General Policies, Representations, and Planning Width

Blai Bonet[1] and Hector Geffner[2]

[1] Universitat Pompeu Fabra, [2] ICREA & Universitat Pompeu Fabra

arXiv:2012.08033

Motivation and Contribution

Width-based search algorithms (e.g. IW, SIW, BFWS, etc) are quite effective in planning. **Why?**

- **Goal:** Address this question by connecting notions of:
  - Bounded width
  - General policies for collections of problems
  - Decomposition of problems into subproblems
- **Explanation:** General policies underlie notion of width; roughly, bounded number of features implies bounded width
- **Also:**
  - General formulation of decomposition and serialized width
  - General effective lenguage for expressing decompositions

Long paper (proofs + ext. discussions) in **arXiv:2012.08033**

Basic Algorithms: IW(1), IW(k), and IW

- IW(1) is breadth-first search that **prunes states** that don't make a feature true for first time (from set F of boolean features)
- IW(k) is like IW(1) but over set F^k of conjunctions of up to k features in F
- **IW(k) expands up to $|F|^k$ nodes and runs in polytime** $O(|F|^{2k-1})$
- Bounded search and exploration based on state structure
- Classical Planning: F is set of ground atoms
- IW runs IW(1), IW(2), …, IW(k) until solved, or $k = k_{max}$

Variations of IW

- SIW (Serialized IW): use IW greedily to decrease number of unachieved goals $\#g$ (assumes conjunctive top goal)
- BFWS(m): **complete best-first** guided by width-based novelty measure m
- Dual-BFWS: **incomplete** BFWS followed by (complete) BFWS

Definition of Width

Width of P bounded by k, $w(P) \leq k$, if there is admissible chain of atom tuples $\theta = (t_0, t_1, \ldots, t_n)$ such that $|t_i| \leq k$, and:

- t_0 holds at initial state s_0 of P
- any optimal plan for t_i can be extended with an action into opt. plan for t_{i+1}
- any optimal plan for t_n is an optimal plan for P

Set $w(P) := 0$ if goal can be reached in 0 or 1 step, and $w(P) := N+1$ if P has no solution

Width of class $\mathcal{Q}$ bounded by k if $w(P) \leq k$ for each P in $\mathcal{Q}$

Theorem (Lipovetzky and G., 2012)
If $w(P) \leq k$, IW(k) solves P **optimally**

Generalized Planning: Features and Policies

Features over class $\mathcal{Q}$ are **state functions:** Boolean p and numerical n (assumed to be linear in number N of atoms)

Policy π_Φ is set of **policy rules** $C \mapsto E$ over features Φ:

- Boolean conditions in C: p, $\neg p$, $n=0$, $n>0$
- Effects in E: p, $\neg p$, $p?$, $n\downarrow$, $n\uparrow$, $n?$

Transition (s, s') in P **compatible with** π_Φ if for some $C \mapsto E$:

- feature valuation $f(s)$ satisfies condition C
- pair $(f(s), f(s'))$ is compatible with effect E

Definition (Solutions)
Policy π_Φ **solves** P if all maximal trajectories that are compatible with π_Φ reach the goal. π_Φ solves class $\mathcal{Q}$ if it solves each P in $\mathcal{Q}$

Example: Blocksworld

Policy π_Φ for solving $\mathcal{Q}_{clear}$ of problems where goal is to get $clear(x)$ and hand-empty:

$\{\neg H, n>0\} \mapsto \{H, n\downarrow\}$ (pick top block above x)
$\{H\} \mapsto \{\neg H\}$ (put held block away)

Features $\Phi = \{H, n\}$ are 'holding' and 'number of blocks above x'

Policy π_Φ solves class $\mathcal{Q}_{clear}$ **optimally**; also:

- Features Φ **distinguish the goals:** $n=0$ and $\neg H$ iff goal
- π_Φ is **Markovian** (see paper)

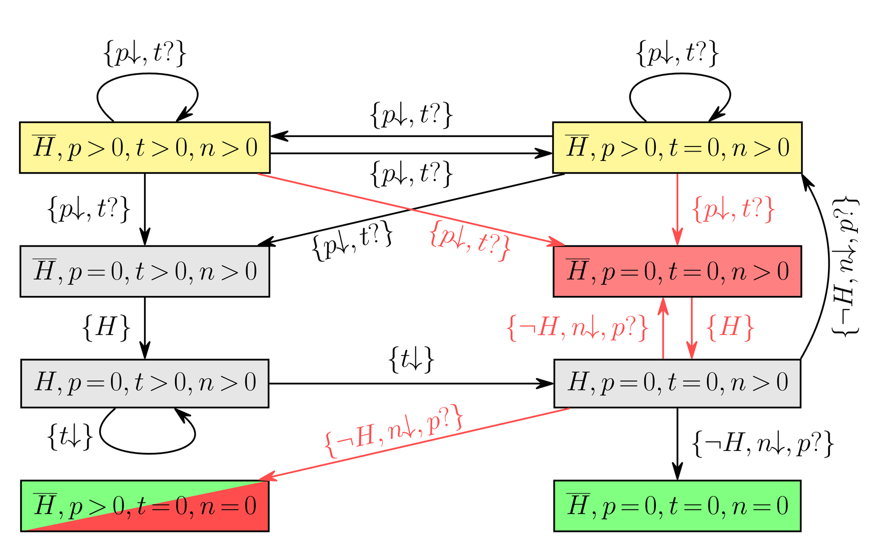
Example: Delivery

Policy π_Φ solves class $\mathcal{Q}_D$ of problems with packages that have to be delivered to target cell:

$\{\neg H, p>0\} \mapsto \{p\downarrow, t?\}$ (go to nearest pkg)
$\{\neg H, p=0\} \mapsto \{H\}$ (pick package)
$\{H, t>0\} \mapsto \{t\downarrow\}$ (go to target cell)
$\{H, t=0, n>0\} \mapsto \{\neg H, n\downarrow, p?\}$ (drop package)

Features $\Phi = \{H, p, t, n\}$ are 'holding package', 'distance to nearest package', 'distance to target', and 'number of undelivered packages'

- Features Φ **distinguish the goals:** $n=0$ iff goal state
- Policy **optimal** and **Markovian** in subclass $\mathcal{Q}_{D1} \subset \mathcal{Q}_D$



Policy graph for Delivery. Yellow/green nodes stand for initial/goal states, and red nodes/edges for states/transitions that don't arise in instances. Graph is **terminating** and **goal connected**. Policy is **closed** and **sound** for $\mathcal{Q}_D$ and $\mathcal{Q}_{D_1}$.

Relation of General Policies and Width

Theorem
Let Φ be set of features over $\mathcal{Q}$ that **distinguish goals**, and π_Φ **optimal** and **Markovian** policy for $\mathcal{Q}$. For any P in $\mathcal{Q}$:

- IW$_\Phi$ solves P **optimally** in polytime $O(N^{|\Phi|})$, where IW$_\Phi$ is like IW but works with feature valuations $f(s)$ instead atoms
- $w(P) \leq |\Phi|$ if features in Φ are **represented in** P
- $w(P) \leq k$ if for any **feature valuation** f_i reached by π_Φ, there is **atom tuple** t_i such that $|t_i| \leq k$ and optimal plans for t_i and f_i are the same

Example: General plan for $\mathcal{Q}_{clear}$ + Theorem yields $w(\mathcal{Q}_{clear})=1$

Admissible Chains from Policies

Let $\theta = (t_0, t_1, ..., t_n)$ be a chain of atom tuples

Features: $\bar{t}_i$ is feature so that $s \vDash \bar{t}_i$ iff $s \vDash t_i$ and $s \nvDash t_j$ for $j > i$

Policy: π_θ given by rules $\{\bar{t}_i\} \mapsto \{\bar{t}_{i+1}, \neg\bar{t}_i\}$, $i = 0, 1, \ldots, n-1$

Theorem (Characterization of admissible chains)
θ is admissible for P iff policy π_θ is **optimal** and **Markovian** for P, and θ is **feasible**: t_0 holds in initial state and the optimal plans for t_n are of length n and also optimal for P

Theorem
Let π_Φ be an optimal policy for $\mathcal{Q}$. If for any P in $\mathcal{Q}$, there is feasible chain θ of size $\leq k$ so that π_θ is a **projection** of π_Φ in P that is Markovian, $w(\mathcal{Q}) \leq k$

Example: General plan for $\mathcal{Q}_{D1}$ + Theorem yields $w(\mathcal{Q}_{D_1}) = 2$

Decomposition of Problems: Serializations

Definition (Serialization)
A **serialization** is pair $(\Phi, \prec)$ of features Φ and **strict partial order** $\prec$ over Φ-tuples that is **well founded** and goal valuations are **$\prec$-minimal**

Subproblem $P[s, \prec]$: problem of finding state s' reachable from s such that s' is goal or $f(s') \prec f(s)$ (i.e., state s' "improves" s)

Decomposition of P into **collection of subproblems** $P[\prec]$:

- $P[s_0, \prec]$ in $P[\prec]$ for initial state s_0
- $P[s', \prec]$ in $P[\prec]$ if $P[s, \prec] \in P[\prec]$, $f(s') \prec f(s)$, and no such s'' or goal is closer from s than s'

Serialized width of problem P: $w_\Phi(P) \leq k$ if $w(P') \leq k$ for all P' in $P[\prec]$. Likewise, $w_\Phi(\mathcal{Q}) \leq k$ if $w_\Phi(P) \leq k$ for all P in $\mathcal{Q}$

Theorem
If $w_\Phi(\mathcal{Q}) \leq k$, algorithm SIW_Φ (Serialized IW$_\Phi$) solves any P in $\mathcal{Q}$ in **polynomial time** (exponential in k and $|\Phi|$)

SIW_Φ: Improves state iteratively with IW$_\Phi$ until finding plan

Where do serializations come from?

Serializations from General Policies

Policy graph for π_Φ: nodes for each Boolean feature valuation b, edges $b \to b'$ labeled with E iff (b, b') compatible with rule $C \mapsto E$

Policy π_Φ is **terminating** if for any cycle $b_1, \ldots, b_m$ in graph, there is a numerical feature n decremented but not incremented in cycle

Theorem
A **terminating policy** π_Φ with features that distinguish goals determines a serialization $(\Phi, \prec)$ where $\prec$ is transitive closure of pairs (f', f) such that (f, f') is compatible with a policy rule.

Sound + **goal-connected** policy yields serialization of width 0

What about partially specified (incomplete) policies?

Policy Sketches: Language for Serializations

Partially specified policy is **policy sketch:** set of sketch rules $C \mapsto E$

Sketch rules have **same syntax** but **different semantics:**

- Policy rules **filter transitions** (s, s'): $(f(s), f(s'))$ compatible with some rule
- Sketch rules **define subproblems:** reach s' from s such that $(f(s), f(s'))$ is compatible with some rule

Terminating sketches specify **serializations:** $\prec$ is transitive closure of pairs (f', f) such that (f, f') is compatible with some sketch rule

- Width of induced serialization bounded by **width of sketch**
- Serializations of bounded width solved by SIW in **polytime**

Sketches ("Incomplete Policies") in Delivery

Features $\Phi = \{H, p, t, n\}$: holding, distance to nearest package, distance to target, number of undelivered packages

Sketch ("incomplete policy")	$w_\Phi(\mathcal{Q}_{D_1})$	$w_\Phi(\mathcal{Q}_D)$
$\sigma_0 =$ empty	2	*unb*
$\sigma_1 = \{\{H\} \mapsto \{\neg H, p?, t?\}\}$	2	*unb*
$\sigma_2 = \{\{\neg H\} \mapsto \{H, p?, t?\}\}$	1	*unb*
$\sigma_3 = \sigma_1 \cup \sigma_2$	—	—
$\sigma_4 = \{\{n>0\} \mapsto \{n\downarrow, H?, p?, t?\}\}$	2	2
$\sigma_5 = \sigma_2 \cup \sigma_4$	1	1
$\sigma_6 = \{\{\neg H, p>0\} \mapsto \{p\downarrow, t?\}\}$	2	*unb*
$\sigma_7 = \{\{H, t>0\} \mapsto \{t\downarrow, p?\}\}$	2	*unb*
$\sigma_8 = \sigma_2 \cup \sigma_4 \cup \sigma_6 \cup \sigma_7$	0	0

Wrap Up, Conclusions, and Future Work

- Why so many domains with bounded width?
  - General policies underlie notion of width
  - Bounded number of features $|\Phi| \implies$ bounded width
  - Good features for IW are those used in general policies
- What about problems with unbounded width?
  - Broad notion of serialization, serialized width, and SIW algorithm
  - General policies and serialized width
  - **Sketches:** a rich language for serializing problems
- Future work:
  - Control knowledge by hand: Sketches vs. HTNs?
  - Learn features and sketches from traces