Lecture 1

AI :- creating machines that solves problems in a way that requires human intelligence

input /output :- interact with the environment

reasoning :-

1 - Deductive Reasoning :

[All people will die]   then if x is a person,
قاعدة عامة   x will die

if A C B & B C C   then A is subset of C

2 - Inductive Reasoning :
form generalizations from specific observations

3 - Abductive :-
best explanation from incomplete observations
غير كاملة

* Logic \ Reasoning is main aspect of AI
* Goal based problem solving is key Aspect of AI

* state $\xrightarrow{operator}$ state

Tree based Search:
    intial State: Root
    leaves; states with no children

Uninformed (Blind) Search:
    not efficient if space is large → | use huiristic
    (it tries All possible states) | it's better

huristic Search

intensification   diversification

← عندو روح زي كده في مكان
← نركز على state المناسبة   نركز على state المناسبين
← يكون تجريب CS و B و C ثم
← ياخذ وقت اكبر   ← لو الحركة من مكان لمكان كان
← وقته local   ← قريب المسافات تجيب اسرع

*** لازم نعمل توازن بين diversification₃ intensification₃

Search techniques

any path   best path   in face of
                 تاخذ اقل المسافة الموجودة   adversaries
              اقل الطرق زي ← اقل ال عدد اطول الطرق
              ناخذ على generations
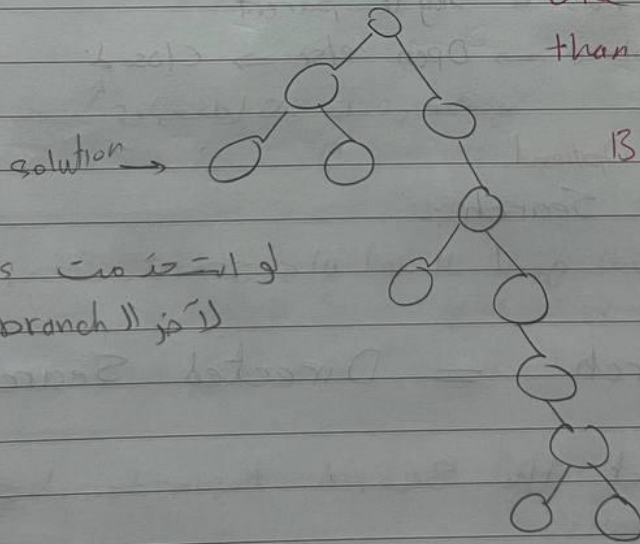   ← (الشطرنج)

difficulty:- find states & operators (successor function)

Branching factor:- all nodes have same number of children

→ need memory to keep track of visited nodes so searching will be terminated before returning to any visited nodes (cycle (مشوار في الحل))

Slide 14 :- Blind Search (All possible states)

_unbalanced tree:-

one branch is bigger than others

solution →

BFs is Better

لو التشتت dfs ممكن لا احتاج لل
branch الغير موجود فيه

BFs :- Queue , DFS:- Stack

lecture 2

Informed Search:- solutions كل على يحتوي ليس عليه نحصل معلومات عندما

Back tracking:- no memory needed (less Memory)
└ only one successor generated

DFS:- children كل ناخد اول 1
وبعدين انزل لأول واحد وأكبر وبعدين انزل الثاني وهكذا

BFS:- Queue FIFO
DFS :- Stack FILO

→ each node has a single parent
→ if node is unexplored ⟹ Open else ⟹ Closed
ما نعتبرها تاني مرة زرناها (علشان) visited

Branch & Bound → Optimal
* bidirectional Search:-
من نقطة start و لو من أي نقطة ال goal لو حاتلاقي مع بعض

* Informed Search = Directed Search

* heuristic:- direct the Search towards better
Solutions

not all nodes are put in the priority Queue ←
unlike uninformed Search

heuristic ← مدى قرب كل node من الهدف؟

evaluation function: ← أعرف الحل الصحيح يكون ولا لأ ؟

heuristic ← الرقم الي بيطلع أقل ساوي الرقم الحقيقي
heuristic function بتاخذ max لأنه كذا heuristic function

يتعامل مع كل problem بشيء معين/مختلف ← huerstic

(A)
| 19
(C) (C→25), B→32, D→35
| 10 F→17, E→19
(F)
| 20
(G)
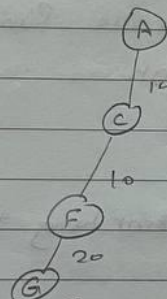
node closest to Goal ← Greedy ← A→C→F→G K=44 not optimal
goal بسبب heuristic تقديري مش goal ، أقصد اني بسبب heuristic

in A* search.

← اساس الزيارة بعتمد على node بدي لها visited؟
كم هو goal؟

← الزيارة بتحدد Cost + heurstic

lecture 3

Slide 8 :-

use an evaluation function to choose next node

في الطرق دي بنحتاج evaluation اللي هي criteria heurstic

في حالة ده بنختار node معين بتحكم في criteria heurstic

Greedy Search :- expand the node closest to the Goal

بيعتمد على heurstic function بشكل كامل

* not, the best solution
    always

Optimal :- min or max according to the nature of
the problem

→ **A* Search :-** Combine Cost & Greedy Search (heurstic)

$$f(n) = g(n) + h(n)$$
$$\quad\quad cost\ so\ far \quad\quad\quad heurstic$$

→ comparison between A* & Greedy

A* :- Complete & optimal

Greedy :- neither

both have Same Time & Space Complexity

lecture 4

Greedy :- heurstic Function

A* :- ~~heuristic~~ evaluation Function Cost + heurstic مع دينا

→ Goal State :- Solution الحل الوصول اليه حتى

→ local Search :- نبحث فقط فى حقلة معينة dass Search حولنا

→ ascent :- maximize   descent :- minimize

Tabu Search :- (مش نوصل لوف) Tabu list مفهوم

اللى احطوطه

Hill Climbing :-

1- From Generation to another there is only One Solution

2- <mark>not</mark> Complete

* neighbouring Solutions :-

ا ← جيب الحلول اللى منه الحل يتناقى (current)

ا ← لما اغنى فى current Solution ← ماتتغير وتيجى neighbouring Solution

* most significant improvemat :- evalution Function لما ترفع

Any time property:

في بعض المشاكل إلزامة كل كويس بس نقط آخر الأمر بيبقي مرضي يتنفع عليه Hill Climbing ⟵

i.e. Real Time Systems :- Auto Cars

→ Doesnot accept any Bad Solutions

مش بتنقل من state للتاني غير لو أكوم أضمن منها

★ ridge

من x_0 نقط لقوم بس متشابتين بروح لـ x_1 (مش بيمشي بقرار)(adv)

parameters in SA:

1- T :- Temerature (Intial)

2- α : $\boxed{\alpha < 1}$ معدل الحرارة

3- T_{mn} :- Temperature (minimum)

⟵ لازم نحدد بس Threshold ايه 0,1

بحيث +فوق لو p أكبر من T نمشي فالقاله Solution لأن أقل للوقته

$\boxed{T = \alpha * T}$ كل مرة بنعمل Temperature update

$T \leftarrow \alpha * T$

lecture 5

→ $p = e^{\left(-\frac{\Delta E}{T}\right)}$

وكلما قلت (Temperature) → الـ probability يقل تقبل

← مع الوقت بنبه انه يقبل حلول سيئة يقل .

Termination of SA could be

١- عدد معين من التكرارات

٢- T توصل لـ T minimum

(T)

كل ما الوقت↑ يقل دا معناه البحث كمان بنشأ لقوة والـ probability تقل

* هيبقى قيمة ثابتة للـ T
بنجرب قيم مختلفة للـ T لغاية ما اوصل لاحسن قيمة ← parameter Tuning
└→ نظبط الـ parameter

* مش لازم T توصل لـ zero ← أوصل لـ T_{min}
T_{min} بنحط لغاية الـ حد الأدنى

* $T_{initial}$ المفروض تكون عالية
* لو كانت قليلة ← الـ probability تكون قليلة منها
يبقى شبه Hill Climbing
يقبل حلول كويسة فقط

* لو كنت عالية جدًا ← هيقبل اكل الحلول
تقريبًا وتقرب للـ random Search

* هيكم انقاص الـ T بشكل خطي
$$T = T + \alpha$$

* وهيكم انقاصها بشكل هندسي
$$T = T \cdot \alpha \qquad \alpha : [0,1]$$

* كل ما زادت قيمة α ← المدة اللازمة للتوقف هتقل
للـ T هتقل ببطء ويوصل للـ Tmin

* عدد التكرارات لكل درجة حرارة
* هيكم تكوين 1 يعني بعد كل لفة اغير درجة الحرارى $T \cdot \alpha \leftarrow T$ الاحتمال

* هيكم العدد يكون ثابت زي اعناك
* هيكم يكون   Dynamic

تكرارات كثيرة في درجات حرارة كثيرة   تكرارات قليلة في درجات حرارة قليلة
Diversefication   intensefication

hill climbing :- intensification

* لو التكرار كثير زاد الاعتراض الحرارى تقل ببطء (ازود α)

Tabu Search:-

   avoid revisiting previously explored Solutions → escape local optima

* Tabu list:- contain previously explored Solutiont

* Tabu Tenure:- عدد التكرارات اللي يظل الحل فيها محظور
السولوشن Solutions حيث يتقفل في Tabu list لفترة معينة يعرب مايفتح

Termination check:-
* حيث قادرة أسرع. كل Solutions موجودة في Tabu list

مش قادر في الآخر بـ