Iterative Depth-First Search for FOND Planning

Ramon Fraga Pereira[1] André Grahl Pereira[2] Frederico Messa[2] Giuseppe De Giacomo[1]

[1]Sapienza University of Rome, Rome, Italy [2]Federal University of Rio Grande do Sul, Porto Alegre, Brazil

Motivation and Contributions

Fully Observable Non-Deterministic (FOND) planning models uncertainty through actions with *non-deterministic* effects with uniform probabilities over the actions' effects.

Existing FOND planners are effective and employ a wide range of techniques. However, most of the existing FOND planners are not robust for dealing with both non-determinism and task size.

The main contributions of this work are as follows:

- **Iterative Depth-First Search** (IDFS) algorithms for FOND Planning:
  - Our algorithms are explicitly designed for solving FOND Planning;
  - Use heuristics to make the iterative process effective in FOND;
  - IDFS is effective to deal with different non-deterministic aspects of FOND Planning;

Background and Notation

Definition (FOND Planning Task)

A FOND *planning task* is tuple $\Pi = \langle \mathcal{D}, s_0, s_* \rangle$:

- $\mathcal{D} = \langle \mathcal{F}, \mathcal{A} \rangle$ is a **non-deterministic domain model**, where $\mathcal{F}$ is a set of *fluents*, and a set of **non-deterministic actions** $\mathcal{A}$.
  Every $a \in \mathcal{A}$ consists of $a = \langle pre, \mathrm{EFFS} \rangle$, where:
  - $pre(a)$ represents the **preconditions**; and
  - $\mathrm{EFFS}(a)$ represents the **set of possible effects** of a;
- s_0 is the **initial state**;
- s_* is the **goal condition**;

The application of $\mathrm{EFFS}(a)$ to a state s generates a set of possible successor states $\mathrm{SUCCS}(s,a) = \{\textsc{succ}(s, \mathit{eff}) \mid \mathit{eff} \in \mathrm{EFFS}(a)\}$.

Definition (FOND Planning Solution Policy π)

A *solution* to a FOND planning task Π is a **policy** π, a *partial function* that maps *non-goal states* s into *actions* $a \in \mathcal{A}$.

A *policy* π induces π-*trajectories*, a non-empty sequence of states $\langle s^0, s^1, \dots s^k \rangle$, such that $s^{i+1} \in \mathrm{SUCCS}(s^i, \pi(s^i)), \forall i \in \{0, 1, \dots, k-1\}$.

A *policy* π is *closed* if any π-trajectory starting from s_0 ends either in a *goal state* or in a state defined in the policy π.

Definition (Strong Policy)

A policy π is a *strong policy* for Π if it is closed and no π-trajectory passes through a state more than once.

Definition (Strong Cyclic Policy)

A policy π is a *strong cyclic policy* for Π if it is closed and any π-trajectory starting from s_0 which does not end in a goal state, ends in a state s' such that exists another π-trajectory starting from s' ending in a goal state.

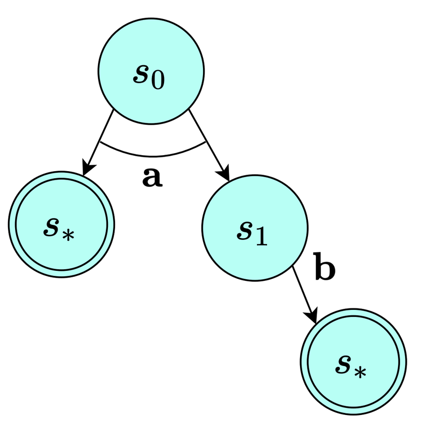


Figure 1. Strong Policy.

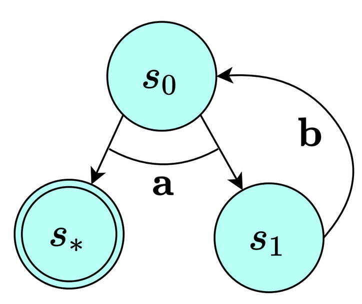


Figure 2. Strong Cyclic Policy.

Evaluation Function $\mathcal{F}$ for FOND

We define the f-value[a] of a state s as:

$$f(s) = g(s) + h(s). \tag{1}$$

We **evaluate the successor states** $\mathrm{SUCCS}(s,a)$ using the *Evaluation Function* $\mathcal{F}_\xi$. $\mathcal{F}_\xi$ uses a function ξ to aggregate the f-values of states in $\mathrm{SUCCS}(s,a)$:

$$\mathcal{F}_{\min}(\mathrm{SUCCS}(s,a)) \equiv \min_{s' \in \mathrm{SUCCS}(s,a)} f(s') \tag{2}$$

$$\mathcal{F}_{\max}(\mathrm{SUCCS}(s,a)) \equiv \max_{s' \in \mathrm{SUCCS}(s,a)} f(s') \tag{3}$$

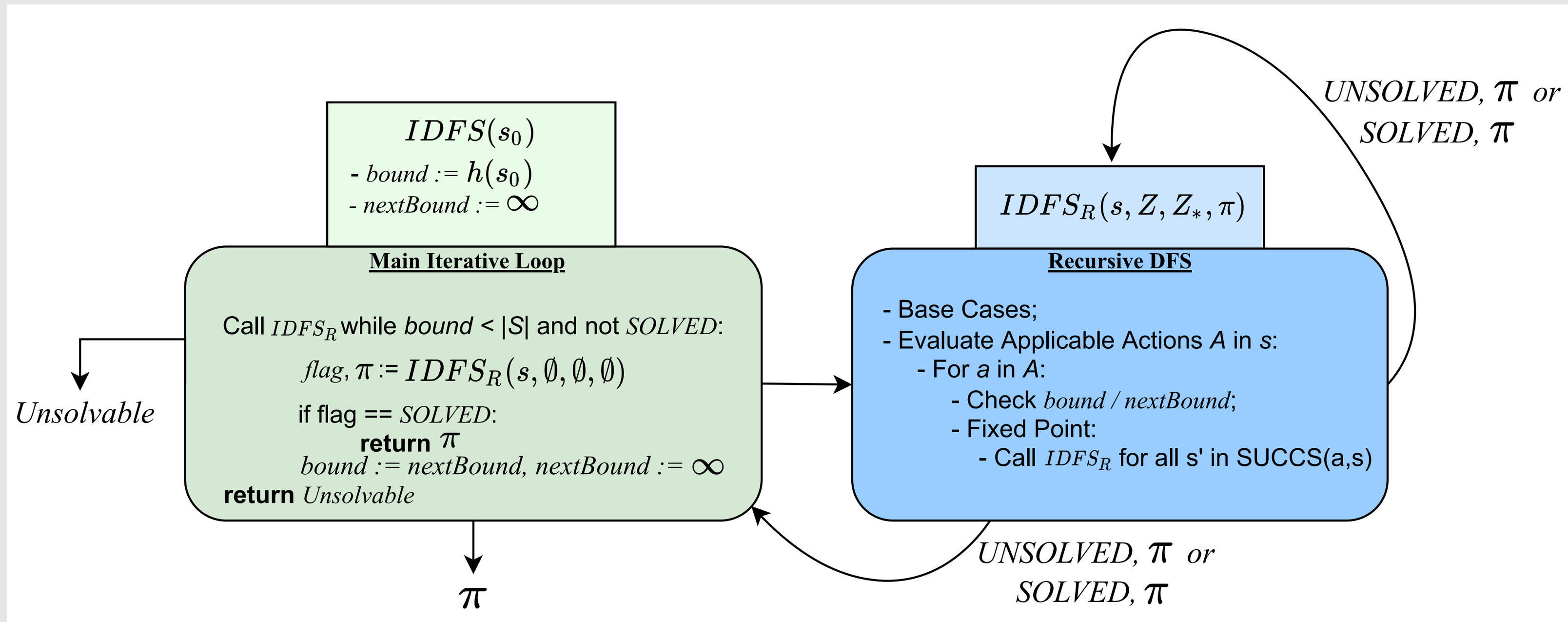
$\mathcal{F}_\xi$ is "pessimistic" when $\xi = \max$, whereas it is "optmistic" when $\xi = \min$.

[a] $g(s)$ represents the search depth from s_0 to s

Iterative DFS for FOND Planning

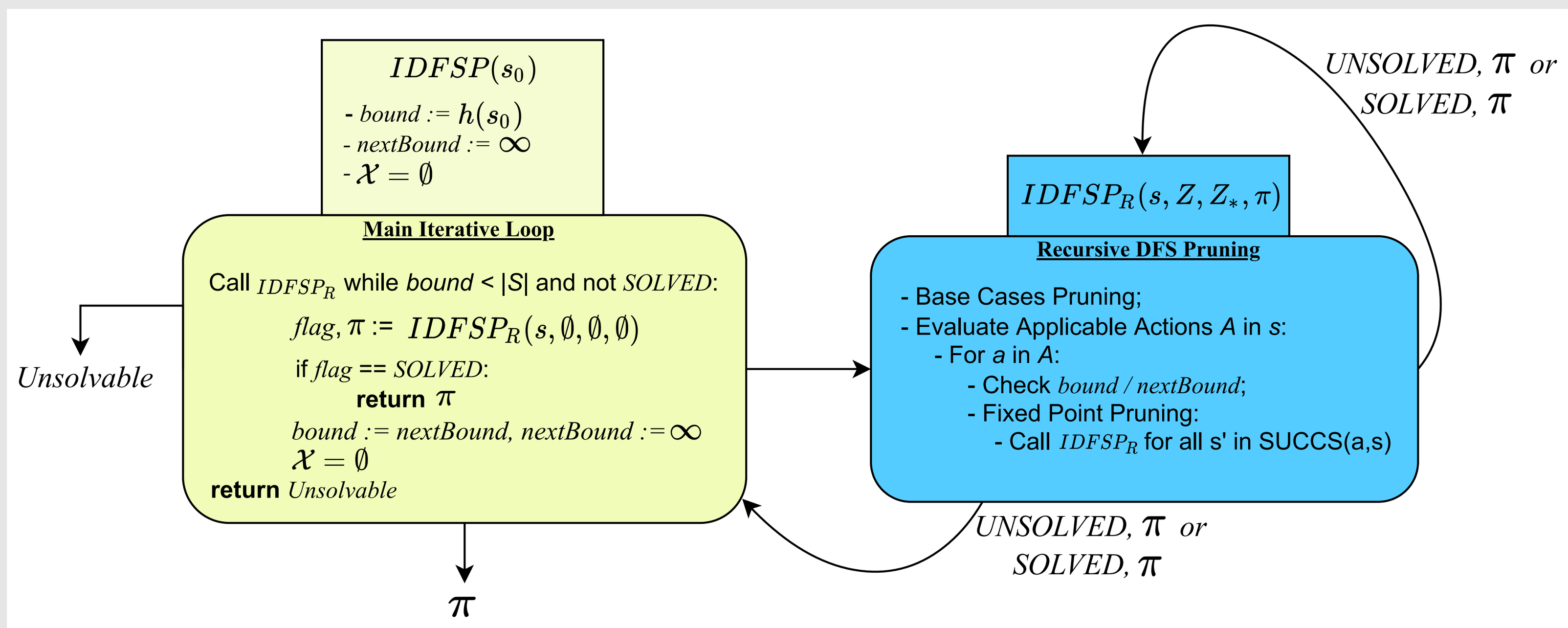
- IDFS performs a series of **bounded** depth-first searches;
- IDFS produces a strong cyclic policy in a **bottom-up way**;
  - It only adds an action to the policy if it determines that the resulting policy with the additional action has the potential to become a strong cyclic policy without exceeding the current search-depth bound.

IDFS



- Z **ancestors** of s; Z_* **ancestors** of s that has achieved s_* through a π-*trajectory*;
- **Base Cases:** Check if $s \models s_*$ OR $s \in Z_*$ OR $\pi(s) \neq \perp$. If so, it returns *SOLVED*; And, if $s \in (Z \setminus Z_*)$ (s visited before), if so, it returns *UNSOLVED*.
- **Evaluate Applicable Actions in** s**:** Check *bound* for $\mathrm{SUCCS}(s,a)$.
  1. If $\mathcal{F}_\xi(\mathrm{SUCCS}(s,a)) > bound)$ and $Z_* = \emptyset$: **discards** a on s, and proceeds to the next action.
  2. If if $g(n) + 1 > bound$: **discards** a on s, and proceeds to the next action.
- **Fixed Point:**
  - It maps $s \mapsto a$ to π **ONLY** if **ALL** recursive calls on states of $\mathrm{SUCCS}(s,a)$ returned SOLVED;
  - If not, it **discards** a on s, and proceeds to the next action.

IDFS Pruning (IDFSP)



IDFSP **prunes** *non-promising* states. E.g., it prunes states whose all successor states for all applicable actions have $\mathcal{F}$ **greater than** the *current bound*.

Experiments and Evaluation

We use two FOND planning benchmarks: IPC-FOND and NEW-FOND.

Our planner is called PALADINUS.

We compare PALADINUS with: MYND, PRP, and FONDSAT.

Planner	Solved Tasks (#590)
PALADINUS IDFSP ($\mathcal{F}_{\min}$, h^{MAX})	337
PALADINUS IDFSP ($\mathcal{F}_{\min}$, h^{FF})	406
PALADINUS IDFSP ($\mathcal{F}_{\min}$, h^{ADD})	411
PALADINUS IDFSP ($\mathcal{F}_{\max}$, h^{MAX})	334
PALADINUS IDFSP ($\mathcal{F}_{\max}$, h^{FF})	380
PALADINUS IDFSP ($\mathcal{F}_{\max}$, h^{ADD})	422
FONDSAT	276
PRP (h^{MAX})	292
PRP (h^{FF})	412
PRP (h^{ADD})	389
MYND (h^{MAX})	180
MYND (h^{FF})	265
MYND (h^{ADD})	289

Table 1. Overall coverage results.

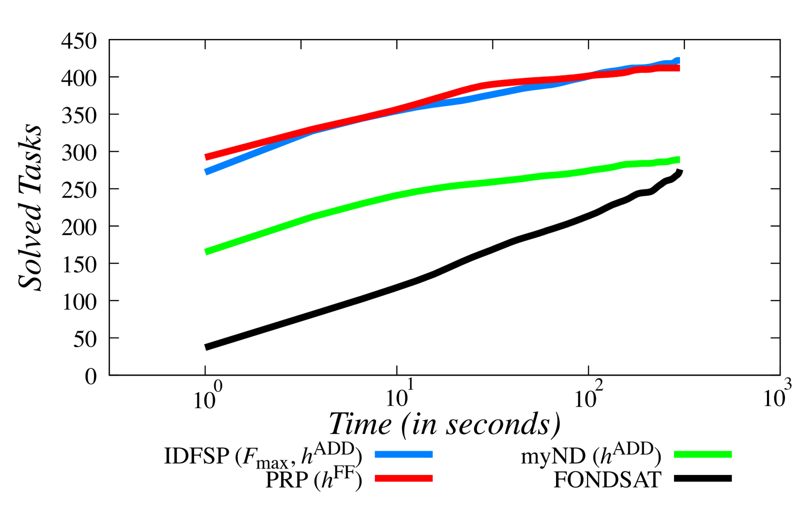


Figure 3. All Benchmarks.

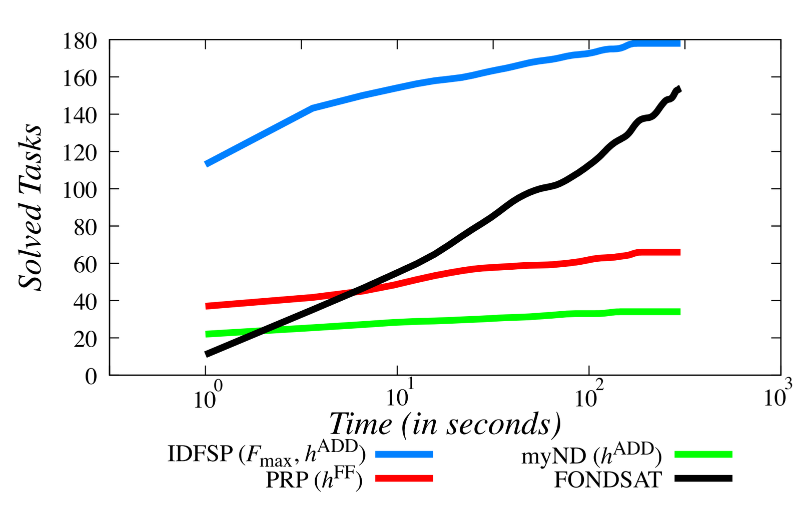


Figure 4. NEW-FOND Benchmarks.

pereira@diag.uniroma1.it