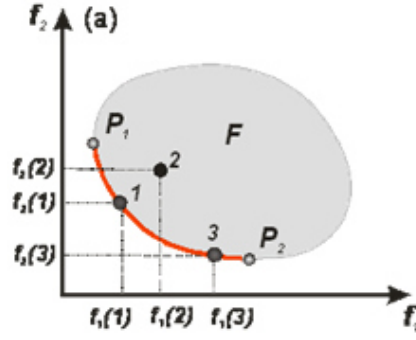
2.1.1.1 Pareto optimal kavramı

Pareto analizi, kalite mühendisleri tarafından en fazla kullanılan istatistiki bir yöntemdir (Squires, 1986). Pareto analizi değişik sayıdaki önemli sebepleri, daha az önemde olan sebeplerden ayırmak için kullanılan bir tekniktir. Pareto optimal kavramı 1900'lü yılların başlarında, İtalyan bir iktisatçı ve sosyolog olan Vilfredo Pareto (1848 – 1923) tarafından bulunmuş olan ekonomik yeni refah teorisinden gelmektedir. Pareto bir ulusun ekonomisinde olası olan denge durumlarını araştırırken, adını kendisinden alan pareto optimal kavramını bulmuştur. Bu teknik, bir olayın grafik yardımıyla gösterilmesiyle, karşılaşılan problemin en önemli sebebi üzerinde dikkati yoğunlaştırarak önceliklerin belirlenmesine yardımcı olduğundan, ekonominin dışında da her alanda kullanılabilir niteliktedir (Ishikawa, 1982).

Pareto-optimal kavramının evrimsel algoritmalara uyarlanmasıyla, çok amaçlı evrimsel eniyileme algoritmaları geliştirilmiştir. En iyi çözüm, amaçların herhangi biri içinde en kötü olmayan ve en azından bir amaç içinde diğerlerinden daha iyi olan çözüm anlamındadır.

Gerçek dünya problemleri genellikle birbiriyle çelişen birden fazla amacın eş zamanlı eniyilemesinden oluşur. Mimari tasarım problemi de hasta-tanımlı bir problem olup, çelişen amaçlara sahip olabilir; bir diğer deyişle amaçların her birini ayrı ayrı sağlayan tek bir çözümün bulunması her zaman mümkün olmayabilir. Hasta tanımlı problemlerde başlangıç aşamasından hedef aşamasına geçen süreç tanımlı değildir; bu tür problemlerin kesin sonuçları da yoktur. Bu nedenle hasta-tanımlı problemlerin çözümünde pareto analizi kullanılabilir. Pareto analizinde problem bilgisine sahip olan karar vericiden, amaçların her biri için kabul edilebilir düzeyde olan alternatif çözümlerden seçim yapması istenir. Bu çözümlerin her birine Pareto-Optimal Çözüm, bu çözümlerin kümesine de Pareto-Optimal Çözümler Kümesi denir. Pareto-Optimal çözüm, çözüm kümesindeki diğer herhangi bir çözüm tarafından bastırılmamış olan çözümdür (Chen,2001).

Şekil 2.2'de örnek bir pareto grafiği verilmiştir. $f_1(x), \dots, f_M(x)$ değerleri birbiriyle çelişen amaç fonksiyonlarıdır ve amaç her iki değeri de minimize etmektir.



Şekil 2.2 : Örnek amaç pareto grafiği(f_1, f_2) (Lahanas, Baltas, Zamboglou, 2003).

F kümesi her iki amaç tarafından da bastırılmıştır P1 ve P2 noktaları arasındaki çözümler pareto öncüleridir hiçbir fonksiyon tarafından bastırılmamışlardır. 1 ve 3 numaralı çözümler de bastırılmamış pareto optimal çözümlerdir. 2 numaralı çözüm ise pareto optimal olarak kabul edilemez, çünkü 1 numaralı çözüm her iki amaç için de daha küçük değere sahiptir.

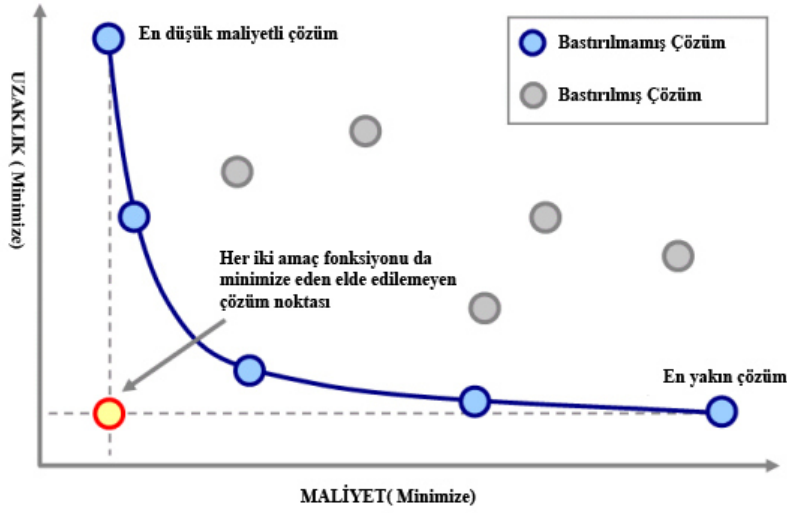
2.1.1.2 Pareto baskınlık ilişkisi, pareto optimal küme ve pareto öncüsü

Üçüncü En iyi takas ya da ferâgat “Pareto Baskınlığı” ile tanımlanır. Problem çözümünde elde iki amaç varsa en iyiler kümesi bir eğri olarak karşımıza çıkar. Üç amaçlı bir amaç uzayında en iyiler kümesi bir yüzeydir. Amaç sayısı "n" olarak tanımlanırsa, Pareto (en iyiler) kümesi n-1 boyutlu olacaktır. (Chen, 2001)

Pareto optimal küme, bütün pareto optimal çözümlerden oluşur. Evrim işlevi altındaki pareto optimal kümenin görüntüsü, pareto optimalin öncüsü, basitçe pareto öncüsüdür. Genellikle pareto optimal öncüsüne uyan gerçek optimal kümeyi bulmak imkansızdır. Pareto öncüsünde (Pareto Frontier) herhangi bir amaç diğer amaçlardan en az birini daha kötü yapmadan iyileştirilemez (Chen, 2001).

Çok amaçlı eniyilemede istenen, amaç değere doğru bir yakınsama ve dağılım gösteren çözümlerin bulunmasıdır. Buna göre bir algoritmanın bulduğu çözümler, diğer algoritmanın bulduğu çözümlerle amaç değere yakınlıklarına göre karşılaştırılır.

Örneğin problem şehir merkezine en yakın en ucuz oteli bulmak olursa, hem uzaklığı hem de maliyeti minimize eden otel alternatiflerini görmek için oluşturulan pareto eğrisi Şekil 2.3'te verilmiştir. Mavi renkle gösterilen çözümler bastırılmamış, yani kabul edilebilir alternatif oteller olarak kabul edilebilir.



Şekil 2.3 : Örnek Pareto Eğrisi(minimum uzaklık-minimum maliyet).

Zitzler (Zitzler, Deb, Thiele, 1999) çok amaçlı evrimsel algoritma için 3 hedef tanımlar:

- Pareto optimal ön yüzüne (pareto front-pareto optimal küme) göre baskın olmayan sonuçlardan kaçınılmalıdır.
- İyi bir dağılım gösteren sonuç istenir.
- Elde edilen baskın olmayan önyüzün (pareto front-pareto optimal küme) dağılımı maksimuma çıkarılmalıdır.

Çok amaçlı problemin matematiksel kurgulamasına örnek olarak Evans'ın(1984) çalışması verilebilir. Çalışmada;

Amaç fonksiyonlarını optimize eden karar değişkenlerinin her biri;

$$S=(x_1, x_2, \dots, x_n)$$

p: amaç sayısı

($p \geq 2$) olmak üzere amaç fonksiyonları; $f_1(x), f_2(x), \dots, f_p(x)$

Bu fonksiyonların her birinin maksimize edilmesini amaçlayan çok amaçlı problem;

$$\text{Max } f(x) = [f_1(x), f_2(x), \dots, f_p(x)]$$

$$x \in S \text{ ve } p \geq 2$$

olarak tanımlanmıştır.

Amaç fonksiyonlarının her birini ve hepsini birlikte maksimize eden çözüm ($x_s \in S$)

“Üstün Çözüm” olarak adlandırılır. Eğer çok amaçlı problem(I)'in çözümü x_s ,

yalnız ve yalnız $x \in S$ ve $f_i(x) \leq f_i(x^*)$ $i=1,2, \dots, p$ ise, her $x \in S$ sağlayan çözüm “Üstün Çözüm” dür. Üstün çözümlerden birisi “İdeal Çözüm” dür (Evans,1984).

Çok amaçlı programlamada, genellikle birbiriyle çelişen amaçlar söz konusudur ve p amaçtan en az ikisi doğası gereği çelişir. Bu nedenle bütün amaçların en iyi şekilde karşılandığı tek bir optimum çözümün bulunması çok nadir bir durumdur bu nedenle etkin çözüm; problemin optimum çözümü olmadığında, her bir amaca mümkün olduğunca yaklaşan ve amaçlardan en az birini sağlayan çözümdür. Problemin uygun çözüm kümesinde birden çok etkin çözüm olabilir. Bu etkin çözümlerden birisi, karar verici tarafından tercih edilir. Problem(I)’in bir etkin çözümü x^* , $x^* \in S$ ‘yi sağlayan uygun çözümdür ve matematiksel olarak;

$$x^* \in S$$

$f_i(x^*) \leq f_i(x)$ $\forall i$ için, $i=1,2, \dots, p$ ve $f_i(x^*) < f_i(x)$ $\forall i$ için, $i=1,2, \dots, p$ şeklinde tanımlanabilir (Evans, 1984).

2.1.2 Çok amaçlı genetik algoritma örnek modelleri

Eniyileme problemlerinin çözümünde öncelikle doğrusal modeller kullanılmıştır, ancak bu modellerin her problemin çözümü için yeterli olmadığı görülmüştür. Bu alanda son yıllarda birçok yeni algoritma tasarlanmıştır, bunlardan biri olan genetik algoritmalar çok amaçlı eniyileme problemlerinin çözümünde yaygın olarak kullanılmaya başlanmıştır (Osyczka, 1985).

Çok amaçlı eniyileme alanında geliştirilmiş genetik algoritmalar başlıcaları Schaffer (1985) tarafından geliştirilen VEGA, Murata (1997) tarafından önerilen MOGA, Horn ve Nafpliotis (1993) tarafından önerilen NPGA, Srinivas ve Deb (1994) tarafından önerilen NSGA, Zitzler v.d. (1999) tarafından önerilen SPEA ve SPEA2, Knowles ve Corne (2000) tarafından önerilen PESA ve son olarak Deb (2001) tarafından geliştirilen NSGAIİ algoritmalarıdır (Sağ ve Cunkaş, 2009).

Sağ, geliştirdiği Çok Amaçlı Evrimsel Algoritmalar Aracı (MOEAT) adı verilen yazılımla çok amaçlı algoritma örneklerinin performanslarını, literatürden seçtiği sekiz adet test problemi ve iki adet mühendislik problemi kullanarak değerlendirmiştir. Geliştirilen yazılım aracılığı (MOEAT) ile seçilen problemlerin her biri en az beşer defa çalıştırılmış ve nesnel bir karşılaştırma yapılabilmesi açısından tüm algoritmalarda kullanılan temel genetik parametre değerleri ortak

seçilmiştir (Sağ ve Cunkaş, 2009). Bu değerlendirmenin sonuçlarına göre; VEGA bazı problemlerde başarılı sonuçlara ulaşabilmesine rağmen, çoğunlukla sadece Pareto-Optimal yüzeyin uç değer çözümlerini bulabildiği görülmüştür. VEGA'nın, farklı çözümler üretmede verimsiz bir algoritma olduğu saptanmıştır. MOGA popülasyon temelli bir algoritmadır. Pareto yüzeyi VEGA'ya nazaran daha iyi bir dağılım göstermekle beraber diğer algoritmaların sonuçlarıyla karşılaştırıldığında oldukça verimsiz sonuçlar ürettiği görülmektedir. NPGA'nın en büyük dezavantajlarından birisi, karşılaştırma kümesi boyutunun kullanıcıya bırakılmış olmasıdır. NSGA'da hücrelendirme tekniği ile daha verimli sonuçlar elde edilmiştir. Ancak, yüksek hesaplama karmaşıklığı, seçkinlik eksikliği ve bazı parametre değerlerinin ayarlanmasındaki belirsizlik gibi zayıf yönleri bulunmaktadır. Hem pareto yüzeye yakınsama hem de çözümlerin çeşitliliği açısından diğerlerine göre daha başarılıdır; ancak uygunluk ataması, yoğunluk tahmini ve arşiv boyutunun küçültülmesi gibi dezavantajlara sahiptir. Bunları ortadan kaldırmak için SPEA2 geliştirilmiştir. NSGAII, SPEA2 ve PESA farklı problemler üzerinde oldukça benzer davranışlar göstermekle beraber NSGAII diğerlerine göre daha iyi bir yakınsama ve geniş bir dağılım sağlamaktadır (Sağ ve Cunkaş, 2009).

Çizelge 2.1.'de Sağ ve Cunkaş'ın elde ettiği sonuçlar göz önünde bulundurularak çok amaçlı evrimsel algoritma örnek modelleri, kullanılan teknikler, avantaj ve dezavantajları ile açıklanmıştır.

Çizelge 2.1 : Çok amaçlı evrimsel algoritma örnek modelleri.

Model Adı	Yöntem	Avantaj-Dezavantaj
VEGA: Vektör Değerlendirmeli Genetik Algoritma (Schaffer-1985)	Popülasyon amaç fonksiyon sayısı kadar alt-popülasyonlara bölünür. Her bir amaç ayrı ayrı kendi popülasyonunda basit genetik algoritmada olduğu gibi ayrıca araştırılır. Elde edilen çözümler tekrar tek bir popülasyon içinde bir araya getirilir. Son olarak standart genetik algoritma operatörleri olan çaprazlama ve mutasyon bu toplu popülasyona uygulanarak yeni popülasyon üretilir. Sağlanması gereken her amaç ayrı ayrı ele alınır ve tam olarak sağlanan tüm çözümler elde edilmeye çalışılır. Bu durum çoğunlukla Pareto-Optimal yüzeyin uç noktaları ve komşuluklarındaki çözümlerin bulunmasına neden olur (Schaffer-1985).	VEGA bu alanda öncü bir çalışma olmakla birlikte diğerlerine göre ilkel bir yapıdadır. Bazı problemlerde başarılı sonuçlara ulaşabilmesine rağmen, çoğunlukla sadece Pareto-Optimal yüzeyin uç değer çözümlerini bulabildiği görülmüştür. Çünkü arama yönleri amaç uzayın eksenlerine paraleldir. Bir diğer yandan popülasyon temelli olması ve ilkel bir seçim tekniğini kullanması nedeniyle az sayıda ve yalnızca uç noktalardaki çözümlere odaklanan VEGA'nın, farklı çözümler üretmede verimsiz bir algoritma olduğu saptanmıştır (Sağ ve Cunkaş, 2009).
MOGA: Çok Amaçlı Genetik Algoritma (Fonseca ve Fleming-1993)	Popülasyon temelli bir evrimsel eniyileme algoritmasıdır. Uygunluk, aynı dizi içindeki bireyleri ayırt etmek için kullanılır. Etkin çözüm; problemin optimum çözümü olmadığında, her bir amaca mümkün olduğunca yaklaşan ve amaçlardan en az birini sağlayan çözümdür (Fonseca ve Fleming-1993).	MOGA popülasyon temelli bir algoritmadır. Pareto yüzeyi VEGA'ya nazaran daha iyi bir dağılım göstermekle beraber diğer algoritmaların sonuçlarıyla karşılaştırıldığında oldukça verimsiz sonuçlar ürettiği görülmektedir. Belli bir düzeni olmadan rastgele üretilen bu katayılar sayesinde MOGA, VEGA'ya göre daha iyi bir dağılım göstermesine rağmen kötü bir yakınsamaya sahiptir(Sağ ve Cunkaş, 2009).
NPGA: Hücrelendirilmiş Pareto Genetik Algoritma (Horn ve Nafpliotis-1993)	Turnuva seçiminde karşılaştırma setinden yardım alınmıştır. Eğer bir aday baskın olmayan setle ilişkiliyse ve diğeri değilse, turnuvayı biçimlendirici kazanır. Düğümün sonucunda, kazananı belirlemek için uygun sonuç kullanılır. Horn Nafpliotis'in geliştirdiği NPGA, Goldberg'in (Goldberg,1989) çok amaçlı eniyileme problemlerinin çözümü için pareto üstünlüğe dayalı seçim işlemini kullanır. Böylece, popülasyondaki çeşitliliğin korunmasını sağlar (Horn ve Nafpliotis-1993).	Burada bireyler arasındaki üstünlük kavramı, boyutu <i>tdom</i> olan rasgele seçilmiş bir küme ile bireyi mukayese ederek hesaplandığı için, bu algoritmanın başarısı yüksek derecede <i>tdom</i> parametresine dayanır. Uygun bir boyut seçilmezse, doğru pareto optimal noktalar bulunamayabilir. Küçük bir <i>tdom</i> seçilirse, bu birkaç popülasyonda nondominated noktaya yol açar. Yada <i>tdom</i> büyük bir değer seçilirse, erken yakınsamaya neden olabilir. NPGA'nın en büyük dezavantajlarından birisi, karşılaştırma kümesi boyutunun kullanıcıya bırakılmış olmasıdır(Sağ ve Cunkaş, 2009).

Çizelge 2.1 (devam) : Çok amaçlı evrimsel algoritma örnek modelleri.

Model Adı	Yöntem	Avantaj-Dezavantaj
NSGA: Bastırılmamış Sınıflandırılmalı Genetik Algoritma (Srinivas ve Deb-1994)	Aynı uygunluk değerleri, geçerli setteki baskın olmayan bireylere atanır. Baskın olmayan bireyler silinene kadar işlem devam eder. Tüm bastırılmamış bireyler sahte uygunluk değeri ile sınıflandırılır. Aynı kategorideki bireylerin hepsine aynı uygunluk değeri atanarak eşit seviyede yeniden üretilme potansiyeli sağlanır, sonra popülasyonda çeşitliliği sağlamak için paylaşım metodu uygulanır. İlk yüzeydeki bireyler geçici olarak ihmal edilir ve kalan popülasyon aynı süreçten geçirilerek ikinci bastırılmamış yüzey tanımlanır. Önceki yüzeyin minimum paylaşılan sahte uygunluk değerinden daha küçük tutulan yeni sahte uygunluk değeri, yeni yüzeydeki tüm bireylere atanır. Bu işlem popülasyondaki tüm bireyler sınıflandırılınca kadar devam eder(Srinivas ve Deb-1994).	NSGA'da hücrelendirme (niching) tekniği ile daha verimli sonuçlar elde edilmiştir. Ancak, yüksek hesaplama karmaşıklığı, seçkinlik eksikliği ve bazı parametre değerlerinin ayarlanmasındaki belirsizlik gibi zayıf yönleri bulunmaktadır. NSGA da bulunan zayıf yönlerin giderilmesiyle geliştirilen NSGAII algoritması, incelenen algoritmalar arasında en başarılı sonuçları üretmiştir(Sağ ve Cunkaş, 2009).
SPEA: Kuvvet Pareto Evrimsel Algoritma (Zitzler ve Thiele-1999)	İki nüfusa sahiptir, biri geleneksel tür, diğeri de baskın olmayan harici türdür, ikisi de genetik kombinasyona katılır. Harici nüfustaki her birey, eğer geleneksel nüfus içindelerse, daha güçlü olan dayanıklılık değerine atanır. Eğer bireyler nüfustaki diğer bireylerden baskı görüyorlarsa, geleneksel nüfustaki bireylere daha büyük uygunluk değeri verilir. Daha büyük uygunluk değerine sahip birey daha kötüdür. Sonuç olarak, harici nüfusun boyutu kümeleme ile azaltılır (Zitzler ve Thiele-1999).	Hem Pareto yüzeye yakınsama hem de çözümlerin çeşitliliği açısından diğerlerine göre daha başarılıdır. Ancak uygunluk ataması, yoğunluk tahmini ve arşiv boyutunun küçültülmesi gibi dezavantajlara sahiptir. Bunları ortadan kaldırmak için SPEA2 geliştirilmiştir. NSGAII, SPEA2 ve PESA farklı problemler üzerinde oldukça benzer davranışlar göstermekle beraber NSGAII diğerlerine göre daha iyi bir yakınsama ve geniş bir dağılım sağlamaktadır (Sağ ve Cunkaş, 2009).
PESA: Pareto Zarflama- Temelli Seçim Algoritması (Knowles ve Corne -2000)	PESA 1+1 evrimsel stratejidir. SPEA'da olduğu gibi, harici baskın olmayan nüfusu, sadece mutasyonda uğramış hali geçerli bireyin yerine kullanıp kullanmamayı belirlemek için kullanır. Bu yaklaşımın diğer bir yeni özelliği, amaçlanan alanın çiftli bölünmesidir ve kalabalık stratejiyi uygulamaktır. PESA algoritmasında seçim ve çeşitliliğin korunması hiper-grid aracılığıyla kontrol edilir. PESA'nın seçim yöntemi; bastırılan çözümlerin derecelendirilmesine bağlı olan sıkıştırma faktörü (squeeze factor) adı verilen bir katsayı kullanır (Knowles ve Corne -2000).	PESA başlangıçta daha hızlı bir yakınsama sağlamıştır fakat NSGA-II kadar iyi bir yakınsama ve daha geniş bir dağılım gösterememiştir (Sağ ve Cunkaş, 2009).

İncelenen modellerden, NSGA-II ile birlikte SPEA2 ve PESA algoritmalarının genel olarak başarılı sonuçlar ürettiği görülmüştür. Ancak genel olarak NSGA-II'nin, tüm test problemlerinde diğer algoritmalarla göre pareto yüzeyi bulma başarısının daha iyi olduğu söylenebilir (Sağ ve Cunkaş, 2009). NSGA-II 'nin incelenen algoritmalar arasında en başarılı sonuçları üretmesi nedeniyle tez kapsamında geliştirilen SSPM modelinde bu yöntem tercih edilmiştir.

2.2 NSGA-II Evrimsel Algoritması (Bastırılmamış Sınıflandırılmalı Genetik Algoritma)

İncelenen çizelge 2.1'de NSGA'nın bahsedilen dezavantajları göz önünde bulundurularak 2002 yılında NSGA-II algoritması yine Deb ve arkadaşları tarafından geliştirilerek ortaya konulmuştur. Strateji olarak NSGA'ya paralellik göstermektedir, fakat algoritma açısından oldukça farklı ve yeni yöntemler içerir. NSGA-II pareto-optimal yüzeyde daha geniş bir dağılım bulabilmektedir ve daha az karmaşıklığa sahip, hızlı ve seçkinlikli çok amaçlı bir algoritmadır. İşleyişi şu şekilde tanımlanabilir (Deb, 2001):

1. Adım: İlk olarak rastgele ebeveyn popülasyonu oluşturulur, (Po).
2. Adım: Popülasyon, Pareto üstünlüklerine dayalı olarak sıralanır. Her çözümün uygunluğu, kendi bastırılmamışlık seviyesine eşit olarak atanır.
3. Adım: Kalabalık-karşılaştırma (crowding comparision) işlemine dayalı ikili turnuva seçimi, çaprazlama ve mutasyon işlemcileri, N boyutlu Q_0 popülasyonunun bireylerini oluşturmak için ebeveyn popülasyona uygulanır.
4. Adım: Seçkinlik işlemi önceki en iyi bastırılmamış çözümlerle mevcut popülasyon karşılaştırılarak gerçekleştirilir.

İlk olarak birleşik popülasyon oluşturulur:

F1 en iyi bastırılmamış kümedir, birleşik popülasyondaki en iyi çözümlerdir. Bu popülasyon $2N$ boyutundadır. Önceki popülasyonun tümü ve mevcut popülasyonun bireyleri R_t ye eklendiği için, seçkinlik garantilenir. En iyi bastırılmamış küme olan $F1$ 'e ait olan çözümler, birleşik popülasyondaki en iyi çözümlerdir. Eğer $F1$ 'in boyutu N 'den küçükse, yeni popülasyon P_{t+1} için $F1$ kümesinin tüm üyeleri kesin olarak seçilir. Yeni popülasyonun kalan üyeleri, oranlarına göre sonraki

bastırılmamış yüzeylerden, sırasıyla önce F2 kümesinden, sonra F3 kümesinden alınır. Bu işlem P_{t+1} popülasyonu doluncaya kadar devam eder. Genelde, F_1 'den F_t 'e tüm kümelerde çözümlerin sayısı popülasyon boyutundan daha geniştir. Kesin olarak N popülasyon üyesi seçmek için, son yüzey F_t 'nin çözümleri kalabalık karşılaştırma operatörü kullanılarak azalan düzende sıralanır ve tüm popülasyon boşluklarını (slot) doldurmak için ihtiyaç duyulan en iyi çözüm seçilir. N boyutlu yeni popülasyon P_{t+1} , yeni bir N boyutlu Q_{t+1} popülasyonunu oluşturacak seçim, çaprazlama ve mutasyon için kullanılır. Burada kalabalık-karşılaştırma (crowding comparision) işlemine dayalı ikili turnuva seçim işlemi kullanılmaktadır (Sağ ve Cunkaş,2009).

Gerçek dünya problemlerinde araştırılan optimal çözüm genellikle tek amaca değil birden fazla amaca bağlıdır ve bu amaçlar genellikle birbiriyle çelişirler. Son zamanlarda çok amaçlı genetik algoritmalar birçok tasarım probleminin optimize edilmesinde yaygın olarak kullanılmaya başlanmıştır. Bu tez çalışmasında da; pareto optimum kavramının evrimsel algoritmalarla uyarlanmasıyla geliştirilen çok amaçlı evrimsel optimizasyon algoritmalarından en başarılısı olarak kabul edilen NSGA-II algoritması kullanılarak sürdürülebilir konut yerleşmesi tasarımı yapan SSPM modeli geliştirilmiştir.

3. TASARIMDA SÜRDÜRÜLEBİLİRLİK

Özellikle son yıllarda çevre bilinçli tüketimin yaygınlaşması sonucu ekolojik ürün kullanımı oldukça yaygınlaşmıştır. Buna paralel olarak ortaya çıkan sürdürülebilir bina; yaşam döngüsü süresince arazi seçimi, yerleşim, tasarım, yapım, işletim, bakım ve yıkım süreçleriyle insan sağlığı ve çevre üzerindeki negatif etkileri azaltılmış, enerji, su ve malzemenin etkin olarak kullanımına odaklanmış bir tasarım sonucu ortaya çıkan ürün olarak tanımlanmaktadır (Cassidy v.d., 2003).

3.1 Sürdürülebilirlik Kavramı ve İlkeleri

Sürdürülebilirlik pek çok alanı içinde barındıran bir kavram olarak karşımıza çıkmaktadır. İlk kez 1987’de yayımlanan Birleşmiş Milletler’in Brundtland Komisyonu raporunda kavram, “Ekonomik, çevresel ve toplumsal gereksinmelerin, gelecek kuşakların yaşam koşullarına zarar vermeden karşılanması” olarak tanımlanmıştır.

Uluslararası Mimarlar Birliği (UIA) ve Birleşmiş Milletler Eğitim Bilim ve Kültür Örgütü (UNESCO) de 1996 yılında hazırlanan Mimarlık Eğitimi Şartı’nda sürdürülebilir mimarlık hedeflerini:

- Binaları kullanan bütün insanlar için, insanlığa yaraşır bir yaşam kalitesi;
 - İnsanların, sosyal, kültürel ve estetik gereksinimlerine saygılı bir teknik uygulama; yapılı çevrenin ekolojiye duyarlı ve sürdürülebilir gelişimi;
 - Herkesin kendi malı ve sorumluluğu olarak görüp değer verdiği bir mimari;
- olarak belirtmiştir.

Yeşil bina terimi sürdürülebilir mimarlık anlayışının ürünü olan bir etiket olarak kullanılmaktadır. Yeşil yapılar doğal ışık ve iyi bir iç mekân hava kalitesiyle kullanıcıların sağlığını ve üretkenliğini korur ve geliştirirken, yapım ve kullanım sırasında doğal kaynakların tüketimine duyarlıdır. Yenilenebilir enerji kaynaklarına (güneş, rüzgâr ve toprak) yönelerek yeşil binanın kendi enerjisini üretmesi

hedeflenir. Gün ışığından maksimum faydalanma ve etkin ısı yalıtımı öncelikli tasarım kriterlerindendir.

3.2 Sürdürülebilir Arazi Yerleşim Kriterleri

Sürdürülebilir arazi başlığını kapsayan sürdürülebilir kent kavramına öncelikle değinmek yerinde olacaktır. 1987 yılında Birleşmiş Milletler 42. Genel Kuruluna sunulan “Ortak Geleceğimiz” adlı raporda "Sürdürülebilir Gelişme" kavramı kullanılmış ve “bugünün ihtiyaçlarını, gelecek kuşakların da kendi ihtiyaçlarını karşılayabilme olanağından ödün vermeksizin karşılamak” olarak tanımlamaktadır. Bu yaklaşım kent yönetimine de yansımış ve kentlerin sürdürülebilir gelişimi gündeme gelmiştir. Sürdürülebilir kentleşme kavramının ön plana çıktığı oluşumların başında 27 Mayıs 1994 tarihinde, Danimarka’nın Aalborg kentinde gerçekleştirilen Avrupa Sürdürülebilir Kent ve Kasabalar Konferansı gelmektedir. Konferansın sonunda Sürdürülebilirliğe Doğru Avrupa Kentler ve Kasabalar Şartı oluşturulmuştur. Sürdürülebilirlik açısından kentsel politikalar şu şekildedir;

- Sürdürülebilirlik için her kente özgü kararlar alınmalıdır.
- Doğal servet olarak kabul edilen su rezervlerinin, toprağın, biyolojik çeşitliliğin korunması ve yenilenemez enerji kaynaklarının kullanımının azaltılması sağlanmalıdır.
- Sosyal adaletin sağlanması gerekmektedir.
- Arazi kullanım ve imar planlamasında önemli çevresel etkenler göz önünde bulundurulmalıdır.
- Erişilebilirlik artırılarak toplumsal refahın artırılması ve daha az ulaşım ihtiyacı doğuracak biçimde yönlendirilmesi gerekmektedir.
- Yenilenebilir enerji kaynakları tek sürdürülebilir seçenektir.
- Halkın etkin katılımı sağlanmalıdır.
- Kentsel yönetim araçlarının oluşturulması gereklidir.

Aalborg Şartı’nın ortaya koymuş olduğu bu ilkeler sürdürülebilir kentsel gelişmenin gerek ilkelerini gerekse de uygulama yöntemlerini belirlemesi açısından büyük önem taşımaktadır. Bu ilkeler sürdürülebilir kentleşmenin hedeflerini oluşturmuştur. İnsan yerleşimlerinin iyileştirilerek herkes için barınma imkânının sağlanması,

sürdürülebilir arazi kullanım planlaması ve yönetimi, uygun altyapı hizmetlerinin sağlanması, sürdürülebilir enerji ve ulaşım sistemleri, afetlere maruz alanlarda yerleşme planlaması, insan yerleşimlerinin gelişmesi için kapasite oluşturulması gibi başlıklar sürdürülebilir kentleşme hedefleri olarak kabul edilebilir.

Kentsel açık alanlar, toplum için planlanan, düzenlenen veya kendiliğinden oluşmuş, toplumun yararlandığı alan olarak da tanımlanabilir. Konut dışı açık alanlar ise, büyük insan kitlelerini gruplayan, belli hizmetleri kolay ulaşılabilir mesafelere getiren, değişik küçük yerleşmelere kişilik kazandıran, aileler arasında komşuluk kurmak zorunda kalmadan da bu ünitelerde yaşayacak insanların ortak yaşamdan yararlanabilmesini sağlayan mekânlar şeklinde de tanımlanabilir. Açık alanlar; aktif ve pasif rekreasyon imkânı sağlamaları yanında, spor ve eğlence donatılarının yerleşimine olanak sağlayarak doğaya yakın ilişki imkanı verirler. Ekolojik olarak baktığımızda ise, açık alanlar kent ve konut çevresinde hava akımlarına olanak verirler ve zararlı gazları temizleyerek ışık ve hava sağlarlar; iyi bir gürültü bariyerleridirler. Arazide yer alan yeşil alanlar, araç trafiğini, yaya ve yerleşim mekânlarından ayırarak kütle boşluk ilişkisinin kurulmasına yardımcı olurlar. Sürdürülebilir arazi kullanımında açık alanların pozitif mekânlar olarak tasarlanması amaçlardan biri olmalıdır.

Tez kapsamında gerçekleştirilecek modelin kullanacağı sürdürülebilir arazi yerleşim kriterlerini üç ana başlık altında sınıflandırmak gerekirse;

- İmar planları ve yerel yönetmeliklerden gelen kriterler,
 - Bina sertifika sistemlerinin sürdürülebilir arazi başlığından gelen kriterler,
 - Fiziksel çevre kontrolü başlığı altında ele alınacak çevresel ve iklimsel kriterler,
- olarak tanımlayabiliriz.

3.2.1 İmar planları ve yerel yönetmelikler

İmar Mevzuatı, mimari tasarıma yön veren yasal düzenlemeleri içermesi yönüyle binanın biçimlenmesinde önemli yer tutar. Bu nedenle, mevzuat kapsamındaki imar yönetmelikleri, yeşil bina sertifika programları ve fiziksel çevre kontrolü kriterleri önem sırasına göre modele tanıtılarak birbirleriyle uyumlu çalışabilmesi sağlanacaktır. Günümüzde imar yönetmeliklerindeki arsa üretimi, yapının arsaya yerleşme biçimi, TAKS ve KAKS hesaplamaları gibi binanın dış kabuğunu

belirleyen ilgili maddeler sürdürülebilir bina tasarımını sağlayabilecek yeterlilikte olmadığından, yeşil bina sertifika programlarının kriterlerine ek olarak fiziksel çevre kontrolü kriterlerinin de modele dâhil edilmesiyle sürdürülebilir tasarıma doğru yönelme sağlanacaktır. Yapı yönetmeliğinde verilen minimum ve maksimum değerler sınır olarak kabul edilecek, optimum sonuç sürdürülebilirlik kriterlerine göre model tarafından belirlenecektir. Böylece birinin eksik olduğu noktalarda farklı başlıklara sahip kriterler devreye girerek sürdürülebilir yerleşme tasarımı sağlanabilecektir. Yapı yönetmeliğinden gelen kriterler kurgulanacak modelin kurallar listesinin bulunduğu Bölüm 5.5'te detaylı olarak sıralanmıştır.

3.2.2 Bina puanlama sistemleri

Yeşil bina endüstrisi 1990'lı yıllardan sonra hızla gelişmiş ve birçok örnek proje üretilmiştir. Binaların çevreye verdiği zararlı etkileri azaltmak amacıyla sürdürülebilir bina üretimini destekleyen bağımsız, kar amacı gütmeyen, üçüncü kişiler tarafından yürütülen ve çok katılımlı birçok organizasyon oluşmuştur. Yeşil bina sertifika sistemleri, bina bazındaki projelerin çevre üzerindeki etkilerini ve doğal kaynakları korumadaki duyarlılıklarını göstermede ölçülebilir bir referans sağlamaya çalışan derecelendirme sistemi olarak tanımlanabilir. Bu çalışmada değerlendirme sistemlerinden ikisi olan, Leed ve Breeam üzerinde durulacaktır.

3.2.2.1 Leed (Leadership in energy and environmental design)

Tez kapsamında incelenecek ilk değerlendirme sistemi LEED olacaktır. Sertifikalı değerlendirme sistemlerinden biri olan "Enerji ve Çevre Dostu Tasarımda Liderlik - Leadership in Energy and Environmental Design (LEED)" , ABD'deki Çevre Dostu Binalar Konseyi tarafından geliştirilen bir dizi kriterler listesidir. LEED' in ana hedefi; yapıların, yaşam döngüsü sürecinde oluşturdıkları çevresel etkilere dikkat çekmek ve yapı sektörünün bu etkiyi en aza indirecek ürünler üretmelerini teşvik etmektir. LEED değerlendirme kriterleri; sürdürülebilir arazi, su kullanımında verimlilik, enerji ve atmosfer, malzeme ve kaynaklar, iç mekân yaşam kalitesi, tasarımda yeni buluşlar olmak üzere altı ana kategoriden oluşmaktadır (Url-2).

LEED kapsamında farklı bina tipleri için ayrı sürümler çıkartılmıştır. 1998 yılında çıkarılan LEED – NC, yeni yapılan veya büyük yenileme geçiren ticari ve kurumsal binalar için kullanılan, özellikle ofisler üzerine yoğunlaşmış bir sürümdür. Bu sürüm okullar, apartmanlar, imalathaneler ve laboratuvarlar ve bunlar gibi daha birçok bina

tipiyle de kullanılmış ve oldukça güvenilir sonuçlar elde edilmiştir. LEED – EB, var olan binalar üzerine değerlendirme yapmak için geliştirilmiş bir sürümdür. LEED – NC’den sonra bu sürüm tamamlayıcı nitelikte çıkartılmıştır. Yapı elemanlarından çok, binada kurulu sistemlerin bakımı ve iyileştirilmesi konularını kapsamaktadır. LEED – CI, ticari iç mekânlar için geliştirilmiş bir sürümdür. Bu sürüm kiracılar ve iç mekân tasarımcılarına, sürdürülebilir seçimler yapabilmeleri için rehber olması amacıyla çıkarılmıştır. Daha sonra, bu sürüme yardımcı ve tamamlayıcı olarak bina kabuğu ve çekirdeği ile ilgili konuları kapsayan LEED – CS sürümü de geliştirilmiştir. LEED – H, sistemin konutlar için kullanılan sürümüdür. Amaç, konut yapımcıları, ev sahipleri ve yerel yönetimler için, ekolojik açıdan sağlıklı, kaynaklarını idareli kullanan yaşam alanları yaratacak bir rehber oluşturmaktır (Url-2). Bütün diğer LEED sürümlerinde arazi seçimine sadece birkaç kredi ayırıp, ağırlık yeşil bina tasarım ve uygulamasına verilmişken, bu sürümlerden farklı olarak LEED–ND’de ise ağırlık arazi seçimine ve doğru büyümeye verilmektedir. Bu sürüm sayesinde binadan önce arazinin özellikleri ve koşulları incelenebilmekte ve arazi karakterlerine uygun tasarımla daha etkili yeşil bina tasarımlarına ulaşmak mümkün olmaktadır (Url-2).

LEED binaları;

- Sürdürülebilir Arazi,
- Su kullanımında Verimlilik,
- Enerji ve Atmosfer,
- Malzeme ve Kaynaklar,
- İç Mekân Yaşam Kalitesi,
- Tasarımda Yeni Buluşlar.

olmak üzere altı ana çevresel kategori ve bunların alt kategorilerinden oluşmaktadır (Url-2). Bu tez kapsamında LEED kriterlerinden sürdürülebilir arazi başlığı altında yer alan arazi yerleşim kriterleri göz önünde bulundurulacak ve bu kriterler, kullanılan pareto genetik algoritmanın uygunluk fonksiyonlarının bir kısmını oluşturacaktır.

Sürdürülebilir arazi kriterleri genel yaklaşım olarak; yeşil alanlar ile daha önceden yerleşim yapılmamış alanlarda yeni bir yerleşim yapılmasından ve tarım alanlarında,

doğal habitata zarar verecek, yerel ya da bölgesel erozyona sebep olacak şekilde yerleşim yapılmasından kaçınılması gerektiğini belirtir. Yeni yerleşimlerin mevcut yerleşimlere, ulaşım ağlarına yakın olması tercih edilmelidir. LEED sürdürülebilir arazi kriterleri;

1. Çevredeki yapıların yoğunluk oranına uygunluk,
2. Açık alanların maksimum tutulması,
 - Arazi yönetmeliğinde verilen değerin %25'inden fazlasının açık alan olması,
 - Arazi yönetmeliğinde açık alan belirtilmemişse arazinin %20 sinin açık alan olması,
 - Herhangi bir yönetmelik oranı olmayan yerde bina oturma alanı kadar yeşil alan bırakılması,
 - Sert zemin açık alan sayılmadığı için geçirimsiz yüzeylerin (sert zeminlerin) azaltılması,
3. Önceden bina yapılmış yere bina yapma kuralları:
 - İlk binanın oturma alanı dışında kalan arazinin %50'sinin korunması ya da tüm arazinin %20' sininin korunması,
 - Isı adası etkisini azaltmak için arazideki geçirimsiz yüzeylerin (asfalt, beton yol) %50'sinin örtü sistemi ile kapatılması,
 - Çatı alanının %50 sinin yeşil çatı yapılması, ya da güneş paneli kullanılması olarak sıralanabilir (Url-2).

3.2.2.2 Breeam (Building research establishment environmental assessment methodology)

Tez kapsamında incelenecek sertifika sistemlerinden bir diğeri olan BREEAM (Yapı Araştırma Kurumu Çevre Değerlendirme Yöntemi), İngiltere Yapı Araştırma Kurumu (BRE) tarafından 1990 yılında geliştirilen ve hala kullanılan çevresel değerlendirme yöntemidir (Larsson ve Cole, 2001). Sistemin amacı bina performanslarının sürdürülebilirlik bağlamında değerlendirilmesidir. BREEAM ile yapılar yönetim, sağlık ve konfor, enerji, ulaşım, su, malzeme, atıklar, arazi kullanımı ve ekoloji, kirlilik olmak üzere dokuz ana kriter ve bunların alt kriterleri çerçevesinde incelenmektedir (Url-1).

BREEAM değerlendirme kriterleri;

- Yönetim,
- Sağlık ve konfor,
- Enerji,
- Ulaşım,
- Su,
- Malzeme,
- Atıklar,
- Arazi kullanımı ve ekoloji,
- Kirlilik

olmak üzere dokuz ana kriterdir (Url-1). Tez kapsamında kullanılacak pareto genetik algoritmanın uygunluk fonksiyonlarında yer alacak BREEAM kriterleri arazi kullanımı, ulaşım ve kirlilik olacaktır. Bu kriterler:

1. Arazide daha önceden yerleşim yapılmış ise kirlenilmiş alanların yeniden kullanımı sağlanmalı, bina metrekaresi buna göre şekillenmelidir. Yeni yapılacak binanın oturma alanının % 75 'i arazinin son 50 yıl içerisinde endüstriyel, ticari ya da konut işlevli olarak yapılaşmış olan bölgesinde olmalıdır. Daha önce rekreasyon alanı, park, tarım arazisi, ormanlık olan arazilerde bu kriter geçerli değildir.
2. Yerleşimden önce tüm negatif etkilerden arındırmak suretiyle kirlilik içeren alanlara yerleşme teşvik edilmelidir.
3. Toplu ulaşım noktalarına bina girişinden uzaklık minimum olmalıdır.
4. Market, bankamatik, postane gibi hizmet noktalarına 500 metreden daha uzak olmama şartı aranmaktadır.
5. Bisiklet kullanımını teşvik etmek amacıyla güvenli ve belirtilen sayıda bisiklet park alanı, duş, soyunma odaları tahsis edilmelidir.
6. Bina tipine ve kullanıcı profiline göre binaya yaya, bisiklet ve diğer ulaşım sistemleri ile engelli ulaşımını gösteren bir ulaşım planı hazırlanmalıdır.

7. Toplu taşımayı özendirmek ve tekil araç kullanımını engellemek amacıyla otopark kapasitesi limitli hale getirilmelidir. Her 3 ya da 4 kullanıcıya bir araç park yeri sağlanmalıdır.

8. Çevrede sese hassas yerleşimler varsa bunlara minimum 800 m çaplı mesafede yerleşim yapılmalıdır.

Bina sertifikasyon sistemlerine eleştirel yaklaşımlar mevcuttur. Özellikle tüm kriterlerin farklı iklim ve coğrafyaya sahip olan ülkelerde uygulanabilirliği mümkün olmamaktadır. Her farklı coğrafyaya göre değişken kriterlere sahip sertifikasyon sistemlerinin geliştirilmesi, tüm ülkelerde kullanılabilirliklerini artıracaktır. Sertifikasyon sistemlerinin rekabet ortamının olduğu inşaat sektöründe kullanılması, binaların ve kullanılan malzemelerin hem kalitesini artırmakta hem de inşaa aşamasından başlayarak daha çevreci şantiye ve daha çevreci bir bina anlayışına katkı sağlamaktadır.

3.2.3 Fiziksel çevre kontrolü kriterleri

Sürdürülebilir mimarlık “yapı içi”, “yapı kabuğu” ve “fiziksel çevre” olarak üç farklı alanda gelişme göstermiştir. Oluşan bu üç alan birbirlerini doğrudan etkilemekte ve ilgilendirmektedir. Mimari tasarımın temel amaçları bağlamında yer alan, tasarlanan yapma çevrelerin kullanıcı konforunu en üst düzeyde gerçekleştirme amacına yönelik olarak mimarlık ve mühendislik çalışmalarının arakesitinde yer alan konular, bu başlık içinde yer alır. İklimlendirme, aydınlatma, akustik ve gürültü kontrolü, ısıtma-havalandırma, sıhhi tesisat ve yangın güvenliği konularında mevcut yönetmelik ve mimari tasarım bilgileri fiziksel çevre kontrolü kriterleri olarak adlandırılabilirler. Tasarımın arazinin çevresel özelliklerine adaptasyonu sürdürülebilir tasarım için önemli bir kriterdir. İklim yapay çevreyi oluştururken ve şekillendirirken tasarımı etkileyen en önemli etkenlerden biridir. Sürdürülebilir bina tasarımını etkileyen iklimsel veriler şu şekilde sıralanabilir;

- Güneş ışınımı,
- Rüzgâr ve hava hareketleri,
- Sıcaklık ve nem.

Dolayısıyla sürdürülebilir mimari, en önemli enerji kaynağı olan güneşten faydalanmayı öngörmektedir. Ancak iklim tipine ya da bina kullanım amacına göre

güneş ışınlamalarına yapıda kimi zaman gerek duyulur, kimi zamansa güneş ışınlarının ısıtıcı etkisinden korunmak için önlemler almak gerekir. Örneğin, sıcak iklimlerde yüzeylerde oluşabilecek ısı kayıplarını arttırmak amacıyla parçalı ve dış cephe alanı fazla bina formları, soğuk iklimlerde ise bina dış cephelerinde oluşabilecek ısı kayıplarını önlemek açısından daha kompakt ve düz formlar tercih edilmelidir.

Bu çalışmada toplu konut birimlerinin seçilen iklime, hâkim rüzgâr yönüne ve manzaraya göre yönelimleri, enerji kaybını en aza indirecek bina formlarına sahip olmaları, sürdürülebilir tasarımın önemli adımları olarak kabul edilmiş ve bu kriterler geliştirilen modelde uygulanacak kurallar listesinde detaylı olarak sıralanmıştır.





4. SÜRDÜRÜLEBİLİR TASARIMDA KULLANILAN BENZETİM MODELLERİ VE MEVCUT EVRİMSEL MODELLER

4.1 Sürdürülebilir Bina Tasarım Sürecinde Kullanılan Benzetim Modelleri

Günümüzde bilgisayar teknolojilerindeki gelişmeler ve kullanıcı odaklı modeller bilgisayarlı bina tasarımında ve performans analizlerinde sıklıkla kullanılan bir araç ve bir karar destek sistemi haline getirmiştir (Al-Homoud, 2001).

Bina benzetim modellerinin mimari tasarım sürecindeki işleyişi, iklim tipi gibi eldeki verilerin sisteme girilerek benzetiminin gerçekleştirilmesi ve sonrasında elde edilen sonuçların değerlendirilerek tasarıma yön vermesi şeklindedir (Harputlugil, 2009) .

Bina benzetim modelleri, genellikle bina kabuğu ve iç mekânın ısı konforunu belirleme, bina ve yapı bileşenlerinin biçim ve boyutunu saptama gibi özel amaçlara yönelik olarak hazırlanmış ve genel olarak sürdürülebilir bina tasarımında kullanılan, performans programlarıdır. Çizelge 4.1’de piyasadaki bina benzetim modellerinin işlevleri, girdi ve çıktı değerleri, güçlü ve zayıf yönleri bir araya getirilmiştir. Sürdürülebilir bina tasarımı amacıyla geliştirilmiş bulunan benzetim modellerinin etkilerini kıyaslama yaparak görmek, mimari tasarım sürecinde kullanılabilirmeleri açısından yararlı olacaktır. Bu kapsamda LEED ve BREEAM standartlarını kullanarak sürdürülebilir bina tasarımına katkı sağlamayı amaç edinmiş benzetim modellerine de yer verilmiştir.

Tez kapsamında incelenen benzetim modellerinden ilki olan Athena modeli binanın yaşam döngüsünde çevresel etkileri en aza indirmek ve sürdürülebilir malzeme kullanımına teşvik etmek amacıyla geliştirilmiştir, tasarımın ileri evrelerinde kullanılabilir niteliktedir (Url-5). BEES modeli de malzeme odaklıdır, ek olarak eknoomik performans verilerini içermektedir (Url-6). Building Green House Rating modeli enerjinin etkin kullanımına odaklanmıştır, binalar enerji kullanımlarına göre puanlanmaktadır(Url-7). ECO-BAT modeli, inşaat, imalat, atık yönetimi ve ulaşım olmak üzere bina ömrü boyunca çevresel etkileri değerlendiren bir bilgisayar programıdır (Url-8). ECOTEC modeli kavramsal tasarım aşamasında 3 boyutlu

model üzerinden kapsamlı ışıık, gölge analizi yapmakta, binanın enerji performansını deęerlendirmektedir, fakat sonraki aşamalarında mühendislik danışmanlığı gerektirmektedir (Mahdavi ve El-Bellahy, 2005). ENVEST modeli erken tasarım aşamasında bina yaşam döngüsünde çevresel etkileri tahmin etmek ve seçilen ana malzemeler hakkında bilgi vermek için geliştirilmiştir (Url-9). Çizelge 4.1'de girilen veriler ve alınan çıktılarına bakıldığında, genel olarak bina performans programlarının iç mekân enerji korunumuna ve bina malzemelerine odaklandıkları söylenebilir. Literatürde, binayı arazideki dięer fonksiyonlarla (yaya-taşıt yolları, yeşil alanlar, v.b.) birlikte ele alarak, erken tasarım evresinde kullanılabilecek bir karar destek sisteminin eksikliği göze çarpmaktadır.

Benzetim programlarının çalışabilmesi için ilk aşamada bina modelinin oluşturulması gerekmektedir; bu da tasarım verilerinin tam olarak tanımlanmış olmasını gerektirir. Bu veriler kütle formu ve malzeme bilgisi başta olmak üzere iklimlendirme sistemlerinin özelliklerinden, işletim stratejilerine kadar ayrıntılı pek çok bilgiden oluşmaktadır. Bu programları iç mekân konforu ve enerji verimlilięi sağlamada faydalı bir tasarım aracı olarak kullanan az sayıda mimar vardır. Bunun nedenlerinden biri programların birbirleriyle çelişkili sonuçlar üretebilmeleridir. Dięer bir neden ise bina geometrisinin, nicel bilgiler içermeyen, tasarım boyunca akışkan ve sürekli deęişime uğrayan bir süreçle tasarlanmasıdır. Tasarımın bu aşamasında, yüzey alanlarını hesaplama veya belirli bir malzemenin ısı etkisini araştırma zorunluluęu, tasarımcının sezgisel düşünmesi gerekli noktalarda matematiksel düşünmeye zorlanmasıdır (Harputlugil, 2011).

Çizelge 4.1 : Bina benzetim modelleri.

Model	İşlevi	Güçlü Yönleri	Zayıf Yönleri
Athena Model	Binanın yaşam döngüsünde çevresel etkileri en aza indirmek ve sürdürülebilir kalkınmayı teşvik edecek malzeme seçenekleri sunmak amaçlanmıştır. Yaşam Döngüsü Değerlendirmesi (The Life-Cycle Assessment-LCA) kullanıcı için model tarafından yapılmaktadır. Birden fazla kavramsal tasarım modele tanıtıldıktan sonra model 6 çevresel ölçüte göre tasarımları birbiriyle kıyaslamaktadır (Url-5).	Bölgesel hassas veri tabanları, kullanıcı dostu arayüzü sayesinde tasarım elemanları her yönüyle karşılaştırılabilir.	Yapı malzemeleri ve montajlar ile sınırlıdır. Model sadece ahşap, çelik ve beton ürünleri için veritabanları içerir.
BEES	BEES, ISO 14040 standartlarında belirtilen yaklaşımı kullanarak yapı ürünlerinin çevresel performansını ölçer. Bir ürünün yaşam bölgesindeki tüm aşamaları analiz edilir: Hammadde satın alma, üretim, taşıma, yükleme, kullanım ve geri dönüşüm ve atık yönetimi ve genel bir performans skoru çıkarır. Yaşam döngüsü grafikleri, bina ürün alternatiflerinin çevresel ve ekonomik puanları, çevresel etkilerin ve maliyetlerin grafikleri elde edilir (Url-6).	Çevre bilimi ve ekonominin harmanını sunmaktadır.	23 yapı elemanını kapsayan sadece 200 yapı ürünü için çevresel ve ekonomik performans verilerini içerir.
Building Greenhouse Rating	Ofis binalarını 1 den 5 e kadar yıldızlandırarak değerlendirilmelerini sağlar. Yapılar, Australian Building Greenhouse Rating tarafından tespit edilir, yıldız çokluğuna göre, enerjinin etkili kullanımı ve ucuzluğu artmakta, gaz emisyonları azalmaktadır. Sera gazı emisyonu (kgCO ₂ /m ²) ve enerji tüketimi (MJ/m ²) raporu elde edilir. Raporda, binanın yıldızla derecelendirmesi gösterilir (Url-7).	Milli emlak piyasasına karşı gerçek enerji ve sera performansı oluşturarak enerji verimliliği sağlar.	Normalizasyon eğrileri sadece Avustralya ofis binaları için kullanılabilir.

Çizelge 4.1 (devam) : Bina benzetim modelleri.

Model	İşlevi	Güçlü Yönleri	Zayıf Yönleri
ECO-BAT	ECO-BAT, inşaat, imalat, yedek, atık yönetimi ve ulaşım dâhil olmak üzere, bina ömrü boyunca çevresel etkileri değerlendiren bir bilgisayar programıdır. ECO-BAT bir malzeme, bir öge veya bir binanın farklı değişkenleri arasında karşılaştırma sağlar. Binanın yapısı, farklı malzeme ile oluşturulmuş yeni unsurlar ekleyerek (duvarlar, pencereler, çatılar, vb.) tanımlanabilir.(Url-8)	Bölgesel hassas veri tabanları, kullanıcı dostu arayüzü sayesinde tasarım elemanları her yönüyle karşılaştırılabilir.	Yapı malzemeleri ve montajlar ile sınırlıdır. Model sadece ahşap, çelik ve beton ürünleri için veritabanları içerir.
ECOTEC	Kavramsal tasarımdan detaylı projelendirmelere kadar her tür sürdürülebilir yapı için kapsamlı analiz araçları sağlar. 3 boyutlu model üzerinden kapsamlı güneş analizi yapar. Günümüzde geliştirilen birçok bina simülasyon yazılımı arasında, eğitimde kullanılmak üzere, görsel arayüzü ile öğrencilerin kolaylıkla uyum sağlayabilecekleri, gereğinden fazla veri girişiyle öğrencileri analiz kavramından uzaklaştırmayacak bir programdır. Bu program, erken tasarım evrelerinde enerji performansını ölçmek için tasarlanmıştır (Mahdavi ve El-Bellahy, 2005)	Özellikle kavramsal tasarım aşamasında (değiştirme ve düzenleme için değerlendirilmesi ve test edilmesi gereken, sürekli geri dönüşlerle beslenen bir süreçte) kullanılması planlanan bu bina simülasyonprogramı, deneme/yanılmalara imkân veren eğitsel özellikleri ile ön plana çıkmaktadır.	Projenin sonraki aşamalarında mühendislik danışmanlığı gerektirmektedir.
ENVEST	Erken tasarım aşamasında bina yaşam döngüsünde çevresel etkileri tahmin etmek için geliştirilmiş ilk yazılım. Bu sürüm, ofis binaları ve bina ömrü boyunca ve inşaat sırasında kullanılan malzemelerin ve enerji kaynaklarının çevresel etkilerini göz önünde bulundurur.(Url-9)	Binanın çevresel etkilerini en uygun hale getirir ve seçilen ana malzemeler hakkında bilgi verir.	Basitleştirilmiş yöntemler kullanır.

Tasarım sürecinde bu tür programların gerekliliği ve uygulanabilirliğine ilişkin problemler Yahiaoui ve Harputlugil (2006) tarafından aşağıdaki gibi sıralanmıştır:

1. Program girdileri geniş ve bilimsel olarak oldukça detaylıdır. Tasarımın erken aşamalarında henüz elde edilmemiş veriler, analizi gerçekleştirirken varsayılmak zorunda kalmaktadır.
2. Programlar, çok sayıda çıktı içermektedir. Benzetim sonuçlarının anlaşılması ve yorumlanması zordur.
3. Detaylı enerji benzetim programlarının çoğu araştırma merkezlidir. Bunları kullanmayı öğrenmek zordur ve ustalaşmak uzun bir zaman gerektirir.
4. Programların çoğunda kullanıcı arayüzü bulunmamaktadır veya kullanımı kolay değildir. Özellikle tasarımın ilk aşamalarında kullanılması önem taşıyan programların, katı veri yapısı ve gerekleri nedeniyle, kendisini grafiksel olarak ifade etme yolunda eğitim görmüş tasarımcılar hayal kırıklığına uğramaktadır.
5. Yazılım kullanıcıya belirli ihtiyaçlarla kolayca örtüşebilen programlamayı gerçekleştirecek esnekliği sağlamamaktadır.
6. Çoğu programın geçerliliği ve akreditasyonu düşünülmemiştir. Kullanıcılar hangi programın daha iyi benzetim sonucu verebileceği konusunda kararsız ve kuşkuludur.

Bina tasarımı, tasarım sürecinden yapımına kadar ekip çalışması gerektirmektedir. Günümüz tasarım sürecinde ilk olarak, bina dış kabuğu mimarlar tarafından tasarlanmakta, sonrasında ise iklimlendirme, aydınlatma hesaplamaları mühendisler tarafından yapılmaktadır. Sürdürülebilir bir bina tasarımında ise tasarım süreci bu şekilde ilerlememelidir. Sürdürülebilir bina kriterlerinin tasarımın eskiz aşamasından itibaren değerlendirilmesi gerekmektedir, çünkü en önemli sürdürülebilirlik kararları araziye yerleşimden itibaren alınmaktadır.

Bu nedenle mimari tasarımın ön tasarım aşamasında değerlendirilmesi ve binanın bu doğrultuda biçim alması gereklidir. Mimarlar binalarını sadece tasarlayıp yapım sektöründe var olan uzmanlık alanlarının bakış açıları ile değerlendirmeyi de destekleyen bir bütünleşik tasarım sistemi içinde yer almalıdır. Bu bağlamda geliştirilen benzetim modelleri bu amacı tam olarak karşılayamamaktadır.

4.2 Sürdürülebilir Bina Tasarım Sürecinde Kullanılan Evrimsel Modeller

Sürdürülebilir tasarım için gerçekleştirilen benzetim modellerinin yanında ön tasarım evresinde kullanılmak üzere yeşil bina eniyilemesini hedefleyen evrimsel modeller de geliştirilmiştir. Bu çalışmada üzerinde durulacak modellerden biri olan yeşil bina eniyileme modeli, bina yönleneşmesi ve binanın dış cephe malzemeleri esas alınarak sürdürülebilir tasarım kriterlerinin bir kısmını karşılamaktadır. İncelenecek ikinci çalışma ise, daha büyük ölçekte sürdürülebilir arazi kullanımını hedefleyen ve bunu mekânsal eniyileme yöntemini kullanarak gerçekleştiren bir modeldir.

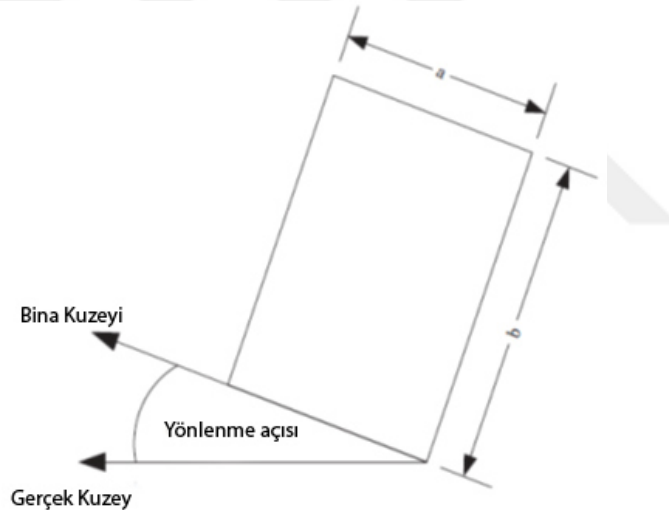
4.2.1 Yeşil bina tasarımı için plan eniyileme modeli

Tez kapsamında literatürden örnek alınarak incelenecek modellerden ilki Weimin Wang, Hugues Rivard ve Radu Zimeureanu (2006) tarafından geliştirilen, yeşil bina tasarımı için plan alternatifleri üreten eniyileme modelidir. Biçim, yeşil bina tasarımında belirleneşmesi gereken önemli kararlardan biridir; çünkü bina biçiminin enerji performansında ve yapı maliyetinde etkisi oldukça büyüktür. Bu çalışmada bina biçimlerinin genetik algoritma kullanılarak eniyilenmesi amaçlanmıştır. Bina oturumu çok kenarlı bir poligon olarak sisteme tanıtılmaktadır. Model tarafından üretilen alternatifler, bina biçimiyle ilgili olan pencere oranlarına ve plandaki girinti çıkıntılara göre birbirileriyle kıyaslanmaktadır. Çok amaçlı genetik algoritma, üretilen plan alternatiflerini duvar malzemeleri ile birlikte değerlendirerek maksimum doğal ışık ve minimum enerji tüketimine göre eleyerek, en iyi alternatifi kullanıcıya sunmaktadır. Sürdürülebilir tasarım bağlamında iklim'e uygun bina biçimi, en iyi duvar malzemesi, minimum enerji tüketimi, doğal ışıktan maksimum faydalanma ve minimum maliyet unsurları göz önünde bulundurularak model tarafından pareto optimal çözüm üretilmektedir.

Çalışmada alternatif bina tasarımlarını tanımlamak için iki tür değişken tipi tanımlanmıştır; kesikli değişken ve sürekli değişken. Pencere tipi, sadece kesikli değişken olabilmektedir; kullanılacak pencere tipi sınırlı sayıdaki değişik pencere tipi listesinden seçilmektedir. Bazı değişkenler de sürekli ya da ayrık olabilmektedir. Örnek olarak bina yönelimi, 0° ile 90° arasında değer alabilmektedir ya da önceden ayarlanmış listelerden 0°, 15° veya 30° arasından seçilebilmektedir.

Bu çalışmada yapılar dikdörtgen şekli ile sınırlıdır. Şekil 4.1'de bina yöneliminin nasıl tanımlandığı gösterilmiştir. Kullanılan tanımlar;

- Bina yöneliminde (orientation) saat yönü pozitif yön olarak kabul edilmektedir.
- Yapı planının en boy oranı, Şekil 4.1'de gösterilen a ve b'nin birbirine olan oranıdır(a/b).
- Pencere tipi (winType), pencere özelliğini tanımlar. Örnek olarak, arasında 13 mm hava boşluğu bulunan çift camlı pencere.
- Duvar tipi (wallType), katman dizileri ile duvar yapılandırmasını tanımlar. Örnek olarak yalıtımlı yığma duvar, dışarıdan içeriye doğru ısı yalıtım malzemesi, buhar bariyeri, yığma duvar ve boya katmanlarından oluşmaktadır.
- Duvarın her bir katmanı, seçilen gerçek malzemeyi tanımlar. Örnek olarak, yalıtım katmanındaki duvar tipi 76 mm kalınlığında fiberglas plakaları veya 101.6 mm kalınlığında mineral yün plakaları olabilir.
- Çatı tipi, bir dizi katman ile çatı yapılandırmasını tanımlar.



Şekil 4.1 : Bina oryantasyonunun tanımlanması (Wang, Rivard ve Zmeureanau, 2006).

Çalışmada bina yönelimi, maksimum doğal ışıktan yararlanılacak şekilde, verilen arazi sınırı içerisinde, sistem tarafından alternatifler olarak üretilir ve bu alternatiflerin elenmesi bina, duvar ve çatı malzemelerinin belirlenmesiyle gerçekleşir. Son ürün hem doğal ışıktan en iyi faydalanabilen, hem de duvar malzemeleri ile çevreye en iyi uyumu sağlayan çözüm olarak karşımıza çıkmaktadır.

Ön tasarım aşamasında tasarımcıları daha iyi çözümlere yönlendirecek araçlar günümüzde önemli rol oynamaktadır. Çok amaçlı eniyileme modeli bu çalışmada en

iyi ya da en iyiye yakın yeşil bina tasarımlarını verilen şartlara uygun olarak üretmektedir. Binanın yaşam döngüsünün çevresel performansı enerjinin verimli tüketimine bağlıdır, bu da modelin amacını oluşturmuştur. Amaçlanan pareto çözümler tasarımcıların ekonomi ve çevresel performansın ilişkisini daha iyi anlamasını sağlayacaktır (Wang, Rivard ve Zmeureanau, 2006).

4.2.2 SMOLA-Mekansal eniyileme yöntemi ile sürdürülebilir arazi kullanım modeli

Sürdürülebilir kent formu oluşturmak arazi planlamada en önemli kriterlerden biridir. Var olan arazi kullanım biçimi ve yoğunluğu ekonomik, sosyal ve çevresel pek çok etkiye sahiptir. Bu çalışmada çelişen iki kriter olan maksimum açık alan ve maksimum kalkınma, komşu yerleşimlere uygunluk ve var olan yerleşim alanlarına yakınlık şartlarını göz önünde bulunduran çok amaçlı mekânsal eniyileme modeli geliştirilmiştir. Zielinska, Church ve Jankowski (2008) tarafından geliştirilen SMOLA (Sustainable Multi Objective Land Use Allocation Model) çok amaçlı mekânsal eniyilemeyi hedeflemektedir. Yerleşim yoğunluklarına göre çözümler üreten model 400 hücreden oluşmakta ve önerilen arazi kullanımı bu kareler üzerinde model tarafından oluşturulan desenlerle tanımlanmaktadır. Çalışmada çok ajanlı benzetimlerle oluşan yerleşim dokuları görsel olarak sunulmaktadır.

Modelde sürdürülebilirlik, ekonomik, sosyal ve çevresel gereklerin ön planda tutulmasıyla hedeflenmiştir. Modelde bina kullanım fonksiyonları homojen hücre tipinde ölçülebilir birimler olarak gösterilmiştir. Arazi kullanımında var olan bitişiklik ve komşuluk üç tür olarak sınıflandırılmıştır: ağ tabanlı bitişiklik, kenar tabanlı kompaktlık ve yakınlık.

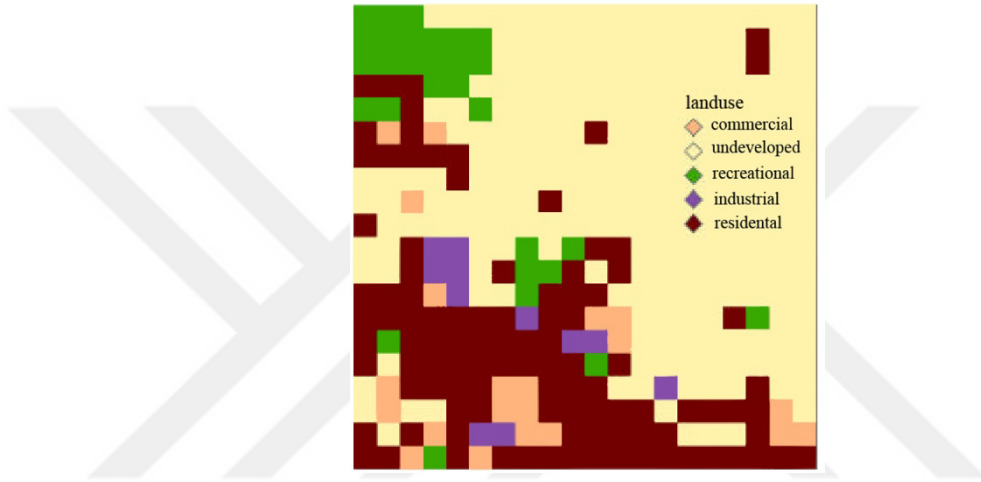
Bunlar göz önünde bulundurularak model formülasyonu için 4 amaç belirlenmiştir;

1. Etkili arazi kullanımını teşvik etmek amacıyla açık alan kullanımının minimize edilmesi,
2. Ekonomik açıdan mekânsal değişimin sağlanması için yeniden gelişimin minimize edilmesi,
3. Çevresel bozulmayı engellemek amacıyla bitişik arazilerde uyumsuzlukların en aza indirilmesi,

4. Var olan yerleşimlere mesafenin en aza indirilmesiyle erişilebilirliğin sağlanması olarak sayılabilir (Zielinska, Church ve Jankowski, 2008).

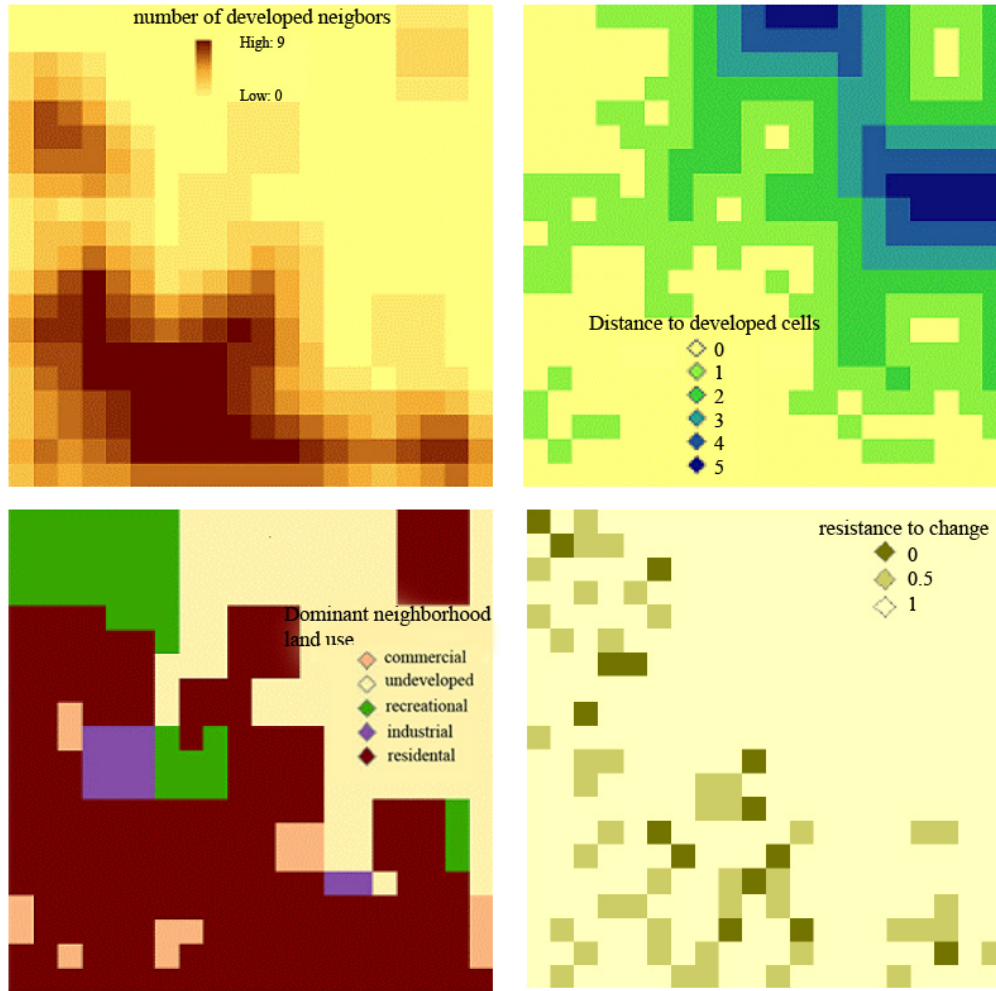
Modelde bir ızgara sistem oluşturulmuştur. Her hücrenin arazi kullanım biçimi homojendir ve yerleşimin gelişme yönünün açık alanlara doğru olması engellenmiştir.

Model 20x20 hücreden oluşan farazi bir örneklerle test edilmiştir. Şekil 4.2'de görüldüğü gibi hücreler kamusal, endüstriyel, konut, rekreasyon ve gelişmemiş alan olarak sınıflandırılmıştır.



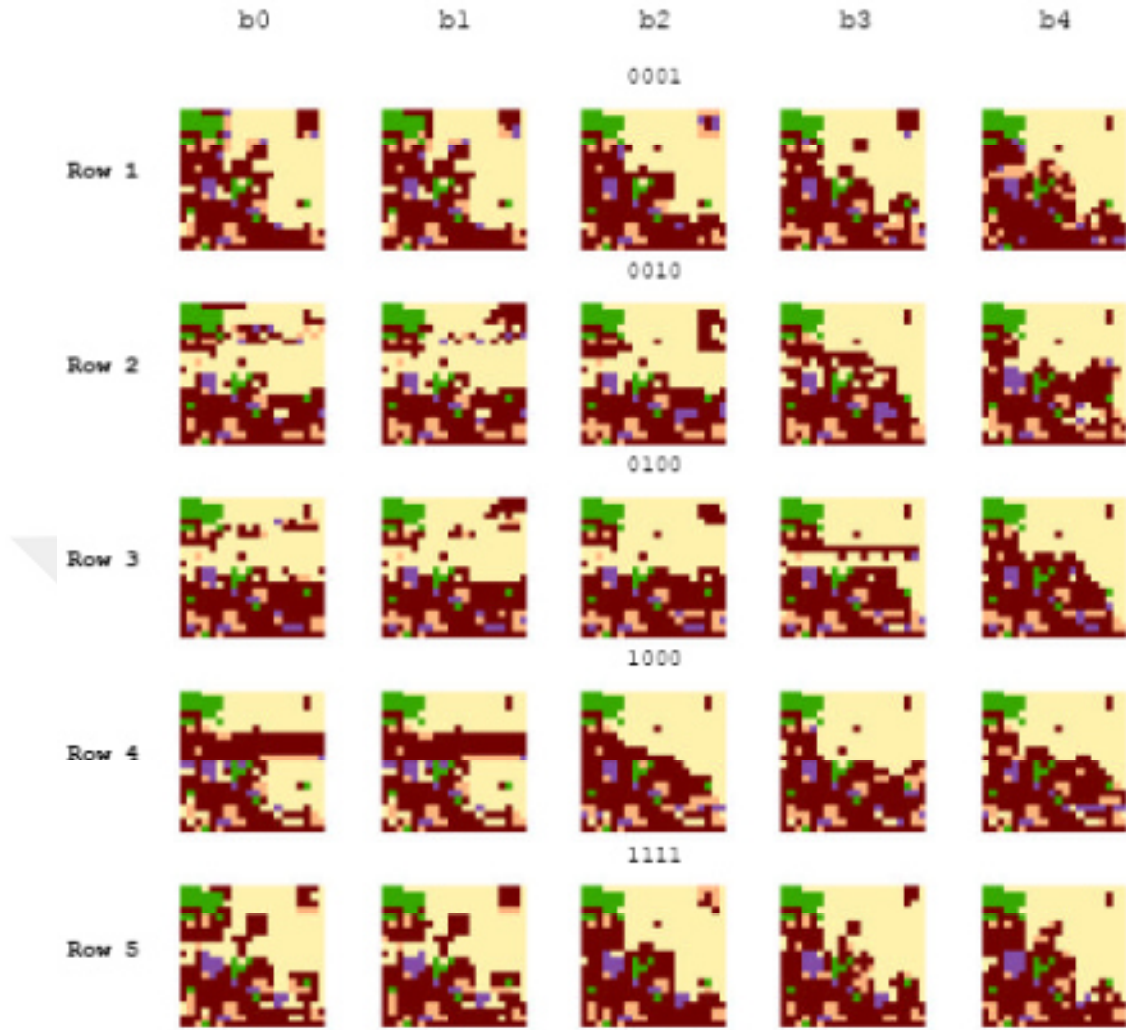
Şekil 4.2 : 20x20 hücreden oluşan bir arazi parçası (Zielinska, Church ve Jankowski, 2008).

Rekreasyon alanı korunmuş alan olarak tanımlanmış ve yeniden gelişimin dışında tutulmuştur. Gelişmemiş olarak gösterilen alanlar açık alan olarak kabul edilerek model çalıştırılmıştır. Arazi kullanım biçimleri Şekil 4.3'te gösterilmiştir. Gelişmiş komşu hücreler ve baskın komşu arazi kullanımı katmanları Python betikleriyle kodlanmıştır.



Şekil 4.3 : Arazi kullanım biçimleri (Zielinska, Church ve Jankowski, 2008).

Sürdürülebilirlik ve kentsel arazi kullanım örüntüleri şehir planlamasında oldukça karmaşıktır. Bu çalışmada matematiksel bir programlama kullanılarak sürdürülebilirlik amaçlarını göz önünde bulunduran, erişilebilir gelişimin yönünü belirleyen bir model tanıtılmıştır. Buna ek olarak model, arazi yoğunluğu sınırlarına göre gelişimi yönlendiren bir sistem olarak da kabul edilebilir. Bir diğer yandan bu model çok analitik ve katı olarak yorumlanabilir ya da henüz tanımlı olmayan kentsel yerleşim problemleri için kısıtlayıcı olarak düşünülebilir; fakat mekânsal eniyileme modellerinin gelişen şehirlerin kentsel formunu yönlendirmede büyük rolü olduğu düşünülmektedir (Zielinska, Church ve Jankowski, 2008). Şekil 4.4'te modelle üretilen arazi kullanım örüntüleri verilmiştir.



Şekil 4.4 : Modelle üretilen arazi kullanım örüntüleri karşılaştırması (Zielinska, Church ve Jankowski, 2008).

Geliştirilen bu iki model sürdürülebilir tasarıma farklı ölçeklerden bakmaları nedeniyle örnek olarak seçilmiştir. Wang, Rivard ve Zimeureanu tarafından 2006 yılında geliştirilen Yeşil Bina Tasarımı İçin Plan Eniyileme Modeli, sürdürülebilir tasarım bağlamında iklim uygun bina biçimi ve yönelimi yanında minimum enerji tüketimi, doğal ışıktan maksimum faydalanma ve minimum maliyet ölçütlerini de göz önünde bulundurarak en uygun pencere tipi ve duvar malzemesi seçiminin yapılmasını sağlamaktadır. Bina biçimi yanında malzeme ve maliyeti göz önünde bulundurması açısından işlevsel bir model olarak tanımlanabilir; fakat erken tasarım evresinde birden fazla binadan oluşan kompleks sistemlerin araziye yerleşiminde yetersiz kalmaktadır. Binalarda enerji korunumuna odaklanarak, binaların birbiriyle ilişkileri, arazideki yeşil alanlar ve yaya yolları gibi elemanları da kapsayan bir arazi

yerleşimi göz ardı edilmiştir. Zielinska, Church ve Jankowski (2008) tarafından geliştirilen SMOLA, şehir ölçeğinde sürdürülebilirlik amaçları doğrultusunda arazi kullanım örüntüleri üretmektedir. Bina kullanım fonksiyonları homojen hücre tipinde renklerle gösterilmiştir, çelişen iki kriter olan maksimum açık alan ve maksimum kalkınma, komşu yerleşimlere uygunluk ve var olan yerleşim alanlarına yakınlık şartlarını göz önünde bulunduran çok amaçlı mekânsal eniyilemeyi hedeflemektedir. Çok amaçlı eniyileme metodunu kullanarak sürdürülebilir arazi yerleşim önerileri üreten model kent ölçeğinde çalışmakta, bina ölçeğinde arazi yerleşimini göz ardı etmektedir.

Tez kapsamında önerilen sürdürülebilir bina modeli sadece enerji etkin bina modeli olarak değil, eskiz aşamasından başlayarak bina ölçeğinde sürdürülebilir arazi kullanımına ve yeşil bina sertifika sistemlerinin ve yapı şartnamelerinin gereklerine öncelik sırasına göre değer veren bütünlük bir modeldir.

5. SÜRDÜRÜLEBİLİRLİK BAĞLAMINDA TOPLU KONUT YERLEŞİMİ TASARLAYAN EVRİMSEL BİR MODEL: SSPM

Tez kapsamında geliştirilen “Sürdürülebilir Yerleşim Planı Modeli (Sustainable Site Planning Model-SSPM)” iki farklı disiplin olan evrimsel algoritmalar ve sürdürülebilir tasarımın ara kesitinde yer almaktadır. LEED ve BREEAM yeşil bina sertifika sistemlerinin arazi yerleşim kriterleri, yerel yapı yönetmelikleri ve seçilen araziye ait iklimsel veriler, modelin uygunluk fonksiyonları olarak kabul edilmiştir. Pareto genetik algoritma tabanlı olarak kurgulanan modelde, NSGA-II (Bastırılmamış Sınıflandırılmalı Genetik Algoritma), kullanıcı tarafından sisteme girilen verilere göre arazi üzerine toplu konut yerleşim alternatifleri üretilmektedir.

Excel'den arazi verisini okuyan modelin yazılımında, Windows ortamında çalışabilmesi, nesneye dayalı olması ve çok gelişmiş bir grafik ortamı bulunması nedenleri ile Processing 2.1 programlama dili kullanılmıştır. Processing, 2001 yılından bu yana çevrimiçi görsel sanatlar odaklı geliştirilen Java üzerine kurulmuş bir programlama dili ve ortamıdır (Url-10).

5.1 Sürdürülebilir Yerleşim Planı Modeli: SSPM

Tez kapsamında önerilen SSPM modelinde, toplu konut tasarımının yapılacağı arazi matrisle bilgisayar ortamına tanımlanmaktadır. Arazi üzerindeki yerleşim alternatifleri pareto temelli NSGA-II algoritması kullanılarak üretilmektedir. Modelin aşamaları aşağıda açıklanmıştır:

Aşama I: Arazinin tanımlanması

Adım 1. Arazi verilerinin girilmesi,

Adım 1.a: Excel’de arazi matrisinin oluşturulması,

Adım 1.b: İklim türünün seçilmesi (a.Sıcak b.Soğuk),

Adım 1.c: TAKS-KAKS-hmax değerlerinin girilmesi,

Adım 1.d: İl yönetmeliğinde belirtilen yeşil alan yüzdesinin girilmesi,

Adım 1.e: Hâkim rüzgar yönünün girilmesi,

Adım 1.f: Manzara yönünün girilmesi,

Adım 2. Tanımlanan arazinin d boyutlu ($d = 1$ metre) ızgara sistem ile temsili;

Aşama II: Birinci Üretim

Adım 3. Üretim: Rastgele ilk yerleşim popülasyonun (N) üretilmesi;

Adım 4. Üretilen arazi bireylerinin gösterilmesi;

Aşama III. Birinci Değerlendirme, Birinci Seçim

Adım 5. Değerlendirme: Popülasyondaki her çözüm kümesi için 5 tane uygunluk fonksiyonu (emsal, taban alanı, iklim, rüzgar, manzara) puanının hesaplanması;

Adım 6. Uygunluk puanları verilen bireylerin toplam skorlarının 0 ile 1 arasında normalize edilmesi (1 en başarılı, 0 en başarısız bireyler) ve turnuva seçilimi uygulanması;

Aşama IV: Çaprazlama ve Mutasyon

Adım 7. Seçilen ebeveyn çiftlerine çaprazlama işleminin uygulanması;

Adım 8. Yeni bireylere mutasyon işleminin uygulanması;

Aşama V. İkinci Değerlendirme, İkinci Seçim

Adım 9. Yeni neslin, ebeveyn havuzuna eklenmesi ($2N$) ve pareto üstünlüklerine dayalı olarak bastırılmamış sınıflandırmaya göre (NSGA-II) sıralama yapılması;

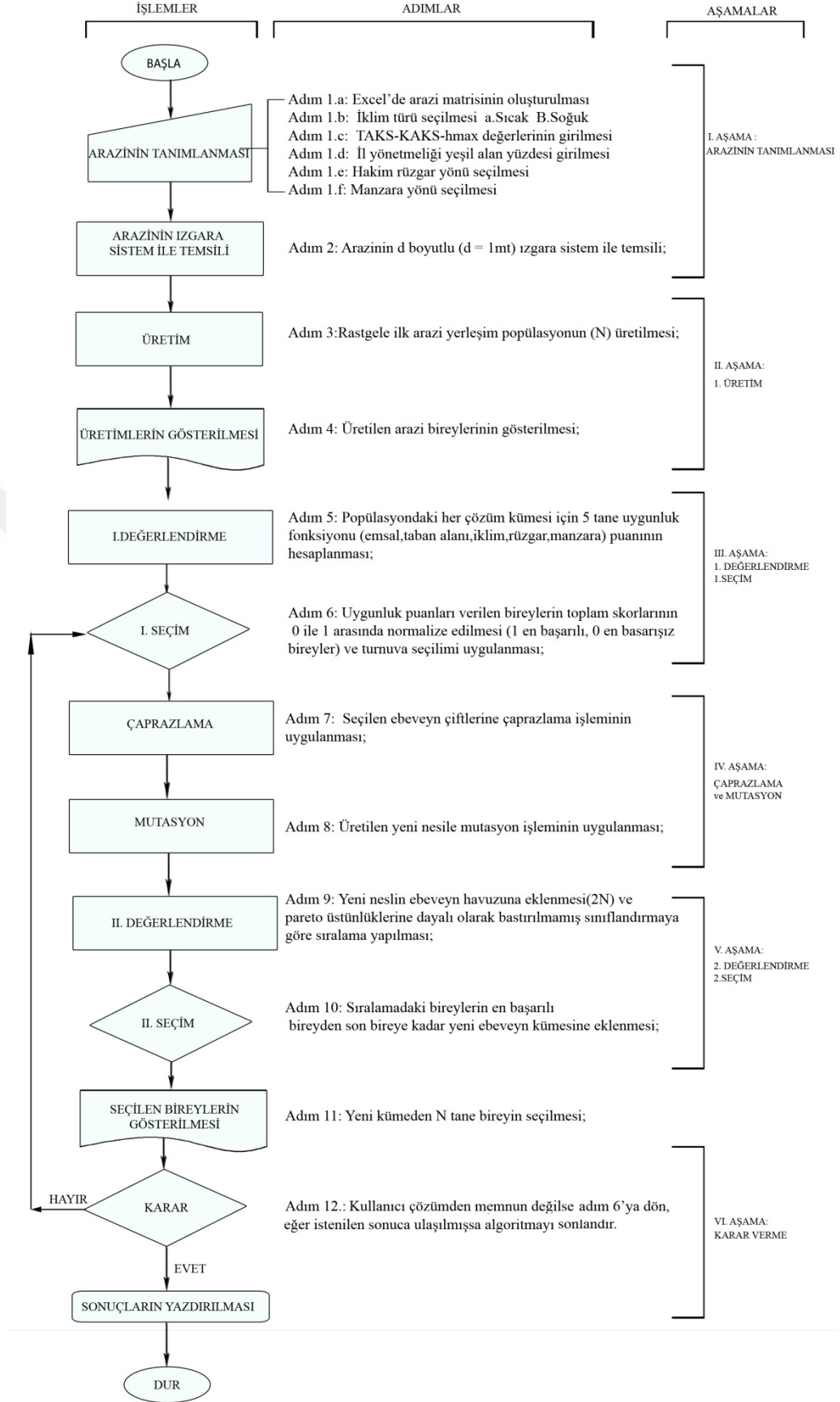
Adım 10: Oluşturulan bu sıralamadaki bireylerin en başarılı bireyden son bireye kadar yeni ebeveyn kümesine eklenmesi;

Aşama VI. Karar Verme

Adım 11: Yeni kümeden N tane bireyin seçilmesi;

Adım 12: Döngü Testi: Kullanıcı çözümden memnun değilse Adım 6'ya dönülmesi, eğer istenilen sonuca ulaşılmışsa algoritmanın sonlandırılması ve sonuçların yazdırılması.

Modelin, bu adımlarını içeren akış diyagramı Şekil 5.1'de görülmektedir.



Şekil 5.1 : Modelin akış diyagramı.

5.2 Arazinin Tanım Tekniđi

Lorem Modelin ilk adımı seçilen arazinin sayısal olarak tanımlanmasıdır. Toplu konut yerleşmesinin tasarlanacağı arazi, matrisle tanım tekniđi kullanılarak temsil edilmiş ve matrisler Excel'de oluşturulmuştur. Arazinin tanımlanmasında Excel programı, veri girişinin pratik ve kolay tanımlanabilir olması ve matrisle tanım tekniđine olanak sağlaması nedeniyle tercih edilmiştir. Matrisin satırlarını ve sütunlarını oluşturan her bir elemanı (hücre), arazinin alanı 1 m² olan bir birimini temsil etmektedir. Arazi köşe noktalarının ve arazi üzerindeki mevcut elemanların x,y koordinatlarına göre matris oluşturulmaktadır.

Her bir Excel hücre koordinat sisteminde bir nokta olarak kabul edilmektedir. Excel'de R1C1 başvuru stili kullanılarak x ve y eksenlerinin sayılarla temsil edilmesi, arazi köşe noktalarının ve arazideki mevcut elemanların koordinatlara göre işlenmesini kolaylaştırmaktadır. Arazinin belirli bir formu olmaksızın tüm girinti ve çıkıntılarıyla modele tanıtılmasına olanak sağlayan bu yöntem, arazi topoğrafyasının da hücresel olarak modele tanıtılmasına olanak sağlamaktadır. Bu sınırlar içerisinde kalan birim kareler, fonksiyonlarına göre numaralandırılarak arazi sayısal olarak bilgisayara tanımlanmaktadır. Ayrıca her hücre fonksiyonlarına göre renklerle gösterilerek, görsel okunabilirlik de sağlanmıştır. Aşağıda, bu sayısal değerlerin temsil ettiği fonksiyonlar açıklanmıştır:

0: Arazi sınırı dışındaki fonksiyonu olmayan boş alan: Hiç bir fonksiyonu yoktur; beyaz renk ile temsil edilir. Arazi köşe koordinatları dışarısında kalan ve Excel çalışma alanını sınır olarak kabul eden birim kareler "0" ile gösterilmektedir, bu şekilde girinti çıkıntılı sınırları olan her türlü arazi tipi kolaylıkla modele tanıtılmaktadır.

1: Arazi sınırları içerisindeki fonksiyonu olmayan boş alan: Farklı fonksiyonlara (konut, yeşil alan) dönüşebilir; bej renk ile temsil edilir.

2: Arazideki korunacak mevcut ağaçlar: Yerleri sabittir, değişmez. Koyu yeşil ile temsil edilir.

3: Üretimden sonra fonksiyonsuz kalan boş hücre.

4: Taşıt yolu hücresi: Gri ile temsil edilir. Korunacak yol yoksa excelde oluşturulan arazide ve modelde yer almazlar.

5: Otopark hücresi: Koyu gri ile temsil edilir. Korunacak otopark yoksa excelde oluşturulan arazide ve modelde yer almazlar.

6 : Yaya yolu hücresi: kahverengi ile temsil edilir. Korunacak yaya yolu yoksa excelde oluşturulan arazide ve modelde yer almazlar.

7: Mevcut anayol hücresi: Arazi sınırları dışarıındadır ve siyah ile temsil edilir.

8: Korunacak su hücresi: Yeri sabittir, değişmez. Mavi renk ile temsil edilir.

9: Korunacak yeşil alan hücresi: Yeri sabittir, değişmez. Açık yeşil ile temsil edilir.

10: Daha önceden kullanılmış kirlilik olan alan hücresi: Sarı ile temsil edilir; bu hücrelere konut yerleşimi önceliklidir.

11: Mevcut toplu taşıma noktası hücresi: T harfi ile temsil edilir, çok büyük ölçekli yerleşimlerde toplu taşıma noktasına yakınlık önceliklidir.

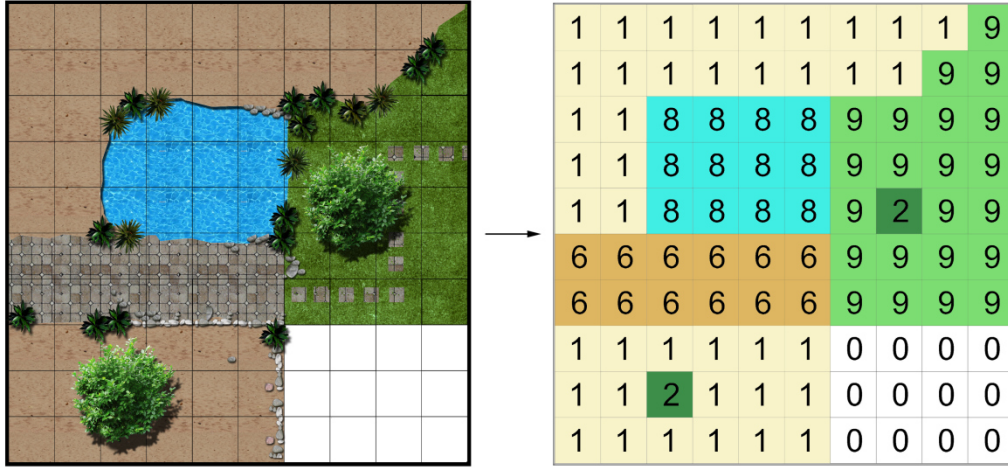
12: Mevcut gürültü kaynağı hücresi: G harfi ile temsil edilir. Çok büyük ölçekli yerleşimlerde gürültü kaynağından uzak olma önceliklidir.

13: Kot farkından dolayı toprak altında kalan birimler: Şeffaf gri ile temsil edilir.

20: Bahçeler: Yan ve arka bahçe hücrelerini temsil eder, model tarafından üretilirler. Yeşil renk ile gösterilirler.

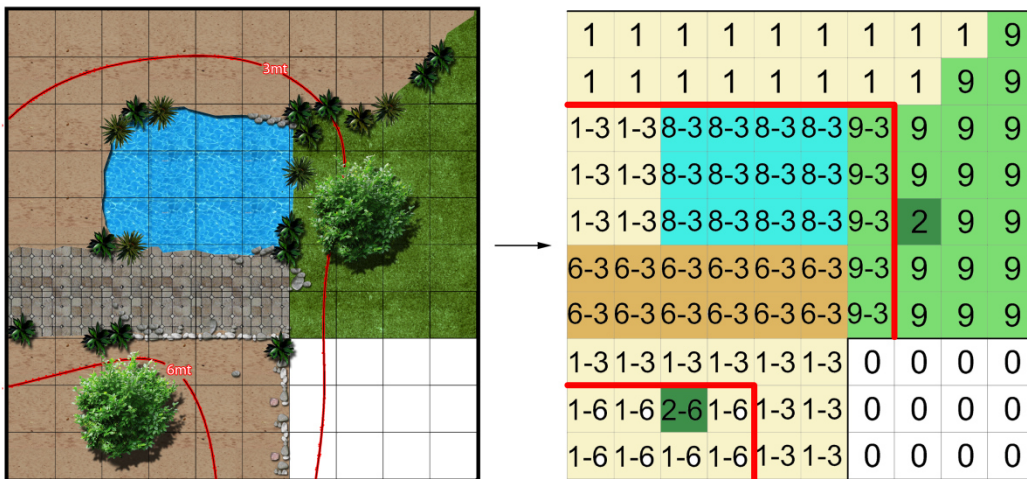
20 üstü: Bina hücresi: Model tarafından üretilen bina hücrelerini temsil eder. Hücrenin üretilen binanın kat verisini göstermesi için çift haneli gösterim tercih edilmiştir. Örneğin, üretilen tek katlı bina hücresi 21; 5 katlı bina hücresi 25 olarak model tarafından tanımlanacaktır, bu şekilde modelin daha hızlı çalışması sağlanmıştır. Kırmızı renk ile temsil edilirler.

Şekil 5.2'de kot farkı bulunmayan, Şekil 5.3' te ise 3 farklı kota sahip örnek arazi matrisleri oluşturulmuştur.



Şekil 5.2 : Örnek arazinin Excel'de matrisle tanımlanması.

Arazinin topografik yapısını ve doğal oluşumları göz ardı etmeden, çözümün gerçekçi sonuçlar verebilmesi için, arazi verilerinin de gerçeğe yakın modellenmesi gerekmektedir. Bunu sağlamak için sayısal araziye eldeki eğim değerlerine göre 3 boyutlu karakter kazandırılmıştır. Yükselti farklarının çözümünde, araziden gelen eğim değerleri kullanıcı isteğine göre farklı kotlardaki parçalara bölünerek modele tanıtılmaktadır. Bu sayede yüksek kot farkına sahip kademeli arazilerde teraslama yapılarak, topografya ile uyumlu arazi yerleşim çözümlerine gidilebilir. Arazinin en düşük kotu sıfır olarak kabul edilerek istenen kotlarda teraslama yapılabilecektir. Farklı kotlardaki bu arazi parçalarının sayısal olarak tanımlanmasında, hücre numarası yanında, sıfır kabul edilen kota göre yükseklik farkı sayısal olarak yazılır.



Şekil 5.3 : Excel'de tanımlanmış 0m-3m-6m kot farklı örnek arazi parçası matrisi.

Örneğin Şekil 5.3'te 1 ile numaralandırılan boş alan hücreleri, sıfır kabul edilen kottan 3 metre yükseklikte ise, hücre numarası 1-3, 6 metre yükseklikte ise 1-6 olarak numaralandırılmıştır. Bu şekilde her hücrenin kotu için görsel okunabilirlik sağlanmıştır. SSPM modeli, her farklı şekil ve topoğrafyaya sahip arazi için hazırlanabilecek excel dosyası yardımıyla, aynı zamanda bir 3 boyutlu topoğrafya tanım modeli olarak da görev yapabilen bütünleşik bir modeldir.

Model araziye oluşturan her birim karenin fonksiyon ve kot verisini hazırlanan excel dosyasından okuyarak araziye iki ayrı matris olarak tanımlar. Bu matrisler, yazılımda "*site height*" ve "*site function*" olarak adlandırılmıştır. "*site height*" arazideki her hücrenin yükseklik bilgisinin yer aldığı matris, "*sight function*" ise arazideki her hücrenin fonksiyon bilgisinin yer aldığı matris olarak tanımlanmıştır.

5.3 Modelin Kullanıcı Arayüzü

Arazinin fonksiyon numaralarına göre Excel'e işlenmesinden sonra, model bu veriyi okuyarak sayısal araziye görsel olarak üretmektedir. SSPM modelinin kullanıcı arayüzü de Processing ile hazırlanmıştır. Şekil 5.4'te verilen modelin ara yüzünde kullanıcının belirleyeceği değerler;

- Bina kütle tipi sayısı,

Arazi yerleşim planındaki bina kütlesi ölçüleri iki farklı şekilde belirlenmektedir. Model tarafından belirlenen rastgele boyutlarda, ya da boyutları kullanıcı tarafından belirlenen üç tip olarak üretilmektedir. Model kullanıcıya "rastgele, tek tip, iki tip, üç tip" olmak üzere farklı seçenekler sunar. Elde var olan bina tiplerine göre üretim yapılması ya da belirli tiplere bağlı kalmadan farklı ebatlarda blok üretimleri yapılması sağlanır. Kullanıcı arayüzünden girilen X,Y, Z değerleri bina kütlelerinin en, boy, yükseklik değerleridir.

- Manzara yönü;
- Hakim rüzgar yönü;
- İklim türü (soğuk, sıcak);
- Yerel yapı yönetmeliklerinden gelen maksimum inşaat alanı oranı (KAKS), toplam bina oturma alanı oranı (TAKS), ve maksimum kat yükseklik (hmax) değerleri;
- Yönetmelikte verilen gerekli açık alan yüzdesi;
- Üretilmesi istenen birey (binalarla kaplı arazi) sayısıdır.

SSPPM_v3

BİNA TİP SAYISI ▾

X Y Z

X Y Z

X Y Z

MANZARA ▾

RUZGAR ▾

IKLİM ▾

0.25 TAKS

1.00 KAKS

5 HMAX

50 YESİL

10 BİREYSAYISI

URET DURDUR

Şekil 5.4 : Verilerin girildiği kullanıcı ara yüzü.

Excel tablosu olarak tanımlanan araziye ait her bir hücre, genetik algoritmanın genlerini oluşturmaktadır. Her bir birey (çözümlerden biri) bu ızgara üzerindeki hücrelere yerleştirilerek farklı popülasyonlar üretilecektir. Arazinin matrisle tanımlanması ve kullanıcı arayüzünden gerekli veri girişinin yapılmasından sonraki aşama, pareto temelli bastırılmamış sınıflandırılmalı genetik algortimanın (NSGA-II) devreye girdiği aşamadır.

5.4 Modelin Değişkenleri

Modelde kullanılan değişkenler aşağıda açıklanmıştır:

site: Excelden okunan arazi verisi.

arazi: Modelde üretilen arazi yerleşimi bireyi.

siteheight: Yükseklik verisini gösteren arazi matrisi.

site function: Fonksiyon verisini gösteren arazi matrisi.

parselPozX: Binanın yan ve arka bahçe mesefelerinin dâhil edildiği bina parselinin x eksenindeki konumu.

parselPozY: Binanın yan ve arka bahçe mesefelerinin dâhil edildiği bina parselinin y eksenindeki konumu.

binaPozX: Üretilen bina kütlesinin X eksenindeki konumu.

binaPozY: Üretilen bina kütlesinin Y eksenindeki konumu.

binaXBoyut: Üretilen bina kütlesinin X eksenindeki hücre sayısı.

binaYBoyut: Üretilen bina kütlesinin Y eksenindeki hücre sayısı.

binaZBoyut: Üretilen bina kütlesinin Z eksenindeki hücre sayısı (bina yüksekliğini göstermektedir).

yanbahçeX: Üretilen bina kütlesinin yan bahçesinin X eksenindeki hücre sayısı.

yanbahçeY: Üretilen bina kütlesinin yan bahçesinin Y eksenindeki hücre sayısı.

arkabahçeX: Üretilen bina kütlesinin arka bahçesinin X eksenindeki hücre sayısı.

arkabahçeY: Üretilen bina kütlesinin arka bahçesinin Y eksenindeki hücre sayısı.

istenenTabanAlani: TAKS katsayısına göre hesaplanan maksimum taban alanı.

dolum2: Üretilen binaların toplam taban alanı.

dolum3: Üretilen binaların toplam inşaat alanı.

farkm2: Üretilen binaların toplam taban alanı değerinden istenen maksimum taban alanı değerinin çıkarılmasıyla elde edilen değer.

farkm3: Üretilen binaların toplam inşaat alanı değerinden maksimum inşaat alanı değerinin çıkarılmasıyla elde edilen değer.

taks: Toplam inşaat taban alanının arazi alanına oranı.

kaks: Toplam inşaat alanının arazi alanına oranı.

hmax: Maksimum bina yüksekliği.

yesil: Üretilen yeşil alan m^2 'si.

emsal: Maksimum inşaat alanı toplam m^2 'si.

caprazlamaNoktasi: Arazide çaprazlamanın gerçekleştirileceği nokta.

spaceCount: Arazideki toplam hücre sayısı(arazi toplam alanını) gösterir.

yon: Yön verisi (1:kuzey, 2:doğu, 3: güney, 4: batı).

score: Tüm uygunluk fonksiyonu puanları toplamı.

mutasyonTip: Uygulanacak mutasyon tipi.

mutasyonOranı: Mutasyonun etkinlik oranı.

5.5 Modelin Üretim ve Uygunluk Fonksiyonları Bağlamında Değerlendirme Sistemi

Genetik algorithmada her hücre bir gen olarak kabul edilmektedir. Başlangıçta rastgele birçok birey yaratılarak, bu bireyler popülasyon kümesi içerisinde toplanmaktadır. Başlangıç popülasyonunu oluşturduktan sonra bu bireyleri doğal seçilime sokarak eleme işlemi gerçekleştirilmektedir. Bu işlemde öncelikle tüm bireylere uygunluk fonksiyonu uygulanmaktadır. Uygunluk fonksiyonu kalıtsal algoritma tasarımlarının en önemli kısmıdır.

SSPM modelinin uygunluk fonksiyonları:

- Yerel yapı yönetmeliklerinden gelen taban oturma alanı, toplam inşaat alanı ve maksimum bina yüksekliği kontrolü;
- Fiziksel çevre kontrolü kriterleri olan iklim, rüzgar ve manzara kontrolü;
- LEED ve BREEAM arazi yerleşim kriterleri kontrolü

şeklindedir.

LEED ve BREEAM sertifika sistemleri her ne kadar tüm dünyada kullanılmakta olsalar da, Amerika ve İngiltere'nin coğrafi, ekonomik ve kültürel şartları ile o ülkelere özgü şartnameler doğrultusunda hazırlandıkları için diğer ülkelerde kullanılmasında zorluklar yaşanmaktadır. Bu bağlamda yeşil bina tasarım kriterleri olarak sadece sertifika sistemlerini ele almak yerine, tasarım arazisinin bulunduğu bölgenin yerel düzeyde yapı yönetmeliklerinin ve iklimine göre değişken kriterlerin de göz önünde bulundurulması önemlidir.

5.5.1 Modelin değerlendirme kriterleri

Sürdürülebilir bir arazi yerleşme planı, tüm bu sürdürülebilir bina kriterlerinin kavramsal tasarım aşamasında ele alınmasıyla gerçekleştirilebilir. Bu uygunluk kriterleri arazi kullanımı Şekil 5.5'te bina yerleşimi ve yönlmesi, bina formu başlıkları altında sınıflandırılmıştır.



Şekil 5.5 : Sürdürülebilir arazi yerleşim modeli ana kavramları.

Kavram 1: Arazi Kullanımı

Yerleşim yeri ve dokusunun belirlendiği bu aşamada binanın çevresiyle ilişkisi ele alınmaktadır. İlk olarak üzerinde durulması gereken konu, yapılacak olan düzenlemelerin arazinin var olan ekolojik değerine zarar vermemesini sağlamak olacaktır. Yerleşim planı oluşturulurken, arazinin ekolojik değerini korumak amacıyla kullanılacak uygunluk fonksiyonları şu şekildedir:

Kural 1.1. Arazide bulunan bitki ve hayvan türleri, su kaynakları korunmalıdır (Url-2). Modele, arazi sınırı köşe koordinatları ile tanıtılarak korunması gerekli elemanlar (ağaçlar, bitkiler ve su kaynakları) koordinatlarıyla belirlenmelidir. Model bu noktalara işlev yüklemeyecektir, açık alan olarak tanımlayacaktır.

Bu kuralın sağlanabilmesi için arazi matrisi üzerinde bulunan ağaçlar 2 ile, su kaynakları 8 ile, yeşil alanlar da 9 ile temsil edilir. Binaların üretilmesi sürecinde arazi matrisinde bu rakamlarla temsil edilen yerlerde üretim gerçekleştirilmez.

Kural 1.2. Mevcut doğal su kaynağına 30 m, sulak alana ise 15 m yapılaşma sınırı vardır (Url-1).

Modelde su kaynağı olan alanların çevrelerine 30 metre sınırı boyunca üretime imkân tanımayan "0" değeri verilir.

Kural 1.3. Korunacak ağaçlar için en az 25 m² alan ayrılmalıdır.

Modelde mevcut ağaç hücresi çevresindeki komşu 2 sıra hücrenin yeşil alan olarak kabul edilmesiyle sağlanacaktır.

Kural 1.4. Arazi yönetmeliğinde verilen değerin %25'inden fazlası açık alan olmalıdır (Url-2).

Kural 1.5. Arazi yönetmeliğinde açık alan belirtilmemişse arazinin %20 si açık alan olmalıdır (Url-2).

Kural 1.6. Herhangi bir yönetmelik oranı olmayan yerde bina oturma alanı kadar yeşil alan bırakılmalıdır (Url-2).

1.4, 1.5 ve 1.6 numaralı kurallar açık alanı tanımlar. Açık alan değeri bu modelde eniyilemesi yapılacak değerlerden bir tanesidir.

Kural 1.7. Arazide daha önceden yerleşim yapılmış ise, kirletilmiş alanların yeniden kullanımı sağlanmalı; bina metrekaresi eniyilemesi buna göre şekillenmelidir. Yeni yapılacak binanın oturma alanının % 75 'i arazinin son 50 yıl içerisinde endüstriyel, ticari ya da konut işlevli olarak yapılaşmış olan bölgesinde olmalıdır. Daha önce rekreasyon alanı, park, tarım arazisi, ormanlık olan arazilerde bu kriter geçerli değildir. Yerleşimden önce tüm negatif etkilerden arındırmak amacıyla, kirlilik içeren alanlara yerleşme öncelikli olmalıdır.(Url-2).

Bu kuralı gerçekleştirebilmek için "arazi" matrisinde üretim yapılmadan önceki "kirli alan"ı temsil eden 10 numaralı hücrelerin toplam sayısı alınır, üretimden sonra bu hücrelerden kaçının doldurulduğu kontrol edilir.

Kural 1.8. Önceden bina olan bir alanda, ilk binanın oturma alanı dışında kalan arazinin %50'sini korumak ya da tüm arazinin %20' sini korumak gereklidir (Url-1).

Bu kuralı gerçekleştirmek için arazide daha önceden yerleşim olup olmadığının tespit edilir. Eğer varsa arazide boş alan olarak 1'le tespit edilen yerlerin %50'si ya da tüm arazinin %20'si üretimden sonra boş kalmalıdır.

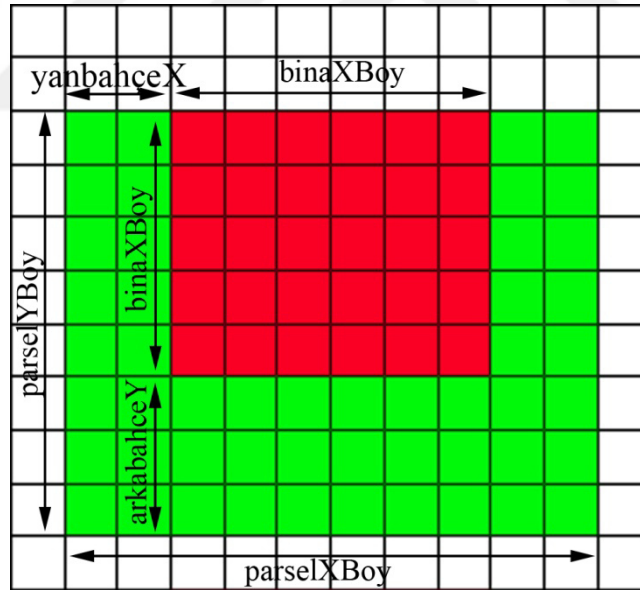
Kural 1.9. Çevrede sese hassas yerleşimler varsa bunlara minimum 800 m çaplı mesafede yerleşim yapılmalıdır (Url-2).

Bu kuralı gerçekleştirebilmek için gürültü kaynağı hücresi çevresine 800 metre boşluk bırakacak şekilde 3 numaralı boşluk hücresi konulur, bu algoritma büyük çaplı arazilerde kullanılabilir.

Kavram 2: Bina Yerleşimi ve Yönlenmesi

Sürdürülebilir tasarım bulunduğu çevre ile bir bütün olmalıdır. Yapının çevre ve doğa ile bağlantısı üzerinde bulunduğu arazi, üzerine oturduğu toprak ile sağlanır. Yapının arazi üzerine doğru şekilde konumlandırılması ve yönlendirilmesi arazi özelliklerinin incelenip analiz edilmesiyle mümkündür. Arazi içerisinde ekolojik değere bağlı değerlendirmeler yapıldıktan sonra bina ya da binaların yerleri ve birbirlerine göre konumları belirlenirken, binaların yapılacağı bölgedeki iklim tipi ve binaların çevrelerinde oluşturacakları gölge, rüzgar ve gürültü etkilerinin yanı sıra imar yönetmeliğinden gelen kurallar da dikkate alınmalıdır.

Model kapsamında üretilecek olan sosyal konutlar ayırık nizam olarak kabul edilmiştir. Hiçbir cephesinden komşu parselle bitişik olmadan inşa edilen yapı tarzına İstanbul İmar Yönetmeliği'nde "Ayrık Nizam" denmektedir (Url-11). Bu çalışmada kat yüksekliği 3.00 metre olarak kabul edilmiş ve yüksekliği 60.50 metreden az olan ayırık nizam binalar için gerekli yönetmelik kararları esas alınmıştır.



Şekil 5.6 : Binanın arkadan ve yandan yaklaşma mesafelerine karşılık gelen bahçe, parsel ve bina boyutu tanımlamaları.

Kural 2.1. Ayırık inşaat nizamına tabi yerlerde; 5 kata kadar yapı yaklaşma mesafesi yollardan en az (5.00) m., komşu parsellerden en az (4.00) m. olup 5 kattan sonraki her ilave kat için yan komşu bahçe mesafelerine (0.50) m. eklenir (İstanbul İmar Yönetmeliği) (Url-11).

$$yanbahçeX = 4 + [binaZBoyut - 5] \times 0.5 \quad (5.1)$$

Kural 2.2. 5 kata kadar (5 kat dâhil) olan binalarda arka bahçe mesafeleri en az 5.00 m'dir. 5 kattan sonraki fazla her kat için bu bahçe mesafeleri 1,00 m artırılır (İstanbul İmar Yönetmeliği) (Url-11).

$$arkabahçeY = 5 + [binaZBoyut - 5] \times 1 \quad (5.2)$$

Kural 2.1 ve Kural 2.2 'de binanın yerleşimi kuzey-güney ve doğu-batı yönlerine göre değiştiğinde aranan bahçe mesafesi değerleri, yanbahçeY ve arkabahçeX olarak değişmektedir.

Kural 2.3. Soğuk iklim bölgelerinde rüzgârdan korunmak için önlemler alınırken, sıcak ve nemli iklim bölgelerinde rüzgârın serinletici etkisinden mümkün olduğunca fayda sağlamak amaçlanmalıdır. Soğuk iklim tipi için kompakt formlar tercih edilmeli, rüzgâr yönüne dar cephesi bakmalıdır. Sıcak iklimlerde rüzgârın dik ve hava giriş-çıkış açıklıklarının geniş olduğu durumlarda doğal havalandırmadan en yüksek oranda yararlanılabilmektedir (TS825) (Url-12). Seçilen iklim tipine göre binaların rüzgar puanı hesaplanır. (Şekil 5.7)

Kural 2.4. Manzaraya yönelim önceliklidir. Üretilen bina kütlelerinin manzara puanı, manzara yönüne bakan hücre sayısı ile hesaplanır. (Şekil 5.4)

Kavram 3: Bina Formu

Bu aşamada önemle üzerinde durulması gereken konu, binanın yapılacağı bölgenin iklim özelliklerine uygun şekilde tasarlanmasıdır. Biçimlenme, binanın enerji performansı üzerinde büyük etkiye sahip faktörlerden biridir. Bu nedenle, bina tasarımında ana kriter, ısı kazanımının ve/veya kaybının bina kabuğu tarafından kontrolü, ana kriteri oluşturmaktadır. Binanın iklime bağlı olarak güneş ışınlamına ve dış havaya açılımı, yapının dış yüzey alanının hacmine göre değişim göstermektedir (Hawkes, McDonald, Steemers, 2002).

Kural 3.1. Bina yüksekliği, arazi için verilen maksimum bina yüksekliği değerini geçemez (İstanbul İmar Yönetmeliği) (Url-11).

Bina yüksekliği modelde her zaman alt değer ve üst değer olarak korunur. Alt değer 1'dir ve üst değer de model arayüzünden girilen arazinin hmax değeridir. Yapılan her üretim bu değerler arasında gerçekleşir.

Kural 3.2. İmar planı veya bu yönetmelikte belirlenen ön bahçe ve bahçe mesafelerinin belirlendiği yaklaşma sınırları içerisinde kalmak şartı ile ayrık nizamda yapı derinliği 30.00 m.'den fazla, taban alanı (TA) 600 m²'den büyük olamaz (İstanbul İmar Yönetmeliği) (Url-11).

$$binaXBoyut \times binaYBoyut \leq 600 \quad (5.3)$$

Kural 3.3. Ayrık nizama tabi yapılarda bina cephesi minimumu 6.00 m, en fazla 30.00 m' dir (İstanbul İmar Yönetmeliği) (Url-11).

$$6 \leq binaXBoyut, binaYBoyut \leq 30 \quad (5.4)$$

Kural 3.4. İki önemli kriter KAKS ve TAKS olarak gösterilebilir. Belediye tarafından belirlenen kadastral parsel içerisinde KAKS ve TAKS kriterlerine uygun bir yerleşim, planlama şartlarına uyan sürdürülebilir arazi tasarımı için önemli bir şarttır. Taban alanı kat sayısı (TAKS); taban alanının imar parseli alanına oranıdır. İmar planlarında aksine bir açıklama bulunmadığı takdirde ayrık nizama tabi arsalar ile imar planında yapı nizamı belirtilmeyen durumlarda binaların taban alanı, iç bahçe alanı dâhil %40'ı geçemez (İstanbul İmar Yönetmeliği) (Url-11).

$$taks = dolum2 / spaceCount \leq \%40 \quad (5.5)$$

$$kaks = dolum3 / spaceCount \quad (5.6)$$

Kural 3.5. Bina dış yüzeylerini büyötmek ısı kaybını da o oranda artıracaktır. İstenmeyen ısı kazanımı ve/veya kaybını önlemek için, binanın dış yüzey alanının hacme oranı mümkün olduđu kadar düşük olmalıdır. Sıcak iklimlerde yüzeylerde oluşabilecek ısı kayıplarını arttırmak amacıyla parçalı ve dış cephe alanı fazla bina formları kullanılmalıdır. Soğuk iklimlerde ise bina dış cephelerinde oluşabilecek ısı kayıplarını önlemek açısından dış cephe alanı azaltılmalıdır (TS825) (Url-12).

Soğuk iklimde binanın daha az ısı kaybı olması için alan çevre ilişkisi, maksimum alan ve minimum çevre şeklinde olmalıdır. Değişkenler Şekil 5.6'da gösterilmiştir.

Soğuk iklimde maksimum alan minimum çevre için kullanılan toplam formülü;

$$\sum_{i=0}^{tüm\ binalar} = binaXBoyut \times binaYBoyut \div 2(binaXBoyut + binaYBoyut) \quad (5.7)$$

Sıcak iklimde maksimum çevre minimum alan için kullanılan toplam formülü ;

$$\sum_{i=0}^{tüm\ binalar} = 2(binaXBoyut + binaYBoyut) \div binaXBoyut \times binaYBoyut \quad (5.8)$$

5.5.2 Modelin uygunluk puanları

Model ilk olarak rasgele populasyon oluşturur. Bu aşama genetik algoritmanın devreye girdiği kısımdır. Üretilen arazi bireyleri 5 ana uygunluk fonksiyonuna göre puanlanmaktadır:

1. Uygunluk Puanı: Taban alanının kullanıcı arayüzünden girilen değere yakınlığına göre bireylere puan verilir. Eğer bireyin toplam taban alanı girilen taks katsayısı ile elde edilen maksimum taban alanı değeriyle eşit ise puanı 1; fazla ya da eksikse 0 -1 aralığında bir değer alır.

$$0 \leq fit1 \leq 1$$

$$farkm^2 = dolum^2 - istenenTabanAlani \quad (5.9)$$

Üretilen bir bireye ait toplam bina oturum alanı ($dolum^2$) istenen taban alanı değerini geçiyorsa, $farkm^2$ 0'dan büyük bir değer çıkar ve bu bireyin alacağı taban alanı uygunluk puanı ($fit1$);

$$farkm^2 > 0 \quad fit1 = 1 - (farkm^2/600) \quad (5.10)$$

olarak hesaplanır. Eğer üretilen bireye ait toplam bina oturum alanı ($dolum^2$) istenen taban alanı değerinden küçükse, $farkm^2$ 0 dan küçük bir değer çıkar ve bu bireyin alacağı taban alanı uygunluk puanı($fit1$);

$$farkm^2 < 0 \quad fit1 = 1 + (farkm^2/1000) \quad (5.11)$$

olarak hesaplanır.

Bu formülden dolayı toplam taban alanı yönetmelik değerinden fazla olan bireyler daha kötü puan alarak seçim şansları düşmektedir.

2. Uygunluk Puanı: Emsal değerinin kullanıcı arayüzünden girilen değere yakınlığına göre bireylere puan verilir. 1.uygunlukta yapılan işlemin aynısı Z boyutu hesaba katılarak yapılır.

$$0 \leq fit2 \leq 1$$

$$farkm^3 = dolum^3 - emsal \quad (5.12)$$

Üretilen bireye ait toplam inşaat alanı ($dolum^3$) istenen maksimum emsal değerini geçiyorsa, $farkm^3$ 0'dan büyük bir değer çıkar ve bu bireyin emsal uygunluk puanı ($fit2$);

$$farkm^3 > 0 \quad fit2 = 1 - (farkm^3/1000) \quad (5.13)$$

olarak hesaplanır. Eğer üretilen bireye ait toplam bina oturum alanı (dolum³) istenen taban alanı değerinden küçükse, farkm³ 0'dan küçük bir değer çıkar ve bu bireyin alacağı emsal uygunluk puanı (fit2);

$$farkm^3 < 0 \quad fit2 = 1 + (farkm^3/2000) \quad (5.14)$$

olarak hesaplanır. Bu formülden dolayı toplam emsali, yönetmelik değerinden fazla olan bireyler daha kötü puan alarak seçim şansları düşmektedir.

3. Uygunluk Puanı: İklim'e göre bina boyutlarının oranları göz önünde bulundurularak hesaplanır. Soğuk iklimde;

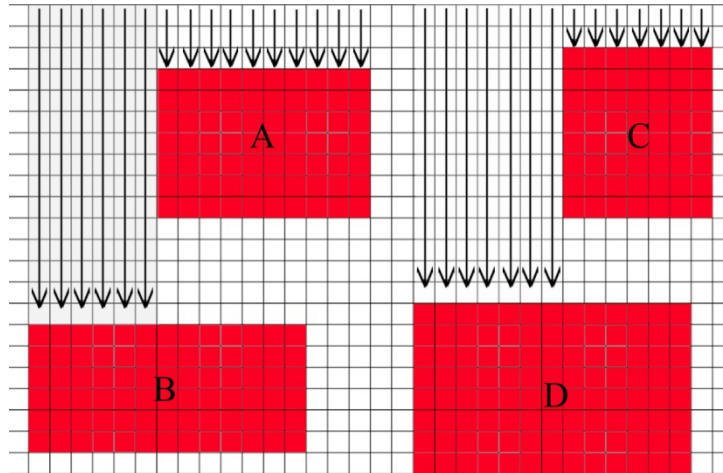
$$fit3 = (binaXBoyut \times binaYBoyut) \div 2(binaXBoyut + binaYBoyut) \quad (5.15)$$

olarak hesaplanır. Sıcak iklimde ise;

$$fit3 = 2(binaXBoyut + binaYBoyut) \div (binaXBoyut \times binaYBoyut) \quad (5.16)$$

olarak hesaplanır.

4. Uygunluk Puanı Üretilen bina kütesinin rüzgâr puanı, rüzgâr alan hücre sayısı ile hesaplanır. Örneğin Şekil 5.7'deki arazi yerleşim parçasında rüzgar puanları; A: 10, B: 6, C:7, D:7 olacaktır, B ve D binalarının rüzgarları A ve C binalarıyla kesilmektedir. Seçilimde önemli olan arazide yer alan tüm binaların toplam puanıdır, üretilen arazi popülasyonunda en yüksek toplam manzara puanına sahip olan arazi 1, en düşük toplam rüzgâr puanına sahip olan arazi ise 0 puan alır. Popülasyondaki diğer arazi bireyleri 0 ve 1 arasında sıralamalarına göre puanlanırlar.



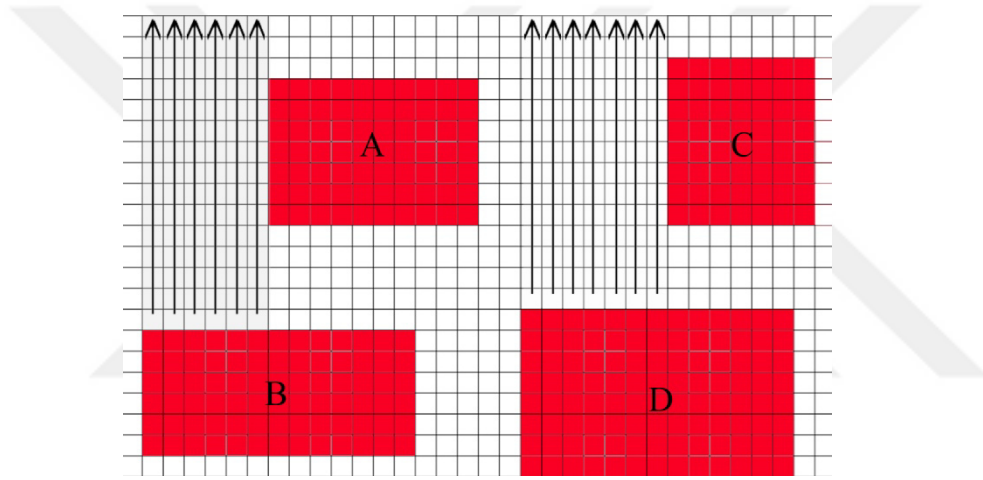
Şekil 5.7 : Rüzgâr puanının gösterimi.

Eğer iklim tipi soğuk iklimse, binaların rüzgâr alması tercih edilmeyecektir. Örneğin sıcak iklimde, en çok rüzgâr alan arazi bireyi 200 puanlı, en az rüzgar alan arazi

bireyi 15 puanlı ise, 200 puan toplayan arazi bireyinin ana puanı 1, 15 puanlı arazi bireyinin puanı 0 olacaktır. Eğer arazi sıcak iklimde ise, 200 puan toplayan bireyin ana puanı 0, 15 puan toplayan bireyin ana puanı 1 olacaktır.

$$\sum_{i=0}^{binasayın} binaRüzgarPuanı = AraziRüzgarPuanı \quad (5.17)$$

5. Uygunluk Puanı: Her bina için manzara yönü grid üzerinden manzarayı en çok gören bireyin en fazla puanı alması prensibine göre hesaplanır. Üretilen bina kütlelerinin manzara puanı, manzara yönüne bakan hücre sayısı ile hesaplanır. Eğer manzara puanı hesaplanacak binanın karşısında başka bir bina varsa, görüşün engellenmediği kenar hücre sayısı kadar manzara puanı elde edilir.



Şekil 5.8 : Manzara puanının gösterimi.

Şekil 5.8'de A ve C binaları, B ve D binalarının manzarasını kesmektedir. Bu örnekte binaların manzara puanları A: 10, B:6, C:7, D:7 'dir. Seçilimde önemli olan arazide yer alan tüm binaların toplam puanıdır.

$$\sum_{i=0}^{binasayın} binaManzaraPuanı = AraziManzaraPuanı \quad (5.18)$$

5.5.3 Modelin veri girişi, üretim ve seçim sistemi

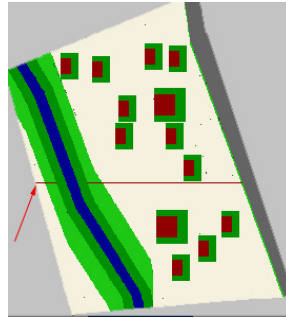
Uygunluk puanları verilen bireylerin toplam skorları 0 ile 1 arasında normalize edilir; 1 en başarılı bireylerdir, 0 en başarısız bireylerdir; bu aşamada turnuva seçilimi uygulanır. Turnuva seçiminde skoru yüksek olan bireylerin üreme şansı daha fazladır. Bireyler kendi skorlarına göre oluşturulan matrizen seçilirler. Bu seçilimi piyangoya benzetebiliriz. Örneğin, 3 bireyli bir durumda 1. Bireyin skoru 0.92, 2.

Bireyin skoru 0.56, 3. Bireyin skoru da 0.23 olsun. Bu değerler arasında seçim yapılıncı 1. Bireyin çıkma olasılığı $92/171 = \%54$, 2. Bireyin çıkma olasılığı $56/171 = \%33$ ve 3. Bireyin çıkma olasılığı $23/171 = \%13$ tür. Bu olasılıklara göre iki birey alınır ve birbirlerine göre sonuçları karşılaştırılır, skoru yüksek olan birey seçilir.

Uygunluk değeriyle doğru orantılı olarak kopyalanan bireyler rastgele ikili gruplara ayrılarak ebeveyn olarak kabul edilir ve bunlara kendi aralarında çaprazlama (gen değişimi) işlemi uygulanır.

Bunun için, ebeveyn olarak seçilen bireylerden ilkinin rastgele olarak belirlenecek sıra ve sayıdaki geni ile, diğer ebeveynin aynı sıra ve sayıdaki genleri karşılıklı yer değiştirilir ve iki yeni birey (çocuk) elde edilir. Mevcut gen havuzunun potansiyelini artırmak üzere, bir önceki kuşaktan daha nitelikli yeni diziler oluşturmak için, çaprazlama işlemi uygulanır.

Çaprazlama işlemi seçilen iki bireyin ortasından rastgele alınan bir noktadan karşılıklı olarak çaprazlama yapılarak gerçekleştirilir. Uygun bir çaprazlama noktası seçildiğinde ebeveynlerden uyum değeri daha yüksek diziler elde edilebilmektedir. Fakat uygun çaprazlama noktası genellikle bilinmediğinden y ekseninde rastsal bir nokta seçilmiştir. (Şekil 5.9)



Şekil 5.9 : Modelde verilen çaprazlama noktası.

Sonraki nesillerde sürekli benzer bireylerin ortaya çıkmasını önlemek amacıyla çaprazlama işlemi sonrası oluşturulan ara popülasyon, mutasyon işlemine tabi tutulur. Programda 4 mutasyon tipi vardır:

- 1-4 arasındaki mutasyonda seçilen bina sağa, sola, yukarı, aşağı oynatılır.
- 5-6 arasındaki mutasyonda seçilen binanın katsayısı artırılır ya da azaltılır.
- 7-10 arasındaki mutasyonda seçilen binanın X – Y değeri artırılır ya da azaltılır.
- 11 yön değiştirilir.

Mutasyon oranları genellikle düşük tutulmaktadır. Bunun nedeni, çaprazlama sonucu elde edilen uyum değeri yüksek dizileri kaybetmemektir.

Bu işlemler iterasyon sayısı kadar sürdürülür. Elde edilen son popülasyondaki uygunluk değeri en yüksek birey, en iyi birey olarak seçilir ve genetik algoritma tamamlanmış olur.

5.5.4 Modelin veri girişi, üretim ve seçim algoritması

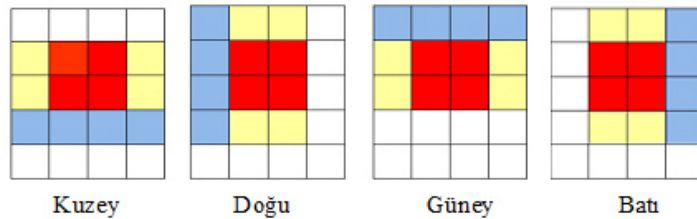
Model kullanıcının bina tipi tercihinine göre farklı olarak ilerlemektedir.

1. Eğer rastgele bina tipi tercih edilmiş ise, binanın yüksekliği rastgele olarak 1-hmax arasında atanır. (Kural 3.1) Eğer bina tip sayısı belirli ve boyutları girildiyse, binanın yüksekliği olarak kullanıcı ara yüzünden Z değeri alınır.
 2. Binanın X ekseninde genişliği 6m-30m olacak şekilde rastgele atanır. (Kural 3.3) Eğer belirli bina tip sayısı tercih edilmişse binanın X eksenindeki genişliği için kullanıcı arayüzünden X değeri alınır.
 3. Binanın Y ekseninde genişliği 6m-30 m olacak şekilde rastgele atanır (Kural 3.3). Eğer belirli bina tip sayısı tercih edilmişse binanın Y eksenindeki genişliği için kullanıcı arayüzünden Y değeri alınır.
 4. 2. ve 3. aşamalar rastgele bina tipi seçilmiş ise çarpım 600 m2den küçük olana kadar tekrar yapılır. (Kural 3.2)
 5. Binanın bakış yönü rastgele olarak atanır. (1:Kuzey, 2:Doğu, 3:Güney, 4: Batı)
- Örneğin, kuzeye bakan bina için;

Kırmızı: Bina;

Mavi: Arka bahçe;

Sarı: Yan bahçe olarak tanımlanabilir.(Şekil 5.10)



Şekil 5.10 : Bina yönlenmesine göre bina yaklaşma mesafelerine karşılık gelen yan bahçe, arka bahçe konumları.

6. Binanın yüksekliğine göre arka bahçe ve yan bahçe genişlikleri belirlenir.

a. Kuzey seçildiyse;

i. Bina kat sayısı 6' dan azsa yan bahçenin X boyutu 4 m ve arka bahçenin Y boyutu 5 m'dir. Kuzey yöneliminde yan bahçenin Y boyutu ve arka bahçenin X boyutu önemli değildir.

ii. Bina kat sayısı 6'dan fazlaysa;

$$yanbahceX = 4 + ((binakatsayısı - 4) \times 0.5) \quad (5.19)$$

Bu değer en yakın tam sayıya yuvarlanır.

$$arkabahceY = 5 + ((binakatsayısı - 5) \times 1) \quad (5.20)$$

b. Doğu seçildiyse;

i. Bina kat sayısı 6' dan azsa, yan bahçenin Y boyutu 4 m ve arka bahçenin X boyutu 5 m'dir.

ii. Bina kat sayısı 6'dan fazlaysa;

$$yanbahceY = 4 + ((binakatsayısı - 4) \times 0.5) \quad (5.21)$$

Bu değer en yakın tam sayıya yuvarlanır.

$$arkabahceX = 5 + ((binakatsayısı - 5) \times 1) \quad (5.22)$$

c. Güney seçildiyse;

i. Bina kat sayısı 6'dan azsa yan bahçenin X boyutu 4 m ve arka bahçenin Y boyutu 5 m'dir.

ii. Bina kat sayısı 6' dan fazlaysa;

$$yanbahceX = 4 + ((binakatsayısı - 4) \times 0.5) \quad (5.22)$$

$$arkabahceY = 5 + ((binakatsayısı - 5) \times 1) \quad (5.23)$$

d. Batı seçildiyse:

i. Bina kat sayısı 6'dan azsa yan bahçenin Y boyutu 4 m ve arka bahçenin X boyutu 5 m'dir.

ii. Bina kat sayısı 6'dan fazlaysa;

$$yanbahceY = 4 + ((binakatsayısı - 4) \times 0.5) \quad (5.24)$$

$$arkabahceX = 5 + ((binakatsayısı - 5) \times 1) \quad (5.25)$$

7. Binanın X ve Y boyutu ve bahçe genişlikleri bulunduktan sonra bu aşamada binanın parseline (parsel bina ve bahçelerin toplam kapladıkları alan) pozisyon atanır.

- a. Atanan değerlerin arazi matrisi içerisinde olması sağlanır;
- b. Bu aşamadan sonra binanın ve parselinin konumu, diğer bir deyişle arazi matrisinde kapladığı yer belli olur.

8. Binanın yerleşeceği konum müsait mi kontrol edilir; o alandaki yerlerin 1 (boş alan) değerinde olup olmadığına bakılır.

9. Eğer o alan boş ise parsel o alana yerleştirilir; değilse tekrar değer atanır.

10. Parsel başarılı bir şekilde yerine yerleştirilirse; taban alanında kaplanan yer kadar değer, toplam taban alanından çıkarılır;

11. Taban alanındaki gerekli alanı kaplayacak kadar bina oluşturulana (1'den 11'e) kadar olan adımlar tekrar edilir;

12. 1'den 12'ye kadar olan adımlar popülasyondaki birey sayısı kadar tekrar edilir ve modelde ilk girilen birey sayısı kadar birey (binalarla kaplı arazi) oluşturulur.

13. Oluşturulan her birey için uygunluk fonksiyonlarına göre puan verilir. Bu modelde 5 uygunluk fonksiyonu vardır:

- a. Taban alanı
- b. Emsal
- c. İklim
- d. Manzara
- e. Rüzgâr

14. Uygunluk puanları verilen bireylerin toplam skorları 0 ile 1 arasında normalize edilir; 1 en başarılı bireylerdir, 0 en başarısız bireylerdir. Bu aşamada turnuva seçilimi uygulanır.

15. Çaprazlama gerçekleşir. Çaprazlama için belirli bir nokta seçilir o noktadan yukarıda olan bireyler anneden, aşağıda olan bireyler ise babadan seçilerek yeni üretim elde edilir.
16. Mutasyon – programda oluşturulan değer oranında gerçekleştirilir.
- a. 1-4 arasındaki mutasyonda seçilen bina sağa, sola, yukarı, aşağı oynatılır.
- b. 5-6 arasındaki mutasyonda seçilen binanın katsayısı artırılır ya da azaltılır.
- c. 7-10 arasındaki mutasyonda seçilen binanın $X - Y$ değeri azaltılır ya da artırılır.
- d. 11 yön değiştirilir.
17. Ebeveyn popülasyonu ve yeni üretilen popülasyonlarla beraber başlangıçta N olan birey sayısı bu aşamada $2N$ olur.
18. Popülasyon pareto üstünlüklerine dayalı olarak bastırılmamış sınıflandırmaya göre sıralanır ve her sıra bir pareto sırasını temsil eder.
19. Pareto öncüleri oluşturulur, eğer her uygunluk puanı için en iyi olan bir birey varsa o birey pareto öncüsüdür. Geri kalan bireyler bastırılmış bireylerdir ve kendilerinden daha iyi skorlu bireyler vardır.
20. Bu oluşturulan sıralamadaki bireyler en başarılı bireyden son bireye kadar yeni ebeveyn kümesine eklenir.
21. Ebeveyn kümesine turnuva seçilimi, çaprazlama ve mutasyon tekrar uygulanarak yeni çocuk bireyler oluşturulur ve bu aşamada yine $2N$ birey oluşmuş olur.
22. Oluşturulan yeni bireyler için 18.-22. madde arasındaki aşamalar, modelin çalışması kullanıcı tarafından durduruluncaya kadar tekrar eder.

5.6 Modelin Uygulanması

Modelin uygulanması için İstanbul'un Kağıthane ilçesi seçilmiştir. Kağıthane'nin kentsel dönüşüm kapsamında yeniden projelendirilmesi gereken geniş arazilere sahip olması, bu bölgeden bir arazinin modelin sınanması için tercih edilmesine neden olmuştur. Şekil 5.11'de görülen arazi, sınırları içerisinde bulunan Kağıthane Deresi ile sürdürülebilir arazi planlama modeli için iyi bir test alanı olmuştur.



Şekil 5.11 : Modelin test edildiği Kağıthane arazisi.

İlk olarak arazi üzerinde bulunan korunacak elemanlar ve kullanılmış alanlar tespit edilmiş ve sonucunda Şekil 5.12' de görülen görsel arazi tanımlaması yapılmıştır.



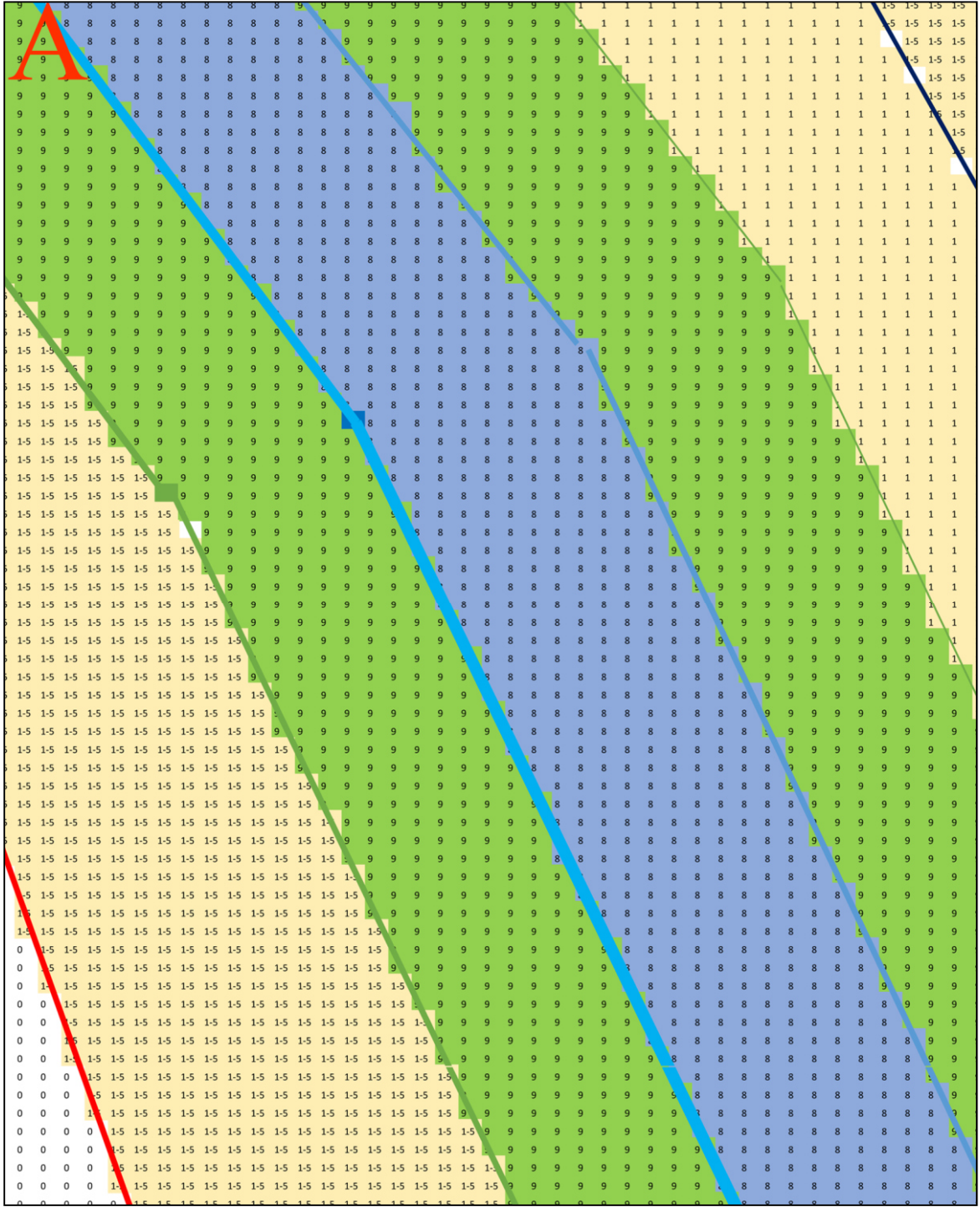
Şekil 5.12 : Seçilen arazinin görsel tanımı.

Sonraki adımda Excel'de arazinin sayısal matrisi hazırlanmıştır (Şekil 5.13, Şekil 5.14, Şekil 5.15). Kağıthane Deresi korunacak olan su elemanı olarak matriste "8"

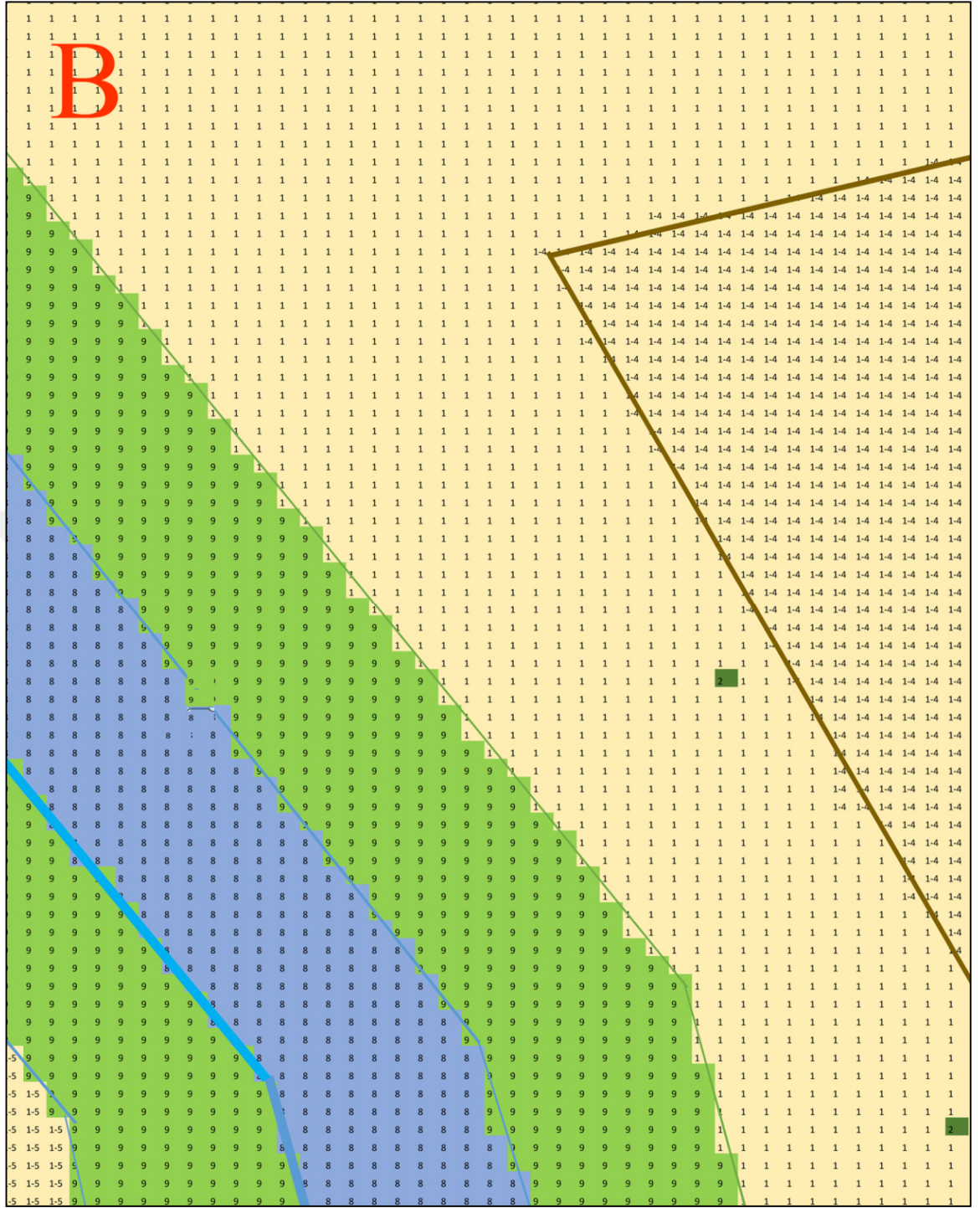
ile temsil edilmiş ve mavi renkte gösterilmiştir. Çevresindeki mevcut yeşil alan "9" ile gösterilmiştir ve bu bölge de koruma altındadır. Matriste "1" sayısı ile yapılaşma olabilecek boş hücreler tanımlanmıştır. Son olarak arazideki mevcut korunacak ağaçlar "2" ile gösterilmiştir. Arazinin Excel’de hazırlanan matrisi Şekil 5.13’te görülmektedir. Arazi matrisinden seçilen 3 farklı bölge daha büyük ölçekte Şekil 5.14, Şekil 5.15 ve Şekil 5.16’da gösterilmiştir. Arazi kendi içerisinde farklı kotlara sahip parçalardan oluşmaktadır. Derenin bulunduğu kot "0.00" kotu olarak kabul edilmiş ve diğer kotlar dereyle aralarındaki kot farkına göre numaralandırılmıştır.



Şekil 5.13 : Seçilen arazinin Excel’de hazırlanan matrisi.



Şekil 5.14 : Arazinin Excel'de hazırlanan matrisinden bir bölüm (A).



Şekil 5.15 : Arazinin Excel'de hazırlanan matrisinden bir bölüm (B).

Arazinin sayısal olarak tanımlanmasından sonraki adım kullanıcı ara yüzüne, seçilen Kağıthane arazisi ile ilgili verileri girmektir. Şekil 5.17’de Kağıthane arazisinde 3 farklı bina tipini kullanarak arazi yerleşim planı üretmek için kullanıcı arayüzünden girilen değerler görülmektedir.

KULLANICI UC TIP GIRER

12	15	6
X	Y	Z
18	20	8
X	Y	Z
20	25	15
X	Y	Z

BATI

DOGU

SICAK

0.25 TAKS

1.2 KAKS

15 HMAX

20 YESIL

100 BIREYSAYISI

URET DURDUR

Şekil 5.17 : Kağıthane arazisi için kullanıcı arayüzünden girilen değerler.

Kullanıcı ara yüzüne girilen araziye ait değerler şu şekildedir:

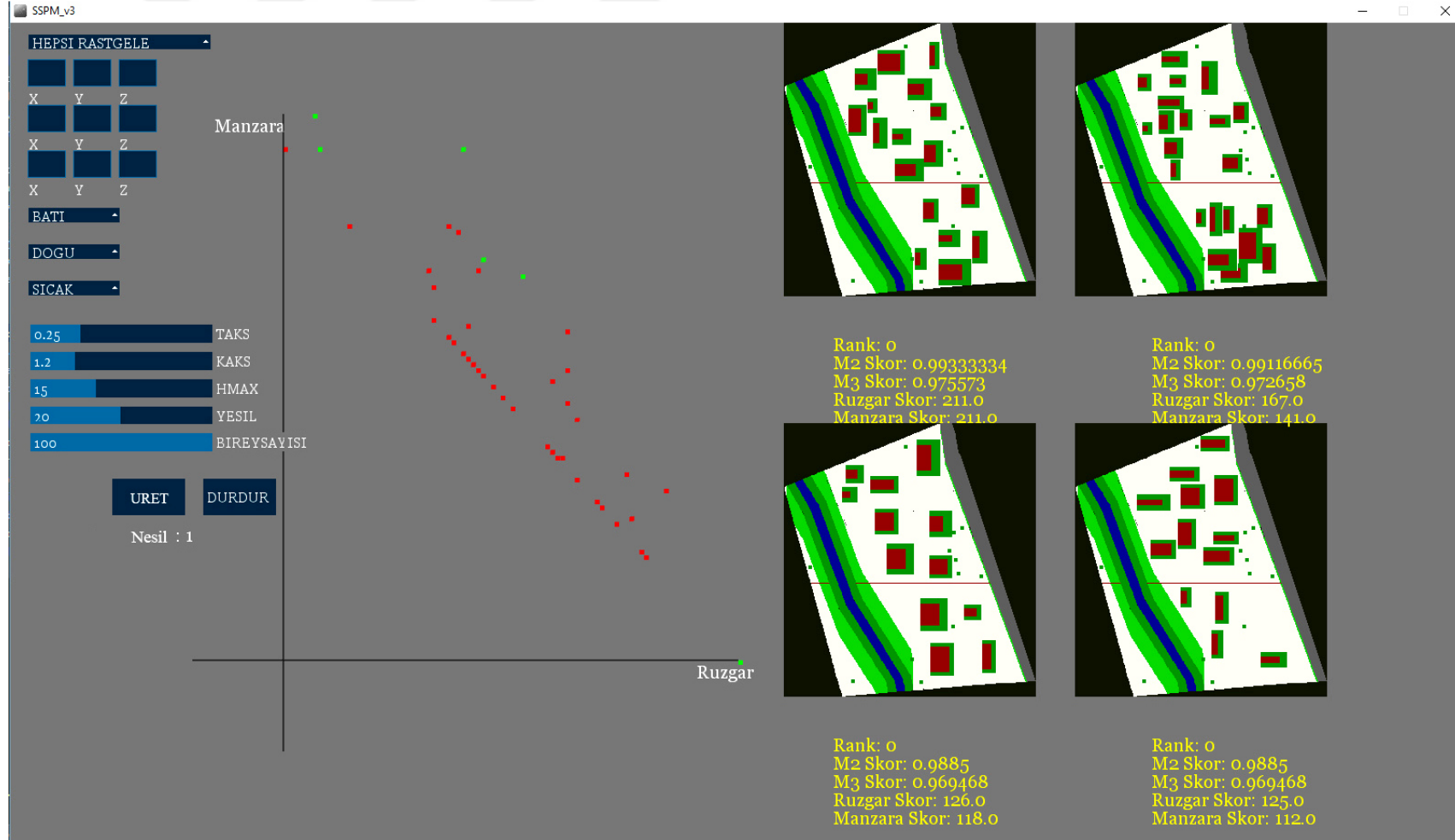
- İklim tipi: Sıcak,
- Hakim rüzgar yönü: Doğu,
- Manzara yönü: Batı,
- TAKS: 0,25,
- KAKS: 1.2,
- Maksimum bina yüksekliği: 45.50 m,
- Açık alan yüzdesi: %20,

Arazi değerlerinin ve her nesilde üretilmesi istenen birey sayısının girilmesinden sonra "ÜRET" tuşu ile ilk arazi bireyleri üretilmeye başlanır. "DURDUR" ile üretim

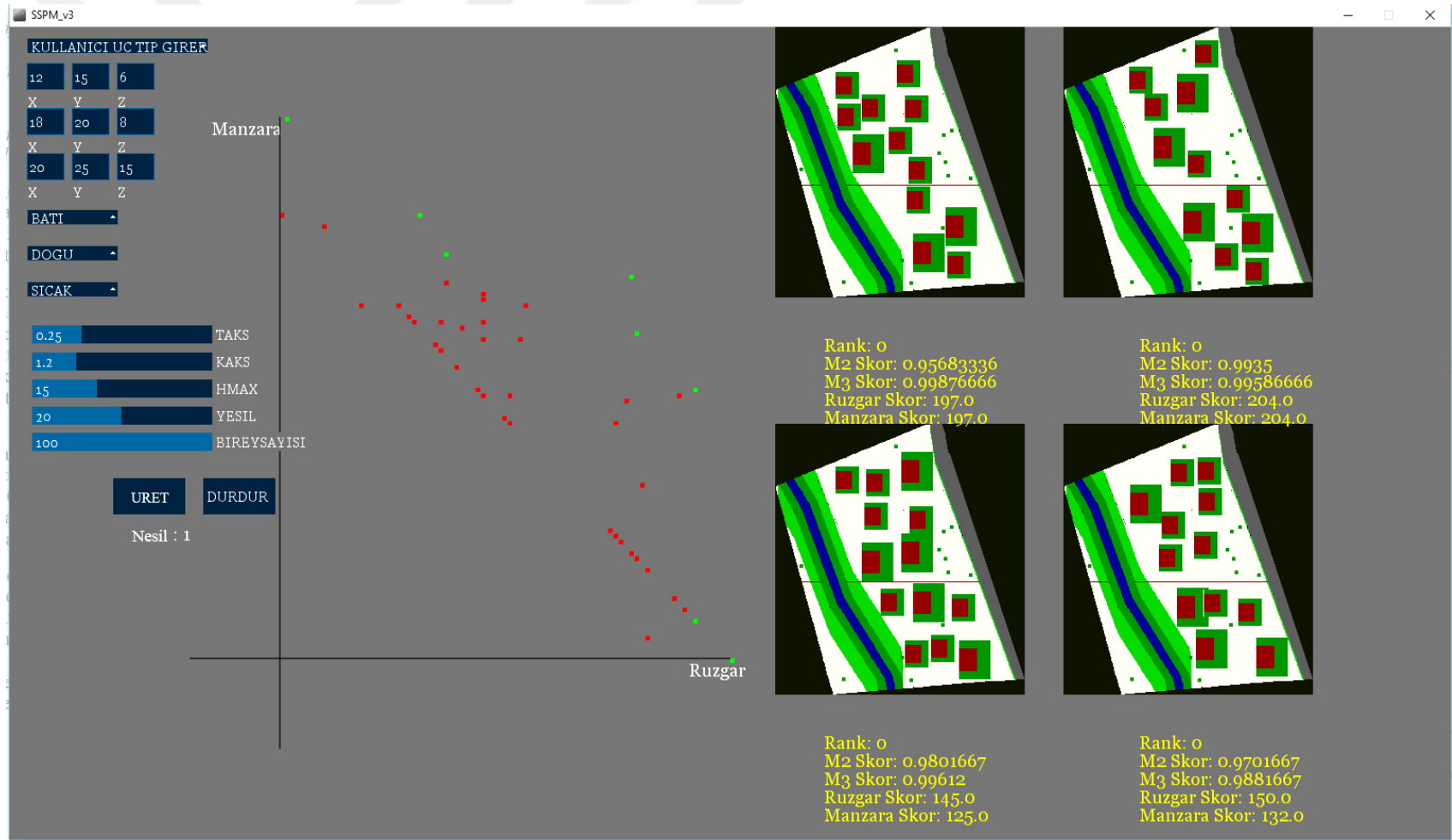
duraklatılarak üretilen bireyler incelenebilir. Her üretimin altındaki butonla 3 boyutlu arazi yerleşimi görülebilmektedir. Tekrar "ÜRET" tuşuna basılarak üretim kaldığı yerden devam ettirilebilir.

İlk olarak Kağıthane arazisinde rastgele tiplerde üretim gerçekleştirilmiştir, her neslin birey sayısı 100 olarak belirlenmiştir. Şekil 5.18'de maksimum yükseklik 11 kat alınarak oluşan ilk neslin en yüksek puana sahip 4 bireyi kullanıcı arayüzünde görülmektedir. Kat yüksekliğinin düşük tutulmasıyla araziye yerleştirilen bina sayısı oldukça fazladır. Üretilen her bireyin altında farklı her fonksiyon için aldığı puanlar görülmektedir. Her neslin manzara ve rüzgâr puanlarına göre oluşturulan pareto eğrisi de arayüzde yer almaktadır. Şekil 5.18'de örnek olarak verilen pareto diyagramında, yeşil noktalar skoru "0" olan pareto optimal bireyleri, kırmızı noktalar ise pareto optimal olmayan, yani başka bir birey tarafından bastırılan bireyleri göstermektedir.

Bastırılmamış pareto optimal birey "0" skora sahiptir, bu da hem rüzgar hem de manzara puanını maksimize eden ve başka hiçbir birey tarafından bastırılmayan birey olduğunu göstermektedir. Skoru "0" olan birey, kullanıcı arayüzünde ilk sıraya yerleştirilir ve başka bir birey tarafından bastırılana kadar yeri korunur. Kullanıcı üretimi durdurana kadar model yeni bireyler üretmeye ve puanlarına göre değerlendirmeye devam eder.



Şekil 5.18 : Rastgele üretilen bina tipleri için arazi yerleşim planı üretimleri 1.Nesil.

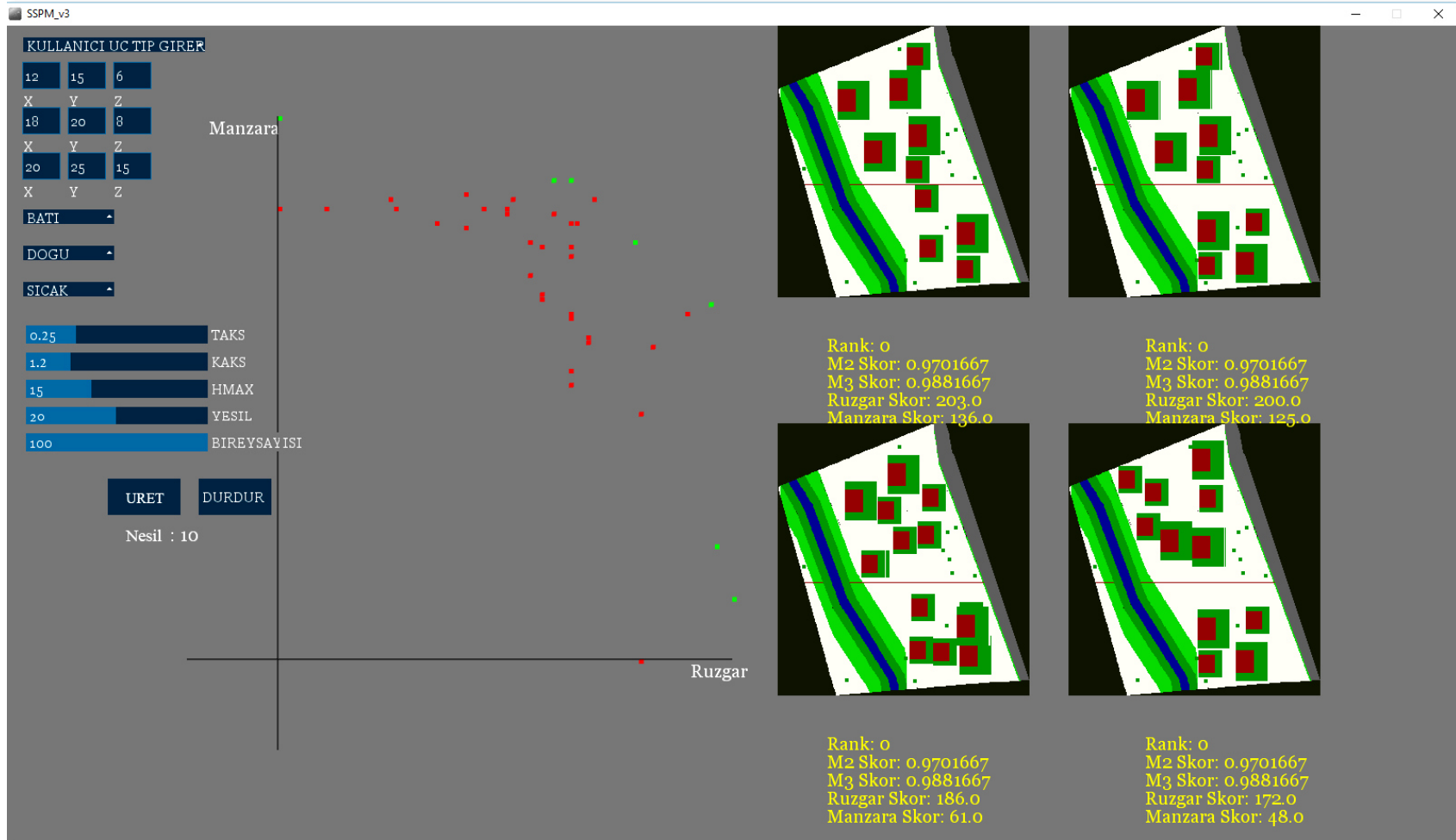


Şekil 5.19 : Üç tip bina için arazi yerleşim planı üretimleri 1. Nesil.

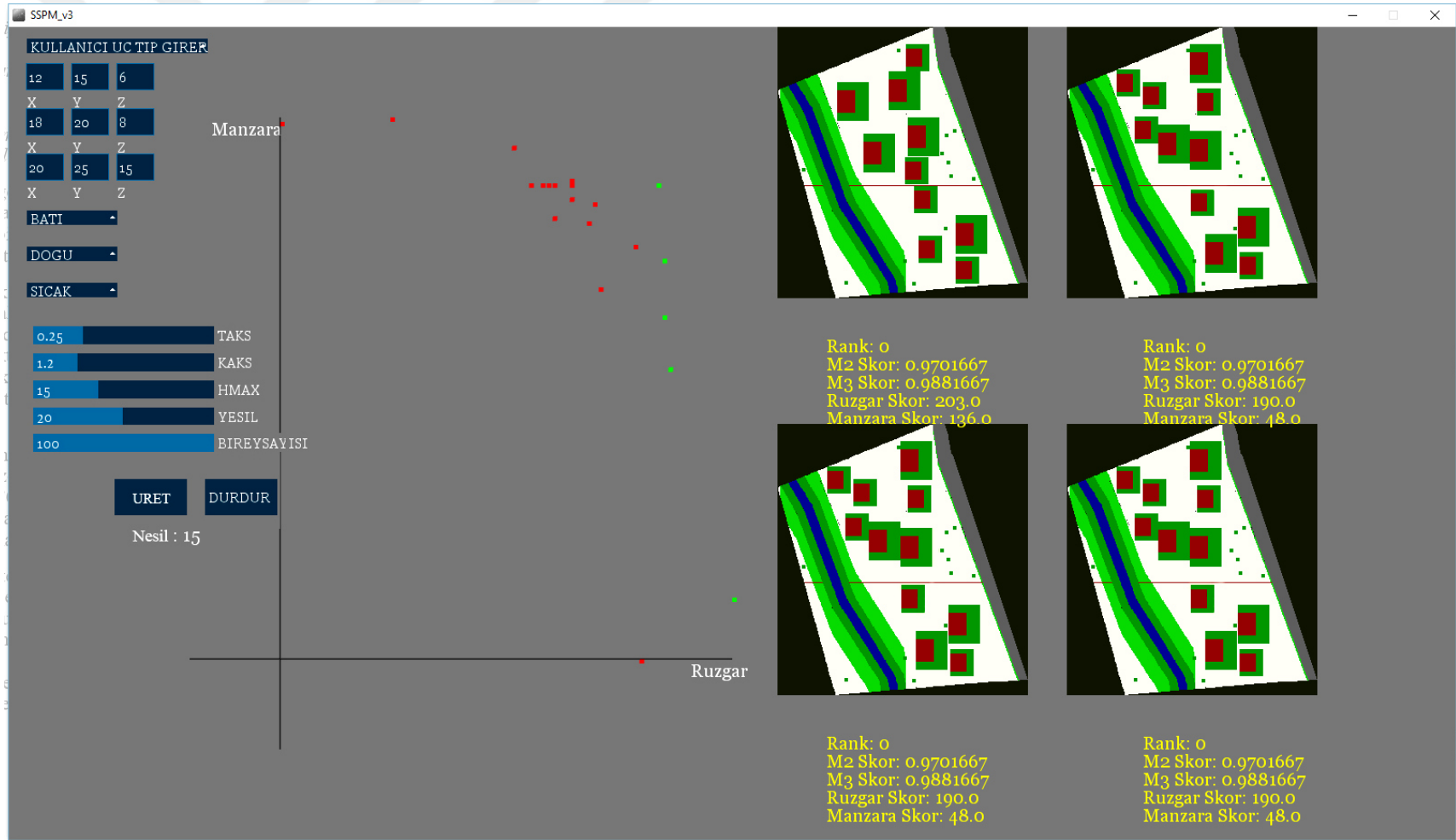
Şekil 5.18'de ilk üretilen bireylere bakıldığında arazideki korunması gerekli ağaçlar dikkate alınarak blok yerleşimleri yapılmış, Kağıthane Deresi ve 30'ar metre çevresi de koruma altına alınarak bloklar bu bölgeden uzağa yerleştirilmiştir. Bloklar manzaraya yönelmiştir. Birbirinin manzarasını kesmeyen fazla puanlı alternatiflerin üretim şansı da fazladır. Sıcak iklimde amaç rüzgârı maksimize etmektir. İlk nesilde, eniyileme tamamlanmadığı için pareto eğrisi daha doğrusaldır; bu nedenle Kağıthane arazisi için manzara ve rüzgar pareto eğrisi nesil ilerledikçe her iki fonksiyonu da maksimize edecek şekilde konkav olarak şekillenecektir.

Şekil 5.19'da 3 farklı boyutta bina tipi kullanılarak üretim gerçekleştirilmiştir. Bina boyutları büyüdüğünde yerleştirilen bina sayıları azalmış ve bina yüksekliğine bağlı olarak arka ve yan bahçe mesafeleri artmıştır. Arka ve yan bahçe mesafeleri yapı yaklaşma sınırı olarak kabul edilmiştir.

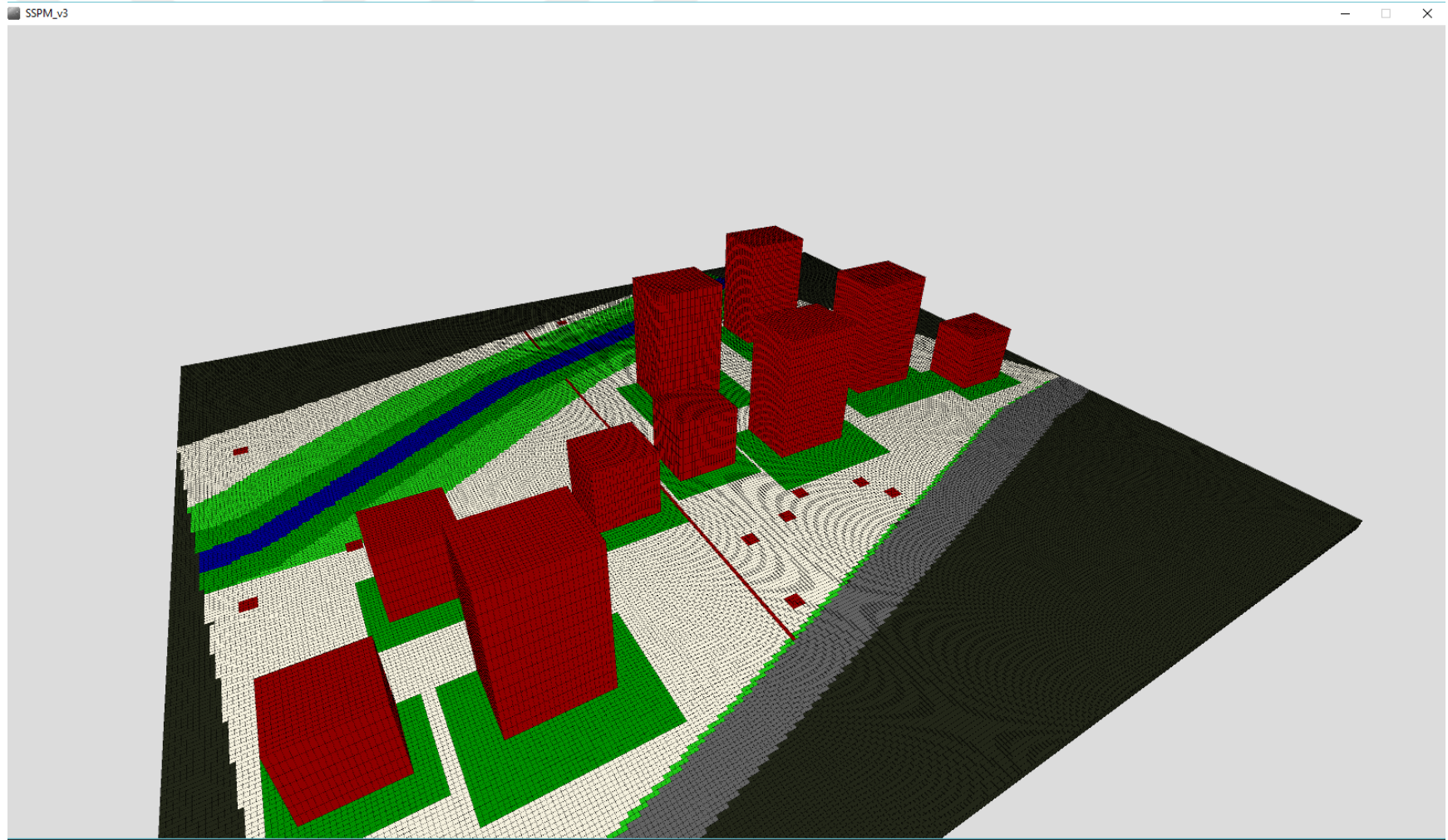
Y ekseninde manzara puanı, X ekseninde ise rüzgâr puanı görülmektedir. Nesil ilerledikçe pareto eğrisi sıcak iklime sahip olan Kağıthane arazisi için hem manzara hem de rüzgar fonksiyonlarını maksimize etmek için konkavlaşacaktır, çünkü her iki fonksiyonda da en yüksek puana sahip bireylerin seçim şansı daha yüksektir. Şekil 5.20 'de üç tip bloktan oluşan bu üretimin 10. nesline ait pareto eğrisi verilmiştir. Şekil 5.21'de ise 15. nesile ait pareto eğrisinin değişimi gözlenebilir. 15. nesile ait pareto eğrisi tamamen konkav olarak görülmekte ve yeşille gösterilen pareto optimal bireyler, hem manzara hem de rüzgarı maksimize etmiş ve üretilmiş diğer hiçbir birey tarafından bastırılmamış bireyler olarak karşımıza çıkmaktadır.



Şekil 5.20 : Üç tip bloktan oluşan üretimin 10. nesil bireyleri ve pareto eğrileri.



Şekil 5.21 : Üç tip bloktan oluşan üretimin 15. nesil bireyleri ve pareto eğrileri.



Şekil 5.22 : 15.nesildeki en yüksek puanlı bireyin 3 boyutlu gösterimi.

"DURDUR" ile üretim duraklatılarak üretilen bireyler incelenebilir. Seçilen herhangi bir bireyin 3 boyutlu arazi yerleşimi görülebilmektedir. Tekrar "ÜRET" tuşuna basılarak üretim kaldığı yerden devam ettirilebilir. Şekil 5.22'de üç tip bloktan oluşan üretimin 15. nesilindeki en yüksek puana sahip bireyi 3 boyutlu olarak görülmektedir.

Elde edilen pareto optimal arazi bireyleri, arazi yerleşim problemlerinde genetik algoritmanın iyi ve tutarlı sonuçlar verdiğini göstermektedir. Üretilen arazi bireylerinde korunacak ağaçlar, manzara ve rüzgâr yönü dikkate alınarak bina yerleşimleri yapılmıştır. Üretilen ilk nesillerde, eniyileme tamamlanmadığı için pareto eğrisi daha doğrusaldır; manzara ve rüzgâr pareto eğrisi nesil ilerledikçe her iki fonksiyonu da maksimize edecek şekilde konkav olarak şekillenmiştir ve daha yüksek puanlı bireyler üretilmiştir. Arazinin 3 boyutlu gösteriminde kot farkları teraslar şeklinde üretilmiştir. Literatürde genellikle iki boyutlu plan eniyileme problemlerinde kullanılan genetik algoritmanın 3.boyutu da kullanarak üretimler yapması sağlanarak hedeflere ulaşılmıştır.



6. SONUÇLAR

Sürdürülebilir Yerleşim Planlama Modeli (SSPM) sadece enerji etkin bina modeli olarak değil, eskiz aşamasından başlayarak sürdürülebilir arazi kullanımına ve yeşil bina sertifika sistemlerinin ve yapı şartnamelerinin gereklerine öncelik sırasına göre değer veren bütünsel bir model olarak ele alınmıştır. Mimarın sadece kendi estetik anlayışına ve sezgilerine göre görsel ağırlıklı ürettiği plan kararlarını uygulaması sürdürülebilir tasarım olarak değerlendirilemez. Mimar, tasarımın biçimsel kararlarını görsel etki, algı veya kullanım gibi konulara ek olarak sürdürülebilir arazi kullanımı ve çevresel faktörler (manzara, iklim, rüzgâr vb.) açısından da sorgulamalıdır. Sürdürülebilir bina tasarımını sadece enerji etkin tasarım olarak değerlendirmek oldukça eksik bir kapsam olmaktadır. Tezde üzerinde durulan sürdürülebilir bina tasarım problemi üç şekilde karşımıza çıkmaktadır:

- Sürdürülebilir bina tasarımı geleneksel bina tasarımına göre daha karmaşık bir sürece sahiptir; birbiriyle çelişen pek çok tasarım kararı erken tasarım evresinde bir arada ele alınmalıdır ve sürdürülebilir bina tasarım problemini çözmeye yardımcı bir karar destek sistemine gereksinim vardır
- Günümüzde sürdürülebilir bina tasarımına yardımcı olmayı hedefleyen benzetim modelleri daha detaylandırılmış ileri tasarım aşamalarında kullanılmaktadır ve çoğu enerji etkin çözümler sunmaktadır; oysa sürdürülebilir bina tasarımında önemli kararların çoğu, tasarımın ilk evresi olan kavramsal aşamada alınmaktadır.
- Sürdürülebilir mimarlığı teşvik etmek amacıyla pek çok ülke tarafından geliştirilen sertifika sistemleri vardır. Türkiye’de henüz bir sertifika sistemi oluşturulmamıştır ve var olan sistemler coğrafi, ekonomik ve kültürel şartları ile hazırlandıkları ülkelere özgü oldukları için kullanımlarında zorluklar yaşanmaktadır.

Bu problemler ışığında, LEED ve BREEAM arazi yerleşim kriterlerine, buna ek olarak yerel yönetmelik ve iklim şartları gibi birbiriyle çelişebilen pek çok ölçüte göre değerlendirme yapılmasını gerektiren sürdürülebilir yerleşme tasarımı problemini çözmeye yardımcı bir karar destek sistemi olan SSPM modeli, çelişen

ölçütlere uygun çözüm önerisi geliştirmeye yönelik bir yaklaşım olan pareto genetik algoritma tabanlı olarak gerçekleştirilmiştir.

Tez kapsamında kullanılan pareto analizi değişik sayıdaki önemli etkenleri, daha az önemde olan etkenlerden ayırmak için kullanılan bir tekniktir. Pareto-optimal kavramının evrimsel algoritmalara uyarlanmasıyla, çok amaçlı evrimsel eniyileme algoritmaları geliştirilmiştir. Bu tez çalışmasında çok amaçlı genetik algortimalar içerisinde en iyi performansı gösteren NSGA-II algoritması pareto temelli olarak kullanılmıştır. Pareto temelli yaklaşımlarda, Pareto optimallik koşuluna göre düzenleme yapmak için baskın olmayanları sıralama teknigi kullanılır. Gerçek dünya problemlerinde araştırılan optimal çözüm genellikle tek amaca değil, birden fazla amaca bağlıdır ve bu amaçlar genellikle birbiriyle çelişirler; bu nedenle, son zamanlarda çok amaçlı genetik algoritmalar birçok tasarım probleminin optimize edilmesinde yaygın olarak kullanılmaya başlanmıştır.

Tezde, sürdürülebilir tasarıma farklı ölçeklerden bakmaları nedeniyle iki farklı evrimsel model örnek olarak seçilmiştir. Wang, Rivard ve Zimeureanu tarafından 2006 yılında geliştirilen Yeşil Bina Tasarımı İçin Plan Eniyileme Modeli, sürdürülebilir tasarım bağlamında iklime uygun bina biçimi ve yönelimi yanında en uygun pencere tipi ve duvar malzemesini minimum enerji tüketimi, doğal ışıktan maksimum faydalanma ve minimum maliyet başlıklarını göz önünde bulundurarak seçim yapılmasını sağlamaktadır. Bina biçimi yanında malzeme ve maliyeti göz önünde bulundurması açısından işlevsel bir model olarak tanımlanabilir, fakat erken tasarım evresinde birden fazla binadan oluşan kompleks sistemlerin araziye yerleşiminde yetersiz kalmaktadır. Binalarda enerji korunumuna odaklanarak, binaların birbiriyle ilişkileri, arazi eğimi, yeşil alanların korunumunu önemseyen sürdürülebilir bir arazi yerleşimi göz ardı edilmiştir. Zielinska, Church ve Jankowski (2008) tarafından geliştirilen SMOLA, şehir ölçeğinde sürdürülebilirlik amaçları doğrultusunda arazi kullanım örüntüleri üretmektedir. Birbiriyle çelişen iki kriter olan maksimum açık alan ve maksimum kalkınma, komşu yerleşimlere uygunluk ve var olan yerleşim alanlarına yakınlık şartlarını göz önünde bulunduran çok amaçlı mekânsal eniyilemeyi hedeflemektedir. Çok amaçlı eniyileme metodunu kullanarak sürdürülebilir arazi yerleşim önerileri üreten model sadece kent ölçeğinde çalışmakta, bina ölçeğinde arazi yerleşimini göz ardı etmektedir.

Sürdürülebilir Yerleşim Planlama Modeli (SSPM) sadece bina yerleşim planlama modeli olarak değil, eskiz aşamasından başlayarak sürdürülebilir arazi kullanımına ve yeşil bina sertifika sistemlerinin ve yapı şartnamelerinin gereklerine öncelik sırasına göre değer veren, bütünleşik bir model olarak ele alınmıştır. Bu tezde sadece enerji korunumuna odaklanmak yerine, binaların birbiriyle ilişkileri, arazi eğimi, yeşil alanların korunumunu önemseyen sürdürülebilir bir arazi yerleşimi hedeflenmiştir.

Modelin sınanması için Kağıthane'de yer alan 4500 m²' büyüklüğünde bir arazi seçilmiştir. Hipotetik bir arazi yerine, daha önceden bir bölümünün kullanılmış olması, sınırları içerisinde korunacak ağaç ve su öğelerini içermesi sebebiyle bu arazi tercih edilmiştir. Arazi matrisle tanım tekniği kullanılarak temsil edilmiş ve matrisler Excel'de oluşturulmuştur. Matrisin satırlarını ve sütunlarını oluşturan her bir elemanı (hücresi), arazinin alanı 1 m² olan bir birimini temsil etmektedir. Arazi köşe noktalarının ve arazi üzerindeki mevcut elemanların x,y koordinatlarına göre matris oluşturulmaktadır. Arazinin belirli bir formu olmaksızın tüm girinti ve çıkıntılarıyla modele tanıtılmasına olanak sağlayan bu yöntem, arazi topoğrafyasının da hücresel olarak modele tanıtılmasına olanak sağlamaktadır. Bu sınırlar içerisinde kalan birim kareler, fonksiyonlarına göre numaralandırılarak arazi sayısal olarak bilgisayara tanımlanmaktadır. Ayrıca her hücre fonksiyonlarına göre renklerle gösterilerek, görsel okunabilirlik de sağlanmıştır. SSPM modelinde arazinin topografik yapısını ve doğal oluşumlarını göz ardı etmeden, çözümün gerçekçi sonuçlar verebilmesi için, sayısal araziye eldeki eğim verilerine göre 3 boyutlu karakter kazandırılmıştır. Sürdürülebilir bina tasarımını amaçlayarak geliştirilen pek çok evrimsel modelden farklı olarak, arazideki kot farklarını ve üretilen bina yüksekliklerini 3. boyutta kullanıcıya gösteren SSPM modeli, gerçek arazi verilerini sayısal olarak kullanarak, arazideki korunacak ağaçları, su öğelerini ve yeşil alanları üretimlerine dâhil etmektedir. Böylece güncel veri grubu üzerinden üretimlerini yaparak gerçekçi sonuçlar vermektedir.

Arazi İstanbul ilinde olması sebebiyle uygunluk fonksiyonlarının bir kısmını İstanbul İli İmar Yönetmeliği kuralları oluşturmuştur. Araziye ait TAKS, KAKS, hmax, manzara, rüzgar ve iklim verilerine göre üretilmiş ilk popülasyon, sonrasında NSGA-II algoritması döngüsü içerisine sokularak pareto optimal arazi yerleşim bireyleri elde edilmiştir. Elde edilen ilk neslin pareto eğrisi eniyileme tamamlanmadığı için

doğrusal olarak elde edilmiştir. Nesil ilerledikçe pareto eğrisi sıcak iklime sahip olan Kağıthane arazisi için hem manzara hem de rüzgar fonksiyonlarını maksimize etmek için konkavlaşmıştır. İleriki nesillerin pareto grafiğinde yeşille gösterilen pareto optimal bireyler hem manzara hem de rüzgarı maksimize etmiş ve üretilmiş diğer hiç bir birey tarafından bastırılmamış en yüksek puanlı bireyler olarak karşımıza çıkmıştır.

Elde edilen pareto optimal arazi yerleşim bireyleri, kütle yerleştirme problemlerinde genetik algoritmanın iyi ve tutarlı sonuçlar verdiğini göstermektedir. Literatürde genellikle iki boyutlu plan eniyileme problemlerinde kullanılan genetik algoritmanın 3.boyutu da kullanarak üretimler yapması bu modelde sağlanarak daha gerçekçi çözümlere gidilmesine olanak sağlamıştır. Hem yerleşilen arazinin eğim verilerini okuyabilen, hem de yerleştirilen bina kütlelerinin yüksekliğini göz önünde bulundurarak yaklaşma mesafelerini hesaplayan model toplu konutların tasarımında yerleşme alternatifleri üretebilmektedir. Özellikle yoğun yapılaşmaya izin veren araziler üzerinde birbiriyle çakışan arazi yerleşim kriterlerini göz önünde bulundurarak bina kütle yerleşimlerini yapmak geleneksel yöntemle oldukça zaman almaktadır. Modelin ürettiği pareto optimal arazi bireylerinin çok amaçlı genetik algoritmanın yardımıyla birkaç dakika içerisinde elde edildiği düşünülürse, bu modelin erken tasarım evresi için önemli bir yardımcı araç olduğu söylenebilir.

Bu çalışmada, sürdürülebilir arazi yerleşim problemi için özgün bir model önerisi yapılmış ve farklı koşullara göre esnekliği sağlanarak anlamlı sonuçlar elde edilmiştir, ancak modelde genetik algoritmanın performansını olumsuz yönde etkileyen parametreler de mevcuttur. Her nesilde üretilen birey sayısı 100 ile sınırlandırılmıştır; daha fazla birey üretimi, kullanılan bilgisayarın teknik özelliklerine bağlı olarak, modelin çalışma hızını oldukça düşürmektedir. Yazılımın geliştirilmesinde dört çekirdekli Core i7-3610M işlemcisinden güç alan dizüstü bilgisayar kullanılmıştır. Depolama için 700 devir hızında çalışan 750GB kapasiteli diske ek olarak G75VW, aynı zamanda 16GB DDR3 kullanılmaktadır. Diz üstü bilgisayar ekran kartı olarak 2GB GDDR5 bellekli bir donanıma sahiptir. Arazi matrisi 257x100 birim kare boyutlarındadır ve bir birey 25700 byte yer kaplamaktadır. Her nesilde üretilen 100 birey için bilgisayar 2570000 byte bellek kullanmaktadır. 2570000 byte, 2,45 GB etmekte ve kullanılan bilgisayar belleğini tamamen doldurmaktadır. Bu nedenle daha yüksek bellekli bir bilgisayarla üretilen

birey sayısı artırılabilir, örneğin 200 bireyden oluşan bir üretim için 4.9 GB belleğe sahip bir bilgisayar kullanılmalıdır.

Arazinin üzerindeki elemanlar ve topoğrafyası kot farklarına göre kullanıcı tarafından teraslanarak bir matris şeklinde elle excel dosyasına girilmiştir. Her bir metrekaire 1 birim kare tarafından temsil edilmiştir. Bu durum arazi veri girişini oldukça yavaşlatmıştır, her bir metrekaire fonksiyonuna ve kotuna göre elle tek tek excel dosyasına işlenmiştir. Farklı bir yaklaşım olarak arazinin uydu görüntüsü kullanılarak, üzerinde bulunan öğelerin renklendirilmesi ve kot farklılıklarında tek bir rengin tonlarının kullanılmasıyla, optik okuyucu yardımıyla arazi matrisinin oluşturulması sağlanabilir. Böylece arazi veri girişi daha hızlı olarak gerçekleştirilebilir.

Modelde arazi kot farklarına göre kullanıcı tarafından tanımlanan teraslar üretilmiş; fakat bina yerleşimlerinde bu teras sınırları göz ardı edilmiştir. Bazı bloklar bu sınırlar üzerine konumlanmıştır. İleriki çalışmalarda binaların yüksek kot farkı bulunan teras sınırlarından belirli uzaklığa yerleştirilmesi sağlanarak daha gerçekçi çözümlere gidilebilecektir.

Model kapsamı daha da genişletilerek, arazi içerisine taşıt ve yaya yolu önerileri, gürültü kaynağına uzaklık ve toplu taşıma nokatsına yakınlık gibi pek çok kriter eklenebilecek ve daha büyük boyutlu arazilerde kullanılarak, 3 tipten daha fazla bina kütlesi yerleşimine de olanak sağlanabilecektir. Böylece SSPM modeli, eklenebilecek farklı uygunluk fonksiyonları ile şehir ölçeğinde de konut bölgelerine yerleşim önerileri getirirken, yardımcı bir karar destek modeli olarak kullanılabilir.

Modelde yerleşim, manzara ve rüzgâr açıları kuzey-güney-doğu ve batı olmak üzere dört ana yöne tasarlanmıştır; açılı yerleşimler modelde üretim dışında tutulmuştur. Modelin sınırlılıklarından bir diğeri de bina gölge ilişkilerinin göz ardı edilmiş olmasıdır. SSPM modeli, imar yönetmelikleri ve setfikasyon problemleri yanında manzara ve rüzgara göre arazi yerleşim planı üreten deneysel bir çalışmadır. Günümüzde bina gölge ilişkileriyle birlikte bina kabuk analizleri yapan profesyonel benzetim modelleri mevcuttur. SSPM modelinin tasarımın daha ileriki aşamalarında kullanılabilen benzetim modellerine, bir arazi yerleşim altyapısı hazırladığı söylenebilir, çünkü hiçbir benzetim modeli TAKS, KAKS ve arazide korunacak

öğelere ve yeşil alan kullanımına yönelik bir uygulama kullanıcılarına sunmamaktadır. Model üzerinde yapılabilecek ileriki çalışmalarda, en iyi puana sahip olan pareto optimal arazi bireyi sanal rüzgar tüneline (flow design) sokularak, iklim koşulları, rüzgar ve manzara yönüne göre konumlandırılan bireyinin rüzgar türbülans etkisi değerlendirilebilir.

Modelin kullanıcı profili, imar yönetmeliğine uygun proje yapma zorunluluğu olan fakat arazi yerleşiminde korunması gereken öğeleri tasarımının odağında tutan, bina malzeme seçiminden önce iklim ve rüzgar verilerine göre öncelikle binayı doğru konumlandırarak sürdürülebilir bir arazi yerleşim planı üretmeyi amaçlayan, mimarlar olarak öngörülmüştür. Doğru planlanmış bir arazi planı, binalarda enerji kaybını minimuma indirerek, hem kullanılam malzemeler açısından hem de inşaa sonrasında tüketilen enerji açısından tasarruf edilerek inşaat ve yaşam maliyeti düşürülebilecektir.

Sayısal ortamda planlanmış ve tasarlanmış bir çevre, tasarımcının yaratıcılığını doğru yönde ilerletmesine olanak sağlayarak oldukça fazla sayıda alternatifi değerlendirmede yardımcı olacaktır. İki farklı disiplin olan evrimsel algoritmalar ve sürdürülebilir tasarım ara kesitinde yer alan bu çalışma ile gerçekleştirilecek sürdürülebilir yerleşme tasarımı modelinde, amaç uzman bir tasarımcının yerini almak değil, tasarımcıya tasarım kararlarının alınması sürecinde yardımcı olmasını sağlamaktır.

Sürdürülebilir tasarım sürecinde, bilgisayar destekli araçlarla problem alanına daha geniş açıdan bakarak çözüm alternatifleri üretmek, tatminkâr çözümlerin elde edilmesini sağlayacaktır. Tasarımın ilk evresinde sayısal ortamda planlanmış ve tasarlanmış bir çevre, oldukça fazla sayıda alternatifi değerlendirmede yardımcı olacaktır. Modelinin, literatürde genellikle iki boyutlu plan eniyileme problemlerinde kullanılan genetik algoritmanın gerçek hayatta var olan 3.boyutu da kullanarak sürdürülebilir arazi yerleşim alternatifleri üretmesi sağlanarak hedefe ulaşılmıştır.

SSPM modeli, disiplinler arası çalışmanın ürünü olan evrimsel algoritmanın yardımı ile daha kısa zamanda etkin çözümleri kullanıcıya sunarak tasarım kararlarının daha bilinçli olarak alınmasıyla, ilerleyen aşamalarda ortaya çıkacak problemlerin en aza indirgenmesini sağlayacak bir karar destek modeli olarak kullanılabilecektir.

KAYNAKLAR

- Al-Homoud, M.S.** (2001). Computer Aided Building Energy Analysis Techniques, *Building and Environment*, 36 (2), 42–433.
- Baouendi, R., Zmeureanu, R., & Bradley, B.** (2004). Energy & Emissions Estimator: A Prototype Tool for Designing Canadian Houses, *ASCE Journal of Architectural Engineering*, 11(2), 50-59.
- Alexander, C. & Chermayeff, S.** (1963). *Community and Privacy: Toward a New Architecture of Humanism*. New York, Doubleday.
- Alexander, C.** (1964). *Notes on the Synthesis of Form*. Cambridge, Harvard University Press.
- Angeline, P.J.** (1995). Adaptive and Self - Adaptive Evolutionary Computations. In M. Palaniswami, Y. Attikiouzel, R. Marks, D. Forgel, T. Fukuda (Eds.), *Computational Intelligence: A Dynamic System Perspective* (pp.152-161). Piscataway: IEEE Press.
- Bentley, P.J.** (1999). *Evolutionary Design by Computers* (pp.1-74). London: Kluwer Academic Press.
- Bentley, P.J.** (1999). Aspects of Evolutionary Design by Computers. <http://www0.cs.ucl.ac.uk/staff/p.bentley/wc3paper.html>
- Cassidy, R., Wright, G. & Flynn, L.** (2003). White Paper On Sustainability: A Report on the Green Building Movement. Retrieved May , 2016, from <https://archive.epa.gov/greenbuilding/web/pdf/bdcwhitepaperr2.pdf>
- Chen, X.** (2001). Pareto Tree Searching Genetic Algorithm Approaching Pareto Optimal Front by Searching Pareto Optimal Tree, (Technical Report NK-CS-2001-002). Tianjin : Department of Computer Science, Nankai University.
- Deb, K.** (2001). Multi-objective Optimization Using Evolutionary Algorithms. Retrieved May 22, 2014, from <http://www.iitk.ac.in/kangal/papers/k2011003.pdf>
- Dritsas, S.** (2012). Dijital ve Fiziksel Duyarlılık. Retrieved November, 2015, from <http://www.mimarlarodasiankara.org/dosya/dosya29.pdf>
- Erdoğan, E. ve Sorguç, A.G.** (2011). Hesaplamalı Modeller Aracılığıyla Mimari ve Doğal Biçim Türetim İlkelerini İlişkilendirmek. *METU JFA*, 28 (2), 269-281.
- Evans, G.** (1984). An Overview of Techniques for Solving Multi-objective mathematical Programs. *Management Science*, 30 (11), 1268-1282.
- Fonseca, C. M. & Fleming, P.J.** (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In

- S.Forrest, (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, (pp.416-423). San Fransisco : Morgan Kaufmann Publishers Inc.
- Goldberg, D.E.** (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston: Addison-Wesley Longman Publishing Co.
- Harputlugil, G.U.** (2009). *Enerji Performansını Öncelikli Mimari Tasarım Sürecinin İlk Aşamasında Kullanılabilecek Tasarıma Destek Değerlendirme Modeli*. (Doktora tezi). Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Harputlugil, G.U.** (2011). Enerji Performansına Dayalı Tasarımda Analiz ve Simulasyon. *Megaron*, 6 (1), 1-12.
- Hawkes, D., Mcdonald, J. & Steemers, K.** (2002). *The Selective Environment: An Approach to Environmentally Responsive Architecture* (pp.416-423). London, Spon Press.
- Holland, J.H.** (1975). *Adaptation in Natural and Artificial Syaytems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Michigan, University of Michigan Press.
- Horn, J. & Nafpliotis, N.** (1993). Multobjective Optimization Using the Niched Pareto Genetic Algorithm, (Illigal Report 93005). Illinois : University of Illinois at Urbana-Champaign.
- Horn, J., Nafpliotis, N. & Goldberg, D.E.** (1994). A Niched Pareto Genetic Algorithm for Multi-objective Optimization. *Proceedings of the First IEEE Conference on Evolutionary Computation*, (Vol.1, pp.82-87). New Jersey : IEEE Computer Society Press.
- Ishikawa, K.** (1982). *Guide to Quality Control*. Tokyo, Asian Productivity Organization.
- Knowles, J.D. & Corne, D.W.** (2000). Approximating the Nondominated front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8 (2), 149-172.
- Larsson, N.K. & Cole, J.R.** (2001). Green Building Challenge: The Development of an Idea. *Building Researchand Information*, 29 (5), 336-345.
- Mahdavi, A. & El-Bellahy, S.** (2005).Efforts and Effectiveness Considerations in Computational Design Evaluation: A Case Study. *Building and Environment*, 40 (12), 1651-1664.
- Mitchell, M.** (1996). An Introduction to Genetic Algorithms. Retrieved from <http://www.boente.eti.br/fuzzy/ebook-fuzzy-mitchell.pdfbooks>
- Mitchell, W. J.** (1977). *Computer Aided Architectural Design*. New York, Petrocelli/Charter.
- Murata, T.** (1997). *Genetic Algorithms for Multi-Objective Optimization*. (Doctoral Thesis). Osaka Prefecture University, Osaka.
- Osyczka, A.** (1985). Multi-criteria Optimization for Engineering Design. In J.S. Gero, (Ed.), *Design Optimization*, (pp.193-227). New Jersey: Academic Press.

- Özcan, S.** (2009). İstatistiksel Proses Kontrol Tekniklerinden Pareto Analizinin Çimento Sanayinde Bir Uygulaması. *Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Dergisi*, 2 (2), 151-174.
- Rivard, H.** (2006). Computer Assistance for Sustainable Building Design, Canada Research Chair in Computer-Aided Engineering for Sustainable Building Design. In Ian F. C. Smith, (Ed.), *G-ICE'06 Proceedings of the 13th international conference on Intelligent Computing in Engineering and Architecture*, (pp.559-575). San Fransisco : Morgan Kaufmann Publishers Inc.
- Rivard, H.** (2006). Computer Assistance for Sustainable Building Design. In F. C. Smith, (Ed.), *Intelligent Computing in Engineering and Architecture* (Vol. 4200, pp. 559-575) . Berlin : Springer-Verlag.
- Sağ, T. & Cunkaş, M.** (2009). Çok Amaçlı Genetik Algoritmalar için Bir Çevrimdışı Performans Değerlendirmesi. *IATS'09 5.Uluslararası İleri Teknolojiler Sempozyumu*, Karabük, Türkiye : 19-23 Mayıs.
- Schaffer, J. D. &** (1985). Multi-objective Optimization with Vector Evaluated Genetic Algorithms. In J.J. Grefenstette, (Ed.), *Proceedings of An International Conference on Genetic Algorithms and Their Applications*, (pp.93-100). Pittsburgh : Lawrence Elbaum, July 24-26.
- Sirinavas, N. & Deb, K.** (1994). Multiobjective Optimization Using Nondominated Sorting Genetic Algorithms. *Journal of Evolutionary Computation*, 2 (3), 221-248.
- Squires, H. F.** (1986). *Pareto Analysis, Quality Management Handbook*. New York, ASQC Qualitypress.
- Url-1** <<http://www.breeam.org>>, erişim tarihi 10.06.2015.
- Url-2** <<http://www.usgbc.org>>, erişim tarihi 10.06.2015.
- Url-3** <<http://www.cedbik.org>>, erişim tarihi 10.06.2015.
- Url-4**<http://www.coe.fau.edu/faculty/cafolia/courses/eme6051/alife/index_files/Page1063.htm >, erişim tarihi 05.04.2013.
- Url-5**< <http://www.athenasmi.org/our-software-data/overview> >, erişim tarihi 17.07.2015.
- Url-6**<http://apps1.eere.energy.gov/buildings/tools_directory/software.cfm/?ID=176&pagename=alpha_list_sub?print >, erişim tarihi 07.07.2015.
- Url-7**< http://apps1.eere.energy.gov/buildings/tools_directory/software.cfm/?ID=398&pagename_submenu=sustainability&pagename_menu=whole_building_analysis&pagename=subjects >, erişim tarihi 07.07.2015.
- Url-8**< http://apps1.eere.energy.gov/buildings/tools_directory/software.cfm/?ID=494&pagename=alpha_list_sub >, erişim tarihi 07.07.2015.
- Url-9** < http://apps1.eere.energy.gov/buildings/tools_directory/software.cfm/?ID=267&pagename_menu=united_kingdom&pagename=countries >, erişim tarihi 07.07.2015.

- Url-10** < <http://www.processing.org> >, erişim tarihi 07.12.2015.
- Url-11** < http://www.ibb.gov.tr/tr-TR/kurumsal/Birimler/ImarMd/Documents/imar_yonetmelik.pdf >, erişim tarihi 12.05.2016.
- Url-12** < <http://www.vitrafix.com/assets/pdf/TS825.pdf> >, erişim tarihi 10.06.2015.
- Url-13** < <http://www.isr.umd.edu/~austin/ence200.d/java-examples.html> >, erişim tarihi 10.06.2015.
- Wang, W., Rivard, H. & Zmeureanau, R.** (2006). Floor Shape Optimization for Green Building Design. *Advanced Engineering Informatics*, 20 (4), 363-378.
- Yahiaoui, A. & Harputlugil, G.** (2006). The Use of Systems Engineering in the Building Design. *CIB Student Chapters Postgraduate Conference*, Ankara, Turkey : March 16-17.
- Yılmaz, B.** (2012). *Türkiye İçin Sürdürülebilir Bina Performans Kriterleri ve Bütünleşik Tasarım Yönetim Modeli Oluşturulması*. (Doctoral Thesis). Istanbul Technical University, Istanbul.
- Zielinska, A. L., Church, R. & Jankowski, P.** (2008). Sustainable Urban Land Use Allocation with Spatial Optimization. *International Journal of Geographical Information Science*, 22 (6), 601-622.
- Zitzler, E., Deb, K. & Thiele, L.** (1999). Comparison of Multiobjective Evolutionary Algorithms on Test Functions of Different Difficulty. In A.S. Wu, (Ed.), *Proceedings of the 1999 Genetic and Evolutionary Computation Conference*, (pp. 121-122). Orlando, July 24-26.
- TS 825 Türk Standartları Enstitüsü Binalarda Isı Yalıtım Kuralları.** (1998). *T.C. Resmi Gazete*, 23725, 14 Haziran 1999.
- İstanbul İmar Yönetmeliği.**(2013). *T.C. Resmi Gazete*,28664,1 Haziran 2013

EKLER

EK A: Excel'den Arazi Verisini Okuma Kodu (arazi.java)

EK B: Araziyi Görsel Oluşturma Kodu (functions.java)

EK C: Bina Tanımlama Kodu (building. java)

EK D: Populasyon Üretme Kodu (population.java)





EK A: Excel'den Arazi Verisini Okuma Kodu (arazi.java)

```
import processing.core.PApplet;

import java.util.ArrayList;

/**
 * Created by 1000891 on 1/8/2015.
 */

// Arazi class
// Bu class arazi ve üzerindeki tüm yapıları temsil eder
// Here, a virtual organism's DNA is an array of character.
// Fonksiyonlar:
// -- convert DNA into a string
// -- calculate DNA's "fitness"
// -- mate DNA with another set of DNA
// -- mutate DNA
public class Arazi {
    PApplet parent;
    int[][] arazi;
    int araziNum;

    ArrayList<Building> binalar;

    int binaPozX; // Binanın arazi üzerindeki X pozisyonu
    int binaPozY; // Binanın arazi üzerindeki Y pozisyonu
    int binaXBoyut; // Binanın X Boyutu
    int binaYBoyut; // Binanın Y Boyutu
    byte binaZBoyut;
    int parselPozX;
    int parselPozY;
    int parselXBoyut;
    int parselYBoyut;
    int yanBahceX;
    int yanBahceY;
    int arkaBahceX;
    int arkaBahceY;
    byte yon;

    int count = SSPM_v3.istenenTabanAlani;

    float fitTabanAlani;
```

```

int doluM2; // Arazide ne kadar m2 uretim oldu
int doluM3;

boolean bos;
int bosluk = 2;

float score; // Butun fitness function scorelari toplami
float manzaraSkor; // Manzara skoru (buyuk daha iyi)
float ruzgarSkor; // Ruzgar skoru (sicaksa buyuk iyi soguksa kucuk iyi)
float m2Skor;
float m3Skor;
float m2m3;
int rank;
float crowdingDistance;
float objective[] = new float[4]; // objectiveveler

// Constructor: Arazinin yapilandirilmesi ve rastgele bir sekilde binalarin
// yerlestirilmesi

Arazi(PApplet p, int num){
    parent = p;
    araziNum = num;
    binalar = new ArrayList<Building>();    // Bina listesi

    // Arazi matrisi olusturulur
    arazi = new
int[SSPM_v3.siteFunction.length][SSPM_v3.siteFunction[0].length];
    for(int iy = 0; iy<SSPM_v3.siteFunction[0].length; iy++){
        for (int ix= 0; ix<SSPM_v3.siteFunction.length; ix++){
            arazi[ix][iy] = SSPM_v3.siteFunction[ix][iy];
        }
    }

    doluM2 = 0;
    doluM3 = 0;
    //System.out.println(SSPM_v3.dropProgramMod.getValue()); debugging

    while(count>0){
        bos = true;
        binaUret();
        binaEkle();
    //    println(binaTabanKontrol()); debugging
        if(binaTabanKontrol()){
            binaYerlestir();
        }
    }
    // System.out.println("bina"); debugging

    fitnessManzara();

```

```

    fitnessRuzgar();
    fitness();

}
// Binanın boyutlarini program Moduna gore olusturur
// program mod 1se rastgele olusturur
// program mod 2se kullanicinin belirledigi 1 bina tipiyle olusturur
// program mod 3se kullanicinin belirledigi 2 bina tipiyle olusturur
// program mod 4se kullanicinin belirledigi 3 bina tipiyle olusturur

public void binaUret(){
    int randomNumber;

    if(SSPM_v3.dropProgramMod.getValue() == 1) {
        binaZBoyut = (byte) (parent.random(1, SSPM_v3.hMax)); // Kural 3.1
        hmaxdan yuksek bina olamaz
        binaXBoyut = (int) (parent.random(6, 30)); // Kural 3.3 30 metreden
        buyuk 6 metreden kucuk bina ayriti olamaz
        binaYBoyut = (int) (parent.random(6, 30)); // Kural 3.3
        while (binaXBoyut * binaYBoyut > 600) { // Kural 3.2 bina taban alani
            600 m2den fazla olamaz
            binaXBoyut = (int) (parent.random(6, 30));
            binaYBoyut = (int) (parent.random(6, 30));
        }
    }
    if(SSPM_v3.dropProgramMod.getValue() == 2) {
        //System.out.println(SSPM_v3.birZDeger.getText()); debugging
        binaZBoyut = (byte)Integer.parseInt(SSPM_v3.birZDeger.getText());
        binaXBoyut = Integer.parseInt(SSPM_v3.birXDeger.getText());
        binaYBoyut = Integer.parseInt(SSPM_v3.birYDeger.getText());
        /*while (binaXBoyut * binaYBoyut > 600) { // Kural 3.2 bina taban alani
            600 m2den fazla olamaz
            binaXBoyut = (int) (parent.random(6, 30));
            binaYBoyut = (int) (parent.random(6, 30));
        }*/
    }
    if(SSPM_v3.dropProgramMod.getValue() == 3) {

        randomNumber = (int) (parent.random(2));
        if (randomNumber == 1) {
            binaZBoyut = (byte)Integer.parseInt(SSPM_v3.birZDeger.getText());
            binaXBoyut = Integer.parseInt(SSPM_v3.birXDeger.getText());
            binaYBoyut = Integer.parseInt(SSPM_v3.birYDeger.getText());
        }
        else {
            binaZBoyut = (byte)Integer.parseInt(SSPM_v3.ikiZDeger.getText());
            binaXBoyut = Integer.parseInt(SSPM_v3.ikiXDeger.getText());
            binaYBoyut = Integer.parseInt(SSPM_v3.ikiYDeger.getText());
        }
    }
}

```

```

    }

    if(SSPM_v3.dropProgramMod.getValue() == 4) {

        randomNumber = (int) (parent.random(3));
        if (randomNumber == 1) {
            binaZBoyut = (byte)Integer.parseInt(SSPM_v3.birZDeger.getText());
            binaXBoyut = Integer.parseInt(SSPM_v3.birXDeger.getText());
            binaYBoyut = Integer.parseInt(SSPM_v3.birYDeger.getText());
        }
        if(randomNumber == 2) {
            binaZBoyut = (byte)Integer.parseInt(SSPM_v3.ikiZDeger.getText());
            binaXBoyut = Integer.parseInt(SSPM_v3.ikiXDeger.getText());
            binaYBoyut = Integer.parseInt(SSPM_v3.ikiYDeger.getText());
        }
        else {
            binaZBoyut = (byte)Integer.parseInt(SSPM_v3.ucZDeger.getText());
            binaXBoyut = Integer.parseInt(SSPM_v3.ucXDeger.getText());
            binaYBoyut = Integer.parseInt(SSPM_v3.ucYDeger.getText());
        }
    }

    yon = (byte)SSPM_v3.dropManzara.getValue();    // Yonler 1 Kuzey 2 Dogu 3
    Guney 4 Bati
}

int FitnessYuzeyBosluk(){
    int yuzeyBosluk = 0;
    for(int x = 0; x<arazi.length;x++){
        for(int y = 0; y<arazi[0].length;y++){
            if((arazi[x][y])==1||((arazi[x][y])==9)){
                yuzeyBosluk++;
            }
        }
    }
    return yuzeyBosluk;
}

// Bina Ekle
void binaEkle(){

    // Binanın yonune uygun olacak sekilde arka bahce ve
    // yan bahce eklenir
    //
    if(yon==1){    // Kuzey
        if(binaZBoyut<6){
            yanBahceX = 4;
            arkaBahceY = 5;
        }
        else{

```

```

        yanBahceX = 4 + (int)((binaZBoyut - 5) * 0.5);
        arkaBahceY = 5 + (binaZBoyut - 5);
    }
}

if(yon==2){ // dogu
    if(binaZBoyut<6){
        yanBahceY = 4;
        arkaBahceX = 5;
    }
    else{
        yanBahceY = 4 + (int)((binaZBoyut - 5) * 0.5);
        arkaBahceX = 5 + (binaZBoyut - 5);
    }
}

if(yon==3){ // guney
    if(binaZBoyut<6){
        yanBahceX = 4;
        arkaBahceY = 5;
    }
    else{
        yanBahceX = 4 + (int)((binaZBoyut - 5) * 0.5);
        arkaBahceY = 5 + (binaZBoyut - 5);
    }
}

if(yon==4){ // bati
    if(binaZBoyut<6){
        yanBahceY = 4;
        arkaBahceX = 5;
    }
    else{
        yanBahceY = 4 + (int)((binaZBoyut - 5) * 0.5);
        arkaBahceX = 5 + (binaZBoyut - 5);
    }
}

if(yon == 1){

    parselPozX = (int)(parent.random(arazi.length - (2 * yanBahceX) -
binaXBoyut)); // Parselin arazi icerisindeki Pozisyonunun X eksenini degeri
    parselPozY = (int)(parent.random(arazi[0].length - arkaBahceY -
binaYBoyut)); // Parselin arazi icerisindeki Pozisyonunun Y eksenini degeri

    binaPozX = parselPozX + yanBahceX;
    binaPozY = parselPozY ;
    parselXBoyut = binaXBoyut + 2*yanBahceX;
    parselYBoyut = binaYBoyut + arkaBahceY;
}

```

```

if(yon == 2){

    parselPozX = (int)(parent.random(arazi.length - arkaBahceX -
binaXBoyut)); // Parselin arazi icerisindeki Pozisyonunun X eksen degeri
    parselPozY = (int)(parent.random(arazi[0].length - (2 * yanBahceY) -
binaYBoyut)); // Parselin arazi icerisindeki Pozisyonunun Y eksen degeri

    binaPozX = parselPozX + arkaBahceX;
    binaPozY = parselPozY + yanBahceY;
    parselXBoyut = binaXBoyut + arkaBahceX;
    parselYBoyut = binaYBoyut + 2*yanBahceY;

}
if(yon == 3){

    parselPozX = (int)(parent.random(arazi.length - (2 * yanBahceX) -
binaXBoyut)); // Parselin arazi icerisindeki Pozisyonunun X eksen degeri
    parselPozY = (int)(parent.random(arazi[0].length - arkaBahceY -
binaYBoyut)); // Parselin arazi icerisindeki Pozisyonunun Y eksen degeri

    binaPozX = parselPozX + yanBahceX;
    binaPozY = parselPozY + arkaBahceY;
    parselXBoyut = binaXBoyut + 2*yanBahceX;
    parselYBoyut = binaYBoyut + arkaBahceY;

}

if(yon == 4){

    parselPozX = (int)(parent.random(arazi.length - arkaBahceX -
binaXBoyut)); // Parselin arazi icerisindeki Pozisyonunun X eksen degeri
    parselPozY = (int)(parent.random(arazi[0].length - (2 * yanBahceY) -
binaYBoyut)); // Parselin arazi icerisindeki Pozisyonunun Y eksen degeri

    binaPozX = parselPozX;
    binaPozY = parselPozY + yanBahceY;
    parselXBoyut = binaXBoyut + arkaBahceX;
    parselYBoyut = binaYBoyut + 2*yanBahceY;

}
}

// binanın oturacagi yer bos mu kontrol eder
public boolean binaTabanKontrol(){
    bos = false;
    if(parselPozX+parselXBoyut<=arazi.length){ // Bina X Boyutunda Arazi
matrisi icerisinde mi kontrol eder
        if(parselPozY+parselYBoyut<=arazi[0].length){ // Bina Y Boyutunda
Arazi matrisi icerisinde mi kontrol eder
            bos = true;
            for(int i=0;i<parselXBoyut;i++){

```

```

        for(int j=0;j<parselYBoyut;j++){
            if(arazi[parselPozX+i][parselPozY+j]==1){ // Binanın oturacağı
alan boş mu kontrol eder
                // bu gereksiz
            }
            else{
                bos = false;
            }
        }
    }
}
return bos;
}

```

// Bina Taban Kontrolun parametreli hali
//

```

public boolean binaTabanKontrol(int prsPozX, int prsPozY, int prsXBoyut, int
prsYBoyut){
    bos = false;
    parselPozX = prsPozX;
    parselPozY = prsPozY;
    parselXBoyut = prsXBoyut;
    parselYBoyut = prsYBoyut;
    if(parselPozX+parselXBoyut<=arazi.length){ // Bina X Boyutunda Arazi
matrisi içerisinde mi kontrol eder
        if(parselPozY+parselYBoyut<=arazi[0].length){ // Bina Y Boyutunda
Arazi matrisi içerisinde mi kontrol eder
            bos = true;
            for(int i=0;i<parselXBoyut;i++){
                for(int j=0;j<parselYBoyut;j++){
                    if(arazi[parselPozX+i][parselPozY+j]==1){ // Binanın oturacağı
alan boş mu kontrol eder
                        // bu gereksiz
                    }
                    else{
                        bos = false;
                    }
                }
            }
        }
    }
    return bos;
}

```

// Binayı Yerleştirir

```

public void binaYerlestir(){
    for(int iy = 0; iy<binaYBoyut; iy++){
        for(int ix = 0; ix<binaXBoyut; ix++){
            arazi[binaPozX+ix][binaPozY+iy]=(byte)(20+binaZBoyut); // Binanın

```

oturduğu yerleri 3 yap

```
    }
    }
    for(int iy = 0; iy<parselYBoyut; iy++){
        for(int ix = 0; ix<parselXBoyut; ix++){
            if(arazi[parselPozX+ix][parselPozY+iy]==1){
                arazi[parselPozX+ix][parselPozY+iy]=20; // Binanın oturduğu
yerleri 3 yap
            }
        }
    }
    count = count - (binaXBoyut*binaYBoyut);
    doluM2 = doluM2 + binaXBoyut*binaYBoyut;
    doluM3 = doluM3 + binaXBoyut*binaYBoyut*binaZBoyut;
    //println("Binalar add parselYBoyut: " + parselYBoyut); debugging
    binalar.add(new Building(binaPozX, binaPozY, binaXBoyut, binaYBoyut,
binaZBoyut, yon, parselXBoyut, parselYBoyut, parselPozX, parselPozY));
}
// *****
// Binayi Yerlestirir (parametrik)
//
//
```

```
public void binaYerlestir(int bnPozX, int bnPozY, int bnXBoyut, int bnYBoyut,
int prsPozX, int prsPozY, byte bnZBoyut, byte binaYon, int prsXBoyut, int
prsYBoyut){
```

```
    int binaPozX2 = bnPozX;
    int binaPozY2 = bnPozY;
    int binaYBoyut2 = bnYBoyut;
    int binaXBoyut2 = bnXBoyut;
    int parselPozY2 = prsPozY;
    int parselPozX2 = prsPozX;
    byte binaZBoyut2 = bnZBoyut;
    byte yon2 = binaYon;
    int parselXBoyut2 = prsXBoyut;
    int parselYBoyut2 = prsYBoyut;
    for(int iy = 0; iy<binaYBoyut2; iy++){
        for(int ix = 0; ix<binaXBoyut2; ix++){
            arazi[binaPozX2+ix][binaPozY2+iy]=(byte)(20+binaZBoyut2); //
```

Binanın oturduğu yerleri 3 yap

```
    }
    }
    for(int iy = 0; iy<parselYBoyut2; iy++){
        for(int ix = 0; ix<parselXBoyut2; ix++){
            if(arazi[parselPozX2+ix][parselPozY2+iy]==1){
                arazi[parselPozX2+ix][parselPozY2+iy]=20; // Binanın oturduğu
yerleri 3 yap
            }
        }
    }
}
```

```

        binalar.add(new Building(binaPozX2, binaPozY2, binaXBoyut2,
binaYBoyut2, binaZBoyut2, yon2, parselXBoyut2, parselYBoyut2, parselPozX2,
parselPozY2));
    }
    // Arazi icerisinde olan bir binayi siler
    public void binaSil(int num){
        Building myBina;
        System.out.println("Debug");
        System.out.println(num);
        System.out.println(binalar.size());
        myBina = binalar.get(num);
        System.out.println("bina Sil - parselYBoyut:" + myBina.parsYBoyut);
        System.out.println("bina Sil - parselXBoyut:" + myBina.parsXBoyut);

        for(int iy = 0; iy<myBina.parsYBoyut; iy++){
            for(int ix = 0; ix<myBina.parsXBoyut; ix++){
                arazi[myBina.parsPozX+ix][myBina.parsPozY+iy]=1; // Binanın
oturduđu yerleri 1 yap boylece bina silinmis olur

            }
        }
        binalar.remove(num);

    }
    /*
    // Arazi icerisinde olan bir binayi kaydirir
    void binaMove(int num){
        Building myBina;
        myBina = binalar.get(num);
        if(myBina.yon)
    }*/

    //
    *****

    // Caprazla
    // 2 Arazi nesnesini caprazlar
    // -----

    public Arazi caprazla(Arazi es) {
        // Yeni bir cocuk uretir
        Arazi cocuk = new Arazi(parent, 3);

        // Yarisi anneden alinir, yarisi babadan alinir
        for (int j= 0; j<SSPM_v3.caprazlamaNoktasi; j++){
            for (int i = 0; i<this.arazi.length; i++) {
                cocuk.arazi[i][j] = this.arazi[i][j];
            }
        }
    }

```

```

Building mybuil;
for(int i=0; i<this.binalar.size();i++){
    mybuil = this.binalar.get(i);
    if(mybuil.parsYBoyut<SSPM_v3.caprazlamaNoktasi){
        cocuk.binalar.add(mybuil);
    }
}

for (int j= SSPM_v3.caprazlamaNoktasi; j<this.arazi[0].length; j++){
    for (int i = 0; i <this.arazi.length; i++) {
        cocuk.arazi[i][j] = es.arazi[i][j];
    }
}

for(int i=0; i<es.binalar.size();i++){
    mybuil = es.binalar.get(i);
    if(mybuil.parsYBoyut>SSPM_v3.caprazlamaNoktasi){
        cocuk.binalar.add(mybuil);
    }
}

//System.out.println("Caprazlandi");
//System.out.println("Yeni cocuk " + cocuk.binalar.size());
return cocuk;
}

// *****
// getFitness

public void fitness(){

    int toplamM2 = 0;
    int toplamM3 = 0;

    Building myBuil;
    for(int k = 0; k<this.binalar.size();k++) {
        myBuil = this.binalar.get(k);
        toplamM2 = toplamM2 + myBuil.binaTabanM2;
        toplamM3 = toplamM3 + myBuil.binaM3;
    }

    int farkM2 = toplamM2 - SSPM_v3.istenenTabanAlani;
    int farkM3 = toplamM3 - SSPM_v3.emsal;
    //System.out.println("farkm3 " + farkM3);

    // System.out.println("fark m2 " + farkM2); // debugging
    //System.out.println("Dolu m2 " + toplamM2); // debugging

    if(farkM2>0){

```

```

        m2Skor = (float) (1.0-(farkM2/6000.0));
    }
    else{
        m2Skor = (float) (1.0+(farkM2/10000.0));
    }
    //println("Population Score: " + score); // debugging

    if(farkM3>0){
        m3Skor = (float) (1.0-(farkM3/600000.0));
    }
    else{
        m3Skor = (float) (1.0+(farkM3/1000000.0));
    }

    m2m3 = m3Skor + m2Skor;
}

//
*****

// fitnessManzara
// Secilmis manzara yonune bakan kac kutucuk var hesaplar
// -----

public void fitnessManzara(){

    int toplamManzara = 0;
    boolean onuBos = true;

    Building myBuil;
    for(int k = 0; k<this.binalar.size();k++){
        myBuil = this.binalar.get(k); // Her bina tek tek ele alinir (Once birinci
        bina sonra ikinci bina vs...)

        if(SSPM_v3.dropManzara.getValue() == 1){ // Manzara Yonu Kuzeyse
            for(int i = myBuil.builPosX; i<myBuil.builPosX+myBuil.builXBoyut;
            i++){
                onuBos = true;
                for(int j = myBuil.builPosY-1; j> 0; j--){

                    if(this.arazi[i][j] > 20){
                        onuBos = false;
                    }
                }
                if(onuBos) {
                    toplamManzara++;
                }
            }
        }
        if(SSPM_v3.dropManzara.getValue() == 2){ // Manzara Yonu Doguysa

```

```

        for(int j = myBuil.builPosY ; j<myBuil.builPosY+myBuil.builYBoyut;
j++){
            onuBos = true;
            for(int i = myBuil.builPosX + myBuil.builXBoyut+1; i<
this.arazi.length; i++){

                if(this.arazi[i][j] > 20){
                    onuBos = false;
                }
            }
            if(onuBos) {
                toplamManzara++;
            }
        }
    }
    if(SSPM_v3.dropManzara.getValue() == 3){ // Manzara Yonu Guneyse
        for(int i = myBuil.builPosX ; i<myBuil.builPosX+myBuil.builXBoyut;
i++){
            onuBos = true;
            for(int j = myBuil.builPosY + myBuil.builYBoyut+1; j<
this.arazi[0].length; j++){

                if(this.arazi[i][j] > 20){
                    onuBos = false;
                }
            }

            if(onuBos) {
                toplamManzara++;
            }
        }
    }

    if(SSPM_v3.dropManzara.getValue() == 4) { // Manzara Yonu Batiysa
        for (int j = myBuil.builPosY; j < myBuil.builPosY +
myBuil.builYBoyut; j++) {
            onuBos = true;
            for (int i = myBuil.builPosX - 1; i > 0; i--) {

                if (this.arazi[i][j] > 20) {
                    onuBos = false;
                }
            }
            if (onuBos) {
                toplamManzara++;
            }
        }
    }
}

```

```

    }
    //System.out.println("Toplam Manzara = " + toplamManzara);
    manzaraSkor = toplamManzara;
}
//
*****

// fitnessRuzgar
// Secilmis ruzgar yonune bakan kac kutucuk var hesaplar
// -----

public void fitnessRuzgar(){

    int toplamRuzgar = 0;
    boolean onuBos = true;

    Building myBuil;
    for(int k = 0; k<this.binalar.size();k++){
        myBuil = this.binalar.get(k);    // Her bina tek tek ele alinir (Once birinci
        bina sonra ikinci bina vs...)

        if(SSPM_v3.dropRuzgar.getValue() == 1){    // Manzara Yonu Kuzeyse
            for(int i = myBuil.builPosX; i<myBuil.builPosX+myBuil.builXBoyut;
            i++){
                onuBos = true;
                for(int j = myBuil.builPosY-1; j> 0; j--){

                    if(this.arazi[i][j] > 20){
                        onuBos = false;
                    }
                }
                if(onuBos) {
                    toplamRuzgar++;
                }
            }
        }
        if(SSPM_v3.dropRuzgar.getValue() == 2){    // Manzara Yonu Doguysa
            for(int j = myBuil.builPosY ; j<myBuil.builPosY+myBuil.builYBoyut;
            j++){
                onuBos = true;
                for(int i = myBuil.builPosX + myBuil.builXBoyut+1; i<
                this.arazi.length; i++){

                    if(this.arazi[i][j] > 20){
                        onuBos = false;
                    }
                }
                if(onuBos) {
                    toplamRuzgar++;
                }
            }
        }
    }
}

```

```

    }

    }
    if(SSPM_v3.dropRuzgar.getValue() == 3){ // Manzara Yonu Guneyse
        for(int i = myBuil.builPosX ; i<myBuil.builPosX+myBuil.builXBoyut;
i++){
            onuBos = true;
            for(int j = myBuil.builPosY + myBuil.builYBoyut+1; j<
this.arazi[0].length; j++){

                if(this.arazi[i][j] > 20){
                    onuBos = false;
                }
            }

            if(onuBos) {
                toplamRuzgar++;
            }
        }
    }

    if(SSPM_v3.dropRuzgar.getValue() == 4) { // Manzara Yonu Batiysa
        for (int j = myBuil.builPosY; j < myBuil.builPosY +
myBuil.builYBoyut; j++) {
            onuBos = true;
            for (int i = myBuil.builPosX - 1; i > 0; i--) {

                if (this.arazi[i][j] > 20) {
                    onuBos = false;
                }
            }
            if (onuBos) {
                toplamRuzgar++;
            }
        }
    }
}

if(SSPM_v3.dropIklim.getValue() == 1) {
    ruzgarSkor = toplamRuzgar;
}
else{ // Soguk Iklim ruzgar coksa skor negatif
    ruzgarSkor = -1 * toplamRuzgar;
}
}

// *****
//
public float getFitness(){

```

```

    return score;
}

// *****
//

public void mutasyon(float m) {
    Building bina;
    byte mutasyonTip;
    mutasyonTip = 1; // debugging

    for (int i = 0; i < binalar.size(); i++) {
        if (parent.random(1) < m) {

            bina = binalar.get(i);
            boolean mutasyonBos = false;
            if(mutasyonTip==1){ // Mutasyon tip 1: Binayi bir saga kaydirir
                // Binanın sagi bos mu kontrol eder
                if(bina.parsPozX+bina.parsXBoyut+1<arazi.length){ // Bina X
                    Boyutunda Arazi matrisi içerisinde mi kontrol eder
                        mutasyonBos = true;
                        for(int j=0;j<bina.parsYBoyut;j++){

                            if(arazi[bina.parsPozX+bina.parsXBoyut][bina.parsPozY+j]==1){ // Binanın
                                oturacağı alan boş mu kontrol eder
                                    bos = bos && true;
                                }
                                else{
                                    bos = bos && false;
                                }
                            }
                        }
                    if(bos){
                        binaSil(i);
                        //System.out.println("bina " + i + " silindi");
                        bina.parsPozX = bina.parsPozX+1;
                        bina.builPosX = bina.builPosX+1;

                        binaYerlestir(bina.builPosX,bina.builPosY,bina.builXBoyut,bina.builYBoyut,
                            bina.parsPozX, bina.parsPozY, bina.builZBoyut, bina.yon, bina.parsXBoyut,
                            bina.parsYBoyut);
                        //System.out.println("Mutasyon gerceklesti"); // debugging
                    }
                }
            }
            if(mutasyonTip==2){ // Mutasyon Tip 2: Binayi bir sola kaydirir
                if(bina.parsPozX-1>0){ // Bina X Boyutunda Arazi matrisi
                    içerisinde mi kontrol eder
                        mutasyonBos = true;

```

```

        for(int j=0;j<bina.parsYBoyut;j++){
            if(arazi[bina.parsPozX-1][bina.parsPozY+j]==1){ // Binanın
oturacağı alan boş mu kontrol eder
                bos = bos && true;
            }
            else{
                bos = bos && false;
            }
        }
    }
    if(bos){
        binaSil(i);
        bina.parsPozX = bina.parsPozX-1;
        bina.builPosX = bina.builPosX-1;

```

```

        binaYerlestir(bina.builPosX,bina.builPosY,bina.builXBoyut,bina.builYBoyut,
        bina.parsPozX, bina.parsPozY, bina.builZBoyut, bina.yon, bina.parsXBoyut,
        bina.parsYBoyut);
    }
}

```

```

    if(mutasyonTip==3){ // Mutasyon Tip 3: Binayı bir yukari kaydirir
        if(bina.parsPozY-1>0){ // Bina Y Boyutunda Arazi matrisi
içerisinde mi kontrol eder

```

```

            mutasyonBos = true;
            for(int j=0;j<bina.parsYBoyut;j++){
                if(arazi[bina.parsPozX-1][bina.parsPozY+j]==1){ // Binanın
oturacağı alan boş mu kontrol eder
                    bos = bos && true;
                }
                else{
                    bos = bos && false;
                }
            }
        }
        if(bos){
            binaSil(i);
            bina.parsPozX = bina.parsPozX-1;
            bina.builPosX = bina.builPosX-1;

```

```

        binaYerlestir(bina.builPosX,bina.builPosY,bina.builXBoyut,bina.builYBoyut,
        bina.parsPozX, bina.parsPozY, bina.builZBoyut, bina.yon, bina.parsXBoyut,
        bina.parsYBoyut);
    }
}

```

```

    if(mutasyonTip==4){ // Mutasyon Tip 4: Binayı bir asagi kaydirir
        if(bina.parsPozY+bina.parsYBoyut+1<arazi[0].length){ // Bina Y
Boyutunda Arazi matrisi içerisinde mi kontrol eder

```

```

            mutasyonBos = true;
            for(int j=0;j<bina.parsYBoyut;j++){
                if(arazi[bina.parsPozX-1][bina.parsPozY+j]==1){ // Binanın

```

oturacağı alan boş mu kontrol eder

```
        // Kontrol
    }
    else{
        bos = false;
    }
}

if(bos){
    binaSil(i);
    bina.parsPozX = bina.parsPozX-1;
    bina.builPosX = bina.builPosX-1;

    binaYerlestir(bina.builPosX,bina.builPosY,bina.builXBoyut,bina.builYBoyut,
    bina.parsPozX, bina.parsPozY, bina.builZBoyut, bina.yon, bina.parsXBoyut,
    bina.parsYBoyut);
}
}
if(mutasyonTip == 5){ // Binayi buyut
}

// println("Binalar add parselyBoyut: " + parselyBoyut);
// binalar.add(new Building(binaPozX, binaPozY, binaXBoyut,
binaYBoyut, binaZBoyut, yon, parselXBoyut, parselyBoyut, parselPozX,
parselPozY));

// bina.yon=2;
// bina.builPosX = 1;
// bina.setBuilPosX(bina.getBuilPosX()+1);
// this.arazi

}
}
}
}
```



EK B: Araziyi Görsel Oluşturma Kodu (functions.java)

```
import org.apache.poi.ss.usermodel.*;
import org.apache.poi.xssf.streaming.SXSSFWorkbook;

import java.io.*;
public class functions {

    SXSSFWorkbook swb=null;
    Sheet sh=null;
    InputStream inp=null;
    Workbook wb=null;

    String[][] importExcel(String filepath) {
        String[][] temp;
        try {
            inp = new FileInputStream(filepath);
        }
        catch(Exception e) {
        }
        try {
            wb = WorkbookFactory.create(inp);
        }
        catch(Exception e) {
        }
        Sheet sheet = wb.getSheetAt(0);
        int sizeX = 277; // Buraya excel dosyasinin satir buyuklugu girilir
        int sizeY = 300; // Buraya excel dosyasinin kolon buyuklugu
        for (int i=0;i<sizeY;++i) {
            Row row = sheet.getRow(i);
            for (int j=0;j<sizeX;++j) {
                try {
                    Cell cell = row.getCell(j);
                }
                catch(Exception e) {
                    if (j>sizeX) {
                        sizeX = j;
                    }
                }
            }
        }
        temp = new String[sizeX][sizeY];
        for (int i=0;i<sizeY;++i) {
            for (int j=0;j<sizeX;++j) {
                Row row = sheet.getRow(i);
```

```

        try {
            Cell cell = row.getCell(j);
            if (cell.getCellType()==0 || cell.getCellType()==2 ||
cell.getCellType()==3)cell.setCellType(1);
            temp[j][i] = cell.getStringCellValue();
        }
        catch (Exception e) {
        }
    }
}
System.out.println("Excel file imported: " + filepath + " successfully!");
return temp;
}

void exportExcel(String[][] data, String filepath) {
    SXSSFWorkbook ww = new SXSSFWorkbook(100);
    Sheet sh = ww.createSheet();
    int sizeX = data.length;
    int sizeY = data[0].length;
    for (int i=0;i<sizeX;++i) {
        Row row = sh.createRow(i);
        for (int j=0;j<sizeY;++j) {
            Cell cell = row.createCell(j);
            if (cell.getCellType()==0 || cell.getCellType()==2 ||
cell.getCellType()==3)cell.setCellType(1);
            cell.setCellValue(data[i][j]);
        }
    }
    try {
        FileOutputStream out = new FileOutputStream(filepath);
        ww.write(out);
        System.out.println("Excel file exported: " + filepath + " successfully!");
    }
    catch (Exception e) {
        System.out.println("Error in saving the file...sorry!");
    }
}
}
}

```

EK C: Bina Tanımlama Kodu (building. java)

```
public class Building {
    byte builZBoyut;
    byte yon;
    int builIndex;
    int builYBoyut;
    int builXBoyut;
    int builPosX;
    int builPosY;
    int binaTabanM2;
    int binaM3;
    int parsPozX;
    int parsPozY;
    int parsXBoyut;
    int parsYBoyut;

    Building(int builX, int builY, int XBoyut, int YBoyut, byte ZBoyut, byte
binaYon, int parselXBoyut, int parselYBoyut, int parselPozX, int parselPozY){
        builZBoyut = ZBoyut;
        builYBoyut = YBoyut;
        builXBoyut = XBoyut;
        builPosX = builX;
        builPosY = builY;
        yon = binaYon;
        parsXBoyut = parselXBoyut;
        parsYBoyut = parselYBoyut;
        parsPozX = parselPozX;
        parsPozY = parselPozY;
        binaTabanM2 = builXBoyut*builYBoyut;
        binaM3 = binaTabanM2 * builZBoyut;

    }

    // Binanın arazideki X Pozisyonu
    int getBuilPosX(){
        return builPosX;
    }

    int getBuilPosY(){
        return builPosY;
    }

    void setBuilPosX(int num){
        builPosX = num;
    }
}
```

```
}  
  
byte getYon(){  
    return yon;  
}  
  
}
```



EK D: Populasyon Üretme Kodu (population.java)

```
import processing.core.PApplet;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.List;

/**
 * Population
 *
 * Populasyon tanımlanır, Populasyon tüm arazi nesnelerinin butunudur
 *
 */

// Bu class'ta Populasyon tanımlanır
//
public class Population {
    PApplet parent;
    float mutasyonOrani;           // Mutasyon Orani
    Arazi[] population;           // Populasyonun tutulduğu array
    ArrayList<Arazi> uremeHavuzu;
    int generasyon;

    // Initialize the population

    Population(PApplet p, float m, int num){
        parent = p;
        mutasyonOrani = m;
        population = new Arazi[num];
        uremeHavuzu = new ArrayList<Arazi>();
        generasyon = 0;
        // Populasyonu doldurur
        for (int i = 0; i < population.length; i++) {
            population[i] = new Arazi(parent,i);
            //System.out.println("Population number: " + i); // debugging
        }
    }

    // Calculate fitness for each creature
    public void fitness() {
```

```

    for (int i = 0; i < population.length; i++) {
        population[i].fitness();
    }
}

// *****
// Fast non dominated Sort algoritmasi
// -----

public void nondominatedSort() {

    // dominateMe[i] iyi domine eden tum cozumleri icerir
    int[] dominateMe = new int[population.length];

    // iDominate[k] k tarafından domine edilen tum cozumleri iceren liste
    List<Integer> [] iDominate = new List[population.length];

    // front[i] i.inci fronta dahil olan tum bireyleri tutmak icin liste
    List<Integer> [] front = new List[population.length+1];

    // Initialize the fronts
    for (int i = 0; i < front.length; i++) {
        front[i] = new LinkedList<Integer>();
    }

    for(int p=0; p<population.length;p++) { // Populasyonun tum uyeleri icin
        iDominate[p] = new LinkedList<Integer>();
        dominateMe[p] = 0;
    }

    for(int p=0; p<population.length-1;p++){
        for (int q = p+1; q < population.length; q++) {

            if(population[p].manzaraSkor>population[q].manzaraSkor&&population[p].m2
m3>population[q].m2m3)
            {
                iDominate[p].add(q);
                dominateMe[q]++;
            }

            if(population[p].manzaraSkor<population[q].manzaraSkor&&population[p].m2
m3<population[q].m2m3)
            {
                iDominate[q].add(p);
                dominateMe[p]++;
            }
        }
    }
    for (int p = 0; p < population.length; p++) {

```

```

        if (dominateMe[p] == 0) {
            front[0].add(p);
            population[p].rank=0;
        }
    }

    //Obtain the rest of fronts
    int i = 0;
    Iterator<Integer> it1, it2 ; // Iterators
    while (front[i].size() != 0) {
        i++;
        it1 = front[i-1].iterator();
        while (it1.hasNext()) {
            it2 = iDominate[it1.next()].iterator();
            while (it2.hasNext()) {
                int index = it2.next();
                dominateMe[index]--;
                if (dominateMe[index]==0) {
                    front[i].add(index);
                    population[index].rank=i;
                }
            }
        }
    }
}

// *****
// Fast non dominated Sort algoritmasi
//
// -----

public void nondominatedSort2(){

    // dominateMe[i] iyi domine eden tum cozumleri icerir
    int[] dominateMe = new int[uremeHavuzu.size()];

    // iDominate[k] k tarafindan domine edilen tum cozumleri iceren liste
    List<Integer> [] iDominate = new List[uremeHavuzu.size()];

    // front[i] i.inci fronta dahil olan tum bireyleri tutmak icin liste
    List<Integer> [] front = new List[uremeHavuzu.size()+1];

    // Initialize the fronts
    for (int i = 0; i < front.length; i++) {
        front[i] = new LinkedList<Integer>();
    }

    for(int p=0; p<uremeHavuzu.size();p++) { // Populasyonun tum uyeleri icin
        iDominate[p] = new LinkedList<Integer>();
    }
}

```

```

        dominateMe[p] = 0;
    }

    for(int p=0; p<uremeHavuzu.size()-1;p++){
        for (int q = p+1; q < uremeHavuzu.size(); q++) {

            if(uremeHavuzu.get(p).manzaraSkor>uremeHavuzu.get(q).manzaraSkor&&uremeHavuzu.get(p).m3Skor>uremeHavuzu.get(q).m3Skor)
            {
                iDominate[p].add(q);
                dominateMe[q]++;
            }

            if(uremeHavuzu.get(p).manzaraSkor<uremeHavuzu.get(q).manzaraSkor&&uremeHavuzu.get(p).m3Skor<uremeHavuzu.get(q).m3Skor)
            {
                iDominate[q].add(p);
                dominateMe[p]++;
            }
        }
    }
    for (int p = 0; p < uremeHavuzu.size(); p++) {
        if (dominateMe[p] == 0) {
            front[0].add(p);
            uremeHavuzu.get(p).rank=0;
        }
    }

    //Obtain the rest of fronts
    int i = 0;
    Iterator<Integer> it1, it2 ; // Iterators
    while (front[i].size()!= 0) {
        i++;
        it1 = front[i-1].iterator();
        while (it1.hasNext()) {
            it2 = iDominate[it1.next()].iterator();
            while (it2.hasNext()) {
                int index = it2.next();
                dominateMe[index]--;
                if (dominateMe[index]==0) {
                    front[i].add(index);
                    uremeHavuzu.get(index).rank=i;
                }
            }
        }
    }
}

/*
// Ureme havuzu olusturulur

```

```

public void selection() {
    // Clear the ArrayList
    uremeHavuzu.clear();

    // Calculate total fitness of whole population

    //float maxFitness = getMaxFitness();
    //System.out.println("Max Fitness: " + maxFitness);
    // Calculate fitness for each member of the population (scaled to value between
    0 and 1)
    // Based on fitness, each member will get added to the mating pool a certain
    number of times
    // A higher fitness = more entries to mating pool = more likely to be picked as a
    parent
    // A lower fitness = fewer entries to mating pool = less likely to be picked as a
    parent
    for (int i = 0; i < population.length; i++) {
        float fitnessNormal = parent.map(population[i].score, 0, maxFitness, 0, 1);
        System.out.println("Fitness Normal" + fitnessNormal);
        int n = (int) (fitnessNormal * 100); // Arbitrary multiplier
        for (int j = 0; j < n; j++) {
            uremeHavuzu.add(population[i]);
        }
    }
}*/

// Ureme
public void reproduction() {
    // Refill the population with children from the mating pool
    //System.out.println("debugging");
    uremeHavuzu.clear();
    for (int i = 0; i < population.length; i++) {
        uremeHavuzu.add(population[i]);
    }
    for (int i = 0; i < population.length; i++) {

        // Sping the wheel of fortune to pick two parents
        int a = (int)(parent.random(population.length));
        int b = (int)(parent.random(population.length));

        // Anne babayi sec
        Arazi anne = population[a];
        Arazi baba = population[b];

        // Mate their genes
        Arazi cocuk = anne.caprazla(baba);
        // Mutate their genes
        // cocuk.mutasyon(mutasyonOrani);
        // Fill the new population with the new child
        uremeHavuzu.add(cocuk);
    }
}

```

```

    }
    generasyon++;
    System.out.println("Generasyon: " + generasyon);
}

// Find highest fitness for the population
public float getMaxFitness() {
    float record = 0;
    for (int i = 0; i < population.length; i++) {
        if (population[i].getFitness() > record) {

            record = population[i].getFitness();
            System.out.println(record);
        }
    }
    return record;
}

// Ffff
public void reduce() {
    int enKucuk = 0;
    for (int i = 0; i < population.length; i++) {
        for (int j = 0; j < uremeHavuzu.size(); j++) {
            if (uremeHavuzu.get(j).rank == enKucuk && i < population.length) {
                population[i] = uremeHavuzu.get(j);
                uremeHavuzu.remove(j);
                i++;
            }
        }
        enKucuk++;
    }
}

public void crowdingDistance() {
    for (int p = 0; p < population.length; p++) {
        population[p].crowdingDistance = 0;
    }
}
}

```

ÖZGEÇMİŞ



Ad-Soyad : Yazgı Badem Aksoy
Doğum Tarihi ve Yeri : 12.11.1980 Ardahan
E-posta : yazbadem@hotmail.com

ÖĞRENİM DURUMU:

- **Lisans** : 2004, İstanbul Teknik Üniversitesi, Mimarlık Fakültesi, Mimarlık
- **Yükseklisans**: 2008, İstanbul Teknik Üniversitesi, Bilişim Anabilim Dalı, Mimari Tasarımda Bilişim Programı

DOKTORA TEZİNDEN TÜRETİLEN YAYINLAR VE SUNUMLAR:

Kitap içinde Bölüm

- **Aksoy,Y.B.**, Çağdaş, G., Balaban, Ö. (2015). A Model for Sustainable Site Layout Design of Social Housing with Pareto Genetic Algorithm: SSPM, In Celani, G., Sperling, D.M., Franco,J.M.S.(Eds.), *The Next City: New Technologies and the Future of the Built Environment*, (Vol.527, pp: 113-133. Brasil: Springer

Uluslararası Hakemli Dergide Makale

- **Aksoy, Y.B.,** Çağdaş, G., Balaban, Ö. (2016). Sürdürülebilir Toplu Konut Yerleşmesi Tasarımı için Pareto Genetik Algoritmaya Dayalı Bir Model Önerisi: SSPM, *Megaron Dergisi*, Yıldız Teknik Üniversitesi.(2016 yılı içindeki sayılardan birinde yayınlanmak üzere, 04.04.2016 tarihinde kabul yazısı alınmıştır.)

Uluslararası Sempozyumda Bildiri

- **Aksoy, Y.B.,** Çağdaş, G. (2014). A Model for Sustainable Site Layout Design with Pareto Genetic Algorithm: SSPM. *In Fusion, Proceedings of the 32nd International Conference on Education and research in Computer Aided Architectural Design in Europe*, (pp:227-238), UK, Northumbria University, September 10-12.

Ulusal Sempozyumda Bildiri

- **Aksoy, Y.B.,** Çağdaş, G. (2014). Pareto Genetik Algoritma ile Sürdürülebilir Mimari Tasarım için Bir Model Önerisi: SSPM. *8.Mimarlıkta Sayısal Tasarım Ulusal Sempozyumu*, (s.103-114), Türkiye, İzmir İleri teknoloji Üniversitesi, 26-27 Haziran.

Araştırma Projesi

- **Aksoy, Y.B.,** Çağdaş, G. (Yürütücü), “Sürdürülebilir Toplu Konut Yerleşmesi Tasarımı için Pareto Genetik Algoritmaya Dayalı Bir Model Önerisi: SSPM”, İTÜ Bilimsel Araştırma Projeleri Lisansüstü Tezleri Destekleme Programı (BAP) (Başlangıç tarihi: 10 Mayıs 2012, No: 36394)