

Системи, основани на знания – 2020/2021 учебна година

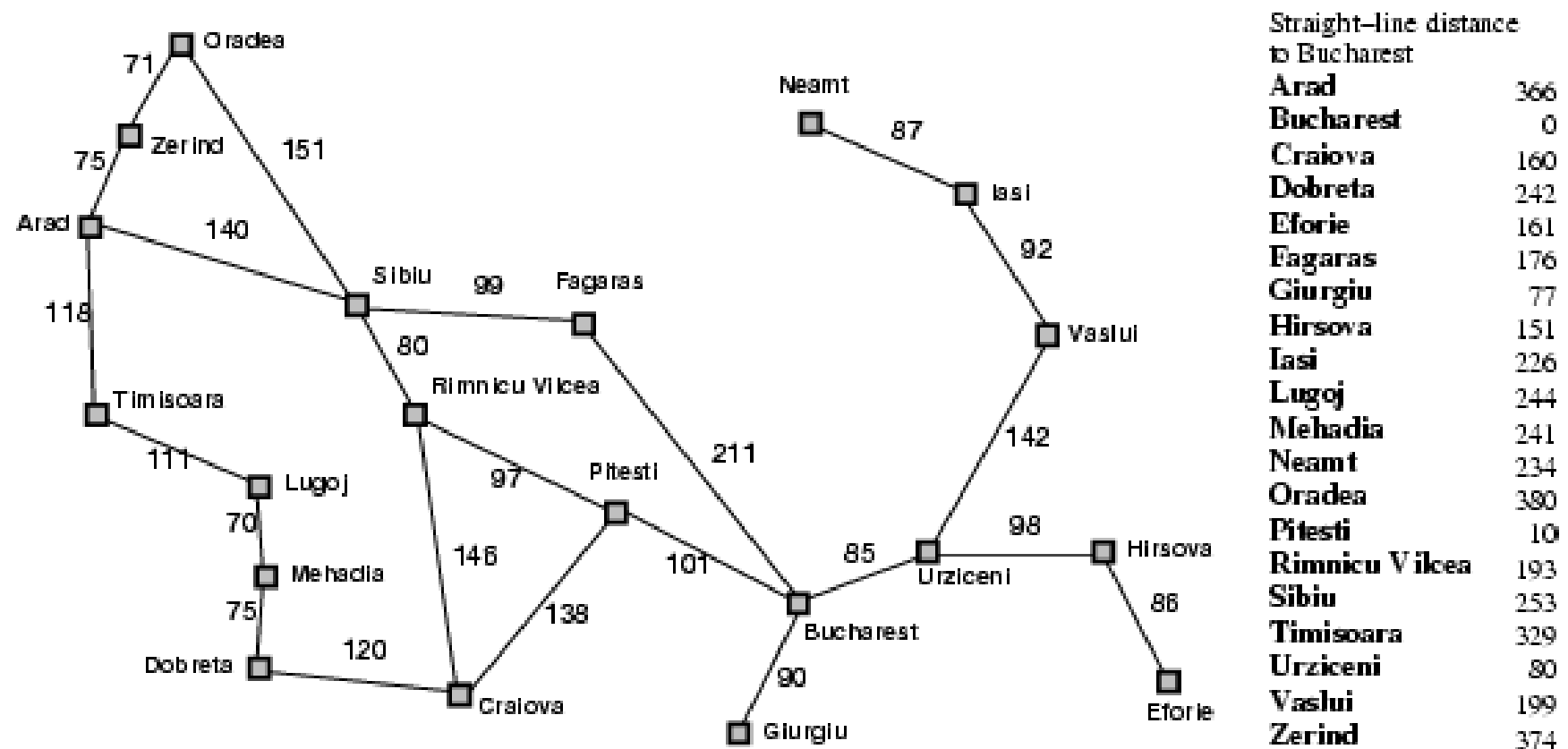
***Тема 3:
Методи за евристично търсене
на път до определена цел***

Обща характеристика: реализират пълно изчерпване по гъвкава стратегия или търсене с отсичане на част от графа на състоянията. Приложими са при наличие на специфична информация за предметната област, позволяваща да се конструира оценяваща функция (евристика), която връща небулева оценка (числова оценка в предварително определен интервал). Тази оценка може да служи например за мярка на близостта на оценяваното състояние до целта или на необходимия ресурс за достигане от оценяваното състояние до целта.

Най-често се използва евристична оценяваща функция ***h***, която връща като резултат приблизителната стойност на определен ресурс, необходим за достигане от оценяваното състояние до целта.

Програмна реализация: чрез използване на работна памет (списък Open/frontier/fringe на т. нар. открити възли или списък от натрупани/изминати пътища, започващи от началния възел – фронт на търсенето).

Примерна задача: търсене на път между две селища.
Пътна карта на Румъния:



Ще използваме данните от тази карта с цел илюстриране на работата на някои алгоритми за евристично търсене при намиране на път от Arad до Bucharest.

Евристичната оценяваща функция ще връща като резултат разстоянието по права линия (straight-line distance) между текущо достигнатия град и целта (Bucharest).

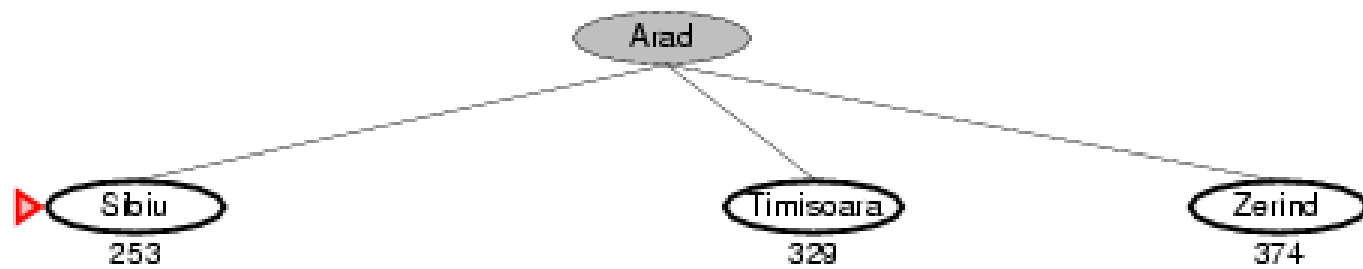
1. **Метод на най-доброто спускане** (търсене на най-добър път, ***best-first search***) - сортиране на списъка Open/fringe в съответствие с евристичната функция (например $h(node) =$ <оценка на цената на пътя от $node$ до целта>).

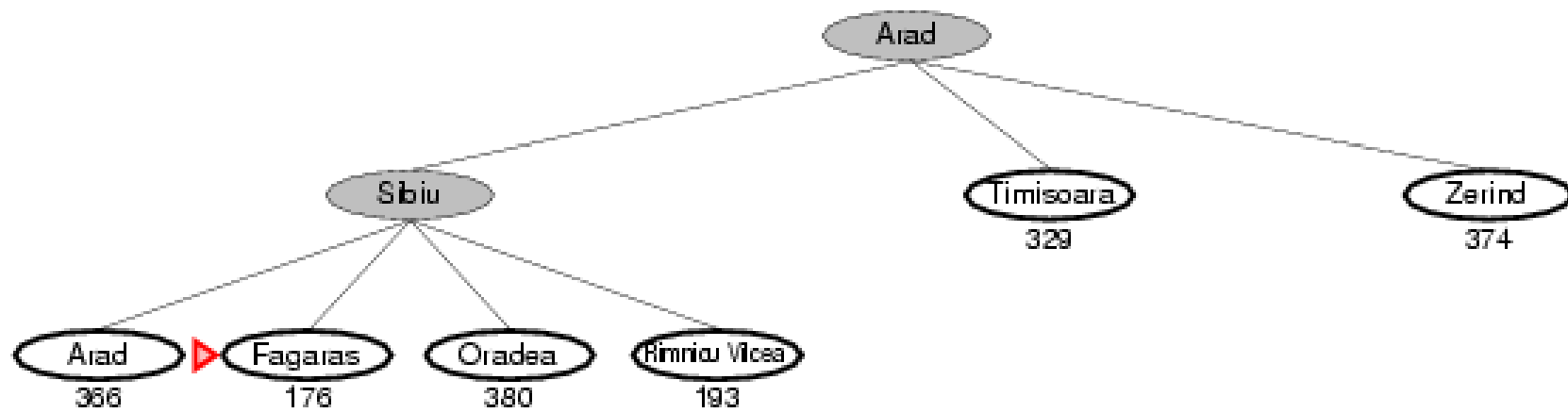
Построяване на пътя като списък от последователни състояния

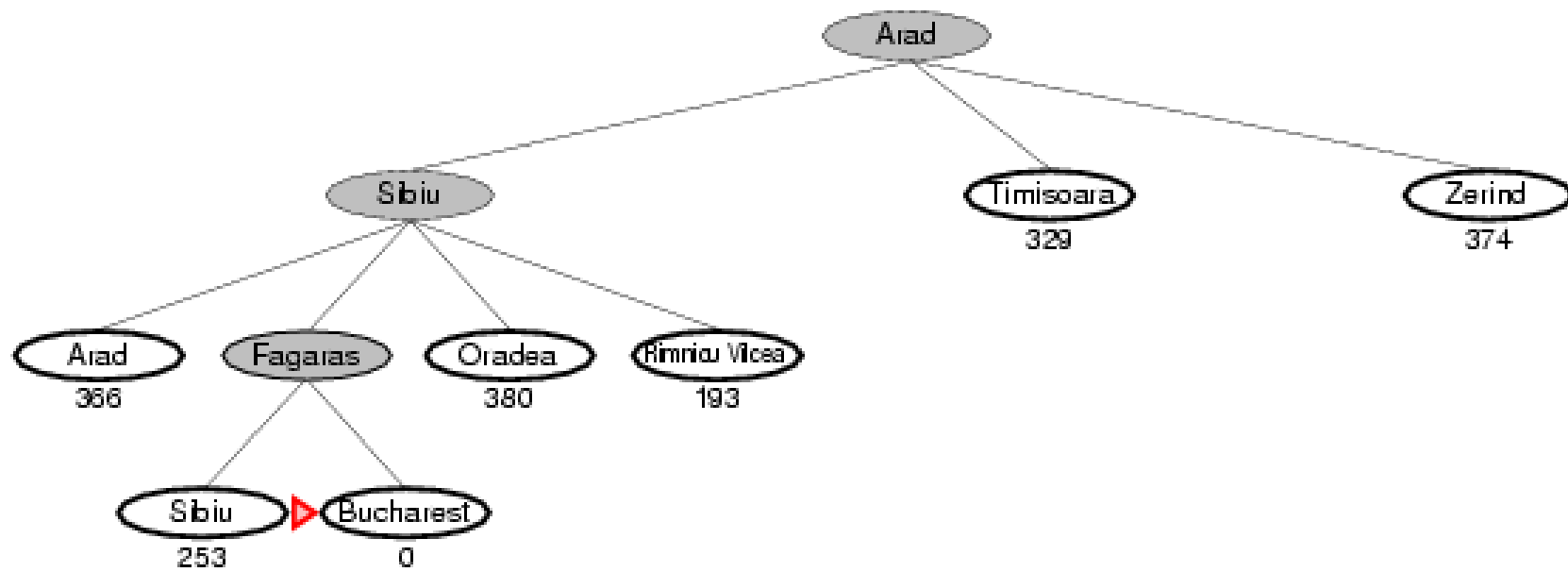
- Фронтът на търсенето е сортиран по отношение на стойностите на евристичната функция списък от пътища (отначало е списък от един път, който включва само началното състояние).
- Ако фронтът има вида $[p_1, p_2, \dots, p_n]$, то:
 - Избира се p_1 . Ако p_1 е довел до целта, **край**.
 - Ацикличните пътища $p_{11}, p_{12}, \dots, p_{1k}$, които разширяват (продължават) p_1 , се добавят към фронта и новополученият фронт се сортира в нарастващ ред на оценките на пътищата, в резултат на което придобива вида $[q_1, q_2, \dots, q_{n+k-1}]$.
 - На следващата стъпка се обработва пътят от фронта, който има най-добра евристична оценка, т.е. q_1 .

Пример: работа на алгоритъма best-first search при намиране на път върху картата на Румъния









Оценка на метода *best-first search*:

Методът е ефективен, но не е нито пълен (има опасност от зацикляне), нито оптимален.

Времевата сложност на метода е $O(b^m)$, но използването на подходяща евристика може да доведе до съществено подобрене.

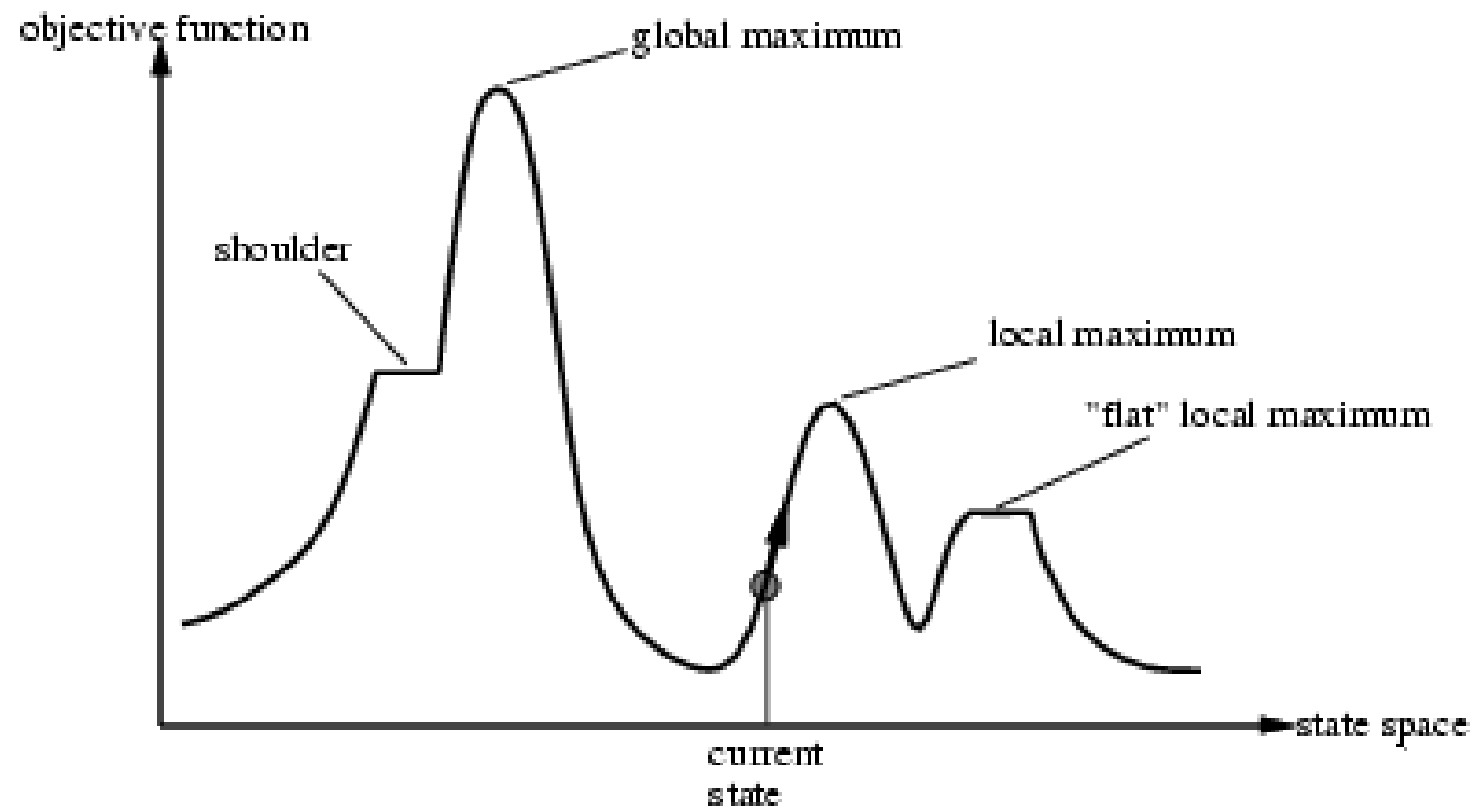
Пространствената сложност на метода е $O(b^m)$, тъй като се съхраняват всички достигнати състояния.

2. **Търсене с ограничена широчина (търсене в лъч, beam search)** - ограничаване на списъка Open/fringe до първите n най-добри възела от него (в съответствие с евристиката). При $n=1$ се доближава до метода на най-бързото изкачване.

3. **Метод на най-бързото изкачване (hill climbing)** - списъкът Open/fringe се ограничава до най-добрия му елемент (в съответствие с евристиката), и то само ако той е по-добър от своя родител. Търсенето е еднопосочно, без възможност за възврат.

Възможни проблеми:

- *локален екстремум* – състоянието е по-добро от съседните (наследниците си), но не е най-доброто в цялото ПС;
- *плато* („плосък“ локален екстремум) – съседните състояния (наследниците на текущото състояние) изглеждат също толкова добри, колкото и текущото;
- *стъпало* – никой от възможните оператори не води до по-добро състояние от текущото, макар че два или повече последователни оператори биха могли да доведат до такова състояние.



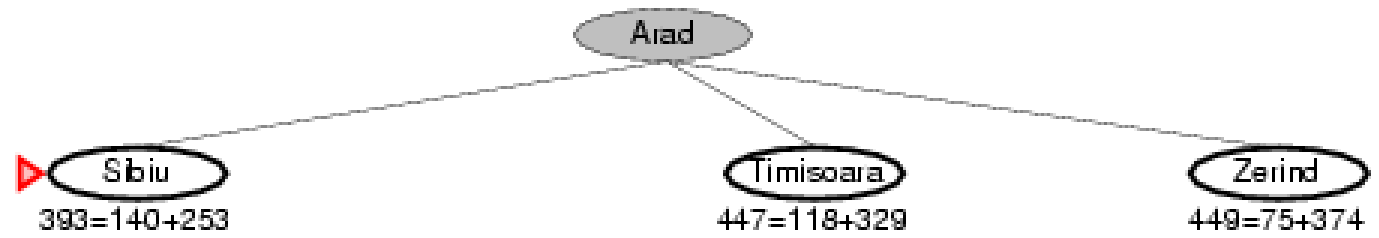
4. **Търсене с минимизиране на общата цена на пътя (A^*)** - комбинира търсене с равномерна цена на пътя с търсене на най-добър път. Списъкът Open/fringe се сортира в съответствие с функцията $f(node) = g(node) + h(node)$. Тук функцията g връща като резултат цената на изминатия път от началния възел до $node$, а евристичната функция h връща като резултат приблизителна стойност на цената на оставащата част от пътя от $node$ до целта.

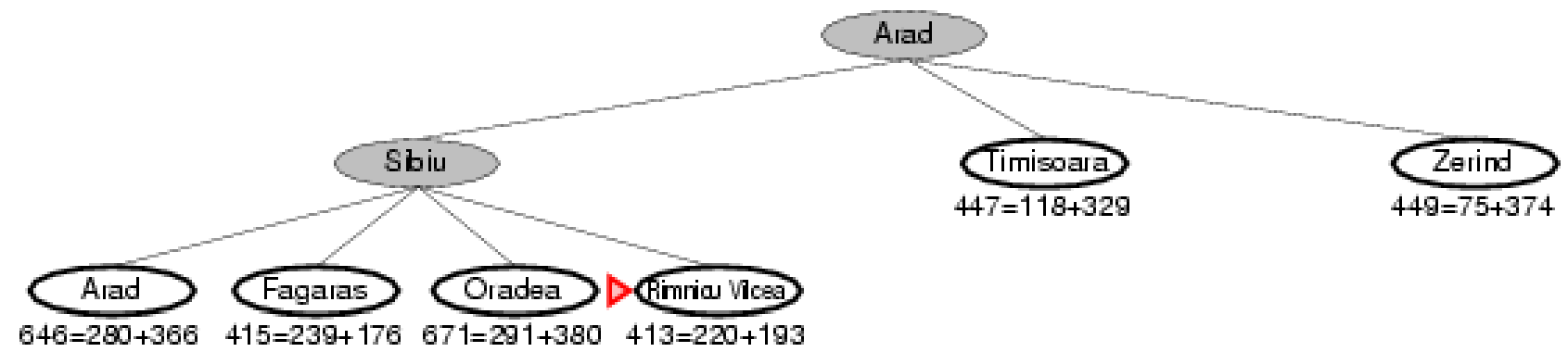
Построяване на пътя като списък от последователни състояния

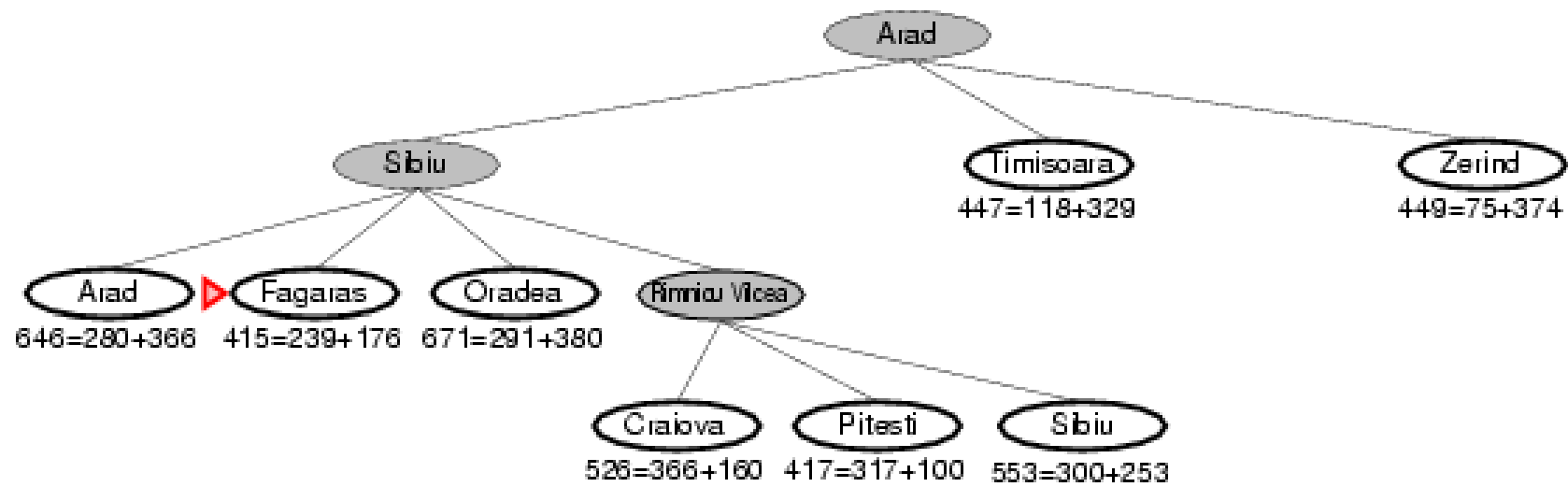
- Фронтът на търсенето е сортиран по отношение на стойностите на функцията f списък от пътища (отначало е списък от един път, който включва само началното състояние).
- Ако фронтът има вида $[p_1, p_2, \dots, p_n]$, то:
 - Избира се p_1 . Ако p_1 е довел до целта, **край**.
 - Ацикличните пътища $p_{11}, p_{12}, \dots, p_{1k}$, които разширяват (продължават) p_1 , се добавят към фронта и новополученият фронт се сортира в нарастващ ред на оценките на пътищата, в резултат на което придобива вида $[q_1, q_2, \dots, q_{n+k-1}]$.
 - На следващата стъпка се обработва пътят от фронта, който има най-добра оценка (най-добра стойност на функцията f), т.е. q_1 .

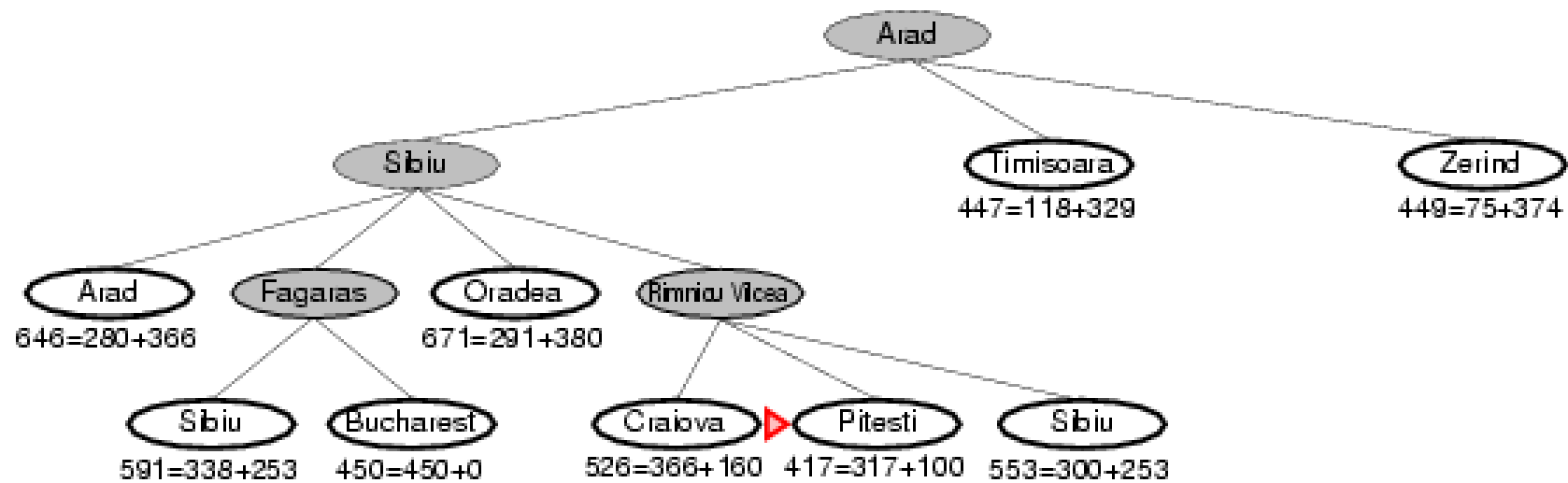
Пример: работа на алгоритъма A^* при намиране на път върху картата на Румъния

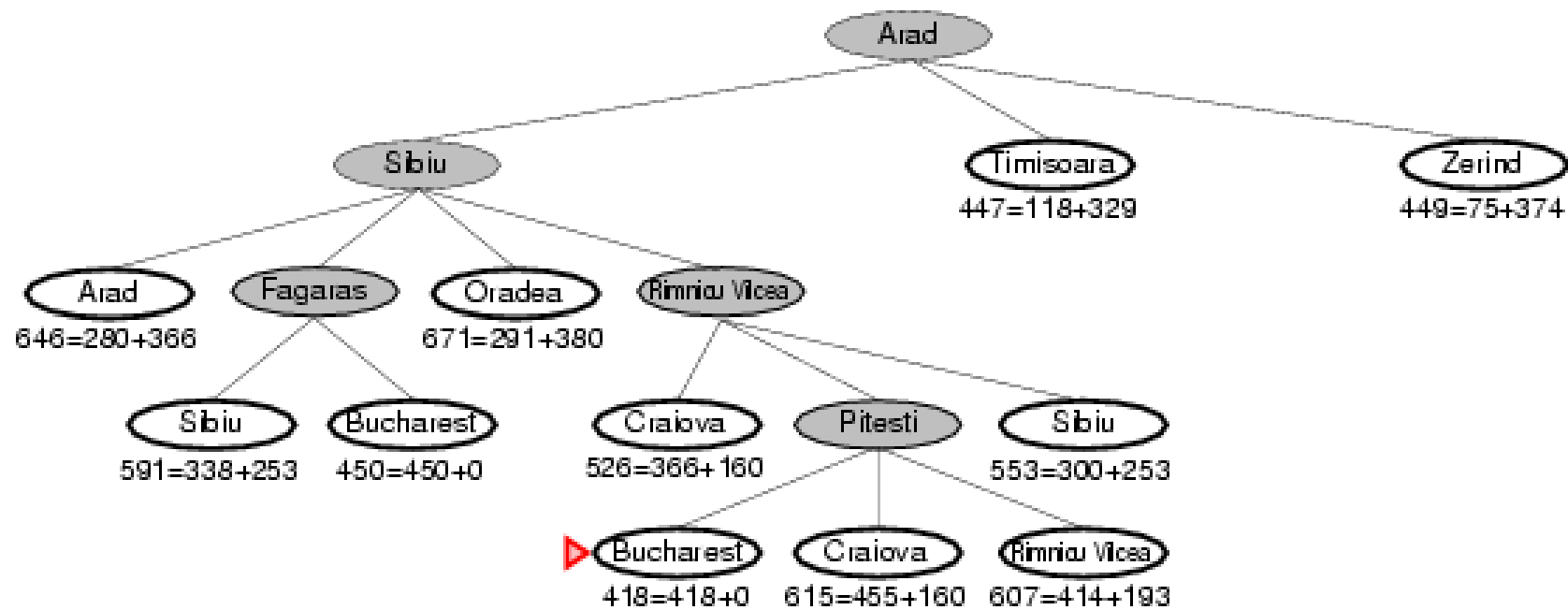












Оценка на метода A^* :

Стратегията е пълна, ако разклоненията (наследниците) на всеки възел от ГС са краен брой и цената на преходите е положителна, и оптимална, ако евристиката е *приемлива* (оптимистична), т.е. никога не надценява стойността (цената) h^* на оставащия път (ако $h^*(node) \geq h(node)$ за всеки възел $node$).

Времевата и пространствената сложност на метода са експоненциални.

Примери за приемливи (оптимистични) евристики

При задачата за 8-те плъзгащи се плочки (the 8-puzzle problem):

- $h_1(S)$ = броя на плочките, чиято текуща позиция е различна от позицията им в целевото състояние
- $h_2(S)$ = тоталното (сумарното) Манхатънско разстояние между текущата и целевата позиция

Манхатънско разстояние (Manhattan Distance) между точките (X_i, Y_i) и (X_j, Y_j) : $D = |X_i - X_j| + |Y_i - Y_j|$.

Пример

7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

Goal State

$$h_1(\text{Start}) = 8$$

$$h_2(\text{Start}) = 3+1+2+2+2+3+3+2 = 18$$