**Iterative Deepening Search (IDS)**

**Abstract**

When making choices, decision makers often either lack information about alternative lack the cognitive capacity to analyze every alternative. To capture these situations, we formulate a framework to study behavioral search by utilizing the idea of consideration sets. Consumers engage in a dynamic search process. At each stage, they consider only those options in the current consideration set. We provide behavioral postulates characterizing this model. We illustrate how one can identify both search paths and preferences.

**1 .Introduction**

Classical choice theory assumes that a decision maker (DM) chooses the best option among all the available alternatives. This might not be practical, especially in situations where (i) aDM cannot easily view all the alternatives, and she must actively seek out alternatives, as when she buys a house or a car (incomplete information), or (ii) there is a menu in front of her but either the menu is too long or her time is too short (limited cognitive capacity).In these situations, the budget set must be explored by the DM. As Herbert Simon pointed out many years ago, exploration of the budget set is one of the most important aspects of real world decision making but is neglected in the standard theory where the whole budget set is assumed to be evaluated simultaneously. This paper proposes a new descriptive model of decision making in which a DM explores her budget set (unlike the standard model) and has a stable preference (as in the standard model). Our procedure is dynamic and incorporates the idea of limited search into a model of decision making. We utilize consideration sets, which have been extensively studied in marketing.

**2.Problem Set**

This project may be undertaken individually or in pairs. If working in a pair, please state clearly the names of the two people undertaking the project and the contributions that each has made. Only one submission should be made per pair.

The purpose is to adapt the Iterative Deepening Search (IDS) method learnt in class to a realistic problem that is of relevance to Industry.

The environment is the warehouse that was used in Project 1, except that it has 15 locations, instead of 10. The 15 locations now represent divisions in the warehouse that stock different types of items. For example, location 1 is a division that stocks Electronics items while division 2 holds Clothes, and so on.

For reasons of efficiency the lead designer in the AI team has decided to map the warehouse in the hope that it will optimize the robot navigation process. Also, it was observed in the trial run undertaken earlier that the sensors were not accurate enough to support a 1-step lookahead search and it was thus decided to use the sensors to perceive the current location only and not the surrounding neighborhood.

**3.Algorithm Module**

In computer science, iterative deepening search or more specifically iterative deepening depth-first search (IDS or IDDFS) is a **state space/graph search strategy** in which a depth-limited version of depth-first search is run repeatedly with increasing depth limits until the goal is found.

// Returns true if target is reachable from

// src within max\_depth

bool IDDFS(src, target, max\_depth)

for limit from 0 to max\_depth

if DLS(src, target, limit) == true

return true

return false

bool DLS(src, target, limit)

if (src == target)

return true;

// If reached the maximum depth,

// stop recursing.

if (limit <= 0)

return false;

foreach adjacent i of src

if DLS(i, target, limit?1)

return true

return false

PseudoCode Run

**unction** Build-Path(s, μ, B) **is**

π ← Find-Shortest-Path(s, μ) *(Recursively compute the path to the relay node)*

remove the last node from π

**return** π

{\displaystyle \circ }

B *(Append the backward search stack)*

**function** Depth-Limited-Search-Forward(u, Δ, F) **is**

**if** Δ = 0 **then**

F ← F

{\displaystyle \cup }

{u} *(Mark the node)*

**return**

**foreach** child **of** u **do**

Depth-Limited-Search-Forward(child, Δ − 1, F)

**function** Depth-Limited-Search-Backward(u, Δ, B, F) **is**

prepend u to B

**if** Δ = 0 **then**

**if** u **in** F **then**

**return** u *(Reached the marked node, use it as a relay node)*

remove the head node of B

**return null**

**foreach** parent **of** u **do**

μ ← Depth-Limited-Search-Backward(parent, Δ − 1, B, F)

**if** μ

{\displaystyle \neq }

**null** **then**

**return** μ

remove the head node of B

**return null**

**function** Find-Shortest-Path(s, t) **is**

**if** s = t **then**

**return** <s>

F, B, Δ ← ∅, ∅, 0

**forever do**

Depth-Limited-Search-Forward(s, Δ, F)

**foreach** δ = Δ, Δ + 1 **do**

μ ← Depth-Limited-Search-Backward(t, δ, B, F)

**if** μ

{\displaystyle \neq }

**null then**

**return** Build-Path(s, μ, B) *(Found a relay node)*

F, Δ ← ∅, Δ + 1

**4. Conclusion**

Iterative Deepening Search (IDS) is an iterative graph searching strategy that takes advantage of the completeness of the Breadth-First Search (BFS) strategy but uses much less memory in each iteration (similar to Depth-First Search).

IDS achieves the desired completeness by enforcing a depth-limit on DFS that mitigates the possibility of getting stuck in an infinite or a very long branch. It searches each branch of a node from left to right until it reaches the required depth. Once it has, IDS goes back to the root node and explores a different branch that is similar to DFS.

**Time & space complexity**

Let’s suppose that we have a tree where each node has b children. We will consider this our branching factor and take d as the depth of the tree.

Nodes at the bottom-most level, which would be level d*d*, will be expanded exactly once; whereas, nodes on level d-1*d*−1 will be expanded twice. The root node of our tree will be expanded d+1*d*+1 times. If we sum all these terms up, it will be:

(d)b + (d-1)b^2 + ... + (3)b^{d-2} + (2)b^{d-1} + b^d(*d*)*b*+(*d*−1)*b*​2​​+...+(3)*b*​*d*−2​​+(2)*b*​*d*−1​​+*b*​*d*​​

This summation will result in time complexity of O(b^d)*O*(*b*​*d*​​)

The space complexity for IDS is O(bd)*O*(*bd*). Here, we assume that b is a constant, and all children are generated at each depth of the tree and stored in a stack during DFS.

**Performance analysis**

It may look like IDS has a large overhead in the form of repeatedly going over the same nodes again and again, but this turns out to be much of a big deal. This is because most nodes in a tree are at the bottom levels, which are visited only once or twice by the algorithm. Because of this, the cost is kept to a bare minimum since the upper-level nodes do not make up the majority of the nodes in a tree.

REFERENCES

1. *Korf, Richard (1985). "Depth-first Iterative-Deepening: An Optimal Admissible Tree Search". Artificial Intelligence.****27****: 97–109.*[*doi*](https://en.wikipedia.org/wiki/Doi_(identifier))*:*[*10.1016/0004-3702(85)90084-0*](https://doi.org/10.1016%2F0004-3702%2885%2990084-0)*.*
2. [**^**](https://en.wikipedia.org/wiki/Iterative_deepening_depth-first_search#cite_ref-3) *David Poole; Alan Mackworth.*[*"3.5.3 Iterative Deepening‣ Chapter 3 Searching for Solutions ‣ Artificial Intelligence: Foundations of Computational Agents, 2nd Edition"*](https://artint.info/2e/html/ArtInt2e.Ch3.S5.SS3.html)*. artint.info. Retrieved 29 November 2018.*
3. ^ [Jump up to:***a***](https://en.wikipedia.org/wiki/Iterative_deepening_depth-first_search#cite_ref-rn3_4-0) [***b***](https://en.wikipedia.org/wiki/Iterative_deepening_depth-first_search#cite_ref-rn3_4-1) [***c***](https://en.wikipedia.org/wiki/Iterative_deepening_depth-first_search#cite_ref-rn3_4-2) [***d***](https://en.wikipedia.org/wiki/Iterative_deepening_depth-first_search#cite_ref-rn3_4-3) [*Russell, Stuart J.*](https://en.wikipedia.org/wiki/Stuart_J._Russell)*; [Norvig, Peter](https://en.wikipedia.org/wiki/Peter_Norvig" \o "Peter Norvig) (2003),*[*Artificial Intelligence: A Modern Approach*](http://aima.cs.berkeley.edu/)*(2nd ed.), Upper Saddle River, New Jersey: Prentice Hall,*[*ISBN*](https://en.wikipedia.org/wiki/ISBN_(identifier))[*0-13-790395-2*](https://en.wikipedia.org/wiki/Special:BookSources/0-13-790395-2)
4. [**^**](https://en.wikipedia.org/wiki/Iterative_deepening_depth-first_search#cite_ref-5) *Russell; Norvig (1994). Artificial Intelligence: A Modern Approach.*