

# Интеллектуальные системы

# **Тема 2. Поиск в пространстве состояний**

- **Поиск в пространстве состояний** (англ. *state space search*) — группа методов, предназначенных для решения задач искусственного интеллекта

# **1 Агенты, решающие задачи**

- Первым шагом в решении задачи является **формулировка цели** с учётом текущей ситуации и показателей производительности агента.

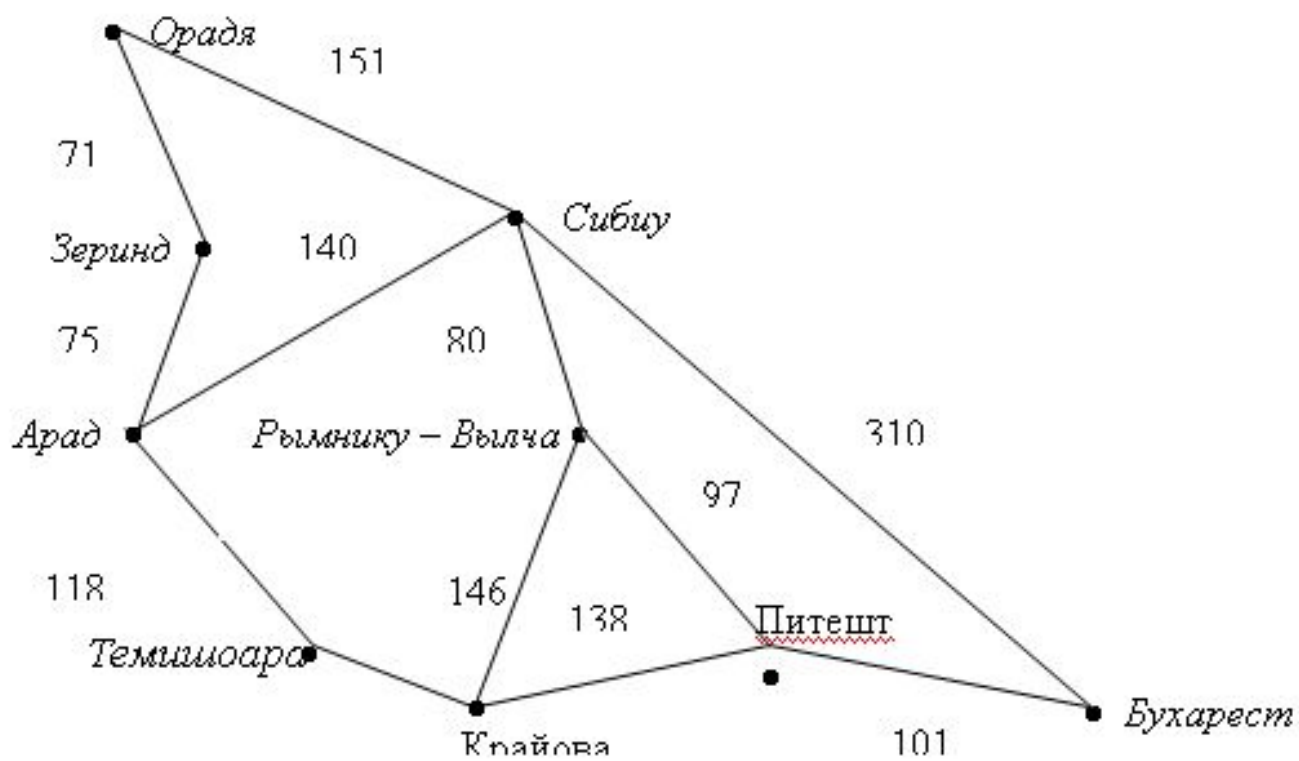
- **Формулировка задачи**  
представляет собой процесс определения того, какие действия и состояния следует рассматривать с учётом некоторой цели.

- Описанный процесс определения такой последовательности называется **ПОИСКОМ**.
- Любой алгоритм поиска принимает в качестве входных данных некоторую задачу и возвращает **решение** в форме последовательности действий.
- После выполнения этого решения агент формулирует новую цель.

# Задача формально определяется с помощью следующих четырёх **КОМПОНЕНТОВ**:

- *Начальное состояние*, в котором агент приступает к работе.
- *Функция определения преемника* – описание возможных действий, доступных агенту.





- Путём в пространстве состояний является последовательность состояний, соединённых последовательностью действий.

- *Проверка цели* позволяет определить, является ли данное конкретное состояние целевым состоянием.

- Функция *СТОИМОСТИ ПУТИ* назначает числовое значение стоимости каждого пути.

- *Решением* задачи является путь от начального состояния до целевого состояния.
- Качество решения измеряется функцией *стоимости пути*, а *оптимальное решение* имеет наименьшую стоимость пути среди всех прочих решений.



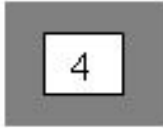


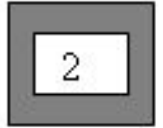








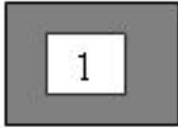


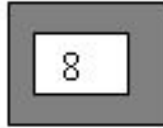
# Примеры задач

- Наиболее известные примеры решения задач подразделяются на два типа: упрощенные и реальные задачи.

- *Упрощенные задачи* предназначены для иллюстрации или проверки различных методов решения задач.
- *Реальные задачи*, которые действительно требуются людям, не имеют, как правило, единого приемлемого для всех описания



# Примеры упрощенных задач

					
					
					
Начальное состояние			Целевое состояние		

- Задачи со скользящими фишками часто используются в искусственном интеллекте для проверки новых алгоритмов поиска.
- Известно, что этот общий класс задач является NP-полным.

- Задачи относятся к классу недетерминированных полиномиальных задач (NP-классу, сокращение от Nondeterministic Polynomial), если существует алгоритм, позволяющий выдвинуть гипотезу о возможном решении, а затем проверить правильность этой гипотезы с помощью полиномиальных затрат времени.

# Примеры реальных задач

- Одними из наиболее распространённых являются *задачи поиска маршрута* в терминах заданных местонахождений и переходов между ними.

- *В задачах автоматического упорядочения сборки сложных объектов роботом цель состоит в определении последовательности, в которой должны собираться детали некоторого объекта.*

# Поиск решений

- Сформулировав определённые задачи, необходимо найти их решения. Такая цель достигается посредством поиска в пространстве состояний.

- Порядок, в котором происходит развёртывание состояний, определяется *стратегией поиска*.

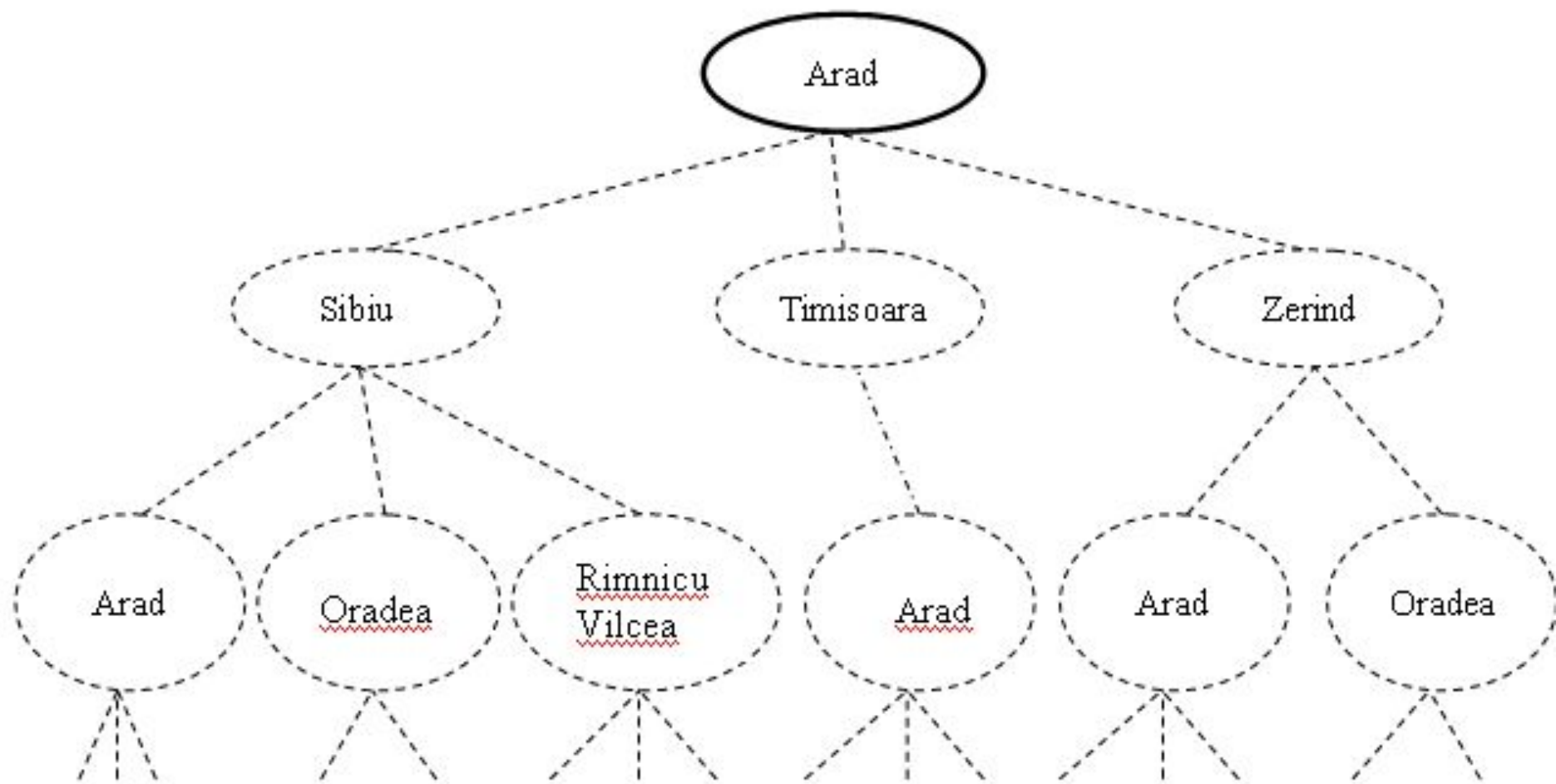


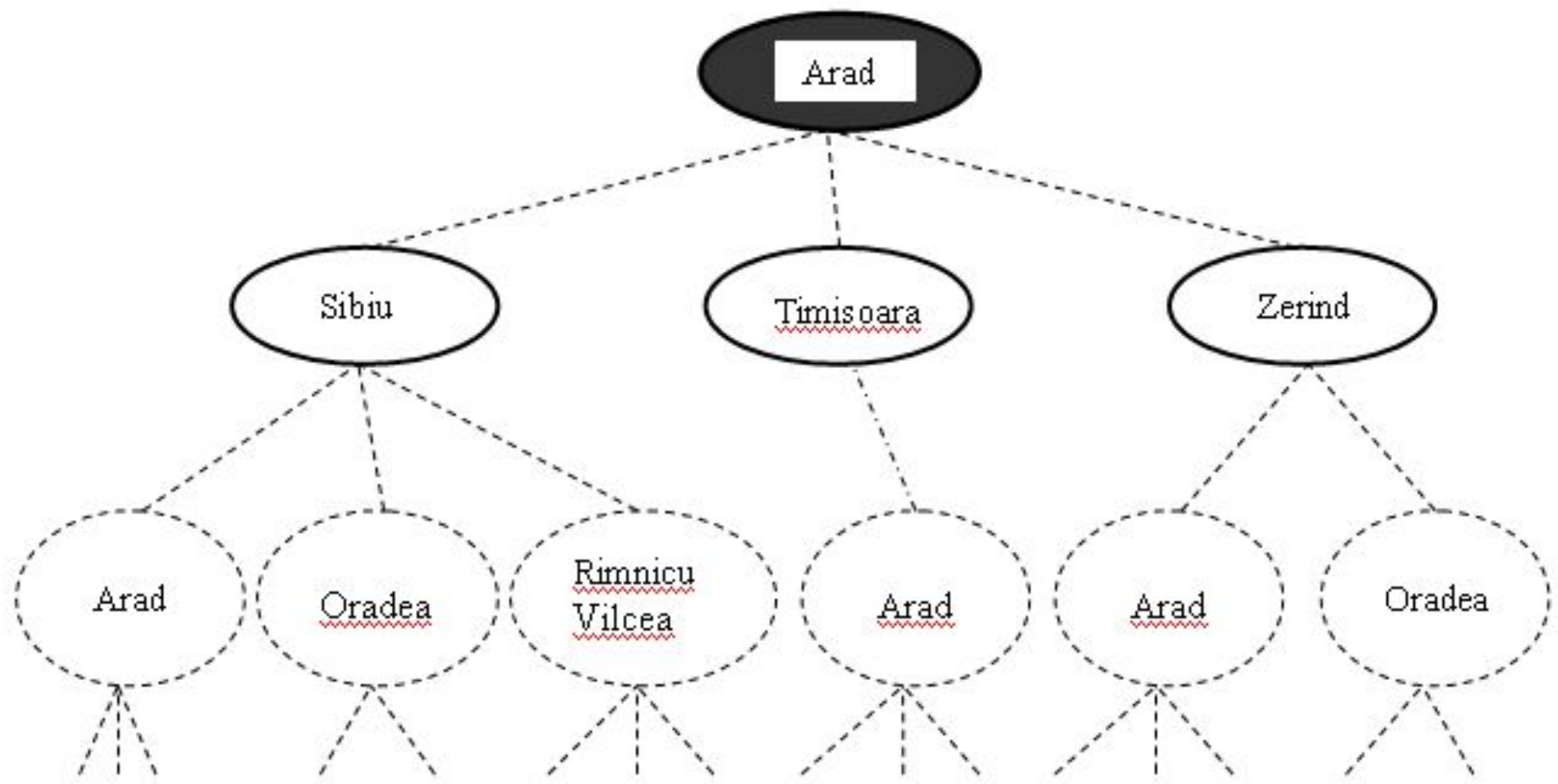
- Корнем этого дерева поиска является *поисковый узел*, соответствующий начальному состоянию
- *Первый этап* состоит в проверке того, является ли это состояние целевым.

- *Развёртывание* текущего состояния приводит к формированию его преемника - нового множества состояний

- *Суть поиска* состоит в том, что пока проверяется один вариант, другие откладываются в сторону на тот случай, когда первый вариант не приводит к решению.

- *Узлы* представляют собой структуры данных, применяемых при построении дерева поиска, а *состояние* соответствует конфигурации мира.





- Каждый узел имеет *родительский узел*, содержит данные о состоянии и имеет различные вспомогательные поля.

# Структура данных может включать в себя:

- *State* - соответствующее данному узлу состояние в общем пространстве состояний;
- Parent-Node – родительский узел;
- Action – действие, которое было применено к родительскому узлу для формирования данного узла;
- Path-Cost – стоимость пути от начального состояния до данного узла;
- Depth – количество этапов (глубина) пути от первоначального состояния до данного узла.



# **Производительность решения задачи**

# Принято оценивать производительность алгоритма с помощью следующих четырёх показателей

- *Полнота.* Гарантирует ли алгоритм обнаружение решения, если оно имеется?
- *Оптимальность.* Гарантирует ли данная стратегия нахождение оптимального решения в соответствие с выбранным критерием?
- *Временная сложность.* Время нахождения решения при данном алгоритме.
- *Пространственная сложность.* Необходимый объём памяти для осуществления поиска.

# **Слепые методы поиска решений**

- Методы поиска в пространстве состояний подразделяются на две группы: методы «слепого» и упорядоченного (эвристического) поиска.

- Методы «слепого» поиска называют также «не информированными» методами, так как в процессе поиска пути на графе не учитывается информация о степени близости текущей и целевой вершин.
- Это приводит к тому, что в методах «слепого» поиска выполняется полный просмотр всего пространства состояний.

# Методы "слепого" поиска можно разделить на три группы:

- Случайный поиск;
- Поиск «в глубину и ширину»;
- Алгоритм равных цен.

# Поиск с частичной информацией

