

Backbones for h^1 Landmarks

Elagkian Rajendram <elagkian.rajendram@unibas.ch>

Universität Basel

1. September 2026

SAT und CNF

$$\varphi = (x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_2 \vee x_3)$$

- › Eine Formel in konjunktiver Normalform
- › Gesucht sind **Modelle**, also Belegungen, die alle Klauseln erfüllen

x_1	x_2	x_3
T	T	T

Diese Belegung erfüllt jede Klausel

SAT und CNF

$$\varphi = (x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_2 \vee x_3)$$

- Eine Formel in konjunktiver Normalform
- Gesucht sind **Modelle**, also Belegungen, die alle Klauseln erfüllen

x_1	x_2	x_3
T	T	T
T	F	T
T	F	F

SAT und CNF

$$\varphi = (x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_2 \vee x_3)$$

- › Eine Formel in konjunktiver Normalform
- › Gesucht sind **Modelle**, also Belegungen, die alle Klauseln erfüllen

x_1	x_2	x_3
T	T	T
T	F	T
T	F	F

x_1 ist in jedem Modell wahr

Backbone

Definition

Der **Backbone** einer erfüllbaren Formel φ ist die Menge der Literale, die in jedem Modell von φ wahr sind.

$$\varphi = (x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_2 \vee x_3) \quad \Longrightarrow \quad B = \{x_1\}$$

- › Test für ein einzelnes Literal: ist $\varphi \wedge \neg \ell$ erfüllbar?
- › Unerfüllbar heisst, ℓ lässt sich nie falsch setzen, also ist ℓ im Backbone
- › Naiv braucht das **einen Solveraufruf pro Literal**

CadiBack

- › Extrahiert den **vollständigen** Backbone mit deutlich weniger Solveraufrufen
- › Baut auf dem SAT Solver **CaDiCaL** auf

Model Rotation

Jedes zurückgegebene Modell schliesst alle Literale aus, die es flippt.

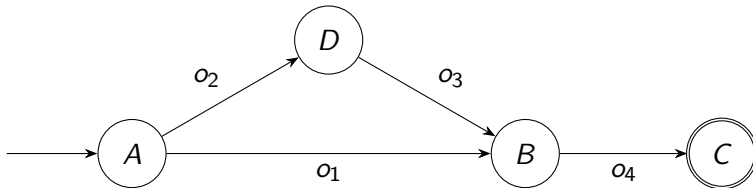
Chunking

Eine Klausel $\rho = (\neg \ell_1 \vee \dots \vee \neg \ell_k)$ testet k Kandidaten gemeinsam. Unerfüllbar bestätigt alle k auf einmal.

Auf diesem Algorithmus baut die Arbeit auf

Planning

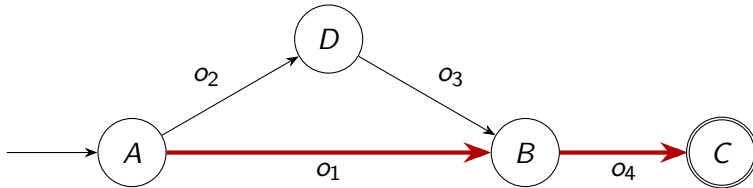
$\Pi = \langle V, O, s_0, s_\star \rangle$ mit Variablen V , Operatoren O , Initialzustand s_0 und Ziel $s_\star$



- $s_0 = \{\langle pos, A \rangle\}$ und $s_\star = \{\langle pos, C \rangle\}$
- Ein Plan ist eine Folge von Operatoren, die von s_0 zum Ziel führt

Planning

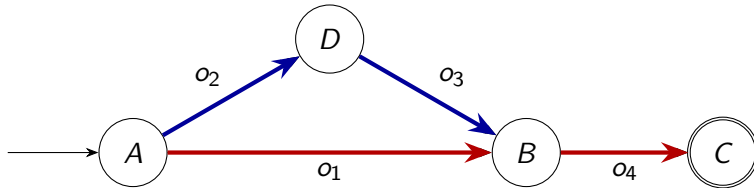
$\Pi = \langle V, O, s_0, s_\star \rangle$ mit Variablen V , Operatoren O , Initialzustand s_0 und Ziel $s_\star$



- > $s_0 = \{\langle pos, A \rangle\}$ und $s_\star = \{\langle pos, C \rangle\}$
- > Ein Plan ist eine Folge von Operatoren, die von s_0 zum Ziel führt
- > Ein möglicher Plan: $\langle o_1, o_4 \rangle$

Planning

$\Pi = \langle V, O, s_0, s_\star \rangle$ mit Variablen V , Operatoren O , Initialzustand s_0 und Ziel $s_\star$



- › Ein Plan ist eine Folge von Operatoren, die von s_0 zum Ziel führt
- › Ein möglicher Plan: $\langle o_1, o_4 \rangle$
- › Ein weiterer: $\langle o_2, o_3, o_4 \rangle$

Landmark

Definition

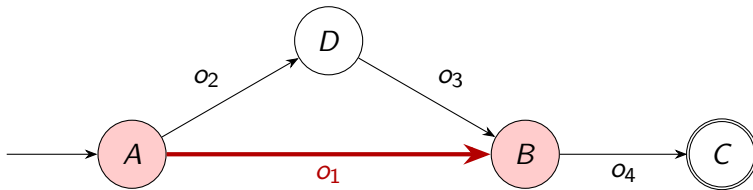
Ein Fakt f ist ein **Landmark** der Aufgabe Π , wenn jeder Plan von Π durch einen Zustand führt, in dem f gilt.

- $\langle pos, B \rangle$ ist ein Landmark
- $\langle pos, A \rangle$ und $\langle pos, C \rangle$ ebenfalls, aber **trivial**: jeder Plan startet im Initialzustand und endet im Ziel

Landmarks werden für Heuristiken und Suchsteuerung genutzt

Delete Relaxation

Wir ignorieren die Delete Effekte der Operatoren



- › Nach o_1 galt vorher nur noch $\langle pos, B \rangle$
- › Jetzt gelten $\langle pos, A \rangle$ und $\langle pos, B \rangle$ **gleichzeitig**
- › Fakten sammeln sich nur an, keiner wird je wieder falsch

h^1 Landmarks

Korrekt

Jeder Plan von Π ist auch ein Plan von Π^+ . Also ist jedes gefundene Landmark wirklich eines.

Unvollständig

Landmarks, die erst durch das Löschen entstehen, werden übersehen.

Wir bekommen eine Teilmenge der Landmarks des Problems

Parallelele

SAT

- › Formel φ
 - › Modelle
 - › Literale, die in **jedem** Modell wahr sind
- ⇒ **Backbone**

Planning

- › Aufgabe Π
 - › Pläne
 - › Fakten, die auf **jedem** Plan gelten
- ⇒ **Landmarks**

Kern der Arbeit

Die Ideen von CadiBack ins Planning übertragen und prüfen, ob sich Landmarks damit schneller finden lassen.

Algorithmus CadiLand

1. Eine relaxierte Exploration gibt den Kandidatenpool $\Lambda = \text{facts}(\pi) \setminus (s_0 \cup s_*)$
2. Chunk Γ der Grösse k wählen und $\text{SOLVABLE}(\Pi^+, \Gamma)$ aufrufen
3. Erreichbar (SAT): $\Lambda \leftarrow \Lambda \cap \text{facts}(\pi')$
4. Unerreichbar (UNSAT) und $|\Gamma| = 1$: bestätigtes Landmark
5. Unerreichbar (UNSAT) und grösser: die Mitglieder einzeln nachtesten

Chunking

Kandidaten nach der ersten Exploration, Chunkgrösse $k = 3$

$$\Lambda = \{ A, \mathbf{B}, C, D, E, F \}$$

Beispiel: der teure Fall

Erster Chunk $\Gamma = \{A, B, C\}$ gemeinsam sperren

Test	Ergebnis	Erkenntnis
$\{A, B, C\}$	UNSAT	mindestens ein Landmark, aber welches?
$\{A\}$	SAT	A fällt weg
$\{B\}$	UNSAT	B ist ein Landmark
$\{C\}$	SAT	C fällt weg

4 Tests statt 3

Beispiel: der günstige Fall

Zweiter Chunk $\Gamma = \{D, E, F\}$ gemeinsam sperren

Test	Ergebnis	Erkenntnis
$\{D, E, F\}$	SAT	kein Landmark darunter, alle drei fallen weg

1 Test statt 3

Landmarkendichte

- › Sei p die **Landmarkendichte**, also der Anteil der Kandidaten, die Landmarks sind
- › Dorfman's gepoolte Bluttests (1943)

Dorfman 1943

Prevalence Rate (per cent)	Optimum Group Size	Relative Testing Cost	Percent Saving Attainable
1	11	20	80
10	4	59	41
12	4	65	35
13	3	67	33
15	3	72	28
20	3	82	18
25	3	91	9
30	3	99	1

Die Modi

Modus	Chunkgrösse	Wachstum
naiv	1	
Model Rotation	1	nutzt jeden Plan wieder
chunk-3, chunk-5	fest	konstant
adaptive-k2	wächst *2	nach UNSAT
adaptive-k2_inv	wächst *2	nach SAT
density-adaptive	$\text{round}(1/\sqrt{\hat{p}})$	aus der laufenden Schätzung

$$> \hat{p} = \frac{\ell + 1}{r + 2}$$

- > **Alle Modi liefern dieselben Landmarks.** Nur die Anzahl Tests unterscheidet sich.

Aufbau der Experimente

- DEFAULT_OPTIMAL_SUITE, **1827 Aufgaben**, verwaltet mit Downward Lab
- sciCORE, 30 min Laufzeit, 3872 MiB pro Lauf
- Metrik: **relaxierte Explorationen geteilt durch Kandidaten**, naiv ist genau 1

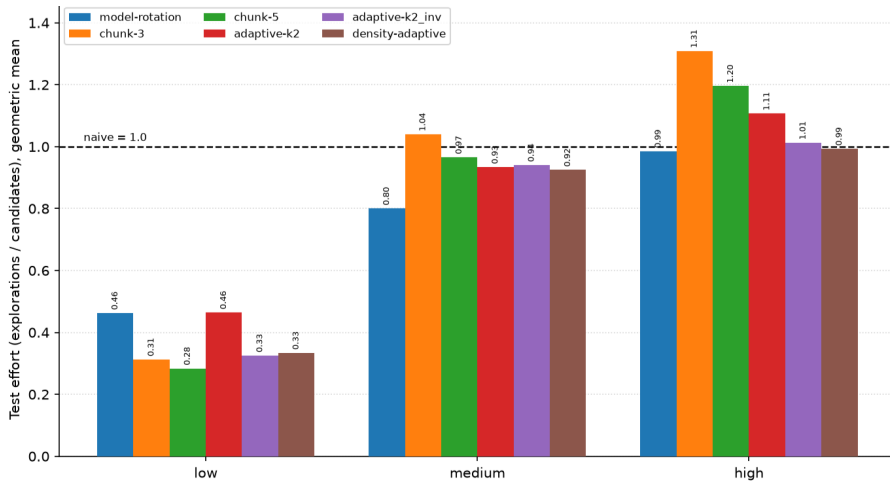
Prototyp gegen native Implementierung

Nativ löst alle 1827 Aufgaben. Der Prototyp schafft höchstens 255, also etwa jede siebte, und liefert auf den gemeinsamen Aufgaben dieselben Landmarks.

Vergleich Dichtekategorie

	tief $[0, 0.3)$ 589 Aufgaben	mittel $[0.3, 0.69)$ 551 Aufgaben	hoch $[0.69, 1]$ 552 Aufgaben
naive	1.00	1.00	1.00
model-rotation	0.46	0.80	0.99
chunk-3	0.31	1.04	1.31
chunk-5	0.28	0.97	1.20
adaptive-k2	0.46	0.93	1.11
adaptive-k2_inv	0.33	0.94	1.01
density-adaptive	0.33	0.92	0.99

Vergleich Dichtekategorie

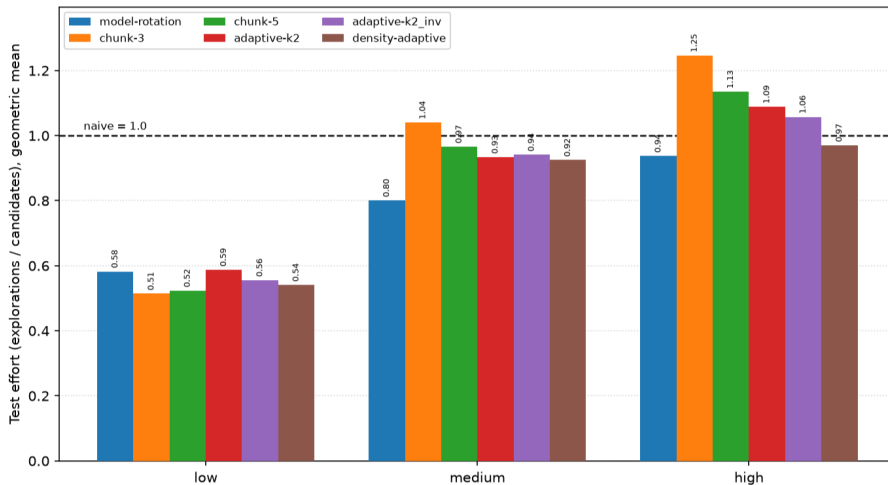


innere Aufgaben

	tief (0, 0.3) 291 Aufgaben	mittel [0.3, 0.69) 551 Aufgaben	hoch [0.69, 1) 122 Aufgaben
naive	1.00	1.00	1.00
model-rotation	0.58	0.80	0.94
chunk-3	0.51	1.04	1.25
chunk-5	0.52	0.97	1.13
adaptive-k2	0.59	0.93	1.09
adaptive-k2_inv	0.56	0.94	1.06
density-adaptive	0.54	0.92	0.97

- › 298 Aufgaben haben Dichte 0, 430 haben Dichte 1, zusammen 43 Prozent
- › Auf diesen ist der beste Modus ohne Messung bekannt

Vergleich Dichtekategorie



Fazit

- **Model Rotation überträgt sich.** Eine zurückgegebene Lösung wiederzuverwenden kostet nichts und zahlt sich auf jeder Aufgabe aus. Im Mittel am besten und nie schlechter als naiv.
- **Chunking überträgt sich nicht.** Lohnt sich nur bei tiefer Landmarkendichte.
- **Density adaptive** kann sich lohnen bei unbekannter Landmarkendichte.

Ausblick

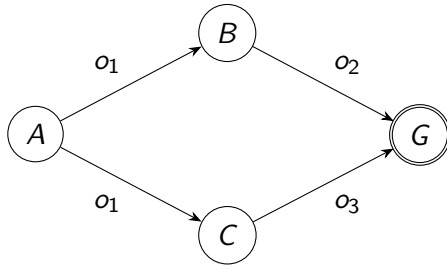
- Wie stark sind die Landmarks innerhalb einer Aufgabe **geclustert**? Davon hängt ab, ob ein Chunk auf einen landmarkfreien Abschnitt fallen kann.
- Ein **besserer Dichteschätzer**, der auf den frühen Kandidaten schneller konvergiert
- Sind die extrahierten Landmarks für einen Planer **nützlich**, und zahlt sich der Aufwand dort aus?

Backup: Vergleich der Modi

Modus	Verhältnis
model-rotation	0.78
adaptive-k2_inv	0.80
density-adaptive	0.81
adaptive-k2	0.83
chunk-5	0.88
chunk-3	0.97
naive	1.00

➤ Die Gesamtdichte liegt bei
 $57854/96352 \approx 0.60$

Backup: Operator mit zwei Effekten



- › B ist ein Landmark, da jeder Plan mit o_1 beginnt
- › Sperrt man nur den unären Operator $A \rightarrow B$, bleibt $A \rightarrow C$ bestehen und das Ziel bleibt erreichbar
- › Sperrt man o_1 als Ganzes, fallen beide Effekte weg und B wird bestätigt

Backup: Die Kodierung

	E1	E2	E3	E4
Fakten insgesamt	505215	603891	676662	807029
Kandidaten	77652	78289	96099	96352
Cadi Landmarks	48796	44011	62915	57854

- E2: –keep-unreachable-facts **senkt** die Anzahl: zusätzliche Erzeuger eröffnen alternative Wege
- E3: –keep-unimportant-variables **erhöht** sie: die neuen Fakten liegen auf dem Plan und sind oft selbst Landmarks