Yapay Zeka

Ders 5 – Bölüm 2: Bilgi ile arama

Doç. Dr. Mehmet Dinçer Erbaş
Bolu Abant İzzet Baysal Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

Bilgi ile arama

- Önceki bölümde gördüğümüz üzere bilgisiz arama metotları
  - sistematik şekilde yeni durumları oluşturur
  - bu durumların hedef olup olmadığını test eder
  - bu şekilde problemlere çözüm bulmaya çalışır.
- Ne yazık ki bu metotlar basit olmakla birlikte birçok problem için oldukça verimsizdir.
- Bu bölümde ise problem ile alakalı bilgiler kullanan bilgi arama metotlarını göreceğiz.
  - Bu algoritmalar bilgi kullanarak daha verimli şekilde problemleri çözebilirler.

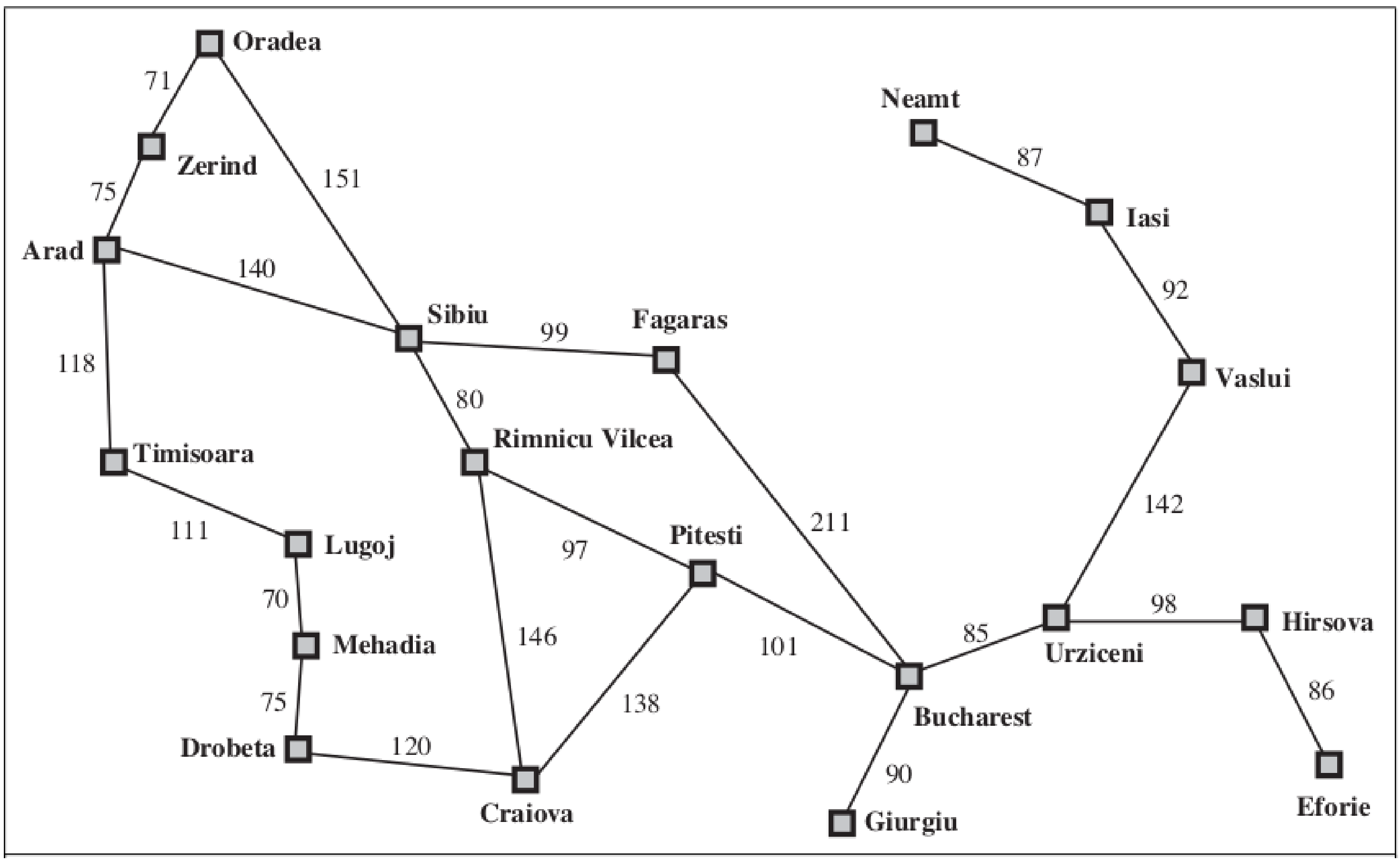
Bilgi ile arama

- Genel olarak en-iyi-öncelikli arama yaklaşımını kullanacağız.
  - Bu yaklaşım daha önce gördüğümüz AĞAÇ-ARAMA veya GRAF-ARAMA algoritmalarının bir versiyonudur.
  - Açılacak düğümler f(n), değerlendirme fonksiyonu ile seçilir.
    - Değerlendirme fonksiyonu bir maliyet tahmini değeri belirler ve en düşük değerlendirmeye sahip düğüm açılmak üzere seçilir.
    - En-iyi-öncelikli arama algoritmasının graf arama olarak implementasyonu daha önce gördüğümüz sabit-maliyetli arama algoritmasına benzer.
      - Sabit maliyet arama algoritmasının implementasyonundaki g değerinin yerini f alır.

Bilgi ile arama

- f fonksiyonun seçimi arama stratejisini belirler.
- Çoğu en-iyi-öncelikli arama algoritması f fonksiyonun bir parçası olarak bir buluşsal fonksiyon (İng: heuristic function), h(n) içerir.
  - h(n) => n durumundan hedef durumuna en ucuz yolun maliyeti
- Örnek verirsek, Romanya tatili örneğimizde haritadaki her şehirden Bükreş şehrine düz-çizgi uzaklığı kullanabiliriz.
- n hedef durumu ise h(n) = 0.

Bilgi ile arama



Bilgi ile arama

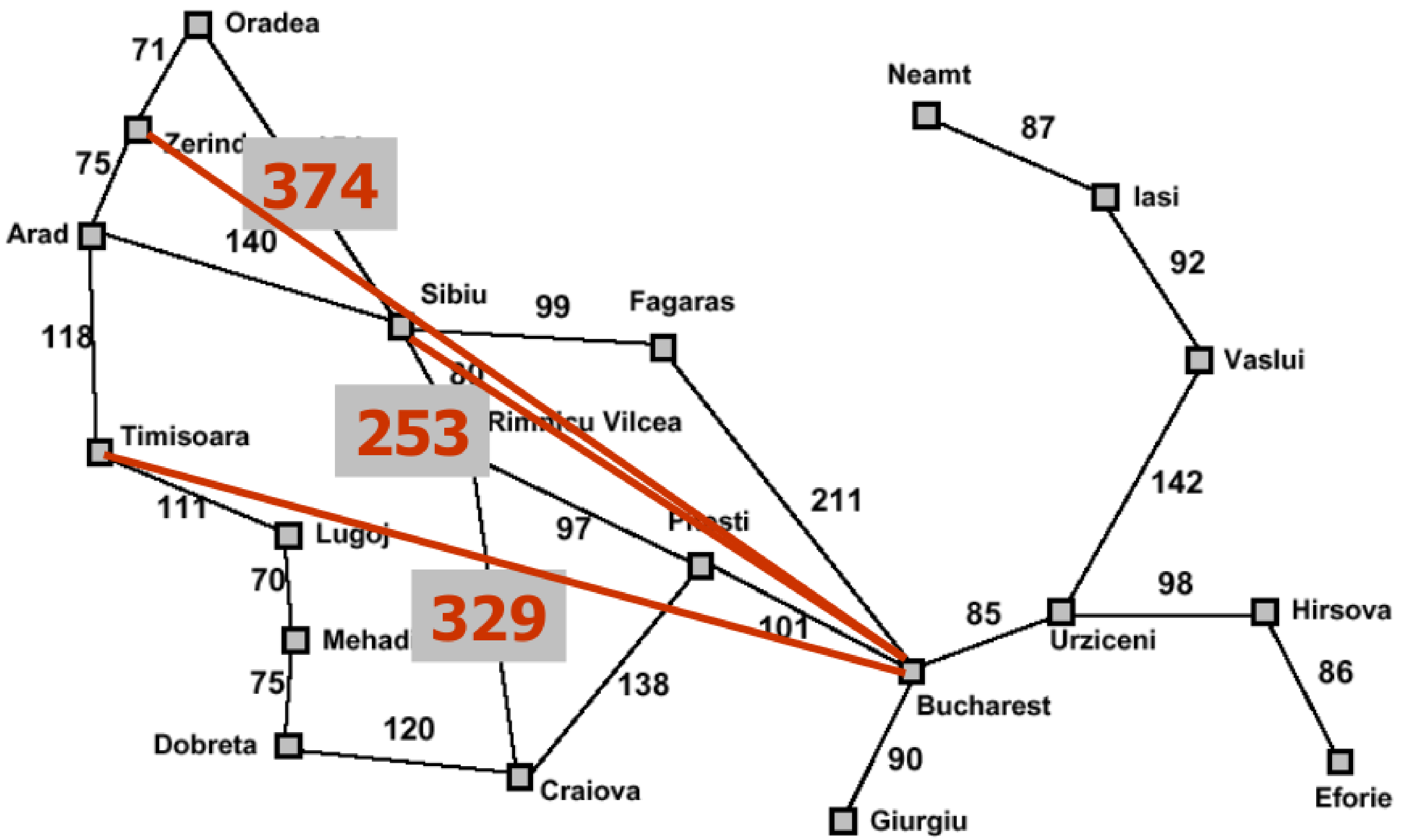
- Açgözlü en-iyi-öncelikli arama (İng: Greedy best-first search)
  - Açgözlü en-iyi-öncelikli arama algoritması hedefe en yakın düğümü açar
  - Hedefe en yakın düğümü açarak hızlıca bir çözüme ulaşmayı hedefler
  - $f(n) = h(n)$
  - Romanya tatili örneğini üzerinde algoritmayı çalıştıralım.
  - Buluşsal fonksiyon olarak şehirlerin Bükreş şehrinden düz-çizgi uzaklığını kullanabiliriz.

Bilgi ile arama

- Açgözlü en-iyi-öncelikli arama

Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

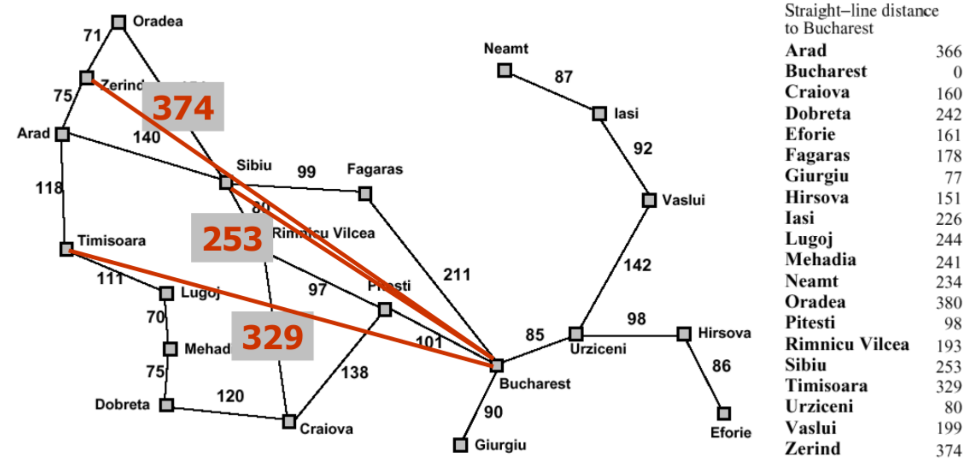
Bilgi ile arama

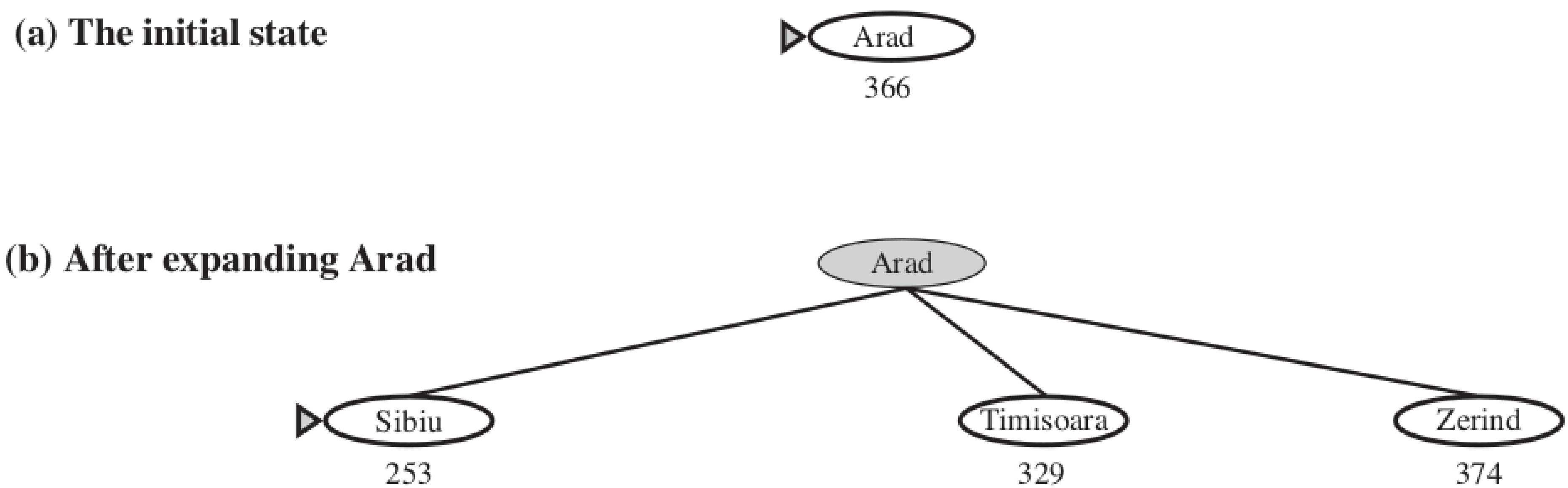


Straight–line distance
to Bucharest

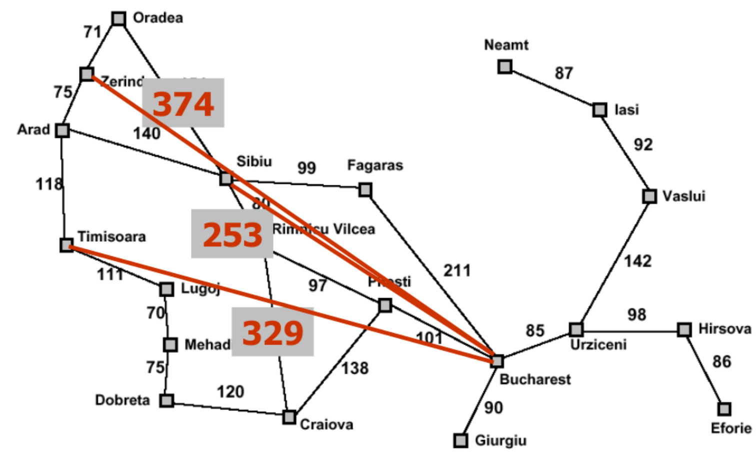
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Bilgi ile arama



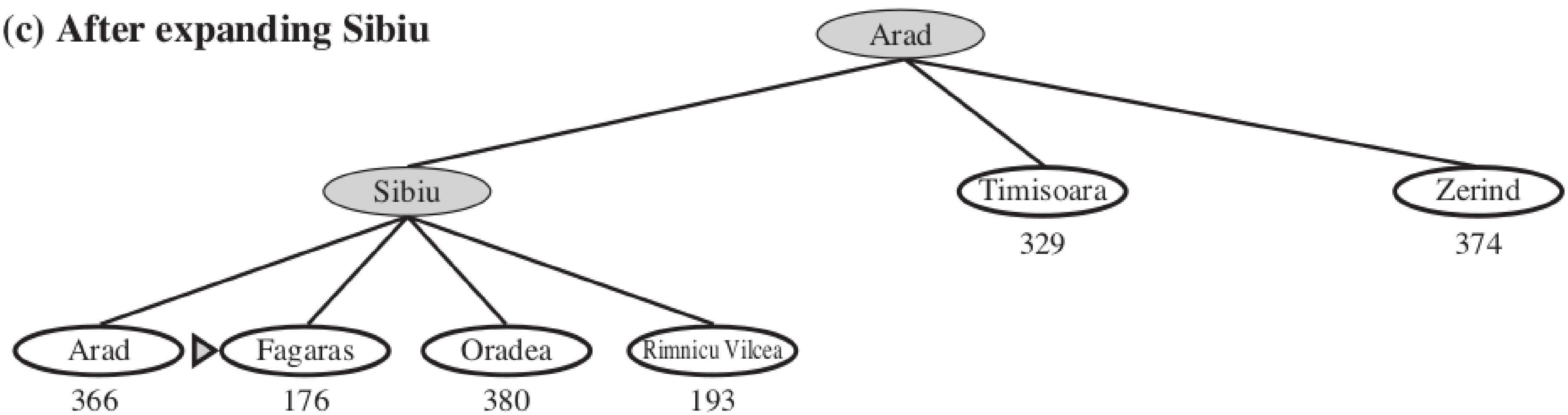


Bilgi ile arama

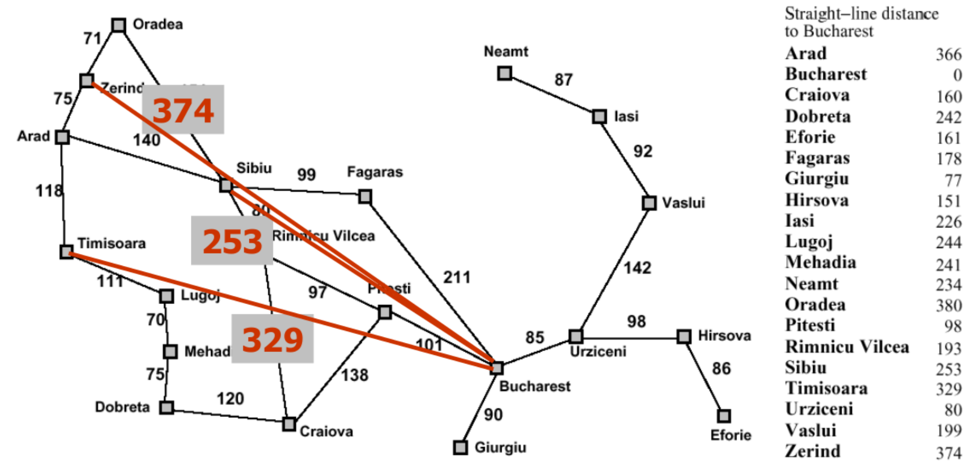


Straight–line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

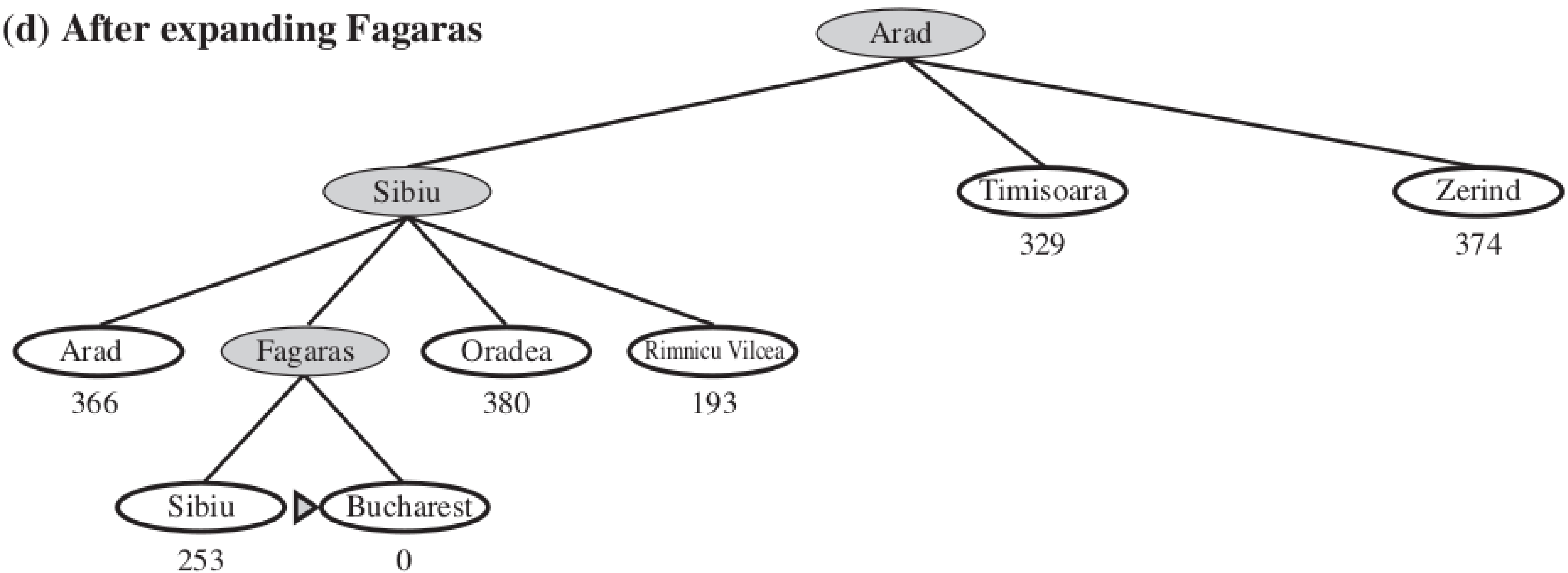
(c) After expanding Sibiu



Bilgi ile arama



(d) After expanding Fagaras



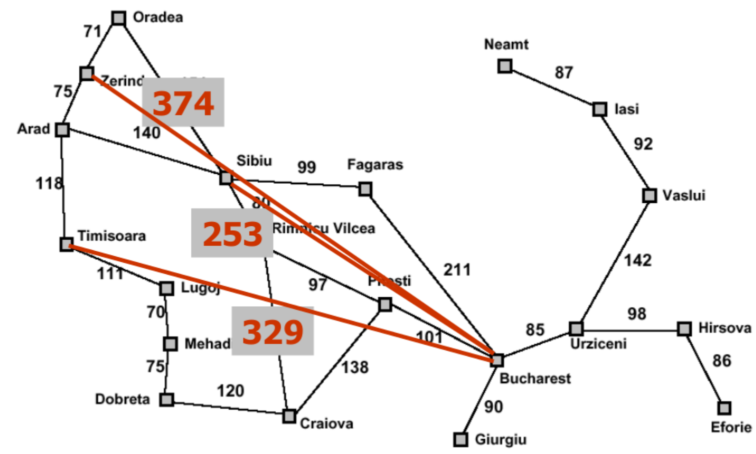
Bilgi ile arama

- Açgözlü en-iyi-öncelikli arama
  - Bu örneğimizde arama operasyonu minimal işlem gerektiriyor
    - açılan her düğüm çözüm yolunun parçası.
  - Ancak bulunan çözüm optimal değil.
    - Aslında bu sebeple “açgözlü” deniyor.
      - Bulunduğu durumda mümkün olduğunca hedefe yakınlaşmaya çalışıyor.
  - Iasi şehrinden Fagaras şehrine gittiğimizi düşünelim
    - Algoritma sonuca ulaşamaz.
    - Graf arama versiyonu sınırlı uzaylarda bütündür, ancak sınırlı olmayan uzaylarda bütün değildir.
  - En kötü durumda zaman ve yer karmaşıklı arama ağacı versiyonu için $O(b^m)$, m arama uzayının maksimum derinliği.
  - İyi bir buluşsal fonksiyon ile karmaşıklık azaltılabilir.
  - Azaltma miktarı problem tipine ve buluşsal fonksiyonun kalitesine bağlıdır.

Bilgi ile arama

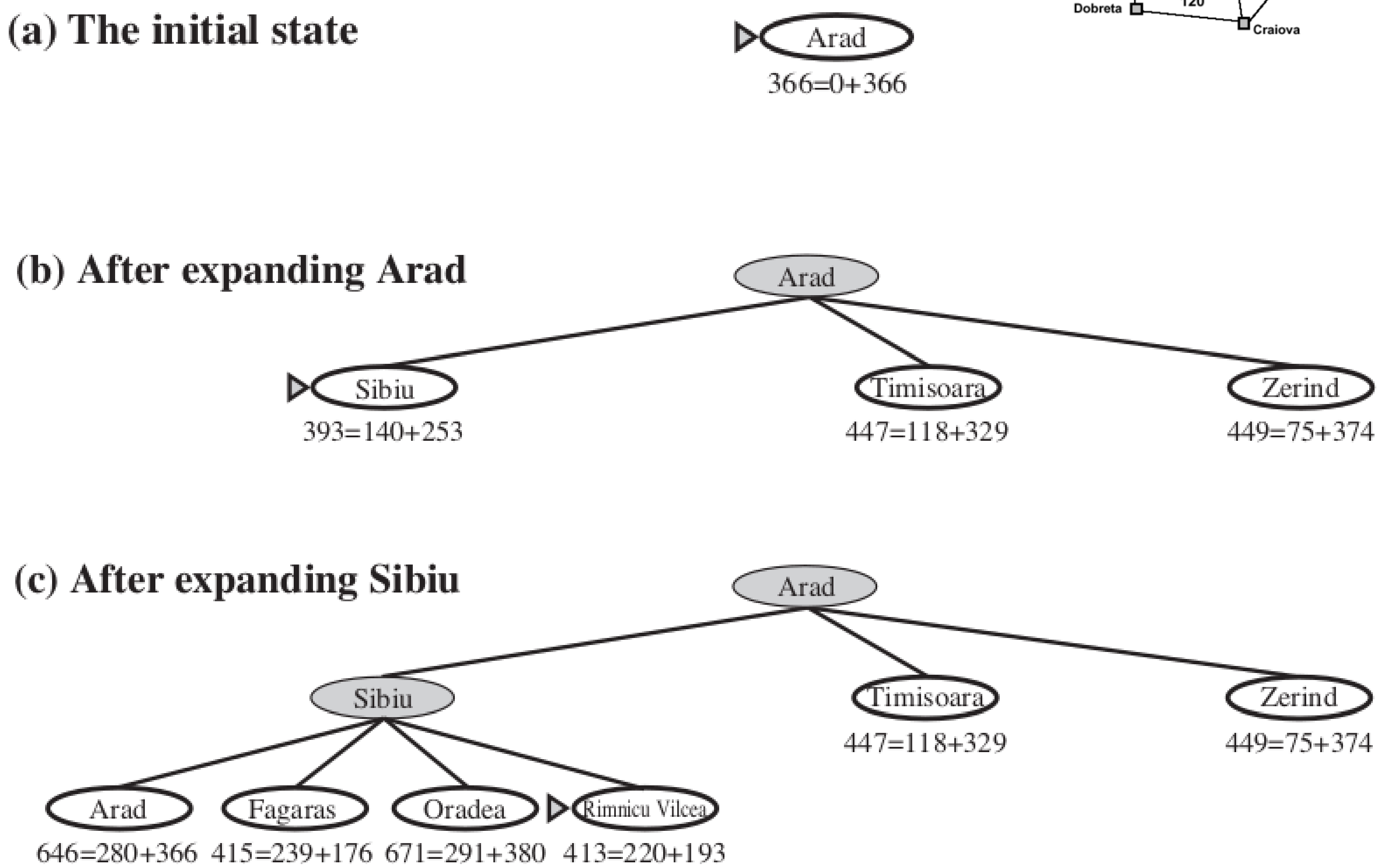
- A* arama
  - A* arama algoritması toplam tahmini çözüm maliyetini minimize etmeyi amaçlar.
    - En bilinen en-iyi-öncelikli arama algoritmasıdır.
  - Düğümleri değerlendirirken g(n), düğüme ulaşma maliyeti ile h(n), düğümden hedefe ulaşma maliyetini birleştirir.
    - $f(n) = g(n) + h(n)$
  - g(n) başlandıç düğümünden düğüm n'e gitminin maliyetini verirken, h(n) düğüm n'den en ucuz yoldak hedefe ulaşmanın tahmini maliyetini verir. Öyleyse
    - f(n) = n düğümünden geçen en ucuz çözümün tahmini maliyeti
  - h(n) belli koşulları sağladığında A* arama hem bütün hem optimaldir.
  - A* arama algoritması, g yerine g + h kullanılması dışında Sabit maliyet arama algoritması ile aynıdır.

Bilgi ile arama

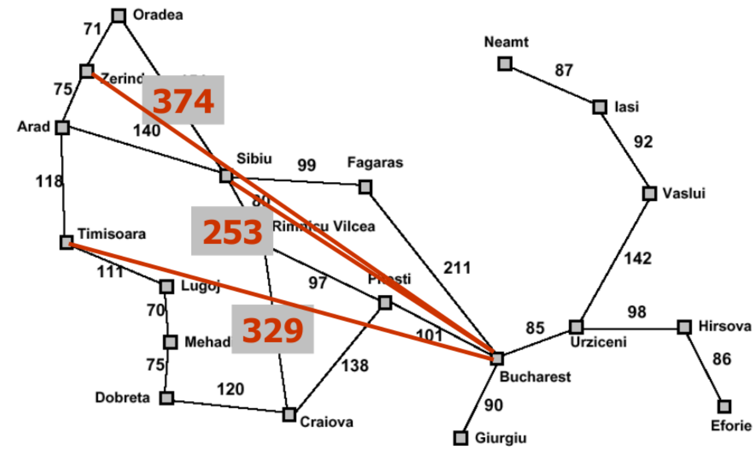


Straight–line distance to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374



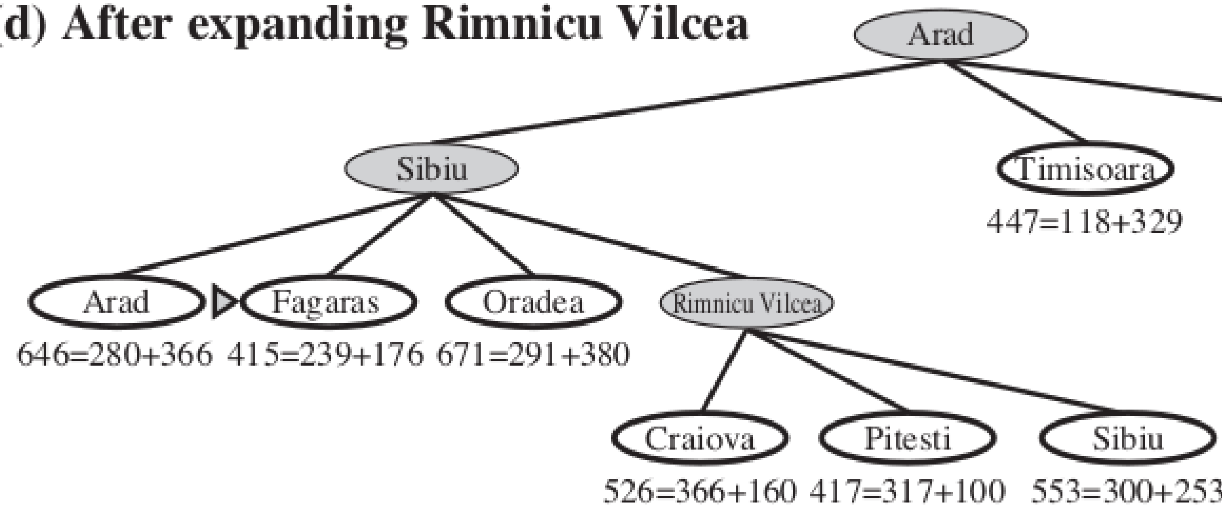
Bilgi ile arama



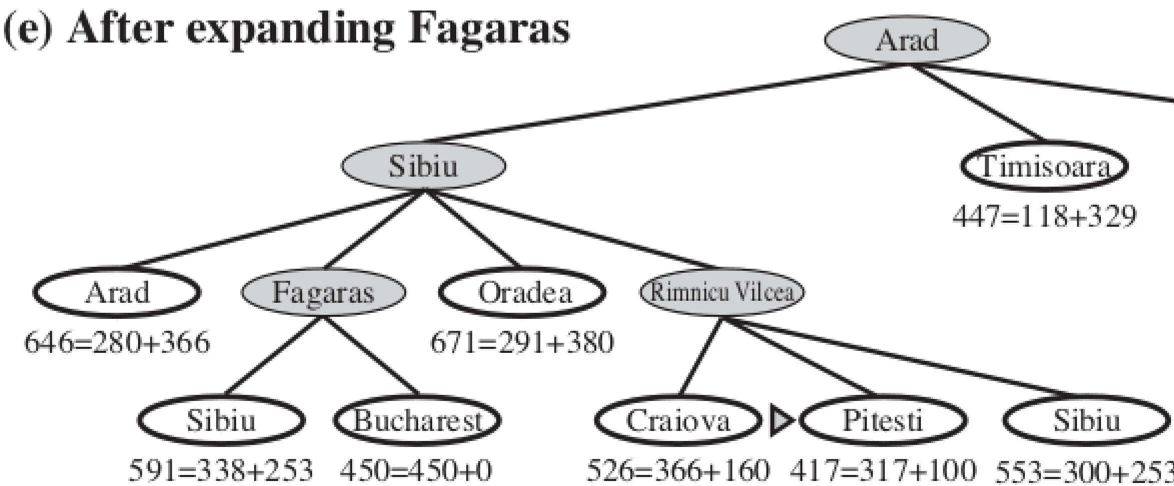
Straight–line distance to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

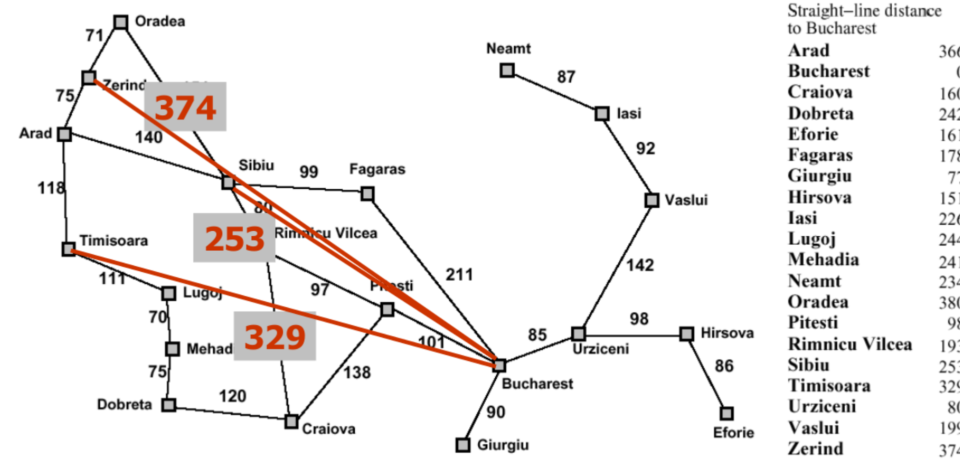
(d) After expanding Rimnicu Vilcea



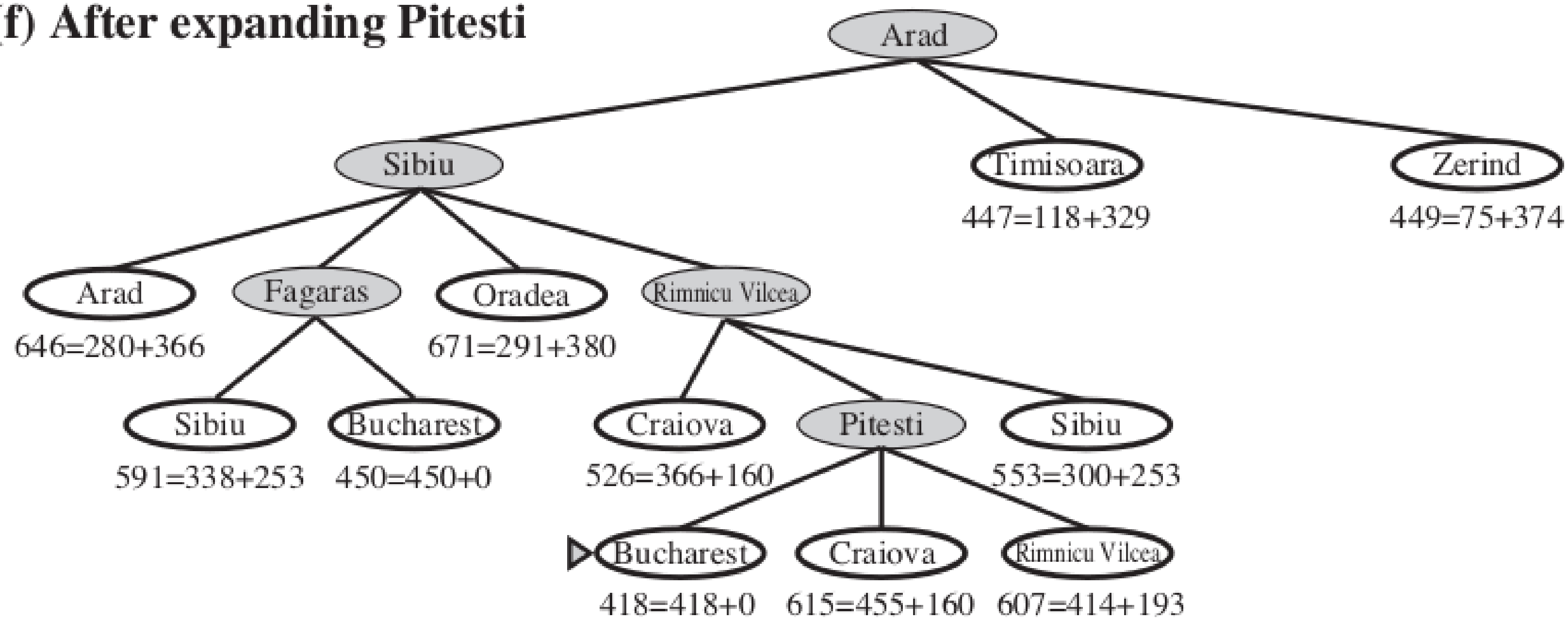
(e) After expanding Fagaras



Bilgi ile arama



(f) After expanding Pitesti



Bilgi ile arama

- A* arama
  - Optimallik için gerekli koşullar: Onanırlık (İng: admissibility), tutarlılık (İng: consistency).
    - Algoritmanın optimal olabilmesi için h(n) onanır olmalıdır.
      - Onanır bir buluşsal fonksiyon asla hedefe ulaşma maliyetini olduğundan fazla tahmin etmez.
      - g(n) n düğümüne ulaşma maliyeti ise ve h(n) n’den hedefe ulaşma maliyetini fazla tahmin etmezse, f(n) de bir çözümün maliyetini asla fazla tahmin etmez.
      - Örneğimiz için kullandığımız doğru çizgi uzaklığı onanır buluşsal fonksiyondur.
    - Algoritmanın optimal olabilmesi için h(n) tutarlı olmalıdır.
      - h(n) fonksiyonunun tutarlı olması için aşağıdaki denklemin geçerli olması gerekmektedir.
        - $h(n) \leq c(n,a,n') + h(n')$
        - Bu eşitsizliğe üçgen eşitsizlik denir.
    - Her tutarlı buluşsal fonksiyon aynı zamanda onanırdır.

Bilgi ile arama

- A* algoritmasının optimalliği
  - Daha önce belirttiğimiz üzere
    - A* algoritmasının ağaç arama versionu h(n) onanır ise optimaldir.
    - A* algoritmasının graf arama versiyonu h(n) tutarlı ise optimaldir.
  - İkinci durumu ispatlayalım
    - Eğer h(n) tutarlı ise, f(n) değerleri bir yol boyunca azalmayandır.
      - n’ n düğümünün ardılı olsun, öyleyse bir a aksiyonu için $g(n') = g(n) + c(n,a,n')$ olmalı. Öyleyse
  - $f(n') = g(n') + h(n') = g(n) + c(n,a,n') + h(n') \geq g(n) + h(n) = f(n)$

Bilgi ile arama

- A* algoritmasının optimalliği
  - İkinci durumun ispatı (devam)
    - A* bir düğümü açmak için seçtiğinde, o düğüme gelen optimal yol bulunmuştur.
    - Durum böyle olmasaydı, n düğümüne gelen optimal yol üzerinde sınırda başka bir n’ düğümü olurdu.
    - Ancak f fonksiyonu az önce gösterdiğimiz üzere azalmayandır, böyle bir n’ düğümü daha az bir f-maliyetine sahip olurdu ve algoritma tarafından daha önce seçilirdi.
  - Bu iki gözlemi birleştirirsek, söyleyebiliriz ki A* tarafından açılan graf araması ile açılan düğümler f(n) tarafından azalmayan sırayla açılır.
  - Öyleyse açılan ilk hedef düğüm optimal çözüm olmalıdır. Çünkü f hedef düğümlerin doğru maliyetini veriyorsa (hedef olduğu için h = 0), sonra açılan hedef düğümlerin her biri en az bu düğüm kadar maliyetli olmalıdır.

Bilgi ile arama

- A* algoritması
  - Belirttiğimiz üzere f üzerinden hesaplanan maliyetlere herhangi bir yol üzerinde azalmayandır. Öyleyse durum uzayında konturlar (eş uzaklık çizgileri) çizebiliriz.