



Optimizasyon Teknikleri

Prof. Dr. Bilal ALATAŞ

Sezgisel Optimizasyon

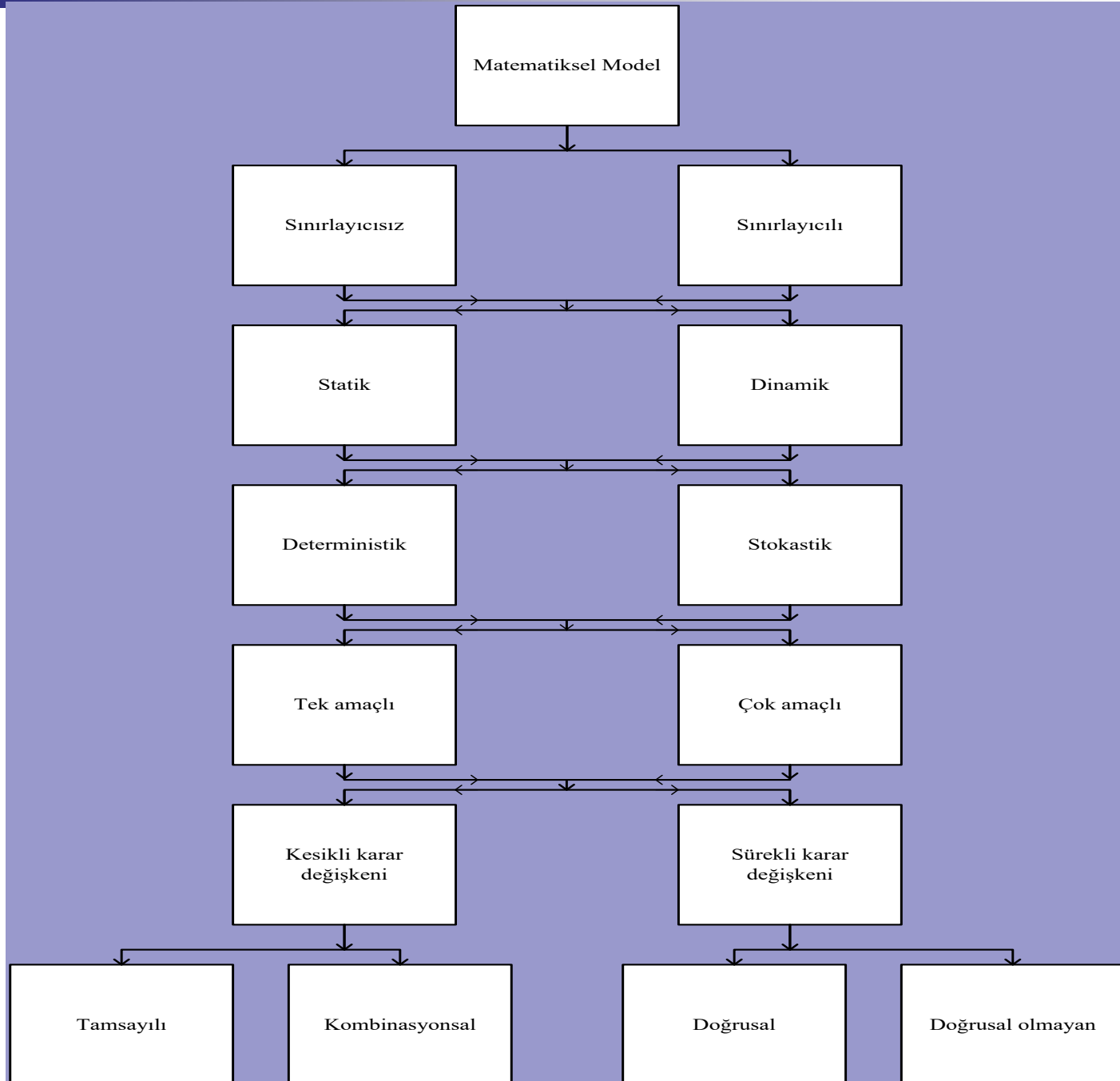
Ders Notları - 7

İÇERİK

- Optimizasyon özet
- Klasik Optimizasyon özet
- Sezgisel Optimizasyona Giriş
 - Tepe Tırmanma Algoritması
 - Tabu Arama Algoritması
 - Isıl İşlem Algoritması
- Metasezgisel algoritmaların tanımı
- Metasezgisel algoritmaların sınıflandırılması
- Çok Noktalı (Popülasyon Tabanlı) Algoritmalara örnekler

OPTİMİZASYON

- Optimizasyon
 - *karar değişkenleri*
 - *amaç fonksiyonu*
 - *sınırlayıcılar*
- Matematiksel modeller



İKİ TEMEL BASAMAK

- Problem modelinin kurulması
- Çözüm üretmek için modelin kullanılması

Klasik Optimizasyon Algoritmaları

- Sistemin modeli ve amaç fonksiyonu için matematiksel modellere ihtiyaç duymaktadır.
- Karmaşık sistemler için matematiksel modelin kurulması çoğu zaman zordur. Model kurulsa bile çözüm zamanı maliyeti çok yüksek olduğundan kullanılamamaktadır.
- Büyük ölçekli kombinyonsal ve doğrusal olmayan problemlerde yetersizdir.
- Verilen bir probleme bir çözüm algoritması uyarlamada etkin değildir.
 - Varsayımlar
- İlgilenilen problem algoritmanın onu idare edeceği şekilde modellenir. Klasik optimizasyon algoritmalarının çözüm stratejisi genellikle amaç ve sınırlayıcıların tipine ve problemi modellemede kullanılan değişkenlerin tipine bağlıdır.

Klasik Optimizasyon Algoritmaları (devam)

- Bunların etkinliliği aynı zamanda problem modellemede çözüm uzayı, karar değişken sayısı ve sınırlayıcı sayısına oldukça bağlıdır.
- Diğer önemli bir eksiklik ise farklı tipte karar değişkenleri, amaç ve sınırlayıcıların olması durumunda problem formülasyonlarına uygulanabilecek genel çözüm stratejileri sunmamalarıdır.

Özetle...

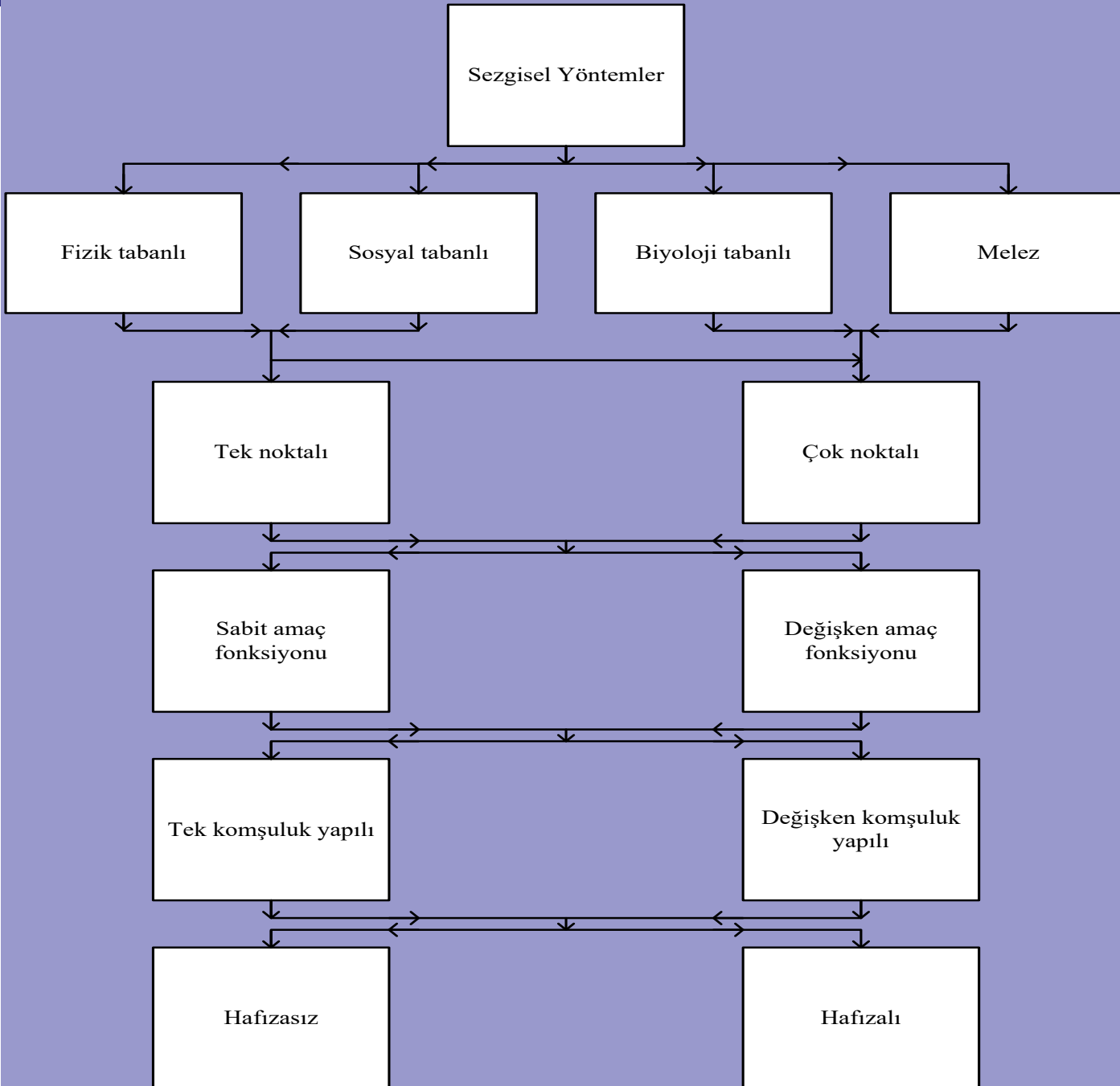
- Hiç biri robust (Yeni problem için yeni bir yöntem tasarlanmalı) değildir.
- Problem matematiksel terimlerle çok iyi tanımlanmalıdır.
- Özellikle gerçek dünya problemlerini çok iyi tanımlamak mümkün değildir.
- Küresel optimumu garanti edebilir ama aşırı derecede hesaplama zamanına ihtiyaç duyarlar.
- Kısa sürede bölgesel optimaya yakınsama eğilimindedirler.

AMAÇ

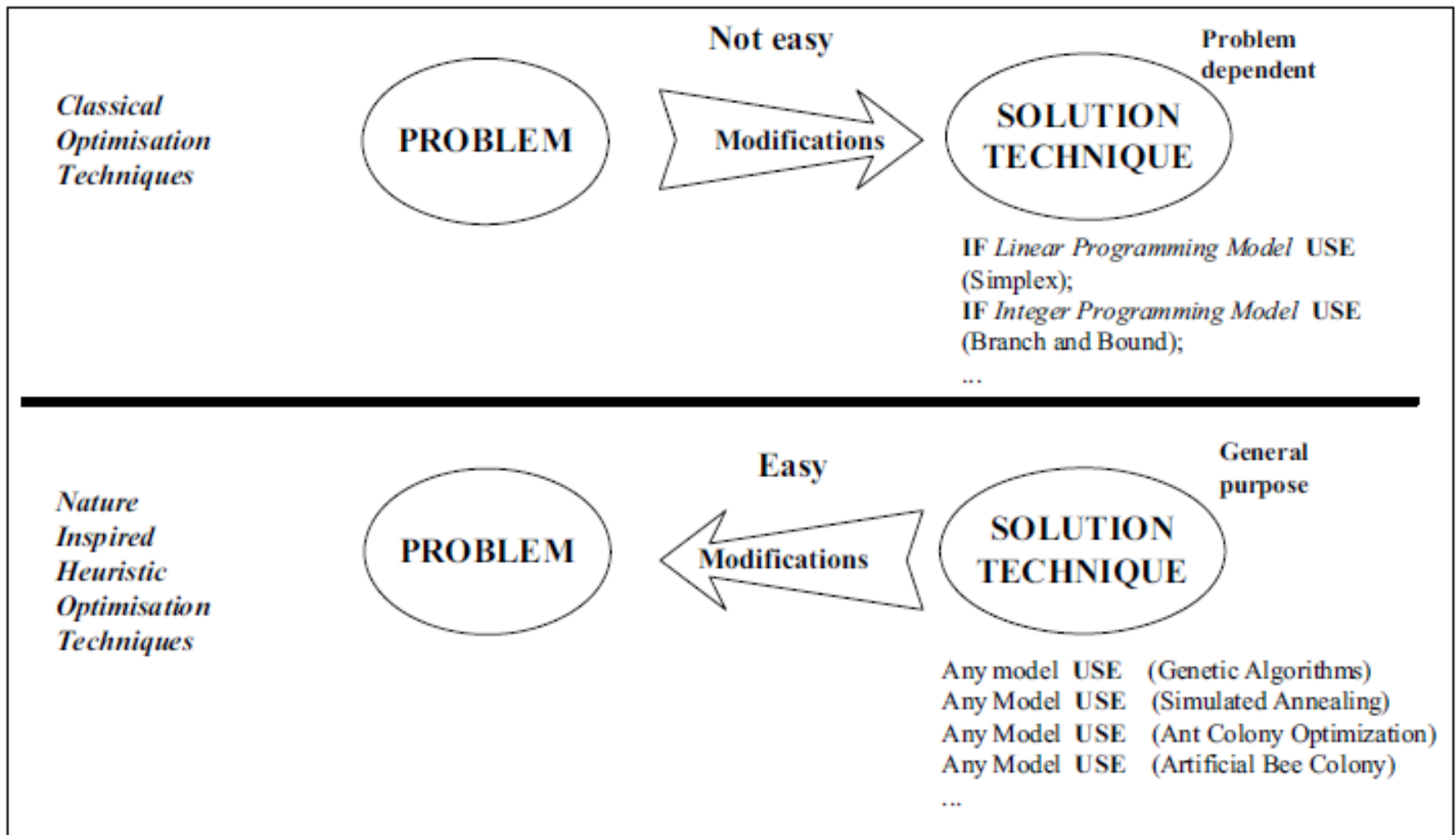
- Erken yakınsamadan kaçış,
- Bölgesel araştırmada optimuma hızlı yakınsama,
- Dinamik gerçek dünya optimizasyonu.

Klasik sezgisel ve genel amaçlı sezgisel optimizasyon algoritmaları

- Hesaplama gücü iyidir.
- Dönüşümleri kolaydır.



Klasik ve sezgisel optimizasyon algoritmalarının karşılaştırılması





KLASİK SEZGİSEL ARAMA ALGORİTMALARI

ANAHAAT

- GİRİŞ
- TEPE TIRMANMA ALGORİTMASI
- ISIL İŞLEM ALGORİTMASI
- TABU ARAMA ALGORİTMASI


ANAHAAT

- GİRİŞ
- TEPE TIRMANMA ALGORİTMASI
- ISIL İŞLEM ALGORİTMASI
- TABU ARAMA ALGORİTMASI

OPTİMİZASYON-ARAMA

- **Optimizasyon**, problemin karar değişkenlerinin mümkün tüm kombinasyonları arasından en iyi performansı (en iyi amaç fonksiyonu değerini) veren kombinasyonun bulunmasıdır. Bu amaçla literatürde birçok optimizasyon algoritması önerilmiştir. Bu algoritmaların çoğu sistemin modeli ve amaç fonksiyonu için matematiksel modellere ihtiyaç duymaktadır. Karmaşık sistemler için matematiksel modelin kurulması çoğu zaman zordur. Model kurulsa bile çözüm zamanı maliyeti çok yüksek olduğundan kullanılamamaktadır.
- **Arama** algoritmaları, algoritmanın yapısına uygun olarak seçilen bir başlangıç çözümünden ya da çözümlerinden başlar ve farklı yapılar kullanarak yeni çözümlere ulaşırlar.

- Ayırık optimizasyon problemleri için genel olarak, tanımlanması kolay çözümü zor problemlerdir denilebilir. Ayırık optimizasyon problemlerini genellikle belirli bir matematiksel fonksiyon olarak ifade etmek güçtür. Bu nedenle, bu türdeki problemleri klasik optimizasyon yöntemleri ile çözmek zordur. Bilinen klasik algoritmalar sadece bu türdeki küçük boyutlu problemleri çözebilir. Bu durumda büyük boyutlu optimizasyon problemleri için, kabul edilebilir sürede optimuma yakın çözümler verebilen heuristik (sezgisel) algoritmalara ihtiyaç duyulur.

- 
- GİRİŞ
 - TEPE TIRMANMA ALGORİTMASI
 - ISIL İŞLEM ALGORİTMASI
 - TABU ARAMA ALGORİTMASI

Tepe Tırmanma

- Sezgisel arama işlemini gerçekleştirmenin en kolay yolu hill climbing olarak adlandırılan bir arama stratejisidir. Tepe tırmanma stratejisinde arama işleminde o anda aktif olan düğüm genişletilir ve çocuk düğümler değerlendirilir. Bunlardan **en iyi** değerle sahip olan düğüm bir sonraki genişletme işleminde kullanılmak üzere seçilir. **Genişletme, ilk en iyi bulunduğunda ya da en iyi bulunduğunda olabilir.**
 - **Sezgi:** Daima daha iyi bur duruma hareket et
- Arama işlemi çocukların hepsinden daha iyi bir duruma erişildiğinde sona erer. Bu sistemde hatalı sezgisel değerler sonsuz yollara, dolayısıyla arama işleminin başarısızlıkla sonuçlanmasına neden olur. Bu yöntemin başarısı için sezgisel değerlerinin doğru belirlenmesi büyük önem taşır. Algoritmada önceki basamaklarla ilgili bilgi tutulmamaktadır, dolayısıyla hata durumu ile karşılaşıldığında eldeki yanlış çözüm üzerinde bir düzeltme yapılabilmesi söz konusu değildir. Algoritma tüm çocuklardan daha iyi durumda bir düğüme erişildiğinde sona erdiğinden yerel maksimum noktaları da sorun yaratmaktadır. İyi mutlak açıdan en iyi olmak zorunda değildir. Tepe tırmanma algoritması yerel ve global maksimum arasındaki ayrımı yapamamaktadır

Tepe Tırmanma

function TEPE-TIRMANMA(*problem*) **returns** bir çözüm durumu

inputs: *problem* // bir problem

local variables: *mevcut* // bir düğüm

sonraki // bir düğüm

mevcut <= DÜĞÜM_YAP(BAŞLANGIÇ-DURUM[*problem*])

loop do

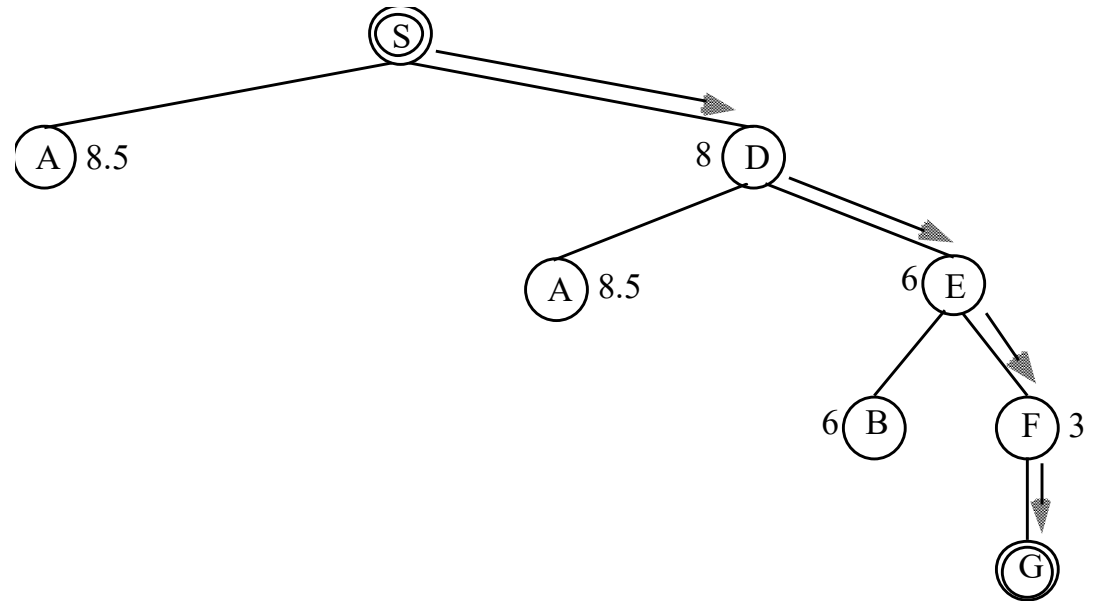
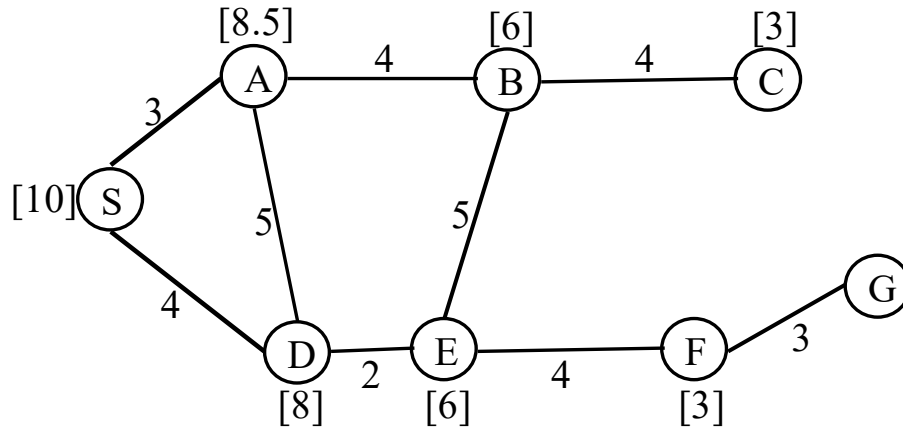
sonraki <= *mevcut*'un en yüksek değerli çocuğu

if DEĞER(*sonraki*) < DEĞER(*mevcut*) **then return** *mevcut*

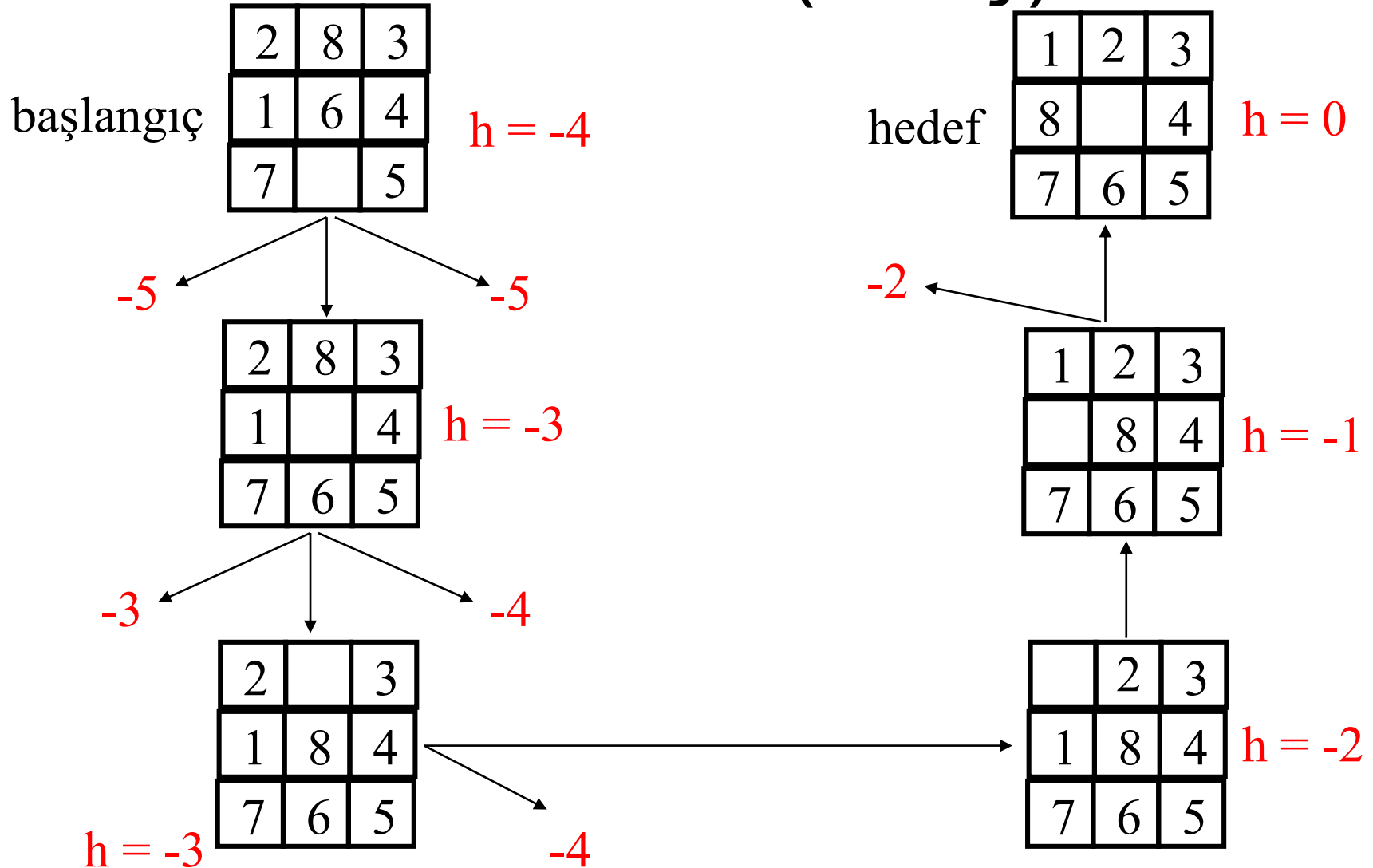
mevcut <= *sonraki*

end

Küçük Bir Örnek...(En kısa yol)

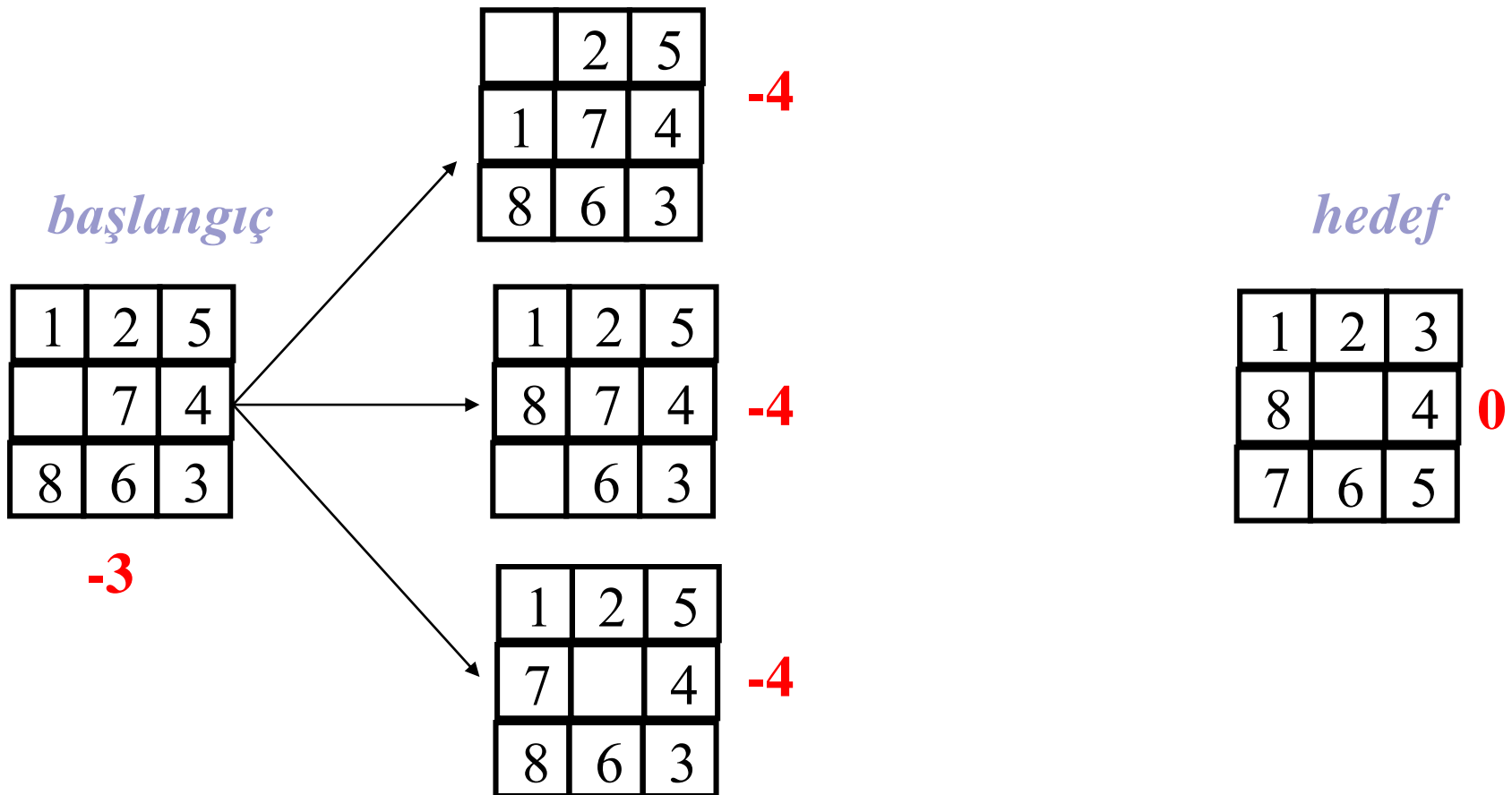


Bir Örnek Daha...(8-taş)



$$f(n) = -(\text{yerinde olmayan taş sayısı})$$

Başka Bir Örnek Daha...?!?!

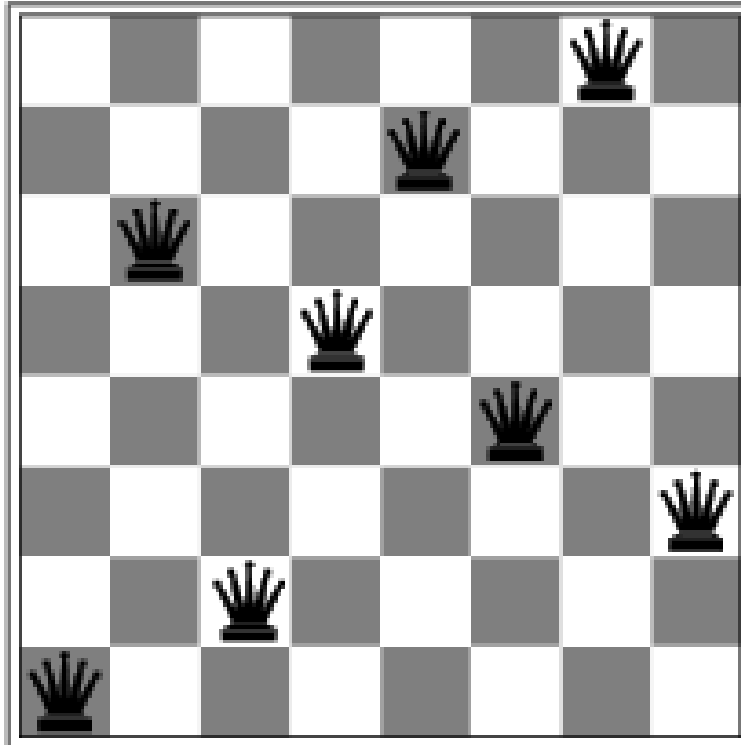


Bir Örnek Daha...(8-vezir)

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	♚	13	16	13	16
♚	14	17	15	♚	14	16	16
17	♚	16	18	15	♚	15	♚
18	14	♚	15	15	14	♚	16
14	14	13	17	12	14	12	18

- h = birbirlerini direkt ya da dolaylı yiyen vezir sayısı
- $h = 17$ (yukarıdaki durum için)

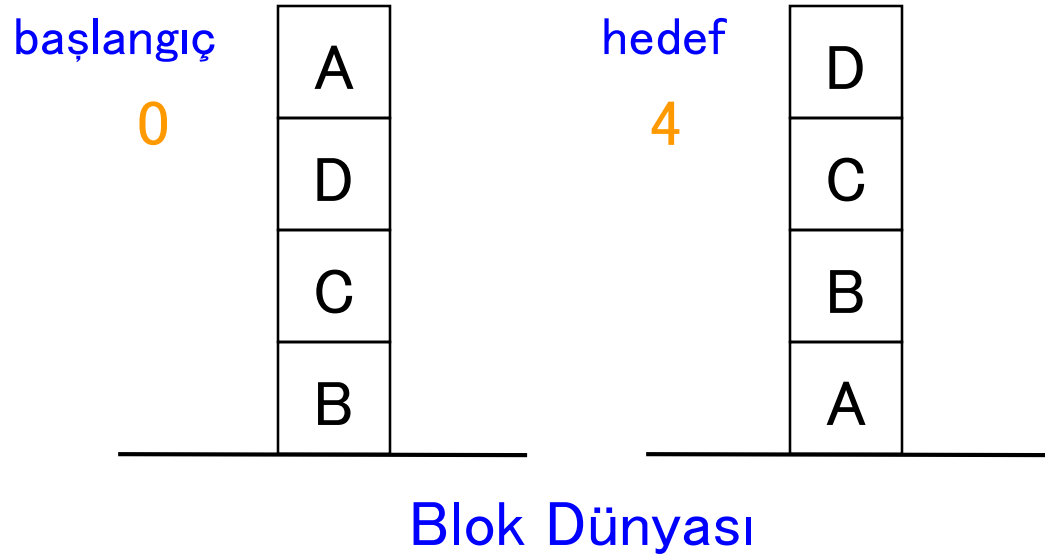
Devam...?!?!



- $h = 1$ li lokal bir minimum



Başka Bir Örnek (Blok Dünyası)

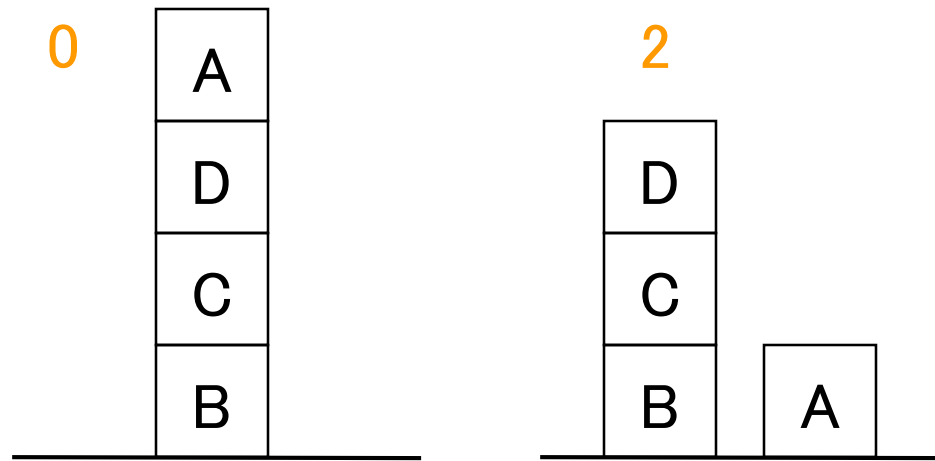


Lokal sezgisel:

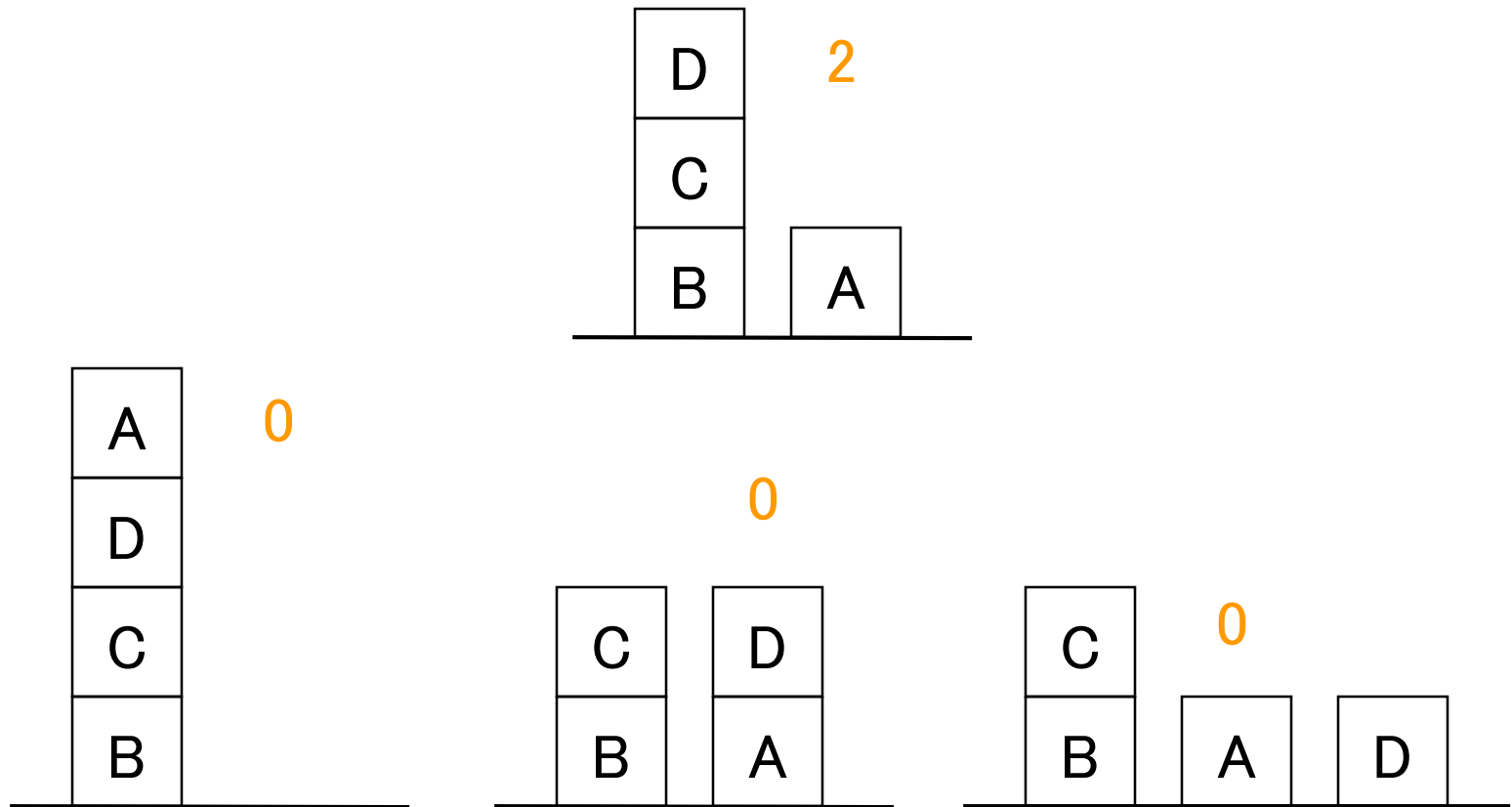
+1 üzerinde olması düşünülen blokta yer alan her blok için.

-1 yanlış yerde yer alan her blok için.

Devam...

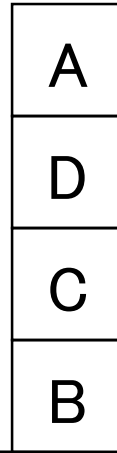


Devam...?!?!



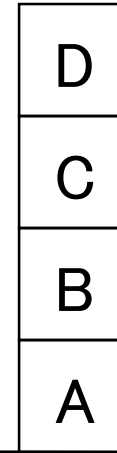
başlangıç

−6



hedef

6

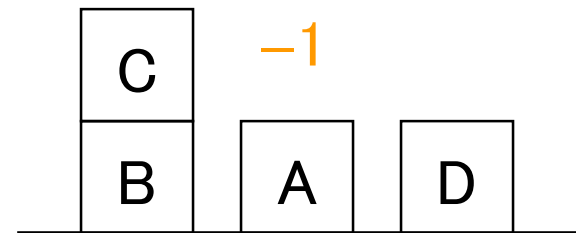
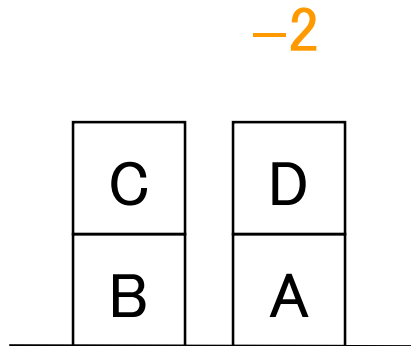
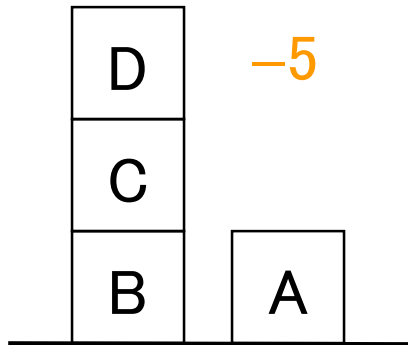
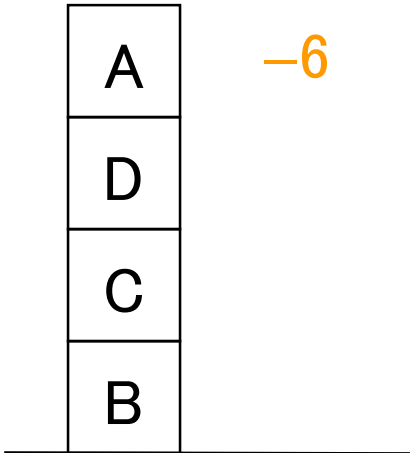


Block Dünyası

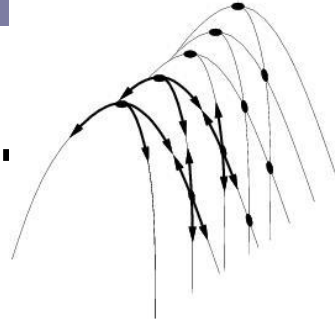
Global sezgisel:

Doğru destek yapısına sahip her blok için : +1 destek
yapısındaki her bloğa.

Yanlış destek yapısına sahip her blok için : −1 t destek
yapısındaki her bloğa

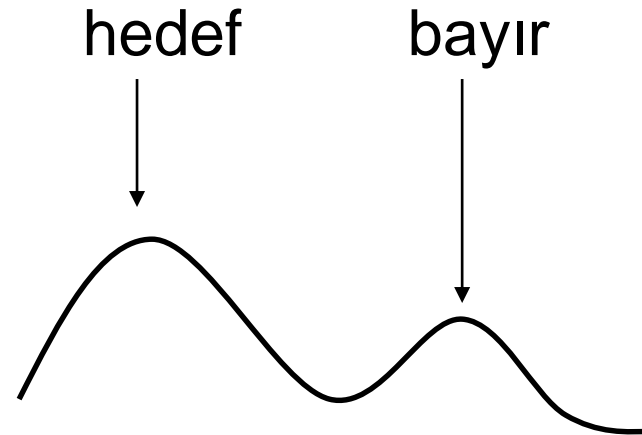


Eksiklikler.



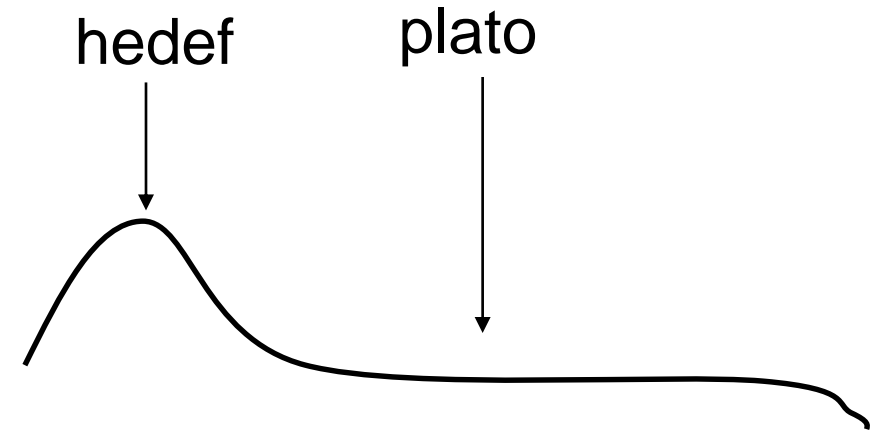
■ Bayır

- Greedy algoritmanın gezinti yapması zor olan lokal maksimumlar sırası



■ Plato (Düz alan)

- Değerlendirme fonksiyonunun düz olduğu durum uzaylarının alanı



Algoritmaya Yapılan Modifikasyonlar

- Stokastik tepe tırmanma
 - Yukarı çıkışlar arasında gelişigüzel seçim
 - Seçim olasılığı yukarı çıkışın dikliğiyle değişebilir.
- İlk-seçimli tepe tırmanma
 - Stokastik tepe tırmanmayı, daha iyi bulunana kadar çocukları gelişigüzel üreterek kullanma
- Rassal-tekrar başlatmalı tepe tırmanma
 - Algoritma, eğer tek tepe varsa çalışır. Lokal maksimumlara takılıp kalmayı önlemek için ve bulunan çözümü iyileştirmek için
 - Tekrarlı olarak tepe tırmanmayı uygula (çoklu tepe varsa tekrar başlatılır)

Sonuç

- Tepe tırmanma **lokal** bir yöntemdir, bir sonraki adımda ne yapacağını sadece seçimlerinin “anlık” sıralarına bakarak karar verir.
- **Global bilgi** sezgisel fonksiyona kodlanabilir.
- Büyük, inişli çıkışlı problem uzayları için çok etkili olabilir. Global sezgisel ile hesapsal karmaşıklık artabilir.
- Diğer yöntemlerle birleştirildiğinde kullanışlı olabilir.

ANAHAAT

- GİRİŞ
- TEPE TIRMANMA ALGORİTMASI
- ISIL İŞLEM ALGORİTMASI
- TABU ARAMA ALGORİTMASI

Isıl işlem algoritması

- Isıl işlem algoritması, katıların belirli bir başlangıç sıcaklığından başlayarak yavaş yavaş soğutulduğu tavlanma sürecinin benzetimi olan stokastik bir arama algoritmasıdır. Kirkpatrick, Gelatt ve Vecchi (1983) ve Cerny (1985) tarafından ayrı ayrı önerilmiştir. Günümüze kadar, bilgisayar tasarımı, görüntü işleme, moleküler fizik ve kimya, çizelgeleme gibi farklı alanlardaki bir çok optimizasyon problemine uygulanmıştır. Son yıllarda birçok araştırmacı bu algoritmayı kombinatoriyal optimizasyon problemlerinde etkili olarak kullanmaktadır.

Isıl işlem algoritması

- Tavlama terimi fiziksel olarak, ısı banyosundaki bir katının yüksek enerji durumlarından başlayarak daha düşük enerji durumlarının elde edilme sürecini temsil etmektedir. Bu süreç çok genel olarak iki işlemden oluşmaktadır:
 - Isı banyosunun başlangıç sıcaklığının katının eriyebileceği bir değere kadar yükseltilmesi.
 - Katılar düşük enerjili durumda, yani düşük sıcaklıkta daha kararlıdır. Yani katıların parçacıkları düşük sıcaklıklarda daha düzenlidir. Bundan dolayı katının parçacıkları kendini düzenleyene kadar ısı banyosunun sıcaklığının giderek azaltılması.

Isıl işlem

- Katının sıvı durumunda tüm parçacıkları gelişi güzel hareket ederler. Katı durumda ise bir kafes şeklinde düzenlenirler. Bu durumda sistemin enerjisi en azdır ve bu duruma yer durumu denmektedir. Bir katının yer durumu, sıcaklık yeteri kadar yükseltilmiş ve soğutma da yeteri kadar yavaş yapılmışsa elde edilir. Aksi halde katının bulunduğu durum yarı kararlı bir durumdur. Eğer soğutma işlemi çok hızlı gerçekleşirse, kristal yapıda bozukluklar ve düzensizlikler ortaya çıkacaktır. Bu nedenle soğutma işleminin dikkatle yapılması gerekmektedir.

Isıl işlem algoritması

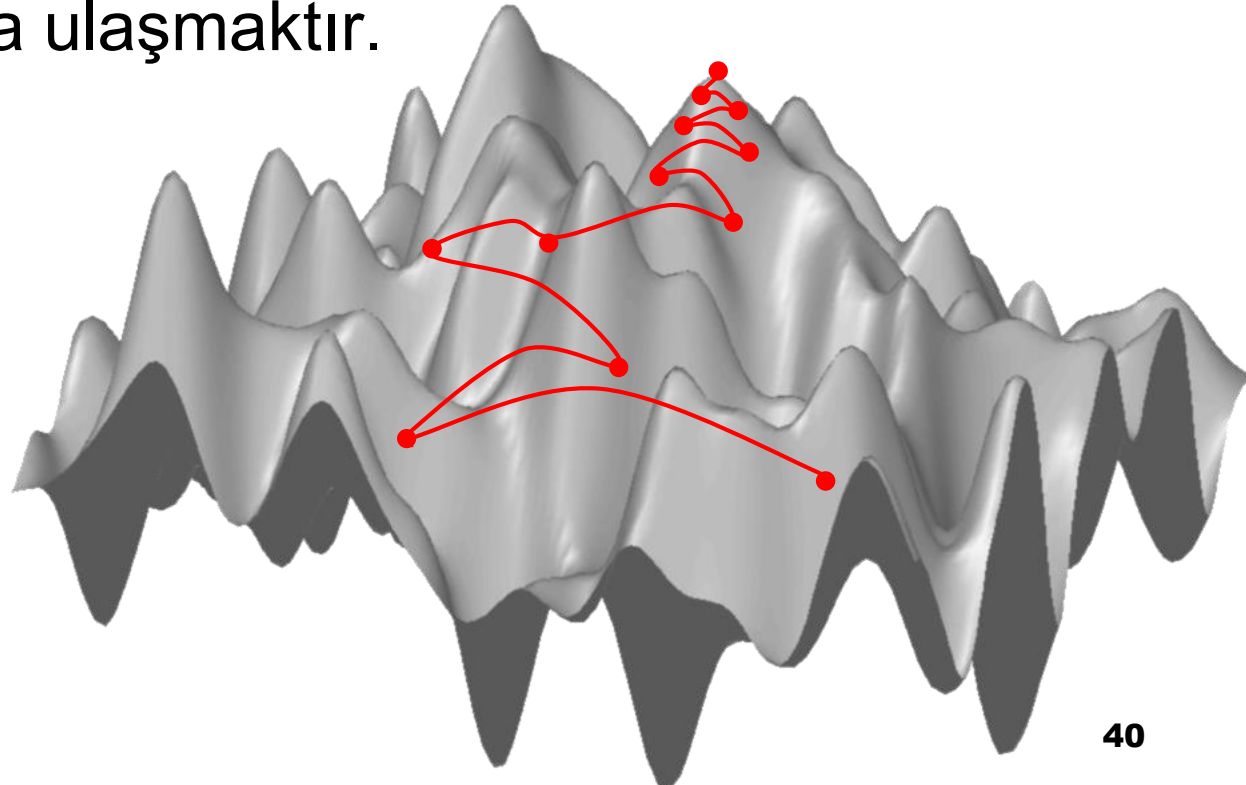
- Optimizasyon problemleriyle Isıl İşlem arasındaki benzerlikler aşağıdaki gibi gösterilebilir:
 - Katının farklı fiziksel durumları problemdeki mümkün çözümlere,
 - Sistemin enerjisi amaç fonksiyonuna,
 - Bir durumun enerjisi bir çözümün amaç fonksiyonu değerine,
 - Yarı kararlı durum yerel en iyi çözüme,
 - Yer durumu genel en iyi çözüme karşılık gelir.

Isıl işlem

- Katıların bu tavlama sürecinde geçirdiği fiziksel durumları benzetebilmek için Metropolis vd (1953) tarafından Monte Carlo tekniği üzerine dayalı olarak "Metropolis Algoritması" geliştirilmiştir. Algoritmada i durumunda bulunan katının enerjisi E_i iken, bir sonraki j duruma geçen katının enerjisi E_j 'dir. Eğer j durumundaki enerji i durumundaki enerjiden küçük veya eşitse, j durumu yeni mevcut çözüm olarak kabul edilir. Aksi takdirde j durumu aşağıdaki formüle göre elde edilen olasılık değeri ile kabul edilir. Uniform dağılımdan $[0,1]$ aralığında rasgele bir δ sayısı üretilir ve bu aşağıda verilen olasılık değerinden küçükse yeni çözüm, mevcut çözüm olarak kabul edilir. Aksi takdirde, mevcut çözüm değiştirilmez. Bu olasılık değeri "**Metropolis Kriteri**" olarak anılır. Burada k_B fiziksel tavlama sürecinde "Boltzman Sabiti" olarak bilinen fiziksel bir sabiti ifade eder.

$$P(E) = \exp\left[\frac{E_i - E_j}{k_B * T}\right]$$

- Yani, yeni çözüümü sonraki iterasyona taşımak istiyorsak, ya eski çözümden daha iyi olması ya da daha kötü olmasına rağmen stokastik kuralı tatmin etmesi gerekmektedir. Stokastik kuralın tatmin edilmesinin sebebi, optimizasyon sürecinin yerel minimumdan kurtulmasını sağlamasıdır. Bu da Isıl işlemin ana özelliklerinden birisidir. Çünkü, Isıl işlemde hedef yerel optimumdan kurtulup genel optimuma ulaşmaktır.



- Eşitliğe göre, yüksek sıcaklıklarda tüm enerji durumları için $P(E)$, 1'e yakınsar. Düşük sıcaklıklarda bile sistemin yüksek enerji seviyesine sahip olması küçük bir olasılıkla görülebilir. Bu nedenle enerjilerin istatistiksel dağılımı, sistemin bir yerel enerji minimumundan çıkmasına izin verir

$$P(E) = \exp\left[\frac{E_i - E_j}{k_B * T}\right]$$

Isıl işlem algoritması

- Isıl işlem algoritması iteratif bir algoritmadır, yani algoritma çözüm uzayında sayıların vektörü formundaki tek bir çözümü sürekli olarak geliştirme şeklinde çalışır. Isıl işlem algoritmasının amacı, tüm mümkün çözüm noktalarının bir alt kümesinde (S) tanımlanmış bir $f(x)$ fonksiyonunu eniyileyecek bir x çözümü bulmaktır. Isıl işlem algoritması rassal olarak seçilen bir başlangıç çözümüyle aramaya başlar. Bundan sonra uygun bir mekanizma ile bu çözüme komşu bir çözüm seçilir ve $f(x)$ 'de meydana gelen değişim hesaplanır. Eğer değişim istenilen yönde ise komşu çözüm mevcut çözüm olarak alınır. Eğer istenen yönde bir değişim sağlanmamışsa, Isıl işlem algoritması bu çözümü "Metropolis Kriteri" ile elde edilen olasılık değeri ile kabul eder. Amaç fonksiyonunda ters yönde bir değişim yaratan bir çözümün belli olasılık değeri ile kabulü, Isıl işlem algoritmasının yerel en iyi noktalardan kurtulmasını sağlamaktadır. Yukarıdaki olasılık değerine göre T değeri yüksek olduğunda amaç fonksiyonunda meydana gelen artışların çoğu kabul edilecektir. T değeri azaldıkça kabul edilme oranı da azalacaktır. Bu nedenle Isıl işlem algoritmasında yerel noktalara takılmamak için başlangıç sıcaklık değeri yüksek seçilerek yavaş yavaş azaltma yoluna gidilmektedir. Isıl işlem algoritması, bir taraftan sıcaklık yavaş yavaş azaltılırken, her sıcaklık değerinde belli sayıda hareket deneyerek arama işlemini sürdürür.

Algoritma temel adımları

function ISIL_ISLEM(*problem*, *plan*) **returns** bir çözüm durumu

input: *problem*, bir problem

plan, zamandan sıcaklığa bir haritalama

local variables: *mevcut*, bir düğüm.

sonraki, bir düğüm.

T, düşen adımları kontrol eden bir “sıcaklık”

mevcut \leftarrow DUGUM_OLUSTUR(İBAŞLANGIC_DURUM[*problem*])

for *t* \leftarrow 1 **to** ∞ **do**

T \leftarrow *plan*[*t*]

if *T* = 0 **then return** *mevcut*

sonraki \leftarrow *mevcut*ün gelişigüzel seçilen bir çocuğu

$\Delta E \leftarrow$ DEGER[*sonraki*] - DEGER[*mevcut*]

if $\Delta E > 0$ **then** *mevcut* \leftarrow *sonraki*

else *mevcut* \leftarrow *sonraki* sadece $e^{\Delta E / T}$ olasılığıyla

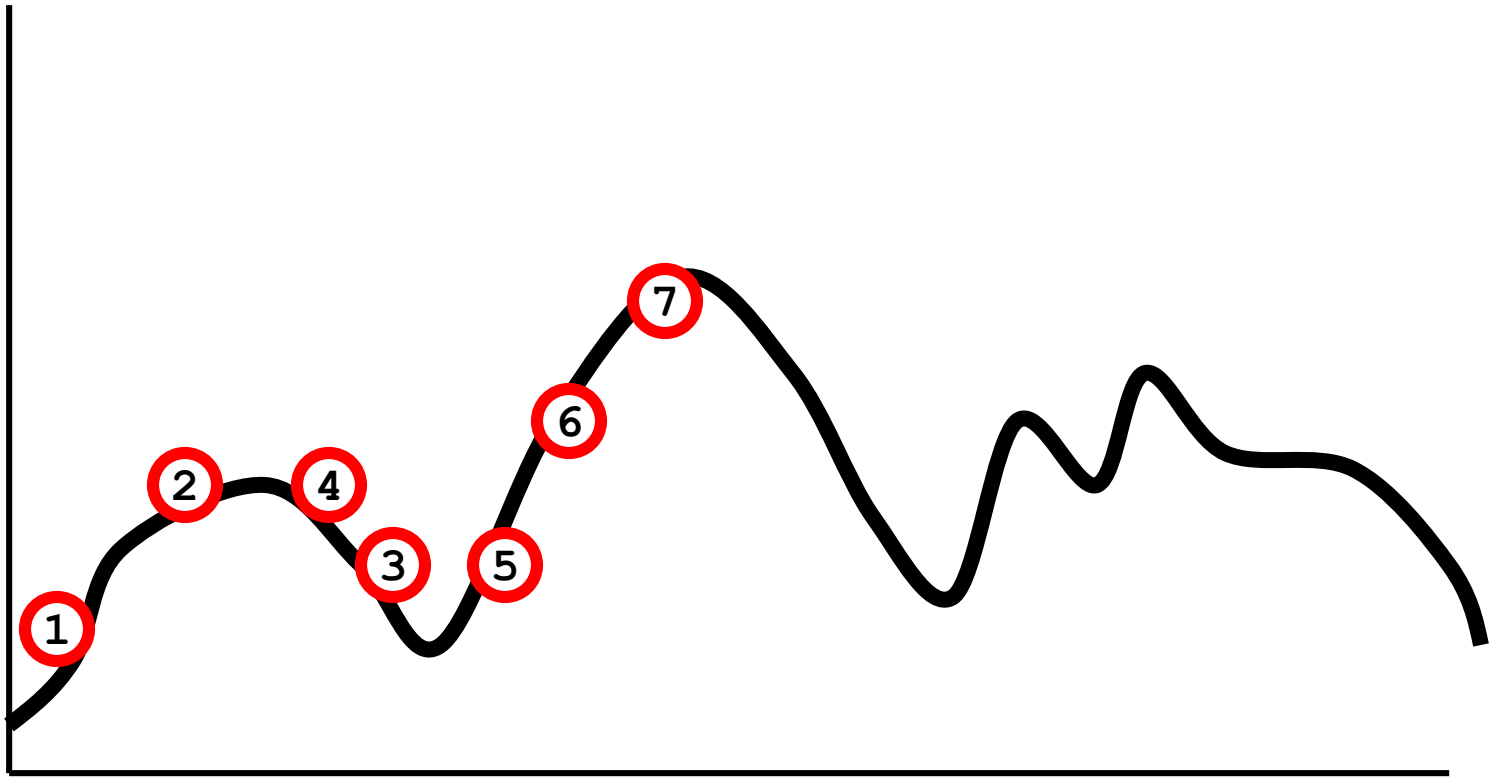
Algoritma

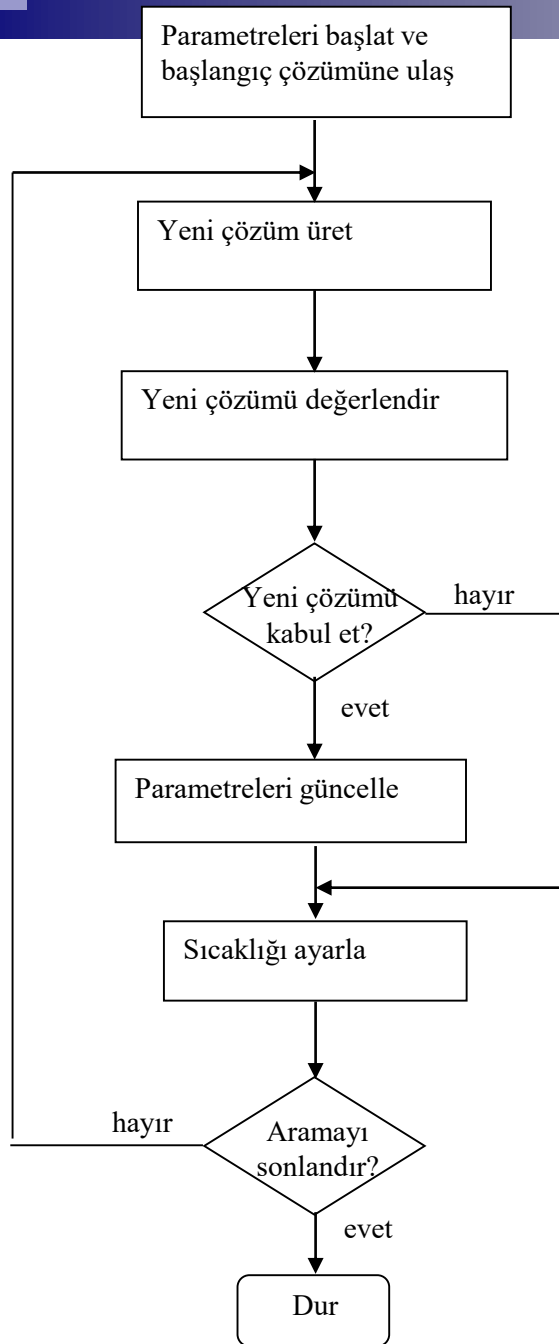
- Algoritma, önceden tanımlanmış iterasyon sayısına bağlı olarak da çalışabilir. İlk önce rassal olarak başlangıç çözüm kümeleri üretilmekte ve kaç iterasyonda sonlandırılacağı belirlenmektedir. Daha sonra Isıl işlem en yüksek ısı (100) ile başlamakta ve $t=0.995t$ fonksiyonu ile ısı 0,01'den küçük olana kadar soğutulmaktadır. Böylece, seçilen ve işlenen çözüm kümesi, tekrardan çözüm kümeleri içine diğerleriyle kıyaslanmak için konmaktadır. Daha sonra çözüm kümeleri içindeki diğer herhangi bir çözüm rassal olarak seçilmekte ve işlenmektedir.


- Isıl işlem algoritmasının performansı büyük oranda seçilen soğutma planına bağlıdır. Günümüze kadar çok çeşitli soğutma planları önerilmiştir. Önerilen en eski plan Kirckpatrick ve arkadaşlarının fiziksel tavlama ile olan benzerliğe dayanarak ileri sürdükleri plandır. Bu tavlama planına göre, maddenin sıvı safhaya ulaştığında tüm parçacıkların rasgele düzenlenmesini taklit etmek için, T sıcaklık parametresinin başlangıç değeri, denenen tüm hareketler kabul edilecek kadar yüksek seçilmiştir. Sıcaklık parametresinin değerini azaltmak için ise, oransal bir sıcaklık fonksiyonu kullanarak sabit bir r için, $T(t+1)=r.T(t)$ dikkate alınmıştır. Burada, r değeri 1' den küçük fakat 1'e yakın bir sabittir ve genellikle değeri 0.8 ile 0.99 arasında bir değer almaktadır. Bu sıcaklık fonksiyonu ile sıcaklık parametresinin değeri, sıfıra yaklaştıkça daha da yavaş azalmaktadır. Sıcaklık parametresinin her değerinde gerçekleştirilecek yeterli tekrar sayısı sabit bir üst sınıra göre belirlenerek problemin, fiziksel tavlamadaki ısı dengeye karşılık gelen bir denge durumuna ulaşması amaçlanmaktadır. Bu tavlama planı ile, sıcaklık parametresinin her değerinde elde edilen çözüm, belli sayıda ardışık sıcaklık değişimleri boyunca aynı kalırsa Isıl işlem algoritması durdurulmaktadır. Buna göre elde edilen son durum, fiziksel tavlamadaki *donma durumu* (*frozen state*) karşılık gelmektedir.

Isıl işlem

Tepe tırmanma algoritmasının aşağı inişleri de olabilen versiyonudur...





- 
- GİRİŞ
 - TEPE TIRMANMA ALGORİTMASI
 - ISIL İŞLEM ALGORİTMASI
 - TABU ARAMA ALGORİTMASI

TABU ARAMA ALGORİTMASI

- Tabu Arama (TA) algoritması ilk defa F. Glover tarafından insan hafızasının çalışmasından esinlenilerek önerilmiş bir yerel arama yöntemidir. Tabu araştırması temel olarak belirli bir problem üzerine elde edilen ilk çözüm etrafındaki komşuluklar oluşturmaktır. Komşuluk mekanizması ele alınarak birbirini izleyen seçilmiş çözümlerden daha iyi olanı Tabu Listesi olarak atanır.

TABU ARAMA ALGORİTMASI

- Tabu arama algoritmasında tabu listesi olarak oluşturulan ilk aday çözüm ve değişken komşu çözüm sayısı, bir tür tabulaştırma görevi yapmaktadır. Kötü sonuç veren bölgelerde daha fazla işlem yapılmaması (bu elemanların komşu sayılarının azaltılması), istenen çözüme daha az hesaplama ile, dolayısıyla daha hızlı ulaşmayla sağlamaktadır. İyi sonuç veren parametrelerin bir sonraki iterasyonda komşu sayıları artmakta, böylece algoritmanın verimliliği de artmış olmaktadır.

TABU ARAMA ALGORİTMASI


- Tabu listesinin en önemli özelliklerinden birisi, mevcut tabu listesinin aday komşu çözümler ile karşılaştırıldıktan sonra bir sıralama ve karşılaştırma işlemi yaparak kendisini yenileyebilmesidir. Bu amaçla, geliştirilen algoritmada, önceki döngülerde elde edilen çözümlerin listesinin tutulduğu bir tabu listesi oluşturulmuştur. Eğer bir komşu çözüm adayı, tabu listesinde yer alan bir çözümle aynıysa (bu durumda aday çözüm, zaten daha önce denenmiştir), bu çözüm değerlendirme dışı bırakılmaktadır. Tabu listesi oluşturulurken her döngüdeki en iyi çözüm listeye alınmakta, listenin dolduğu durumda listedeki ilk kayıtlar (başlangıçtaki çözümler) listeden atılıp, son döngülerde elde edilen çözümler listeye alınmaktadır.
- Tabu Listesi ilk en iyi çözüm kümesinin oluşturularak hafızaya alınma yöntemi ile oluşturulur. Tabu listesi oluşturmanın önemli bir kuralı da giriş değerleri oluşturulurken çeşitli filtreleme işlemlerinden geçirilmesidir.

TABU ARAMA ALGORİTMASI

- Oluşturulan Tabu listesinin en temel özelliği yeni çözüm adaylarını değerlendirmeye alarak dairesel bir döngü içerisinde yol olarak her yeni döngüden sonra yeni özellikler kazanmasıdır.
- Tabu listesinin yeni elde edilen çözümler ile liste uzunluğu artacak ve çözüm aramada büyük bir etki yaparak kısa çözümlerin daireselliği sayesinde nesnel bir çözüm arama tekniği olduğunu gösterecektir. Burada istenilen durum çözüm adaylarının mevcut bu döngüsü ile Tabu listesinin geliştirilmesidir. Tabu listesine dahil edilen yeni çözümler oluşturulurken eğer elde edilen çözüm tabu listesindeki çözümlerden daha iyi ise elde edilen çözüm tabu listesine eklenebilir. Bunun tersi durumunda, yani daha kötü bir çözüm ise tabu listesine eklenmeyecek ve doğal olarak belirli bir bellek kaplamayacaktır. En iyi çözüm bulunana kadar bu işlemler mevcut döngü ile devam edecektir.

TABU ARAMA ALGORİTMASI

- TA ana olarak basit tepe tırmanma (BTT) yönteminin zaafalarını gidermek için düşünülmüştür. BTT eldeki aday çözüme, bir komşuluk işleci uygulayarak, yeni adaylar üretir. Yeni adaylar bir değerlendirmeye tabi tutulur. Değerlendirme çözümün sonuca yakınlığını ölçer. Yeni adaylar ile eldeki eski aday içerisinde çözüme en yakın olan eskinin yerine geçer. BTT bu haliyle kısır bir döngüye sebep olabilir. Komşuluklar kapsamında birbirine eşit değerdeki iki yada daha fazla komşu arasında BTT takılıp kalabilir. Arama yapılan alanın özellikleri yada BTT yönteminin iyi seçilmemesi, herhangi bir denetim mekanizması kullanılmadan yapılan aramayı yerel en iyi noktalara götürebilir. Fakat gerçek hayattaki problemler yerel en iyilerin bol olduğu, fakat global iyinin az, hatta tek olabildiği problemlerdir. Kısacası kısır döngü kaçınılmazdır. Tabu arama yöntemi kısır döngüden kurtulmak için hafıza kullanılmasını, hatırlamayı önerir. Daha önce ziyaret edilmiş ya da herhangi bir nedenle ziyaret edilmesi istenilmeyen aday çözümlerle ilgili özellikler, *tabu listesi* adı verilen, kısa dönem hafızaya benzer bir yapıda tutulur. Yöntem bu listedeki hamlelerin belirli bir süre yapılmasını yasaklar. Böylece arama yerel bir en iyi noktadan kurtularak asıl sonuca yakınsayabilir. Bu temel altyapının yanı sıra, tabu arama yöntemlerinde aramayı belirli bir noktaya yönlendirebilecek uzun dönem hafıza yapıları da kullanılmıştır



Metasezgisel Optimizasyon Algoritmaları

Metasezgiseller

- Çözüm uzayını etkin bir şekilde aramayı sağlayacak temel sezgisel yöntemleri birleştirmeye çabalayan yeni yaklaşıp yöntemler.
- Glover (1986) tarafından literatüre kazandırılan “metaheuristic” kelimesi, Yunanca kökenli iki kelimedenden türetilmiştir:
 - “Heuristic“, bulmak, keşfetmek anlamına gelen “*heuriskein*“ kelimesinden türetilmiştir.
 - “Meta” ise üst seviye anlamına gelmektedir.
- Metaheuristics → “modern heuristics”

Metasezgisel algoritmalar

- Isıl İşlem
- Tabu Arama
- Değişken Komsu Arama
- Genetik Algoritmalar
- Karınca Kolonisi Optimizasyonu
- Parçacık Sürü Optimizasyonu
- Arı Sürü Optimizasyonu
- Yapay Bağışıklık Sistemleri
- ...

Tanımlar...

- [Osman ve Laporte, 1996]
 - “Metasezgisel, arama uzayında araştırma (exploring) ve işletme (exploiting) için farklı kavramları zeki bir şekilde birleştirerek alt seviye sezgisellere rehberlik eden iteratif üretim surecidir”
- [Voß et al. 1999].
 - “Metasezgisel, yüksek kaliteli çözümleri etkin bir şekilde üretmek için alt seviye sezgisellere rehberlik eden bir iteratif üst seviye prosestir. Her iterasyonda, bir çözümü ya da çözümlerin bir topluluğunu kullanır. Alt seviye sezgiseller, basit yerel arama algoritması yada çözüm kurucu bir yöntem olabilir.”

Tanımlar...

- [Stutzle 1999]

- “Metasezgiseller, arama uzayında yüksek kaliteli çözümlere ulaşmak için probleme özgü sezgisellere rehberlik eden yüksek-seviye stratejileridir. Amaç, yerel en iyi tuzaklara yakalanmayı engellemektir. Bu amaca, arama sırasında kötü hareketlere izin vererek yada rassal başlangıç çözümleri yerine çok daha zeki bir yolla üretilen çözümleri kullanarak ulaşılabilir.

- [Dorigo and Stutzle, 2004]

- “Metasezgisel, arama uzayının yüksek kaliteli çözümlerini kapsayan bölgelerinde aramayı gerçekleştirmek için probleme özgü sezgisellere rehberlik etmek amacıyla tasarlanan genel amaçlı sezgisel yöntemdir”

Metasezgisel Yöntemlerin Özellikleri

- Metasezgiseller, arama sürecine rehberlik eden stratejilerdir.
- Amaç, en iyi yada en iyiye yakın çözümleri bulmak için arama uzayını hızlı bir şekilde araştırmaktır.
- Metasezgiseller, basit yerel arama algoritmalarından karmaşık öğrenme proseslerine kadar geniş bir yelpazeyi içermektedir.
- Metasezgiseller, yaklaşık algoritmalar ve genellikle deterministik değildir.
- Arama uzayındaki yerel en iyi tuzaklardan kurtulmak için çeşitli mekanizmaları kullanırlar.
- Metasezgiseller, probleme özgü değildirler.
- Metasezgiseller, üst seviye stratejiler tarafından kontrol edilen sezgisellerde probleme özgü bilgi kullanımına izin verirler.
- İleri seviye metasezgiseller, aramaya rehberlik etmesi amacıyla arama sırasında elde edilen bilgiyi (hafızayı) kullanırlar.

Kısacası, metasezgiseller, farklı metotlar ile arama uzayının araştırılması için yüksek seviye stratejilerdir.

Metasezgisellerin en önemli özelliği

- çeşitlendirme (diversification) ve
- yoğunlaşma (intensification)

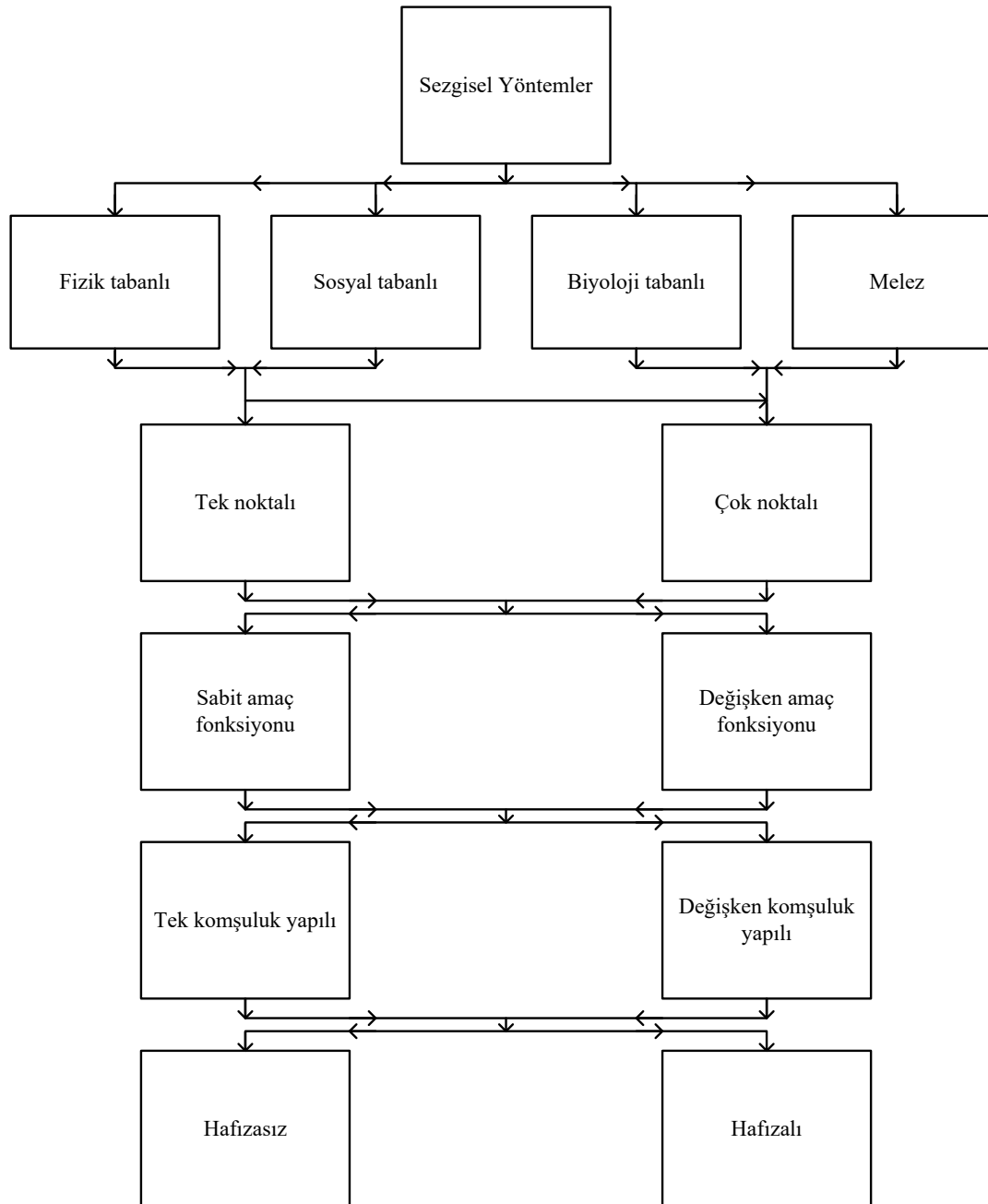
arasındaki dinamik dengeyi oluşturmalarıdır.

- Çeşitlendirme; arama uzayında araştırmayı
- Yoğunlaşma; arama sırasında elde edilen tecrübenin (bilginin) isletilmesi
- Çeşitlendirme → araştırma (exploration)
- Yoğunlaşma → işletme (exploitation)

Metasezgisellerin sınıflandırılması

- Esinlendikleri kaynaklara
- Aramada kullandıkları çözüm sayısına
- Kullanılan amaç fonksiyonuna
- Kullanılan komşuluk yapısına
- Hafıza kullanımına

göre sınıflandırılmaktadırlar.



Esinlendikleri kaynaklara göre

- Genel amaçlı sezgisel yöntemler; biyolojik tabanlı, fizik tabanlı, kimya tabanlı, müzik tabanlı ve sosyal tabanlı olmak üzere beş farklı grupta değerlendirilmektedir. Ayrıca bunların birleşimi olan melez yöntemler de vardır.
- Genetik algoritma, diferansiyel gelişim algoritması, karınca koloni algoritmaları, yapay sinir ağları, arı koloni algoritmaları, yapay bağışıklık sistemleri, kedi sürü optimizasyonu, bakteriyel yem arama algoritması, fidan gelişim algoritması **biyolojik tabanlı**
- Isıl işlem ve elektromanyetizma algoritması **fizik tabanlı**
- Yapay kimyasal reaksiyon algoritması **kimya tabanlı**
- Armoni arama algoritması **müzik tabanlı**
- Tabu arama, emperyalist yarışmacı algoritma ve parlamenter optimizasyon algoritması **sosyal tabanlı** algoritma ve modellerdir.
- Kültürel algoritma da hem biyolojik hem de sosyal tabanlı algoritma olarak sınıflandırılabilir. Parçacık Sürü Optimizasyonu da hem biyolojik hem de sosyal tabanlı algoritma sınıfına girmektedir.

Aramada kullandıkları çözüm sayısına göre

- Algoritma tek bir çözümden başlayıp bunu operatörlerle ilerletiyorsa bunlara **tek noktalı yöntemler** denir ve tabu arama, ısıtma işlemi gibi bütün yerel arama tabanlı sezgisel algoritmalar bu gruba girer. Arama sırasında tek bir çözüm kullanan algoritmalar, yörünge (trajectory) metotları olarak adlandırılırlar. Bu yöntemlerde arama, arama sürecinde arama uzayında tanımlanan bir yörünge üzerinde gerçekleştirilir.
- Çok noktadan yani bir popülasyon üzerinden çözüme başlanıp bu farklı noktalarla optimizasyon yapılıyorsa bunlar **çok noktalı ya da popülasyon tabanlı** yöntemlerdir. Bunlara GA, PSO, karınca koloni algoritmaları, arı koloni algoritmaları, yapay bağışıklık sistemleri ve elektromanyetizma algoritması örnek olarak verilebilir.

Kullanılan amaç fonksiyonuna göre

- Bazı sezgisel algoritmalar problemin gösterimini gerçekleştirirken amaç fonksiyonunu sabit tutar ve **sabit amaç fonksiyonlu** olarak adlandırılırlar. Örnek: Genetik Algoritmalar, Karınca Kolonisi Optimizasyonu, Parçacık Sürü Optimizasyonu, Tabu Atama, Değişken Komsu Arama, İteratif Yerel Arama ve Tavlama Benzetimi arama.
- Bazıları da örneğin rehberli yerel arama algoritmasındaki gibi değiştirir ve **değişen amaç fonksiyonlu** olarak adlandırılırlar. Değiştirmekteki amaç yerel minimumdan kurtulmaktır. Bu mantık, yerel minimumdan kaçmak için bazen diğer sezgisel algoritmalara da uygulanabilmektedir. Örnek: Yönlendirilmiş Yerel Arama (Guided Local Search) metasezgiseli. Amaç, arama uzayında yapılan bu tür bir modifikasyon ile yerel en iyi çözümlere yakalanmayı önlemektir. Dolayısıyla, arama sırasında elde edilen bilgi kullanılarak amaç fonksiyonu değiştirilir.

Kullanılan komşuluk yapısına göre

- Çoğunlukla sezgisel yöntemler tek bir komşuluk yapısında çalışır ve **tek komşuluk yapılı** olarak sınıflandırılabilir. Değişken komsu arama metasezgiseli haricindeki metasezgiseller tek bir komşuluk yapısını kullanırlar.
- Bazıları da değişken komşuluk arama algoritmasında olduğu gibi arama işlemini sistematik bir şekilde değiştirerek birden fazla yerel arama yöntemiyle diğer çözüm alanlarına ulaşmaya çalışır ve **değişken komşuluk yapılı** şeklinde sınıflandırılabilir. Parçacık sürü optimizasyonunun iki versiyonu da bulunabilmektedir. Çeşitli komşuluk yapısını kullanmada amaç, aramada çeşitliliği sağlamaktır.

Hafıza kullanımına göre

- Algoritmalar çalışırken daha önceki durumlar ya da en iyi durumlar hatırlanıyorsa **hafızalı**,
- hatırlanmıyorsa **hafızasız** şeklinde sınıflandırılabilir. Örneğin parçacık sürü optimizasyonu, karınca kolonisi ve tabu arama hafızalı, ısıt işlem hafızasızdır.
- Sınıflandırılmadaki özellik: arama sırasında aramadan elde edilen geçmiş bilgiyi kullanıp kullanmadıklarıdır.
- Hafıza kullanımı iki farklı şekilde gerçekleştirilebilir:
 - Kısa dönem hafızada, yeni yapılan hareketler, çözümler ya da alınan kararlar tutulur.
 - Uzun dönem hafıza ise, arama sırasında elde edilen bilgiler tutulur.

Çok Noktalı (Popülasyon Tabanlı) Algoritmalar

- Evrimsel (Gelişimsel) Hesaplama
 - Genetik Algoritmalar
 - Evrimsel Algoritmalar
 - Diferansiyel Gelişim Algoritması
 - Evrimsel Programlama
 - Evrim Stratejileri
 - Sınıflayıcı Sistemleri
 - Genetik Programlama
- Karınca Koloni Algoritmaları
- Arı Koloni Algoritmaları
- Yapay Bağışıklık Sistemleri
- Parçacık Sürü Optimizasyonu
- -
- -
- -