

これまでの研究業績と今後の抱負

浅井 政太郎 東京大学 総合文化研究科 学振DC2

発表要旨

背景

Fully Automated Cyclic Planning for Large-Scale Manufacturing Domains 1. **ICAPS14.**

Solving Large-Scale Planning Problems by Decomposition and Macro Generation 1. **ICAPS15.**

Tiebreaking Strategies for A* Search: How to Explore the Final Frontier 1. **AAAI16.** (*JSAI*
学生奨励賞)

Tie-Breaking Strategies for Cost-Optimal Best First Search 1. **JAIR 58 (2017): 67-121.**

Exploration Among and Within Plateaus in Greedy Best-First Search 1. **ICAPS17.**

Efficient Optimal Search under Expensive Edge Cost Computation 2. **IJCAI17.**

Classical Planning in Deep Latent Space: From Unlabeled Images to PDDL (and back) 1.
KEPS17.

今後の研究計画

1 Masataro Asai, Alex Fukunaga

2 Masataro Asai, Akihiro Kishimoto, Adi Botea, Radu Marinescu, Elizabeth Daly M, and Spyros Kotoulas

1 背景 - 自律機械とは?



1.1 誰?



1.1.1 誰?

鉄腕アトム



1.1.2 誰?

鉄腕アトム



見て,聞いて,喋って,行動する
自律行動 *Autonomous*



1.1.3 誰?

鉄腕アトム



見て,聞いて,喋って,行動する
自律行動 *Autonomous*

T-52 援竜 @北九州消防局
レスキュー・ロボット



1.1.4 誰?

鉄腕アトム



見て、聞いて、喋って、行動する
自律行動 Autonomous

T-52 援竜 @北九州消防局
レスキュー・ロボット



自律行動 Autonomous
?

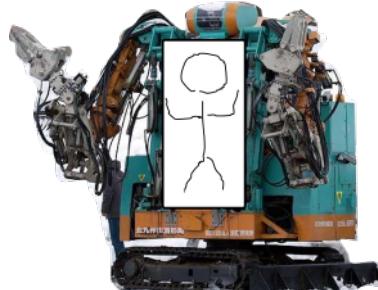
1.1.5 誰?

鉄腕アトム



見て,聞いて,喋って,行動する
自律行動 Autonomous

T-52 援竜 @北九州消防局
レスキュー・ロボット



自律行動 X Autonomous
人間による操作
でかいラジコンショベルカー

1.2 なぜ自律機械? 社会的意義 — 例えば、操縦士が足りない!



- そのままでは役に立たない!

1.2.1 「人」は簡単に増やせない – Human Resource and Training



時間 かかる 訓練に > 100 時間, 必要な時だけ増やす の
は不可能

¥ ¥ ¥ ¥ かかる 訓練官、訓練場所、訓練用具
技術は 維持が重要 定期的な再訓練、長期的コスト
平時は 無駄な技術 普段は意味がない – 無駄な費用!

1.3 目的達成型知能のための自動プランナ（モータ制御）

プランナ=抽象行動列を求める人工知能プログラム



初期状態: 人が生き埋め!
ゴール: 患者の搬送



出力: 被災者を助ける
アクション列 = プラン
出血をとめる
→病院に電話
→患者を助け出す
→救急隊に引き渡し

1.4 目的達成型知能のための自動プランナ（モータ制御）

プランナ=抽象行動列を求める人工知能プログラム



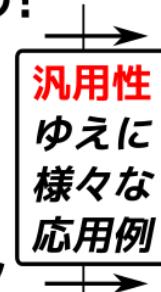
初期状態: 人が生き埋め!

ゴール: 患者の搬送



出力: 被災者を助ける
アクション列 = プラン

出血をとめる
→病院に電話
→患者を助け出す
→救急隊に引き渡し



一階述語論理で入力

初期状態 (STRIPS)

出血(浅井)
on(木材,浅井), on(岩, 木材)

アクション

電話(病院)
救出(浅井)

ゴール

at(浅井, 病院)

引き渡し(浅井, 救急車)

1.5 目的達成型知能のための自動プランナ（モータ制御）

プランナ=抽象行動列を求める人工知能プログラム



初期状態: 人が生き埋め!

ゴール: 患者の搬送



出力: 被災者を助ける

アクション列 = プラン

出血をとめる

→病院に電話

→患者を助け出す

→救急隊に引き渡し

汎用性
ゆえに
様々な
応用例

プランニングの 応用実績

衛星/探査機の自動運行

通信に光速でも20分かかる

→自律対処が必要

Deep Space 1
(Muscettola et al. '98)



Mars Exploration Rover
(Bresina et al. '05)

(企業名)

CORE SECURITY[®]
Thinking Ahead.

(クラッキング)

ネットワークの自動ハッキング
→脆弱性を報告

問題を論理で記述→
目的達成の手段を
自動で計算可能

一階述語論理で入力

初期状態 (STRIPS)

出血(浅井)

on(木材,浅井), on(岩, 木材)

ゴール

at(浅井, 病院)

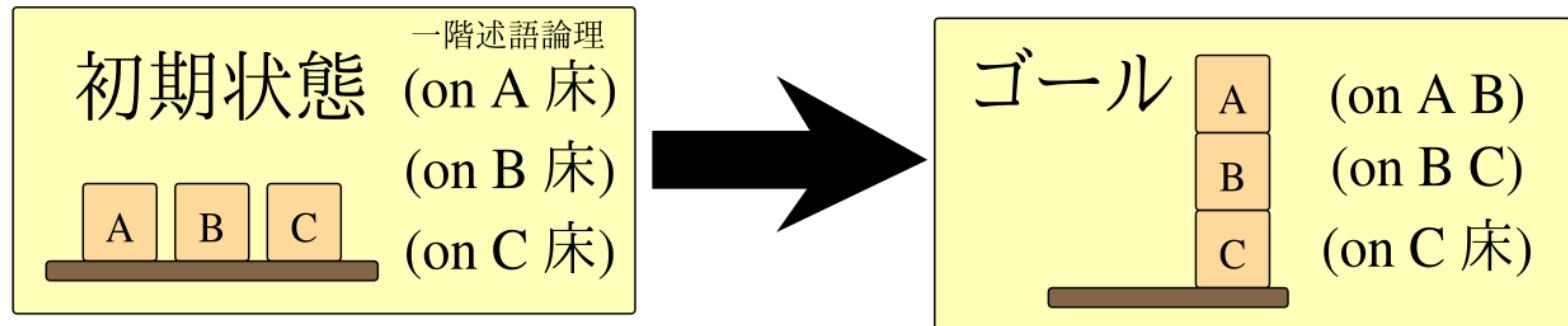
アクション

電話(病院)

救出(浅井)

引き渡し(浅井, 救急車)

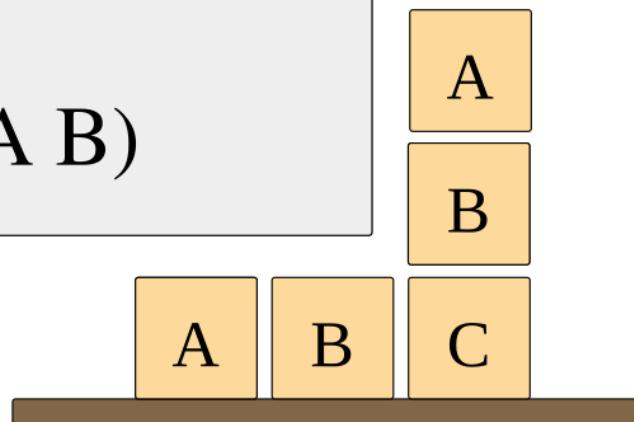
1.6 古典プランニング問題(決定的,完全情報) – Blocksworld



ゴールに至る方法: プランを計算

アクション1
(move B C)

アクション2
(move A B)



非古典的なさまざまな拡張

(並列アクション,POMDP,HTN… どのAIの教科書にものっている)

1.6.1 アクション = 条件付き状態遷移

アクション (move ?X ?Y)

?X, ?Y : 変数。 値 BLOCK-A, BLOCK-B などを適用して使う

条件と効果で構成される

条件：実行に必要な条件を表す命題

(clear ?X) : 積み木 ?X の上が空

(clear ?Y) : 積み木 ?Y の上に空

モデリング言語 PDDL で

効果：前後の状態の 差分 を表す命題

記述

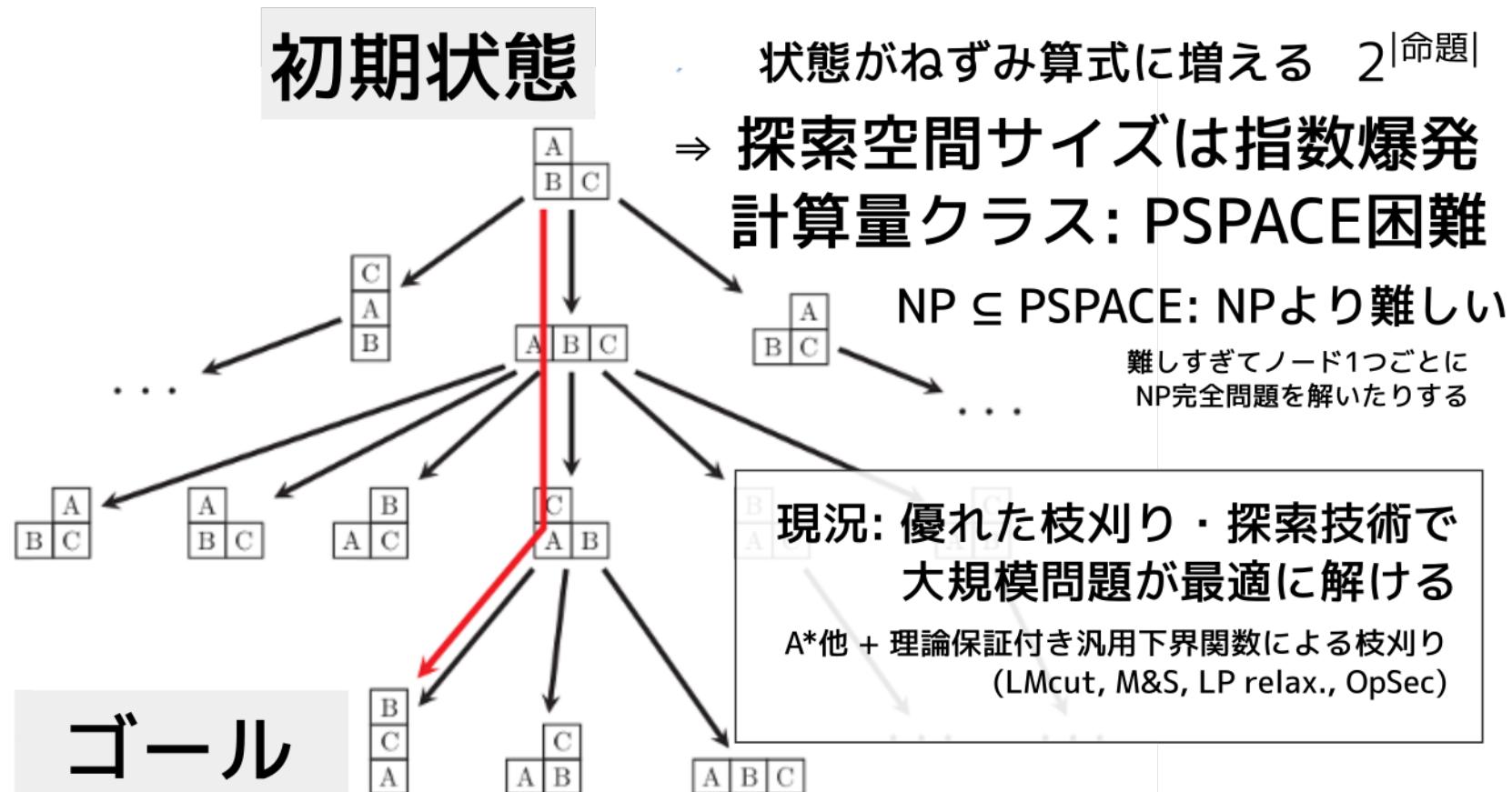
(on ?X ?Y) を 追加 : ?Y の上は ?X

(clear ?Y) を 削除

1.6.2 プランニング = グラフ探索

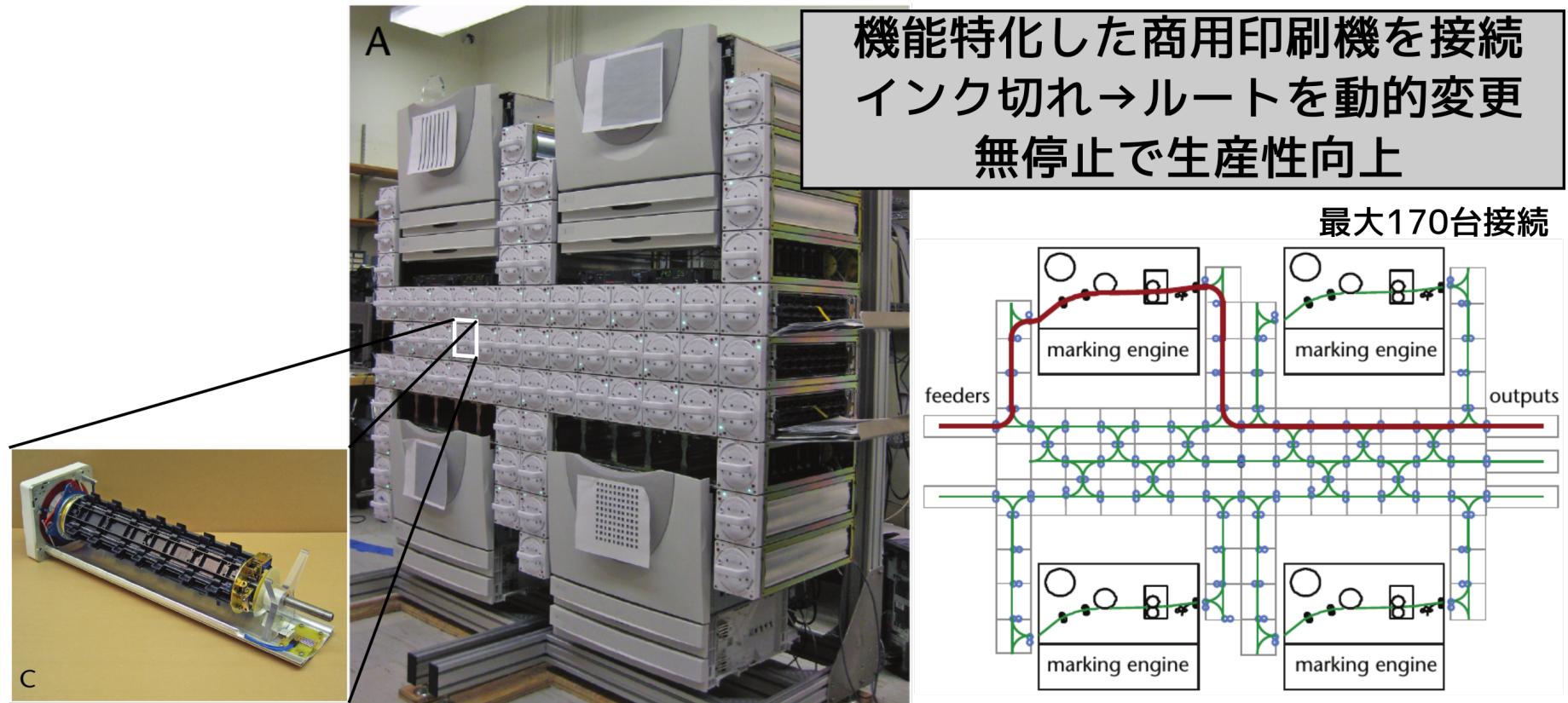
ノード：状態 = 命題の集合 $(\text{on } A \ B), (\text{clear } A)$ など

辺：アクション $(\text{move } A \ B)$ 等

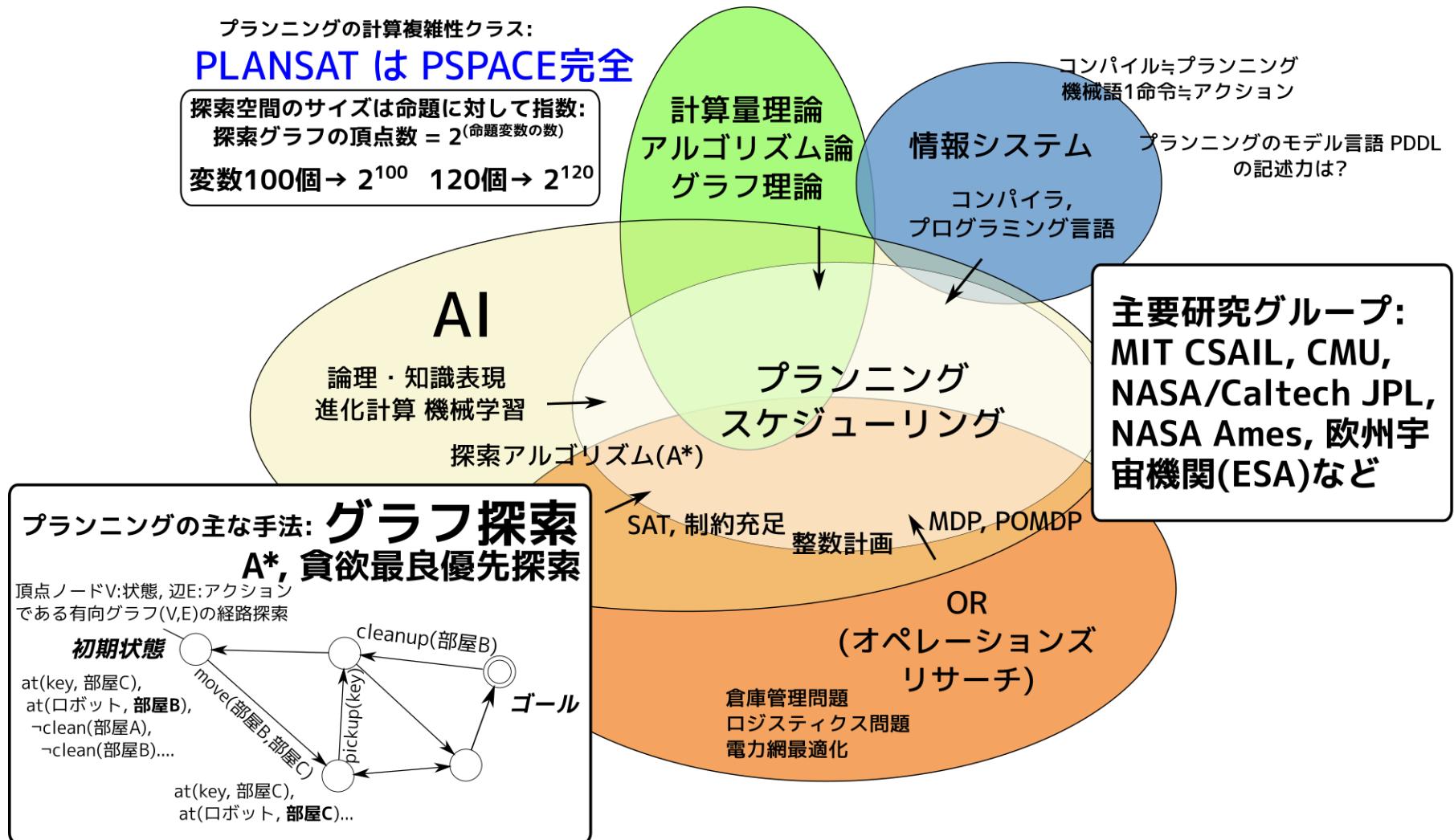


*1 [Helmert, 2006] [Richter, 2010]

1.7 Xerox PARC Printer: *Hypermoduler* Systems



1.8 プランニング(自動行動計画)分野の位置づけ



2 査読論文実績 (抜粋)

AAAI, IJCAI: AI一般の 1st-tier, **ICAPS:** プランニング専門の 1st-tier, **JAIR:** トップジャーナル

ICAPS14 (採択率 33%) (**IHI** 共同研究) Fully Automated Cyclic Planning for ...

ICAPS15 (採択率 33%) Solving Large-Scale Planning Problems by Decomposition ...

AAAI16 (採択率 26%) (**JSAI** 学生奨励賞) Tiebreaking Strategies for A* Search: ...

JAIR17 (採択率 12%) Tie-Breaking Strategies for Cost-Optimal Best First Search.

ICAPS17 (採択率 33%) Exploration Among and Within Plateaus in Greedy Best-First Search.

IJCAI17 (採択率 25%) (**IBM Research** 応用研究) Efficient Optimal Search under ...

KEPS17 (採択率 60%) Classical Planning in Deep Latent Space: From Unlabeled Images ...

応用研究、基礎研究 共に経験あり

2.1 業績1: 査読付き国際学会 ICAPS14 (採択率33%) (IHI共同研究)

セル生産方式: 工場生産方式の一種 (対義語: ライン生産)

ロボットアームなどを含む小規模セルが複数協調して製造

目的: 並列の製造プランを自動計画 + 製作時間を最小化

一台の組み立て問題を解くのはある程度複雑でも可能

研究の対象

✗ アクチュエータの角度制御 (例: $\theta: 0\text{deg} \rightarrow 30\text{deg}$, 1deg/sec)

◎ 行為の計画 (例: 部品2をアーム1で塗装, ネジ3を締結 etc.)

問題点: 扱う対象の数が多くなる場合 (製品が4個程度以上の時)

最適解を返すソルバで直接解く → ✗ 探索空間が広大で探索不可

1つ分の製造手順を複数つなげて後処理最適化 → ✗ 生産時間が長い
(makespan)

手法: 問題の繰り返し構造を「工場, セル生産, 製品」など前提なしに

任意のプランニング問題の論理構造から自動検出

検出した繰り返し構造から効率よい (生産時間の短い) プランを生成

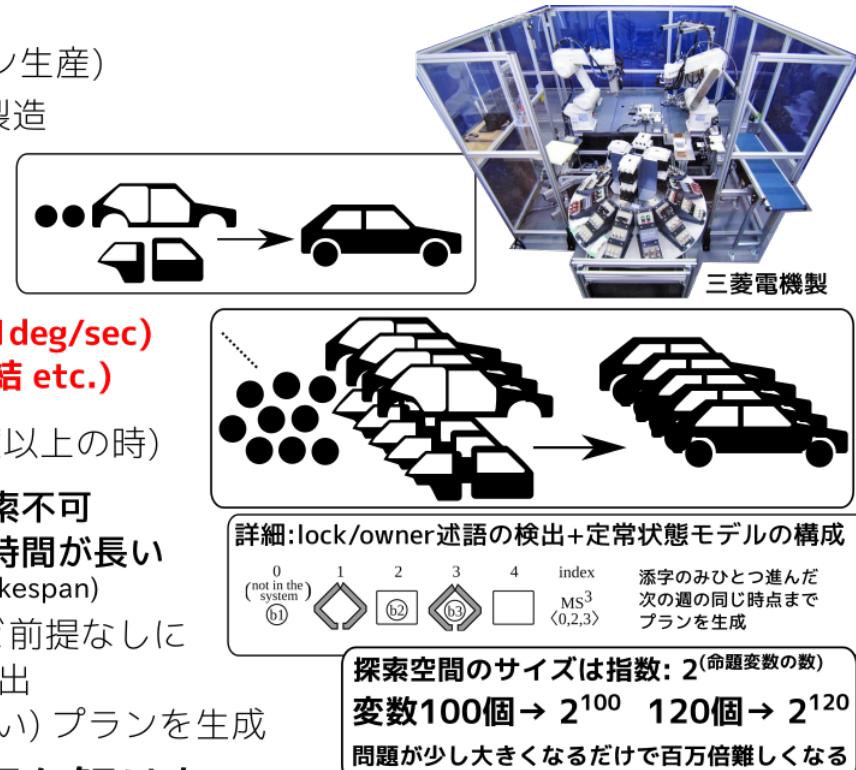
結果: 1000台以上の製品を組み立てる問題を解けた

+ 生産時間を最大3割短縮(後処理最適化と比較)

新規性: 繰り返し構造を自動で抽出するというアイディア+抽出技術

インパクト: 大きな高速化, プランニングの適用規模を広げた

株式会社IHIと共同研究(システムの先行評価として)



→ しかも探索空間はその指数で増加
解ける探索空間のサイズ:

以前: $10^6 \rightarrow$ 提案手法: 10^{274}

2.2 業績2: 査読付き国際学会 ICAPS15 (採択率33%)

スキップ予定

プランニング分野一般の目標:

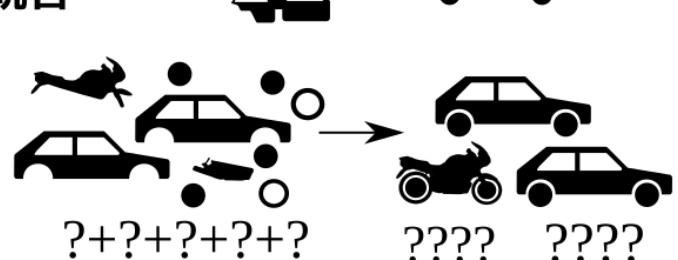
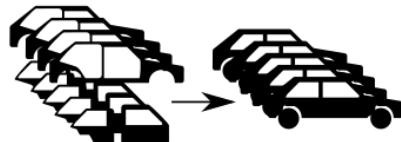
汎用性を失わずに より複雑な問題を解く! → 高度なAI

ICAPS14を一般化すればはるかに困難な問題が解けるハズ

探索空間のサイズは指數: $2^{(\text{命題変数の数})}$

変数100個 → 2^{100} 120個 → 2^{120}

問題が少しだけ大きくなるだけで百万倍難しくなる



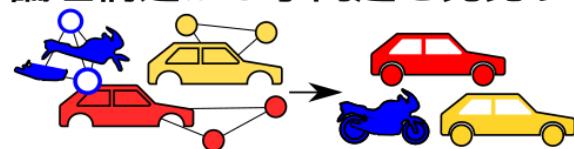
ICAPS14: 大規模な問題を同一種類の小問題に分割+処理+統合

目的: **大規模な問題を複数種類の小問題に分割+処理+統合**

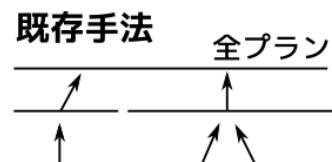
課題: 1つの小問題の構成要素が不明: 自動で発見する能力が必要

分割/統合が本質的に不可能な問題でも動作する汎用性が必要
(問題分割を行うその他の既存研究は、その場合動作しない/想定していない)

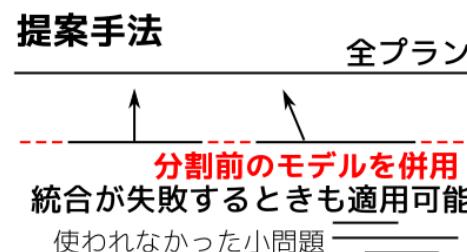
手法: **問題の論理構造から小問題を発見する新手法**



分割なしでは解けない問題
(探索失敗)



より柔軟な統合手法



結果: **広い問題種別で**

3-4倍変数の多い問題を解けた
解ける探索空間のサイズ:

以前: $\sim 10^7$ 提案手法: $\sim 10^{28}$

新規性: 小問題の発見手法 / 柔軟な統合手法

インパクト: 高速化, 小問題分割の汎用性を実証

2.3 業績3: 査読付き国際学会 AAAI16 (採択率 26%)

グラフ探索一般に適用できる高速化手法の開発

グラフ探索: スタートからゴールまでの経路を探す
+ 通る辺の重みの和を最小化する

身近な例: カーナビ 但し、本研究の対象はメモリには収まらない規模のグラフを扱う
例: アミノ酸の多重配列問題(MSA)など
プランニング問題もグラフ探索で解く

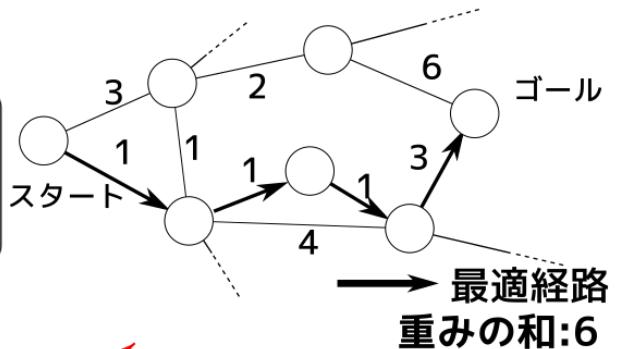
著名なグラフ探索アルゴリズム:
A*(1968) Dijkstra法(1959)
=動的計画+分枝限定

探索の一部の実問題には**重み0の辺が大量にある** (指数的に多い)

例: トランクの配達計画(燃料を最小化) 辺=行為, 重み=燃料
→荷物の積み替えには燃料を使わない

問題点: **重みの和が同じノードが大量に存在**

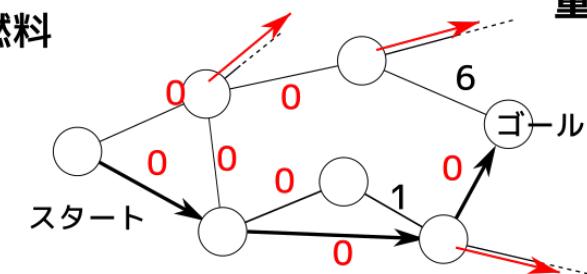
探索を進める方向を見失う → **タイブレーク**の影響大



オリジナリティ: A*(見積もり値を用いた最良優先探索(最良分枝限定法))にて

1. タイブレークに関する**70年代からの定説を覆した**
2. タイブレークの新手法を提案し性能向上

(A*のTiebreakingには
ゴールまでの距離の見積もり値 h を
流用すべきだとされてきた)



結果: 解ける問題規模が拡大 (ベンチマーク1104問:814問 → 867問 (+53問))

解ける問題の探索空間サイズ(例):
(企業ネットワーク脆弱性の診断問題にて) 以前: 10^6 提案手法: 10^{88}

今後のインパクト: 重み0の辺を持つ広範なグラフ探索問題に対して性能向上

AAAI16の一般化: タイブレーク≡非最適探索

問題点: 重みの和が同じノードが大量に存在 (ゼロコスト問題)

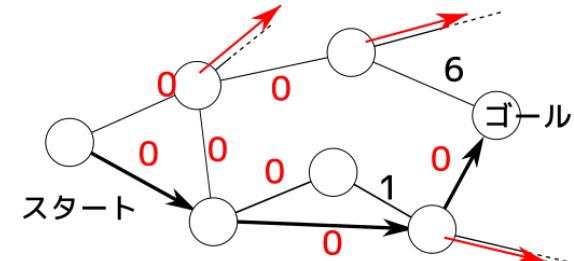
探索を進める方向を見失う → タイブレークの影響大

ノードの下界関数 h の種類

許容的: 常に真のコストより低く見積もる → 最適解

非許容的: 非最適解を見つけてしまう ← 従来は使われない

Unit cost 関数: 辺コストを全て1で代用, 非許容的関数の例



主な発見: タイブレーク とは 許容的関数 h で作ったプラトーの中で非最適探索を行うことである

新規性: 最適探索で非許容的関数を使う方法を初めて示した

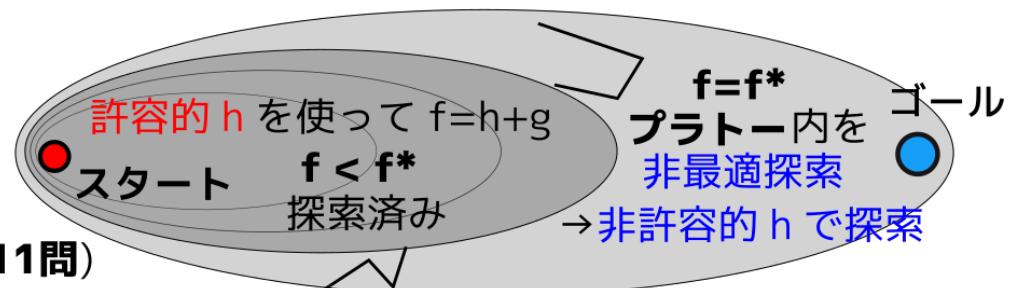
性能評価:

ゼロコスト問題のタイブレークに

非許容的 Unit cost 関数を使う

→ 指数的に高速化 (844問 → 906問)

→ AAAI16と同時使用でさらに高速 (→ 911問)



理論的成果:

無限グラフでのA*の完全性を

様々なタイブレーク法ごとに証明

→ 非最適探索の理論的結果を援用

非最適探索用の全技術を最適探索に援用可能

(従来は非互換と考えられてきた)

→ 今後のさらなる発展が期待できる

非最適探索の均一化手法の理解と改善

非最適探索: GBFS (**貪欲最良** 優先探索) + **非許容的** 下界関数 h

下界関数 h を常に信頼して**(貪欲)**探索を集中させる(exploitation)
 →**非許容的** h は常に正しいとは限らない → h を時折無視して探索を**均一化**→性能向上

問題点: アドホック手法が多い

→どう組み合わせればよいか解らない

目的: →クリーンに評価しなおす

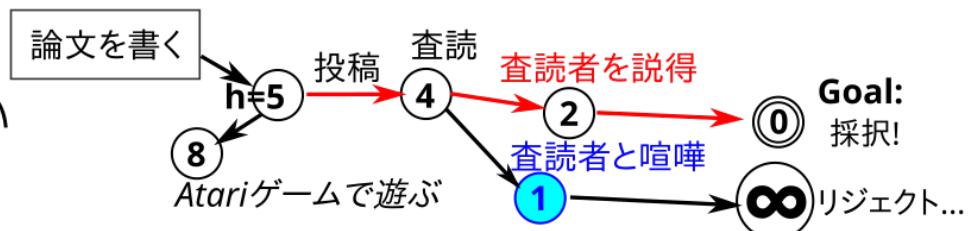
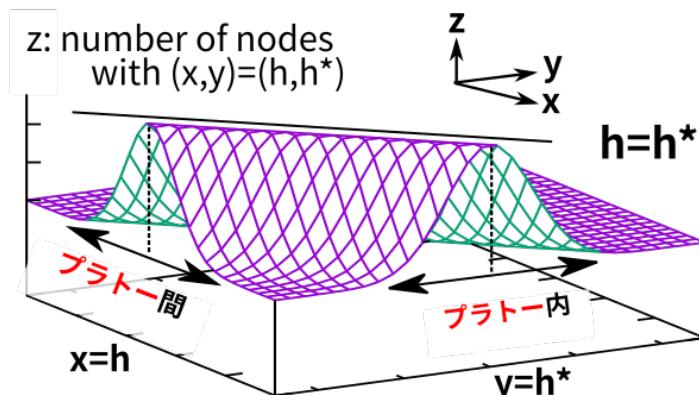
発見1. 2つの直行した均一化方法がある。

理由: 真の値 h^* からの h の誤差は二次元的

→ **プラトー間**均一化と **プラトー内**均一化

→同じ均一化手法を二通りに利用出来る

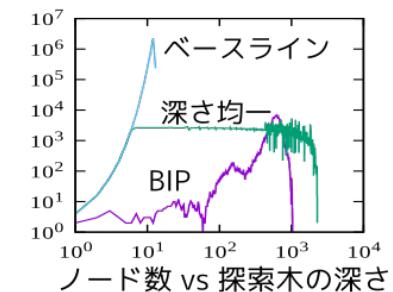
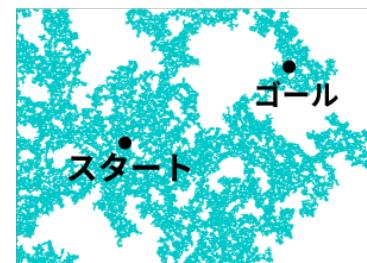
→直行だから同時利用してOK! → 性能向上



発見2. ボンドパーコレーション(BIP)

フラクタルを使った均一化手法

→ 性能向上の理由: 探索木の幅を削減



制限時間内で解けた問題数: 192 → 237.7

参考: [Xie et. al. 2014] 深さ均一化 223.9

1. 未知の均一化アルゴリズムに汎用に適用できる法則
2. フラクタルを用いてより良い探索が出来る

2.6 業績6: 査読付き国際学会 IJCAI17 (採択率 25%) (IBM 共同研究)

辺コスト動的計算が必要なグラフ探索の高速化

(IBM Research Ireland での研究)

例1: 辺コスト(都市間距離)が未知のTSP

都市数Nに対し $O(N^2)$ 回経路探索ソルバを呼び出す

例2: MWRP (複数ワーカー乗り換え問題):

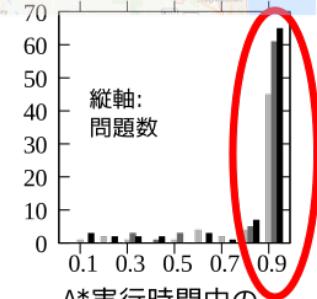
[地図上の全場所] $^2 \times$ [全出発時刻] 個の 二点間乗り換え

→ 全ての辺コストを求めるのは現実的でない

→ コストを求めた後の問題自体もNP困難

在宅訪問診療のモデル問題

医者が患者を予約時間に訪問する最適乗り換え計画



問題点: 一回の辺コストを求める計算(外部ソルバ)が重い

→ 全コストを求めるのが現実的でない

→ A*探索で必要な辺コスト c_a のみを動的に計算

→ それでもなお実行時間の90%を辺コスト計算に使用

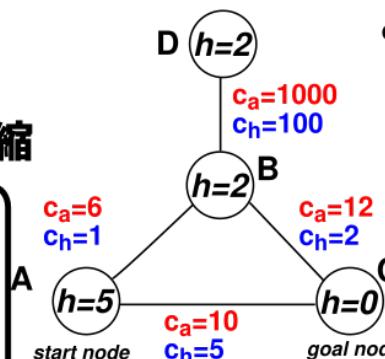
解決法: c_a の下界 c_h で代用 必要になるまで c_a 計算を遅延

c_h : 外部ソルバの下界関数值, c_a より高速に計算可能

同じノードを c_h/c_a で二回展開・キューに挿入

→多くの c_a を計算せずにすむ → 探索時間を最大5倍短縮

多くの実問題は階層構造を持つ
既存外部ソルバを用いて汎用に適用可能



e: expand	d: deleteMin	i: insert	DEA*
A*			D <small>E</small> A*
i A(5)	i A(5)	e B(8)	
d A(5)	d A(5)	i D(106)	
e A(5)	i A(5)	i C(8)	
i B(8)	d A(5)	d C(8)	
i C(10)	e A(5)	i C(18)	
d B(8)	i B(3)	d C(10)	
e B(8)	i C(5)	e C(10)	
i D(1006)	d B(3)	goal	goal
i C(18)	i B(8)		
d C(10)	d C(5)		
e C(10)	i C(10)		
goal	d B(8)		

③ 産業技術総合研究所で行いたい研究(短期目標)

Neural-Symbolic複合システムによる

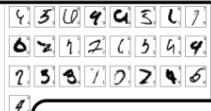
次世代AIシステム

の研究

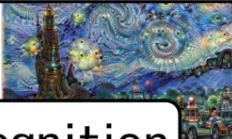
3.1 深層學習

Computer Vision

Classification



Style Extraction/Synthesis



Object Recognition



NLP/Time-Series Analysis

Captioning

The Remote Agent Experiment

Debugging code
from 250 million miles away

Ron Garrett
14 February 2012

Subtitles/CC Options
Off English (auto-generated)
Auto-translate

Machine Translation

Lisp, the Universe and Everything → Lisp、宇宙そしてすべて

Frame Prediction



Prediction: Handshake

3.2 Q. 深層学習と 記号的 AI はどう違う?

A. レイヤが違う

機械学習・**Neural Networks** == 関数近似
for 認識・反射

- 入力は **Subsymbolic** (連続値)
画像、音声、非構造化テキスト:
- 感覚的知能:
反応, 直後の行動の決定
パブロフの犬: 餌を認知 よだれ
自動運転: 赤信号, 人 止まる.
翻訳: 文章 文章
- 囲碁局面の評価関数: 局面 勝率
効率よく 1-to-1 mapping
単純作業

- AlphaGo = Subsymbolic (DLNNによる評価関数) + Symbolic (MCTSによる探索)

推論・探索
for プランニング・ゲーム・定理証明

- 入出力は **Symbolic**
論理 オブジェクト ルール
- 論理・推論による知能:
未来に渡る 戰略の決定
(戦略 = 行動の 列や木)
レスキューロボ: ゴール = 被災者生存
証明器: ゴール = QED
コンパイラ: 命令列の生成
- 囲碁, 将棋: ゴール = 勝利
順序制約+複雑な作業

3.3 記号的AIによる論理推論の重要性

f Conclusions

Y LeCun



深層学習の「創始者」
と呼ばれる一人

- Deep Learning is enabling a new wave of applications
 - ▶ Today: Image recognition, video understanding. vision now works
 - ▶ Today: Better speech recognition: speech recognition now works
 - ▶ Soon: Better language understanding, dialog, and translation
- Deep Learning and Convolutional Nets are being widely deployed
 - ▶ Today: Image understanding at Facebook, Google, Twitter, Microsoft.....
Facebooks computer vision projects medical image analysis robot perception
- We need hardware (and software) for embedded applications
 - ▶ For smart cameras, mobile devices, cars, robots, toys....
- But we are still far from building truly intelligent machines
 - ▶ We need to integrate reasoning with deep learning
 - ▶ We need a good architecture for “episodic” (short-term) memory.
 - ▶ We need to find good principles for unsupervised learning

3.4 業績7: 査読付きワークショップ KEPS (採択率 60%)

深層学習(非記号型AI)と行動計画(記号型AI)の融合

画像で示された問題を

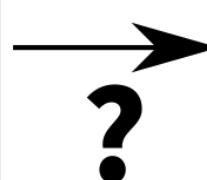
「パネル」「9マス」「動く」などの情報は一切与えられない問題を解いた例も与えられない



入力: 個別の状態遷移データ, 初期画像, ゴール画像



初期画像



ゴール画像



Original
Mandrill image

3.4.1 業績7: 査読付きワークショップ KEPS (採択率 60%)

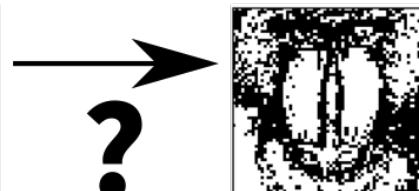
深層学習(非記号型AI)と行動計画(記号型AI)の融合

画像で示された問題を

「パネル」「9マス」「動く」などの情報は一切与えられない問題を解いた例も与えられない



初期画像



ゴール画像

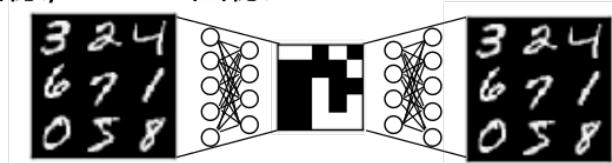


Original
Mandrill image

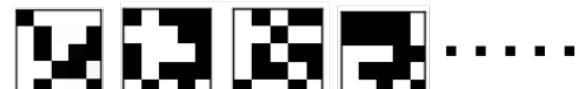
↓ 入力: 個別の状態遷移データ, 初期画像, ゴール画像

Variational AutoEncoder

Gumbel-Softmax を用いたVAEで
記号的命題表現に自動的に変換し
記号的プランニングソルバで解く



ノードの数を減らしてから元のサイズに戻す NN
入力画像と出力画像が一致するように学習
かつ、真ん中の層は0/1値: 何らかの命題の真偽値に対応



3.4.2 業績7: 査読付きワークショップ KEPS (採択率 60%)

深層学習(非記号型AI)と行動計画(記号型AI)の融合

画像で示された問題を

「パネル」「9マス」「動く」などの情報は一切与えられない問題を解いた例も与えられない



初期画像

ゴール画像

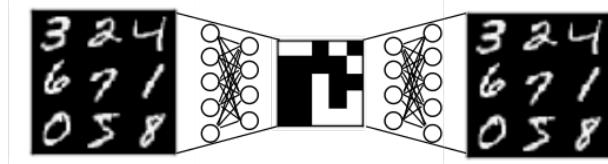


Original
Mandrill image

↓ 入力: 個別の状態遷移データ, 初期画像, ゴール画像

Variational AutoEncoder

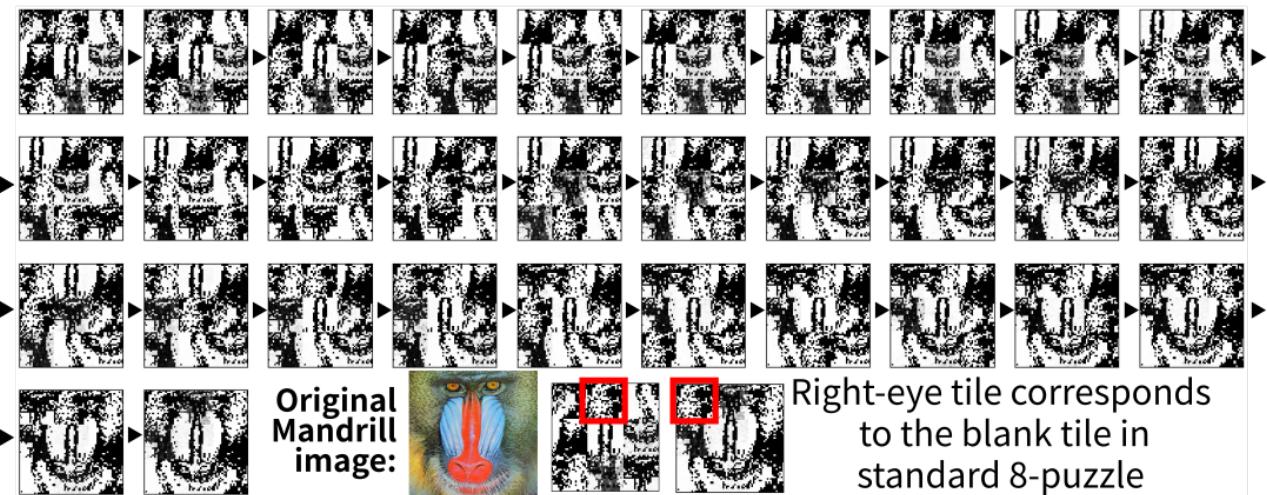
Gumbel-Softmax を用いたVAEで
記号的命題表現に自動的に変換し
記号的プランニングソルバで解く



ノードの数を減らしてから元のサイズに戻す NN
入力画像と出力画像が一致するように学習
かつ、真ん中の層は0/1値: 何らかの命題の真偽値に対応

↓
結果を画像に戻して
プランを返却する
システム

LatPlan

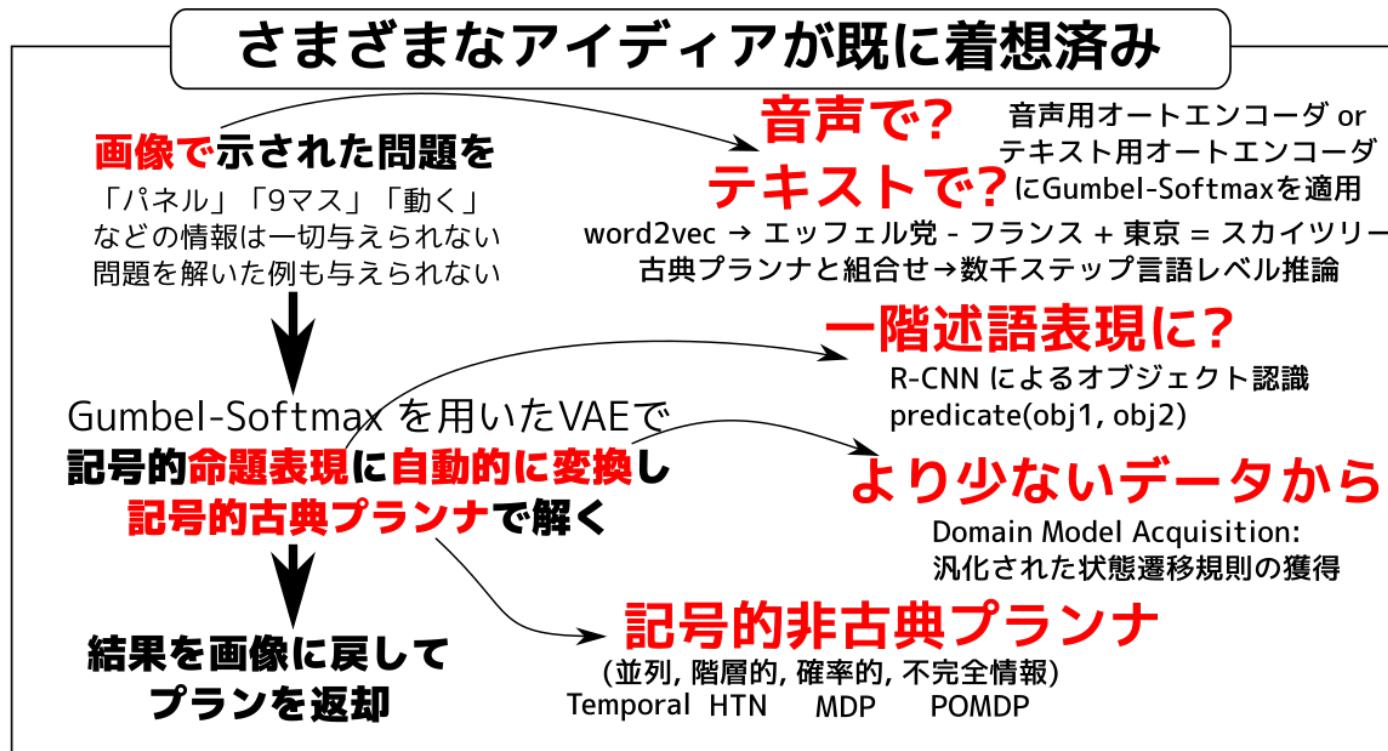


Original
Mandrill
image:

Right-eye tile corresponds
to the blank tile in
standard 8-puzzle

3.5 短期目標: このシステムを様々に拡張

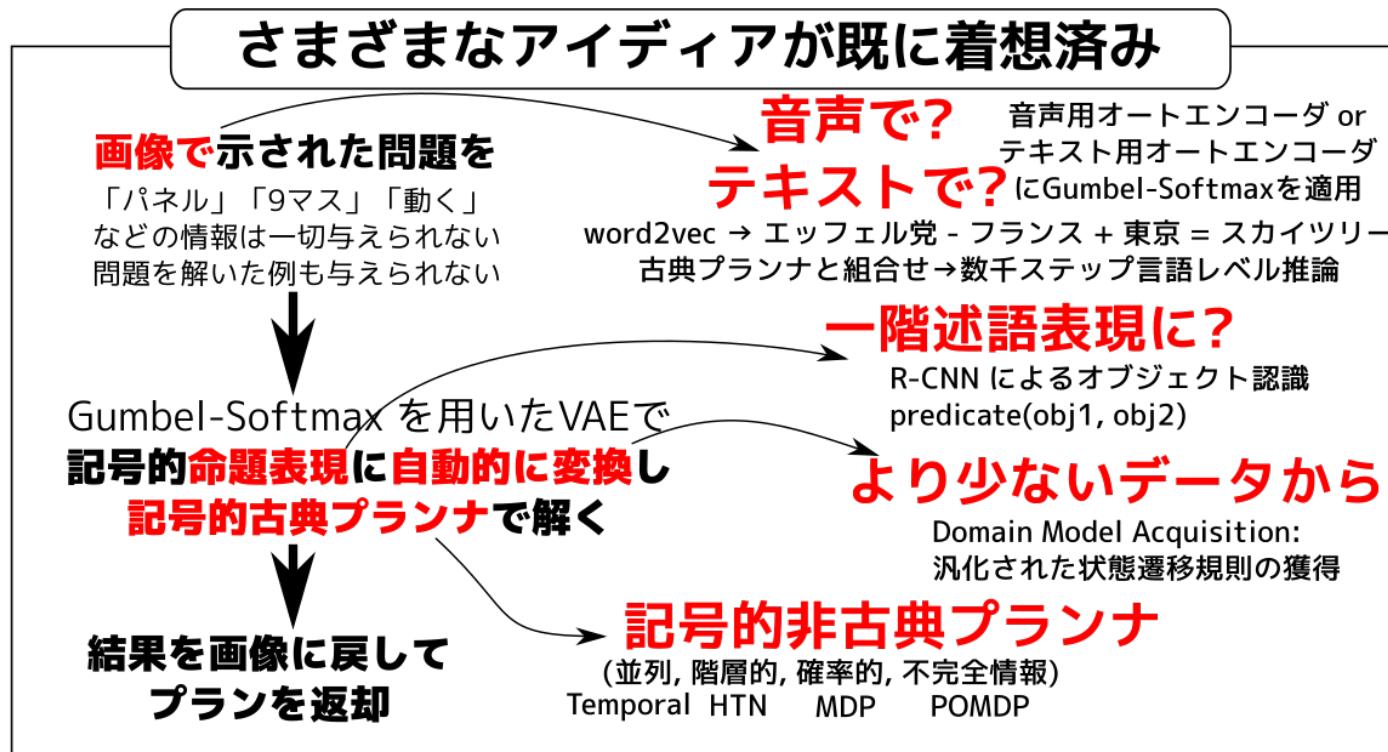
LatPlanの様々な発展形を実現



3.6 長期目標: パラダイムシフト

LatPlanの様々な発展形を実現

目標:
Neural-Symbolic 複合システムの
standard として地位を確立



3.7 産業応用: 深層学習+記号的推論 = ブレークスルー

人工無能以上の対話ボット

- 人工無能: 従来型対話ボットの総称
 - 反射的応答 = 論理を持たない
 - Apple Siri (人の書いた返答データベース)
 - Microsoft りんな (学習した返答ルール)
- テキスト 論理表現 記号的推論 返答
 - 論理に基づき 思考する ボットへ



4 めざす研究者像

分野の創始者 / 1000 以上の citation をもつ重要な発見の著者

- 10年後に AIMA (AIの超有名な教科書) に乗る内容
- いま興味・関心がある分野とは限らない
 - やりたいことはいくらでもある (進化計算/SAT/学習/知識表現/HCI)
- 今の自分は小粒研究者 本物の一流ビッグネームへ。

5まとめ

1. 難関国際会議(33%) Fully Automated Cyclic Planning for Large-Scale Manufacturing Domains. In ICAPS2014.
 - (a) 任意の問題から1種類の繰り返し構造を自動で検出
 - (b) 工場での製造スケジューリング(x1000高速化, 探索空間 $10^6 \sim 10^{274}$)
2. 難関国際会議(33%) Solving Large-Scale Planning Problems by Decomposition and Macro Generation. In ICAPS2015.
 - (a) 複数の繰り返し構造をより柔軟・汎用に組み合わせる手法
 - (b) ベンチマークセット全体で高速化(x3-4高速化, 探索空間 $10^7 \sim 10^{28}$)
3. 難関国際会議(26%) Tiebreaking Strategies for A* Search: How to Explore the Final Frontier. In AAAI-2016. (JSAT 学生奨励賞)
 - (a) コストゼロの辺がグラフ探索に引き起こす問題を解決(探索空間 $10^6 \sim 10^{88}$)
4. 難関論文誌(12%) Tie-Breaking Strategies for Cost-Optimal Best First Search. Journal of Artificial Intelligence Research 58 (2017): 67-121.
 - (a) (3.)に加え タイブレーキングと非最適コスト探索の関連性を指摘, さらに性能向上
5. 難関国際会議(33%) Exploration Among and Within Plateaus in Greedy Best-First Search. In ICAPS2017.
 - (a) 非最適コスト探索をフラクタルを用いて改善
 - (b) プラトー内均一化とプラトー間均一化の直交性を実証
6. 難関国際会議(25%) Efficient Optimal Search under Expensive Edge Cost Computation. In IJCAI-2017.
 - (a) 辺コストの動的計算が必要な問題に対して高速な最適アルゴリズム DEA*
7. 国際ワークショップ(60%) Classical Planning in Deep Latent Space: From Unlabeled Images to PDDL (and back). Knowledge Engineering for Planning and Scheduling (KEPS) Workshop
 - (a) 画像から命題を自動生成して記号的AIで組合せ最適化問題を解き、画像で出力するシステム

6 付録

section.1 subsection.1.1 subsubsection.1.1.1 subsubsection.1.1.2 subsubsection.1.1.3 subsubsection.1.1.4
subsubsection.1.1.5 subsection.1.2 subsubsection.1.2.1 subsection.1.3 subsection.1.4 subsection.1.5
subsection.1.6 subsubsection.1.6.1 subsubsection.1.6.2 subsection.1.7 subsection.1.8 section.2 sub-
section.2.1 subsection.2.2 subsection.2.3 subsection.2.4 subsection.2.5 subsection.2.6 section.3 sub-
section.3.1 subsection.3.2 subsection.3.3 subsection.3.4 subsubsection.3.4.1 subsubsection.3.4.2 sub-
section.3.5 subsection.3.6 subsection.3.7 section.4 section.5 section.6

6.1 既存研究について	40
6.2 AIの倫理について	48
6.3 ベンチャー?	49
6.4 ライフワーク: 人はより自由に創作/研究できるべきである	50
6.5 LatPlan 関連	51

6.1 既存研究について

6.1.1 その研究は…

評価	オリジナリティ	過去のインパクト	未来のインパクト
ACP 難関学会	ループの概念を検出	超大規模問題に初めてタックル	産業応用
CAP 難関学会	問題分割統合手法	より汎用	産業応用
AAAI16 査読者 2/3 人に絶賛 ・ JAIR17 (+ JSAT 学生奨励賞)	同コストのノードの分類	70年代からの定説を覆す 通常と異なる方法で性能改善	コスト0は実応用によ り基礎技術
IJCAI17 難関学会	問題設定(辺コストが未知)		医療福祉での実問題
ICAPS17 難関学会	均一化手法の整理・高速化		
KEPS17 査読者「重要な方向性」	画像から自動で命題を生成	シンボルグラウンディング	ニューラル・記号複合

6.1.2 古典プランニングを研究する意義は?

"古典"プランニングを研究する意義は?

古典プランニングは

様々な拡張のベースとなる **最小限** のフレームワーク

実験は 古典プランニングでやった というただそれだけ

← 古典プランニングでは評価がやりやすい

- ・単純なモデル

- ・詳細が調べつくされている

- ・何が性能向上の理由なのか切り分けやすい(差分が取りやすい)

直接の実用的な
意義もある



Deep Space 1
(Muscettola et al. '98)
宇宙船の自律航行



Mars Exploration Rover
(Bresina et al. '05)
火星探査ロボットの
自律行動システム



Xerox Parc Printer
大規模 商業印刷システムの
自律運用

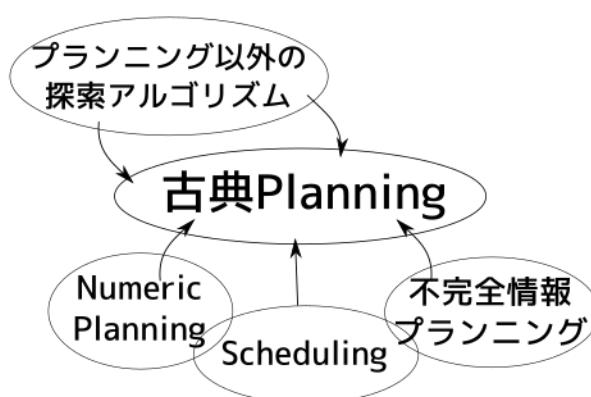


Core Security (サービス企業)
企業ネットワーク脆弱性の自動診断
攻撃プランの自動生成

様々な拡張の「基礎」となる基盤技術

古典プランニングで成功した技術は

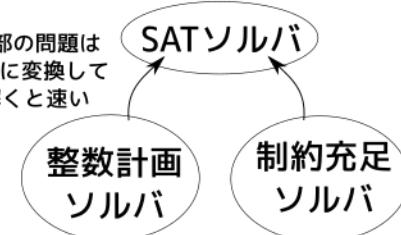
他の問題にも使える!



類似のシナリオ:SATソルバ

同じく基礎的なモデル

一部の問題は
SATに変換して
解くと速い



6.1.3 Explicit Graph と Implicit Graph との違い

カーナビ、ソーシャルグラフなど : Explicit Graph
グラフ全体がメモリ (~ 数ペタバイト) または二次記憶 (~ 数ゼタバイト) に収まる
参考: 2012年の全世界のデジタルデータ: 数ZB (1ZB = 10^{21} バイト)
AI and Web, social graph

プランニングにおける探索グラフ : Implicit Graph
地球上の全計算資源を集めても二次記憶に入らない
グラフのノード数は状態変数に対して 指数的に増加
動的メモリ確保+優れた枝刈りをしないと問題が解けない
探索空間サイズの例:
3x3x3 のルービックキューブ: $4.32 \times 10^{19} = 4$ EB
4x4x4 のルービックキューブ: $7.40 \times 10^{45} > 10^{24}$ ZB
5x5x5 のルービックキューブ: 2.83×10^{74}

6.1.4 第五世代コンピュータとの違いは?

第五世代コンピュータ：並列推論機械(Prologベース, ハードウェア, OS)

ハードの問題ではなく、根本的なソフトウェア技術、探索技術が未発達だった

第五世代	現在
後方全探索 + バックトラック	前方ヒューリスティック探索
問題ごとに手書きの cut で枝刈り	汎用下界関数
Prologベース	C/C++で高度に最適化されたプログラム State packing, 決定木, mutex...

今はベンチマーク問題 1104問のうち 5分で 800問 前後解ける

仮に当時のソフトウェアを現在のハードウェアで動かしたとしても、100問も解けない

6.1.5 似たような研究は誰がやっているか

ICAPS, SoCS : 例年 200 人-300 人の参加者を集めており大変盛況,

AAAI, IJCAI : プランニングに関する論文は例年数十本採択 Proceedings の一つの章

JAIR, AIJ (トップジャーナル) : 同様に多い (JAIR Volume 54: 12 本中 2 本 がプランニング)

主な研究室:

MIT CSAIL (Brian Williams), Carnegie-Mellon,

NASA (NASA Ames および NASA/Caltech JPL のそれぞれに 20 名以上の研究者)

欧州宇宙機関 (ESA)

King's College London, U. Freiburg, U. Bazel, U. Toronto, U. Ben-Gurion

指導教員は NASA JPL AI Lab の元メンバー

6.1.6 国内で誰が似たようなことをやっているか

この専門分野をやっている人はほぼいない

石田 亨先生 (京都大学) — 以前は日本の A* 系の探索手法の代表者, R. E. Korf と共に著

- R.E. Korf — Alex Fukunaga — 浅井 政太郎

岸本先生 (探索) (東工大 IBM Research Ireland)

少し離れているが似ている研究というと

- SAT ソルバ, ASP(解集合) ソルバ, CP ソルバ系, 推論系 – logic and reasoning : 田中 哲朗先生 (総合文化研究科), 鍋島英知先生 (山梨大), 田村直之先生, 平山勝利先生 (神戸大), 田沼先生 (東工大)
- 井上克己先生 (NII) — Lemma Reusing for SAT based Planning and Scheduling (ICAPS10)
- AAMAS マルチエージェント, CSP : 横尾誠先生 (九大)
- ERATO グラフ探索 (explicit graph) 系, 大規模知識処理系 : 河原林健一先生 (NII), 秋葉先生, 濟先生 (北海道)
- ゲーム系 金子先生 (総合文化研究科)

6.1.7 汎用性を失わずに解く?

No Free Lunch 定理: 最適化アルゴリズムの性能は 全問題の平均を取れば 全て同じ

Q. NFL 定理のもとで「汎用性を失わずに高速に解く」というのは不可能?

A. NFL 定理は確かにそのように主張するが、プランニング分野の意味する「汎用性」は
実問題(人間にとて有意義な問題の集合)における汎用性である。

全プランニング問題の集合 人間にとて有意義な問題の集合

従って、全問題の平均を取れば という前提が成り立たない。

6.2 AIの倫理について

- 研究内容は 漠然とした「AI」のうち グラフ探索 の研究
- AIの目的はプランナのゴールで定義される
 - AI自体は善悪の判断を行わない
 - 人が与えるゴールについての倫理的問題(兵器使用)
- 値値判断は与えられる入力の中にある = 使用者の価値観/データのバイアスを反映
- 悪用の問題はある。しかし、自分としては、災害救助ロボットなど、人道的な応用を目指している

6.3 ベンチャー?

ベンチャーで大切なのは 顧客 速さ アイディア

何を やらないか:

- 研究目的のベンチャー *if all you have is a hammer, everything looks like a nail*
- 請け負い / スモールビジネス (国内ベンチャーの多く)
- 多目的の複雑な製品 / 過剰サービス ()

おそらく生活に密接した自動エージェント

6.4 ライフワーク: 人はより自由に創作/研究できるべきである

上記の目標を達成したらの話

- 人/機械 は、 ひとつの課題でキャッシュを有効利用 すると 最大パフォーマンス
- 今でさえ distraction が多すぎる (プレゼン準備, 事務書類, 論文締め切り)
- プrezen、論文 は 知の伝播/蓄積 のために必要だが、 研究の本質 ではない
- プrezen、論文の形式記述化と自動生成
 - 主張を適切な順序で並べるプランニング問題
 - 現代の研究活動で最も「時給単価が高い人間の手作業」を要する分野

6.5 LatPlan 関連

6.5.1 AI プランニングの *Killer App*

人が高価 or 不可能な作業 原発、宇宙空間、
火星、深海

正しさと最適性の理論保証が必要なミッション

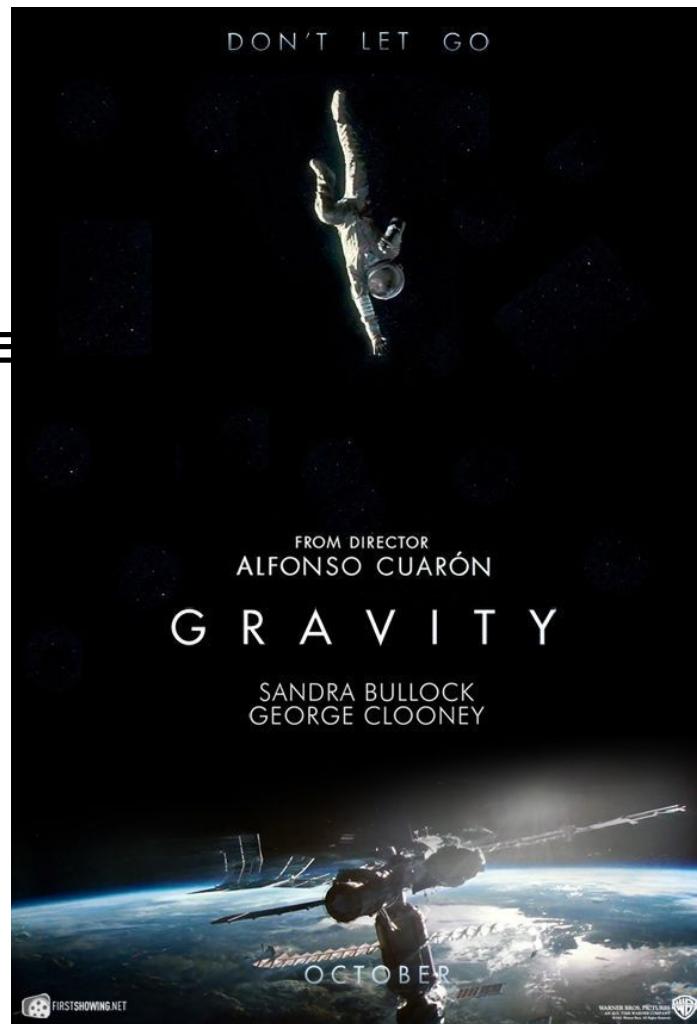
製造システム、運送 (時間=お金)

人工衛星 (燃料使いきれば運用終了)

間違った解は許されない

思考過程を説明可能なシステム レス

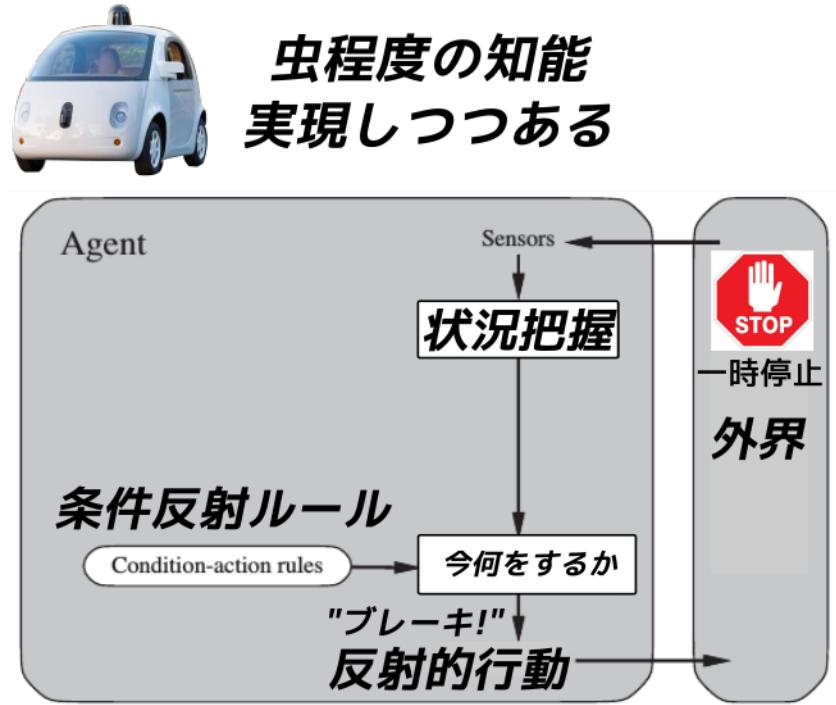
キュー・宇宙船 (人間の安全がかかっている)



6.5.2 知的機械を作る上で深層学習の果たす役割

6.5.3 知的機械を作る上で深層学習の果たす役割

Reflex Agent (反射的機械)



6.5.4 知的機械を作る上で深層学習の果たす役割

Reflex Agent (反射的機械)



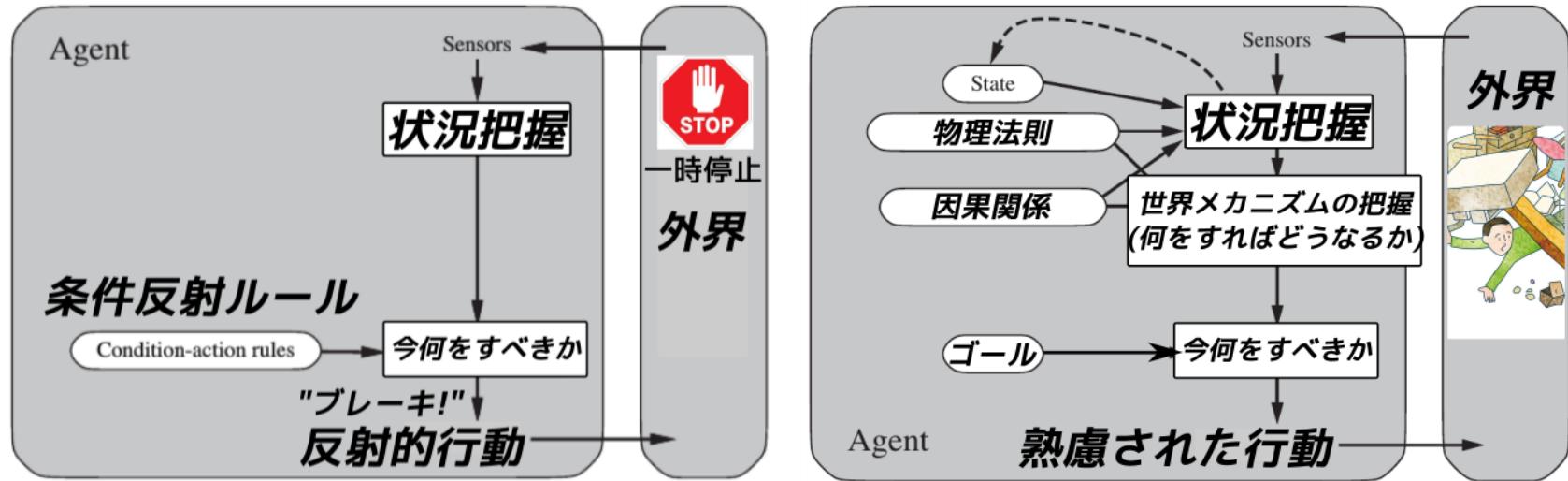
虫程度の知能
実現しつつある



Goal-based Agent (目的達成型 機械)

本当は
コレが
欲しい

目標を達成するために
論理で戦略を熟慮する機械



6.5.5 目的達成型機械

反射的に動くロボットでは複雑なゴールを設定できない

瓦礫に埋もれた骨折・出血している被災者をなるべく早く病院に運ぶ。

救急車を先に呼んでおき、患者を傷つけないように、素早く、瓦礫を崩れないようにどこで止血し、当て木し、担架に乗せて、ちょうどきた救急車に隊員と協力して運び入れる。

- **目的達成/熟慮型の知能が必要**

熟慮: deliberation の訳

6.5.6 知的機械を作る上で深層学習の果たす役割

Reflex Agent (反射的機械)



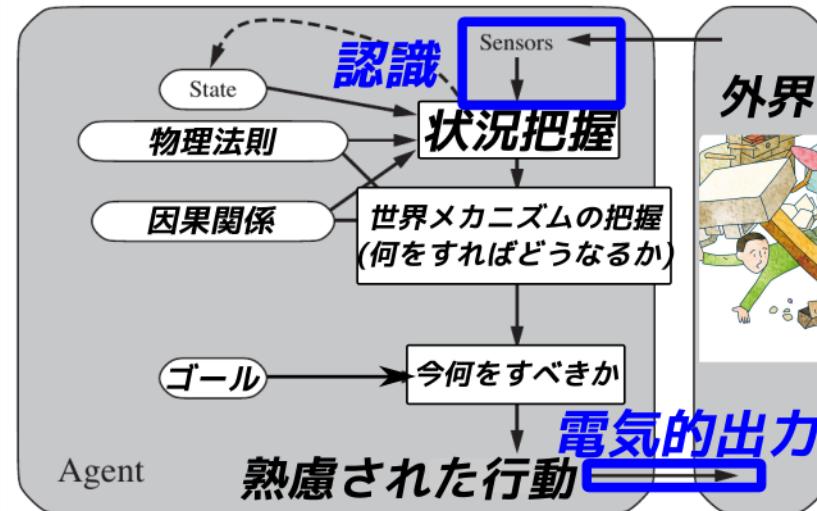
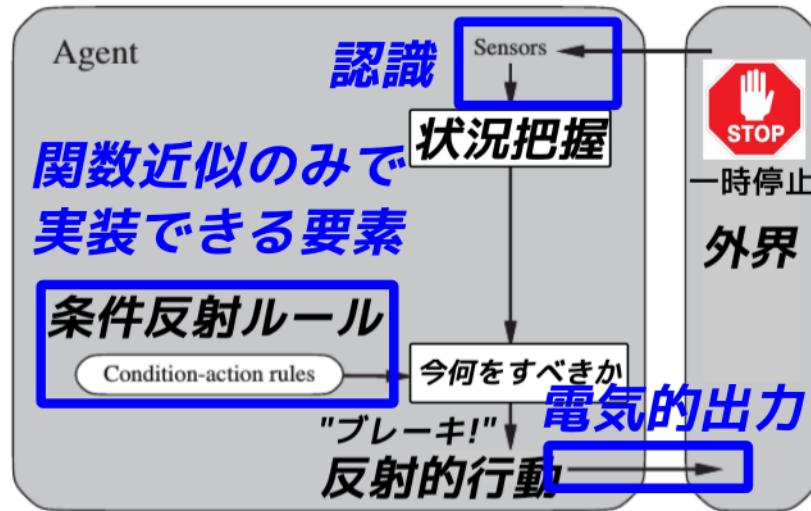
虫程度の知能
実現しつつある



Goal-based Agent (目的達成型 機械)

本当は
コレが
欲しい

目標を達成するために
論理で戦略を熟慮する機械



6.5.7 知的機械を作る上で深層学習の果たす役割

Reflex Agent (反射的機械)



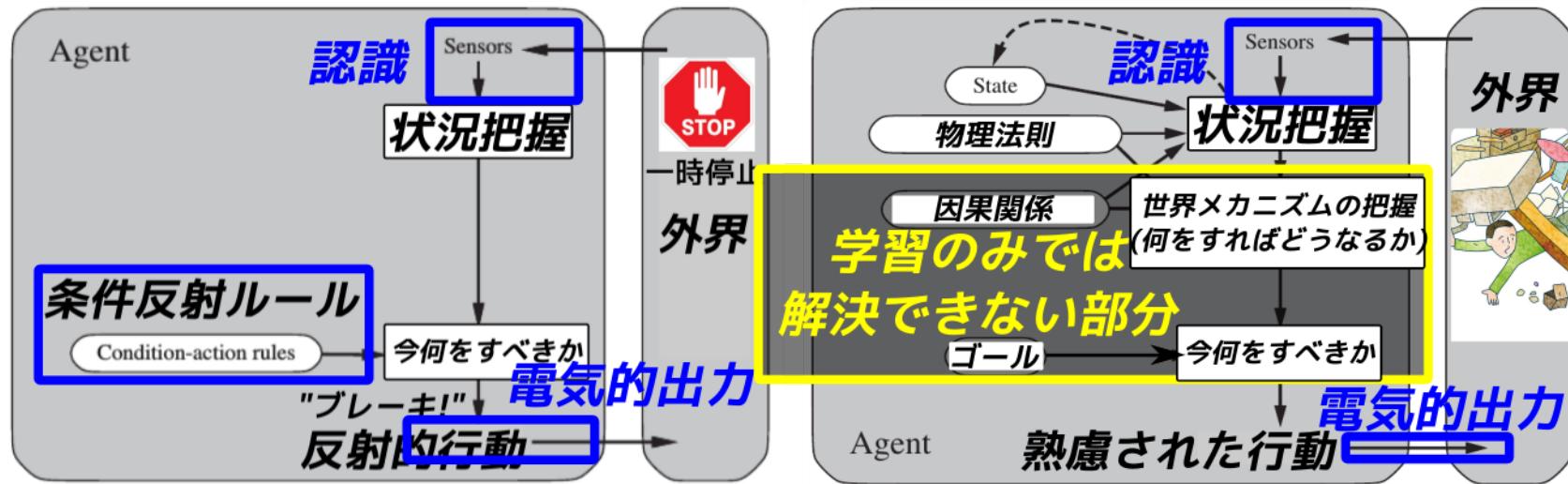
虫程度の知能
実現しつつある



Goal-based Agent (目的達成型 機械)

本当は
コレが
欲しい

目標を達成するために
論理で戦略を熟慮する機械



- 深層学習でカバーできない部分には 記号的 AI(プランニング)が必要

6.5.8 既存システムとの違い(追加)

NNで直接問題を解くシステム

- TSP [Hopfield and Tank, 1985], NeuroSolver [Bieszczad and Pagurek, 1998], Neural Turing Machine [Graves *et. al.*, 2014]
 - NNで解くが、入力はシンボリック(ニューロンが人の与えた状態変数に対応)
 - 完全性、最適性などの保証なし

NNを認識でなく探索中の枝刈り用に使うシステム

- AlphaGo [Sievers 16], Rubik Cube [Arfaee *et al.*, 2011], Planning [Satzger and Kramer, 2013]
 - LatPlan は NNを探索の外で使う

6.5.9 Learning from Observation との違い

主にロボットの経路探索(ローレベル制御) [Argall et al., 2009]

ボードゲームの学習だが「マス目」など強い仮定 [Barbu et al., 2010; Kaiser, 2012; Kirk and Laird, 2016]

Action Segmentation problem がある

- 「映像の観察」を中心とするので、いつアクションが始まる/終わるのか解らない
- LatPlan には関係なし

6.5.10 Gumbel-Softmax

深層学習(非記号型AI)と行動計画(記号型AI)の融合

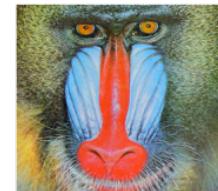
画像で示された問題を

「パネル」「9マス」「動く」などの情報は一切与えられない問題を解いた例も与えられない



初期画像

ゴール画像

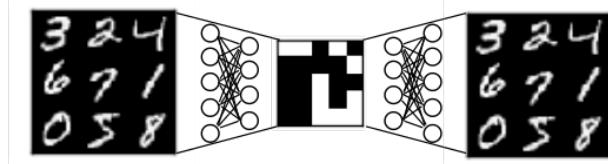


Original
Mandrill image

↓
入力: 個別の状態遷移データ, 初期画像, ゴール画像

Variational AutoEncoder

Gumbel-Softmax を用いたVAEで
記号的命題表現に自動的に変換し
記号的プランニングソルバで解く



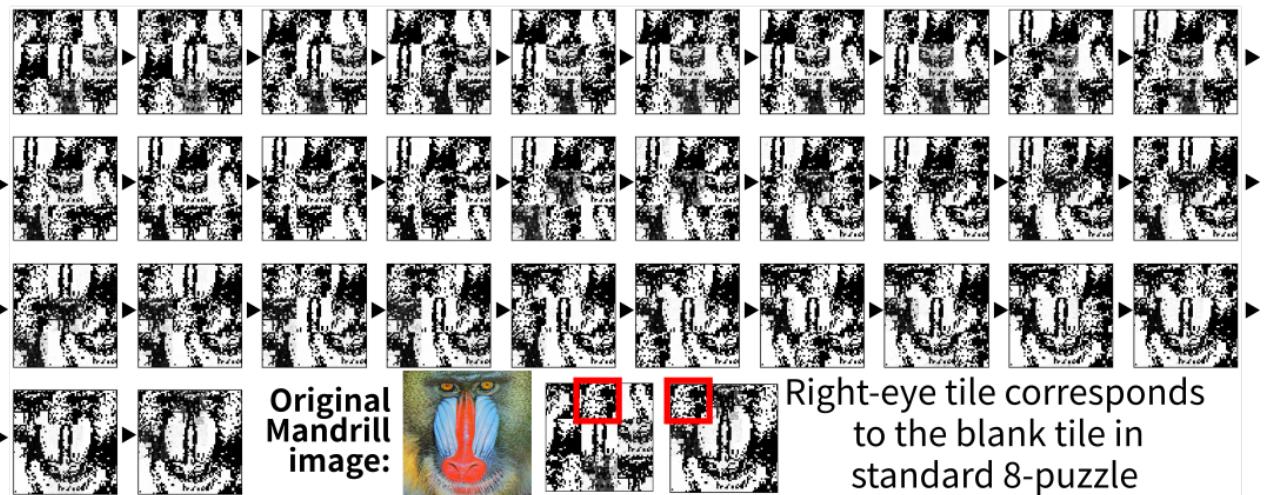
ノードの数を減らしてから元のサイズに戻す NN
入力画像と出力画像が一致するように学習
かつ、真ん中の層は0/1値: 何らかの命題の真偽値に対応

↓
結果を画像に戻して

プランを返却する

システム

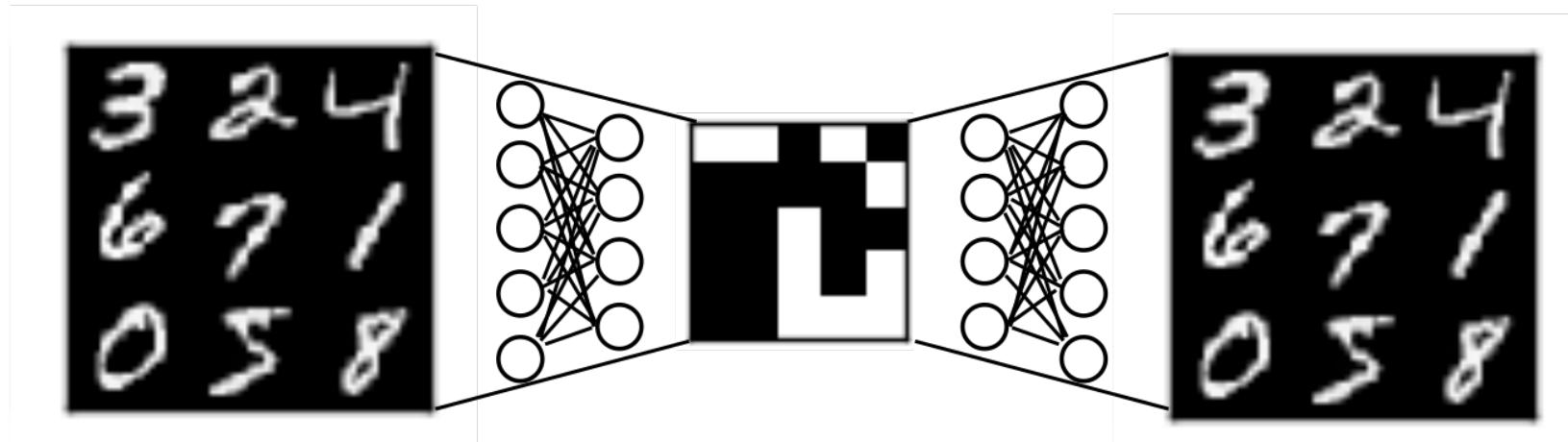
LatPlan



6.5.11 Gumbel-Softmax

AutoEncoder(AE): 入力と出力が一致するように学習するNN
中間層: 記号的システムで扱えない実数値

↓
Gumbel-Softmax AE: 1-hot カテゴリカル分布 を近似するAE
中間層: 0/1値, 何らかの命題の真偽値に対応させて
記号的システムで使える!



6.5.12 LatPlan 実装

詳しくは論文を

Keras + TensorFlow

1764(42x42)

[FC(4000,ReLU) Batchnorm Dropout(0.4)] × 2

FC(49,GumbelSoftmax) (variational loss)

[FC(4000,ReLU) Batchnorm Dropout(0.4)] × 2

1764(42x42) (loss: Binary crossentropy)

なぜ全結合?? 論文の主題は SAE で命題を作る方針がそもそもうまく行くかどうか

余計な要素を省いて限りなくシンプルに

8-パズルでの訓練 可能な全状態 (362880) 中 12000 枚 で訓練 汎化能力あり

Adam optimizer (learning rate:0.001)