PMR-3510 Inteligência Artificial

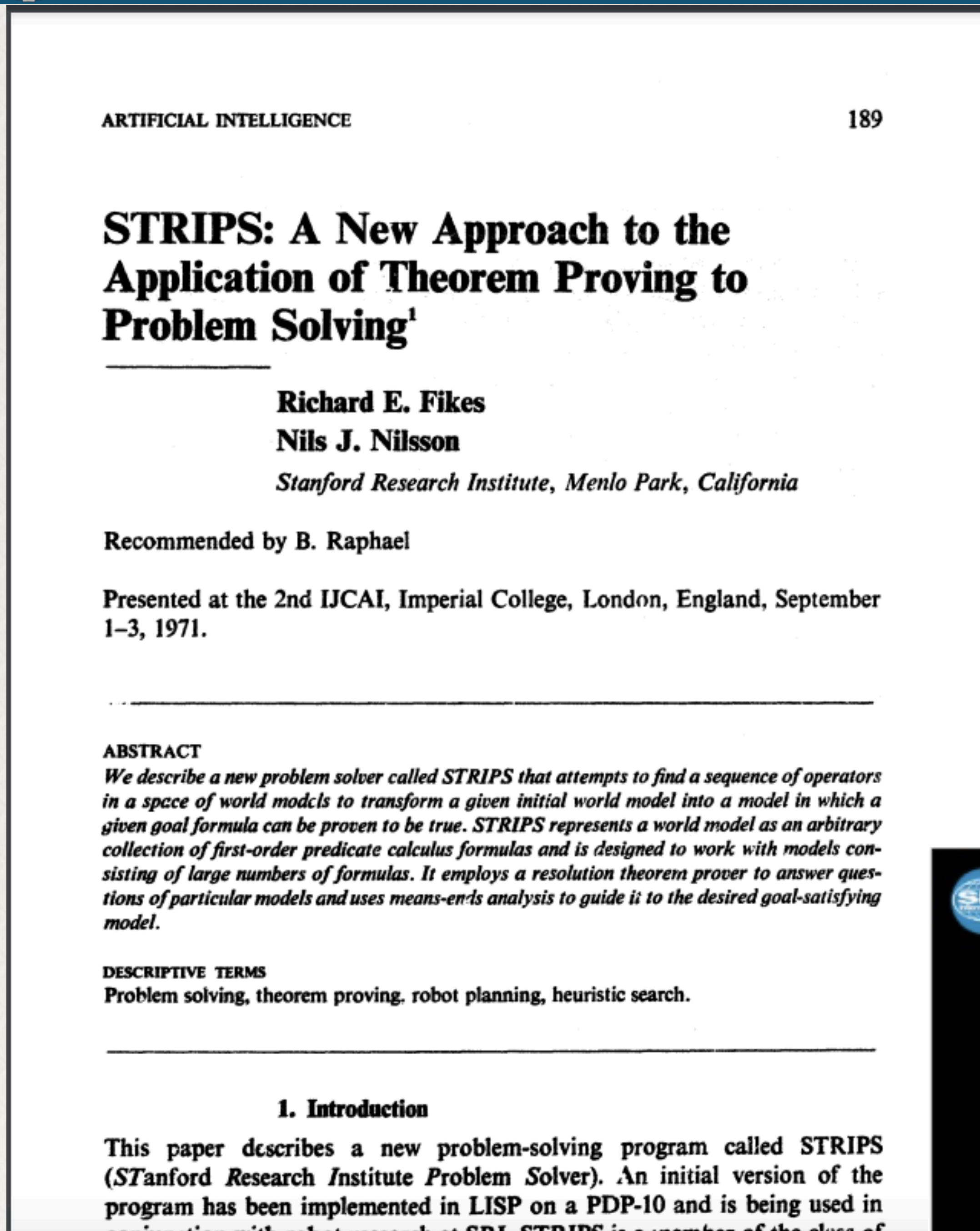
Aula 15 - Exploring STRIPS

Prof. José Reinaldo Silva
reinaldo@usp.br

d-l@b

Planning is present every day

Imagine as ações: put_left_socket, put_right_socket, put_left_shoe, put_right_shoe. A nossa intuição identifica automaticamente as ações que requerem uma ordem parcial.



STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving[1]

Richard E. Fikes
Nils J. Nilsson
Stanford Research Institute, Menlo Park, California

Recommended by B. Raphael

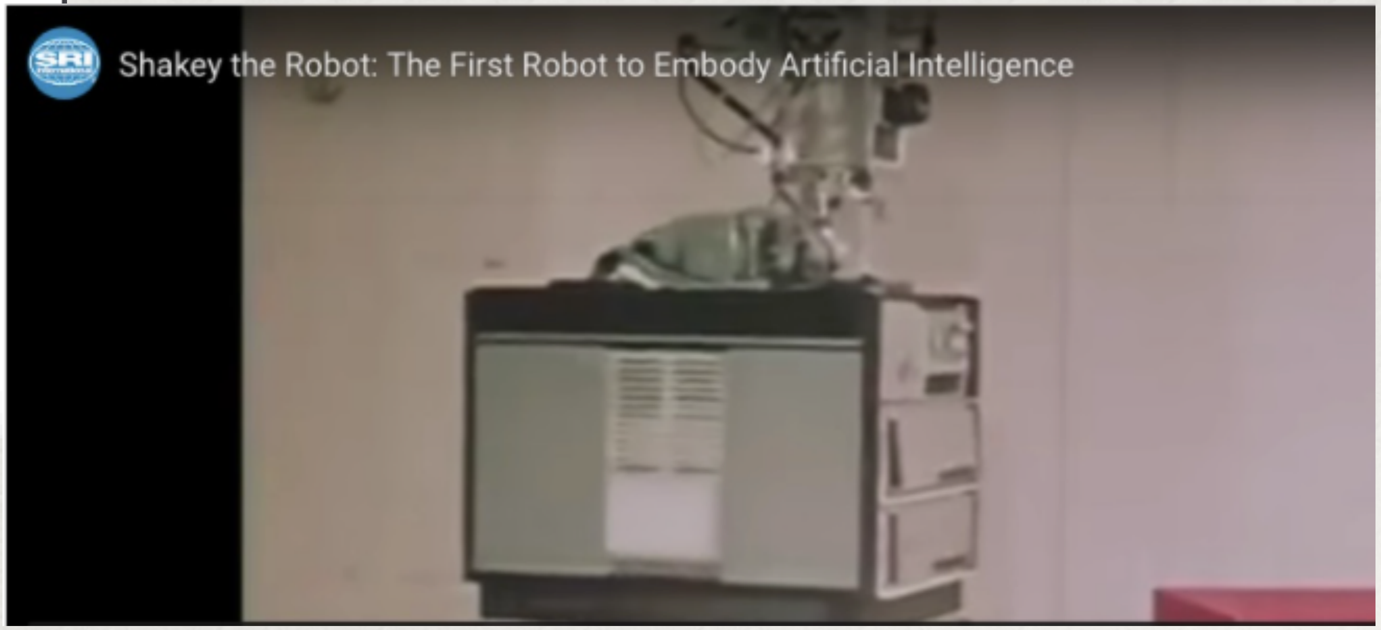
Presented at the 2nd IJCAI, Imperial College, London, England, September 1–3, 1971.

ABSTRACT
We describe a new problem solver called STRIPS that attempts to find a sequence of operators in a space of world models to transform a given initial world model into a model in which a given goal formula can be proven to be true. STRIPS represents a world model as an arbitrary collection of first-order predicate calculus formulas and is designed to work with models consisting of large numbers of formulas. It employs a resolution theorem prover to answer questions of particular models and uses means-ends analysis to guide it to the desired goal-satisfying model.

DESCRIPTIVE TERMS
Problem solving, theorem proving, robot planning, heuristic search.

1. Introduction

This paper describes a new problem-solving program called STRIPS (*ST*anford *R*esearch *I*nstitute *P*roblem *S*olver). An initial version of the program has been implemented in LISP on a PDP-10 and is being used in conjunction with robot research at SRI. STRIPS is a member of the class of



George Pólya
1887-1985

PONTES (2019) HOLOS ISSN 1807 - 1600

MÉTODO DE POLYA PARA RESOLUÇÃO DE PROBLEMAS MATEMÁTICOS: UMA PROPOSTA METODOLÓGICA PARA O ENSINO E APRENDIZAGEM DE MATEMÁTICA NA EDUCAÇÃO BÁSICA

E.A.S.PONTES*
Instituto Federal de Alagoas
edelpontes@gmail.com*

Artigo submetido em 20/12/2017 e aceito em 24/06/2019
DOI: 10.15628/holos.2019.6703

RESUMO
...mundo contemporâneo diversas pesquisas são ...em busca de uma solução eficaz no processo ...matemática, tendo como ...da educação matemática. ...objetivo apresentar uma ...o ensino e aprendizagem ...básica, através da resolução ...do o método de Polya. O método ...n três etapas: Compreender o ...um plano, Executar o plano e Retrospecto do problema. M... apresentados três problem... resolução seguirá o método ... Resolução de Problemas atrav... como prática educacional no... aprendizagem de matemática... facilitador e ao aluno apre... habilidades no intuito de fo... crítico e o raciocínio lógico.

...aprendizagem de matemática, método de Polya, Resolução d...

IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 13, NO. 6, NOVEMBER/DECEMBER 2001 913

Structured Development of Problem Solving Methods

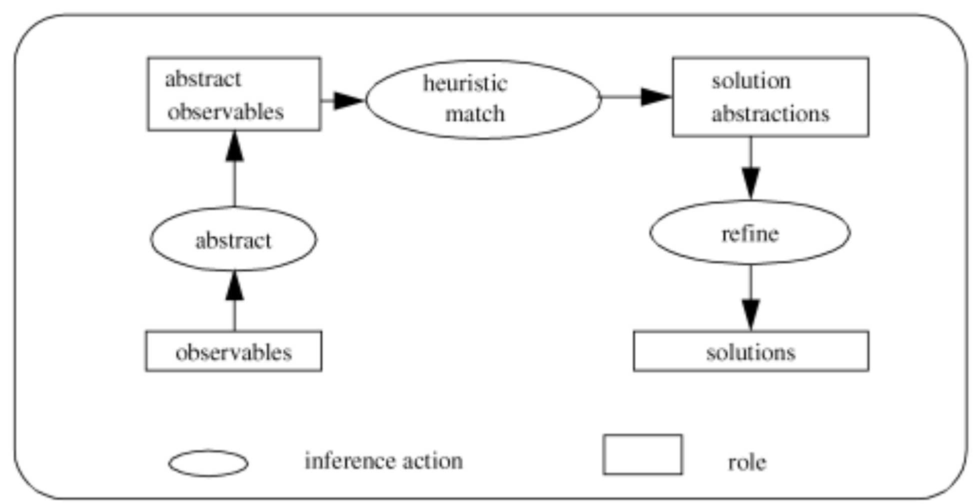
Dieter Fensel and Enrico Motta

Abstract—Problem solving methods (PSMs) describe the reasoning components of knowledge-based systems as patterns of behavior that can be reused across applications. While the availability of extensive problem solving method libraries and the emerging consensus on problem solving method specification languages indicate the maturity of the field, a number of important research issues are still open. In particular, very little progress has been achieved on foundational and methodological issues. Hence, despite the number of libraries which have been developed, it is still not clear what organization principles should be adopted to construct truly comprehensive libraries, covering large numbers of applications and encompassing both task-specific and task-independent problem solving methods. In this paper, we address these "fundamental" issues and present a comprehensive and detailed framework for characterizing problem solving methods and their development process. In particular, we suggest that PSM development consists of introducing assumptions and commitments along a three-dimensional space defined in terms of *problem-solving strategy*, *task commitments*, and *domain (knowledge) assumptions*. Individual moves through this space can be formally described by means of *adapters*. In the paper, we illustrate our approach and argue that our architecture provides answers to three fundamental problems related to research in problem solving methods: 1) what is the epistemological structure and what are the modeling primitives of PSMs? 2) how can we model the PSM development process? and 3) how can we develop and organize truly comprehensive and manageable libraries of problem solving methods?

Index Terms—Knowledge modeling, problem-solving methods, ontologies, knowledge engineering, software engineering, formal languages.

1 INTRODUCTION

P*roblem solving methods* (PSMs) describe the reasoning ... com... behavio... instance... ([56], [9... ized by... "revision... ce—sche... solving... supporti... edge en... framewo... ([55], [8...



Lecture Notes for 6.824, Artificial Intelligence
 Rev. November, 1994

Graph Search and STRIPS Planning

1 Introduction to Graph Search

Consider the eight puzzle show in figure 1. This puzzle is played on a three by three square board containing eight tiles and an empty square. Any tile which is next to the empty square can be slid into the empty square leaving an opening (empty square) in the place that used to be occupied by the moved tile. Figure 1 shows how one state of the puzzle can be transformed into another configure by sliding a tile into the empty square. Given an initial state of the puzzle the objective is to find a sequence of legal moves that transform the initial state into some desired goal state. Figure 2 also shows a typical goal state.

The problem of finding a sequence of moves that transforms a given initial state into a given goal state can be formulated as a graph search problem. The nodes of the graph are the possible states of the puzzle and the arcs of the graph correspond to legal moves that transform one state into another. The problem is to find a path in this graph from a given initial state to a given goal state.

In the eight puzzle there are four types of legal moves. The empty square can move up, move down, move right, or move left. This allows the puzzle to be formulated in terms of four operations U, D, L, and R that move the empty square up, down, right and left respectively. If the empty square is already on the upper edge of the board then we define the operation U to

Definition: A *STRIPS planning problem* consists of a STRIPS operator specification, a set Σ of initial propositions and a set Ω of goal propositions.

Definition: A *solution* to a STRIPS planning problem is a plan (sequence of operators) α such that, in every graph where the STRIPS operator specification holds, we have $\Sigma \rightarrow [\alpha]\Omega$.

Definition: A *STRIPS operator specification* consists of a set of *operator symbols* where each operator symbol is associated with a *prerequisite list*, an *add list* and a *delete list* each of which is a set of proposition symbols.

Definition: A STRIPS operator specification is said to *hold* (or be *valid*) in a graph search problem if for each operator o_i, and each node n such that every prerequisite of o_i is true at n, we have the following conditions.

- All propositions on the add list of o_i are true at the node $o_i(n)$.
- If P is a proposition that is *not* on the delete list of o_i, and P is true at n, then P is true at $o_i(n)$.

STRIPS

The planning process matches the graph-oriented approach used in Engineer and particularly in automation.

General Planning approach

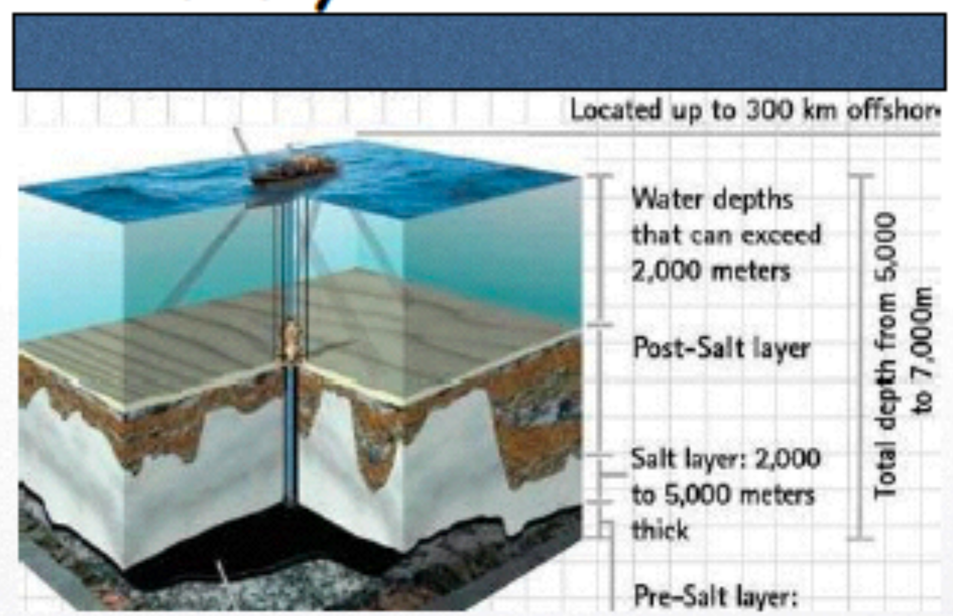
Domain independent

Knowledge Base

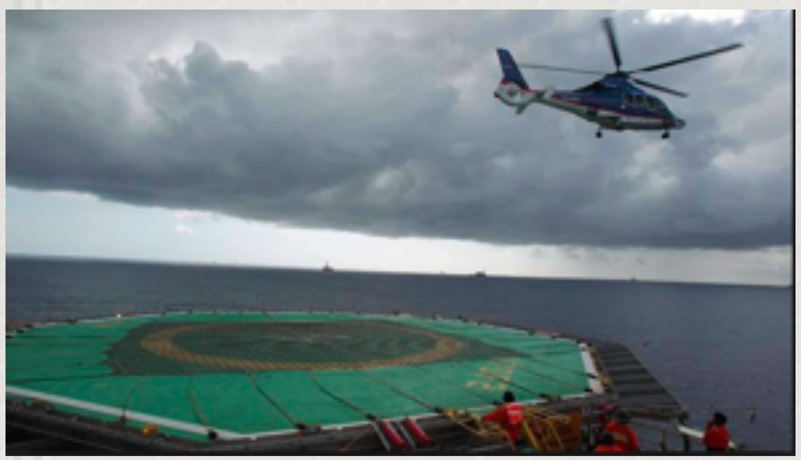
Domain dependent

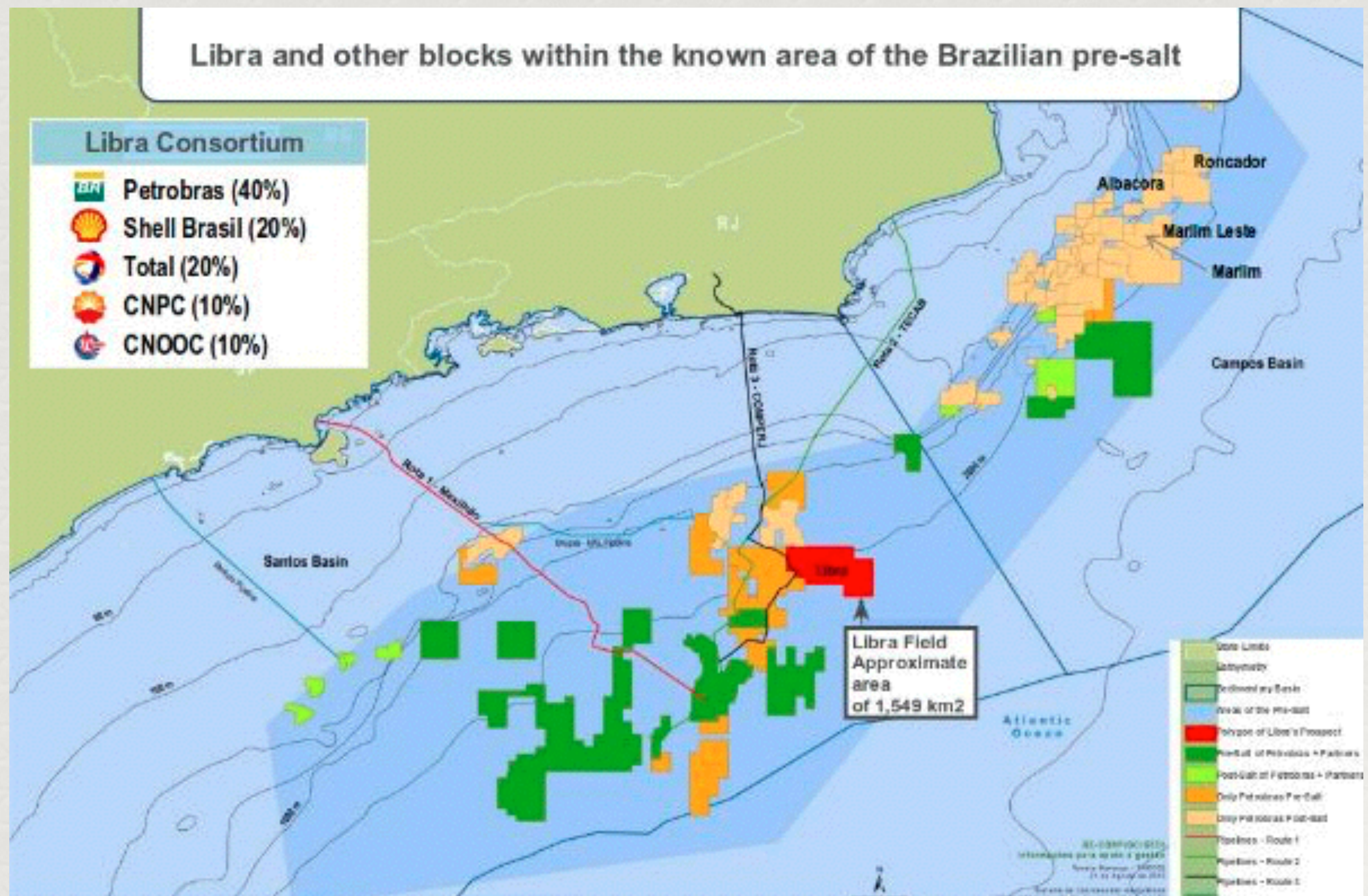
Exemplos: SIPROV (Sistema Inteligente para Programação de Vôo)

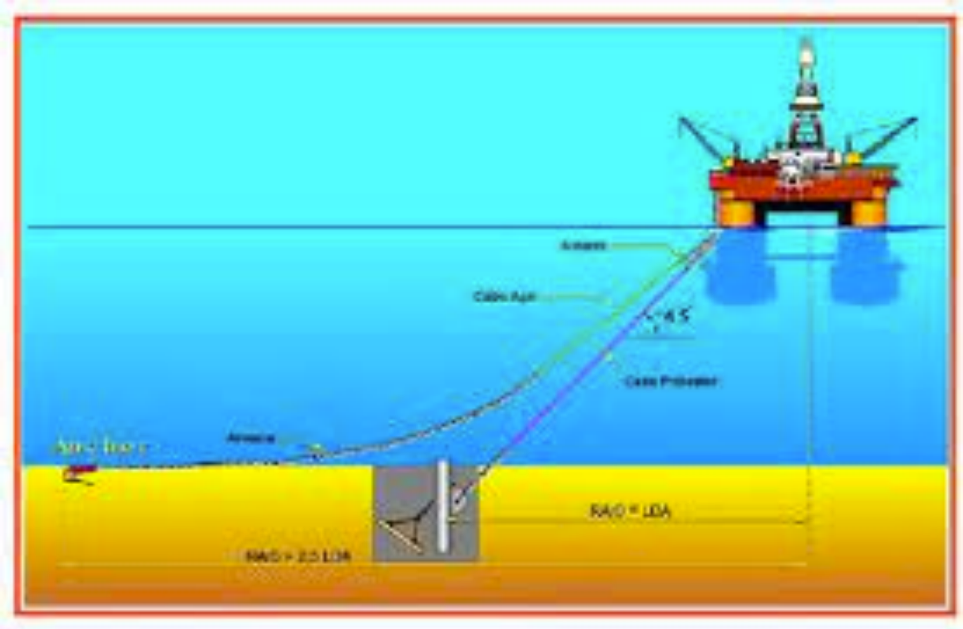
Petroleum exploitation in the pre-salt layer



Exemplos: SIPROV (Sistema Inteligente para Programação de Vôo)





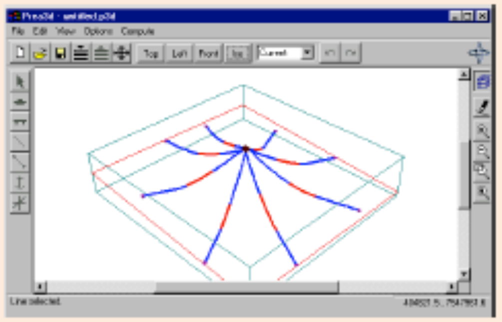


SEANC

PETROBRAS

SEANC-BD
Oracle

Prea3D

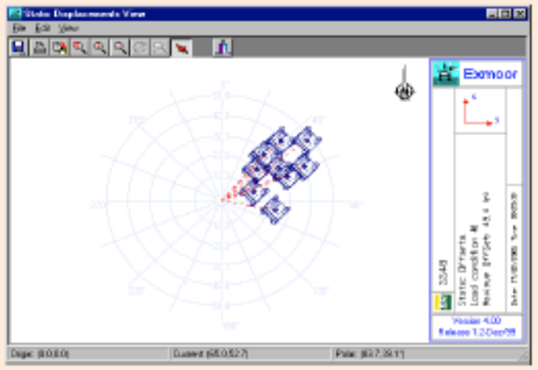


Calculo de linha
Visualização 3D
LANC

SEANC

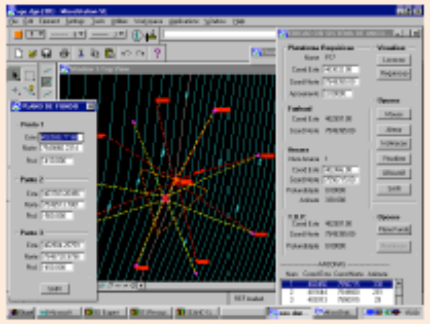
Controle
Sistemas
Inteligentes
Preparação de dados

Exmoor



Calculo de linha
Validação de criterios

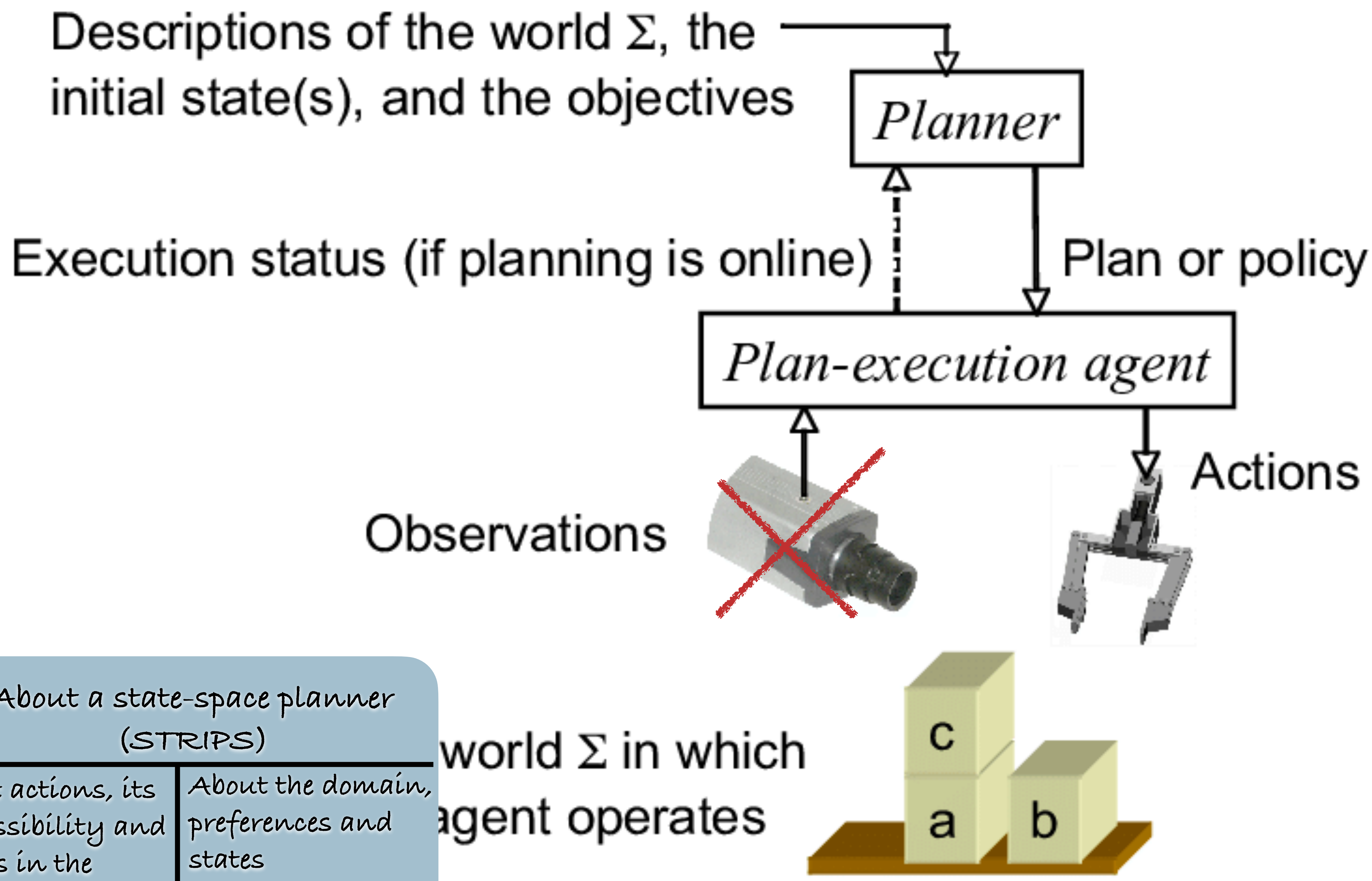
SGO



Visualização do fundo
obstáculos
posição de outros vasos

Existe um métodos geral, independente de domínio, para resolver problemas, em especial o problema de agentes com objetivo?.

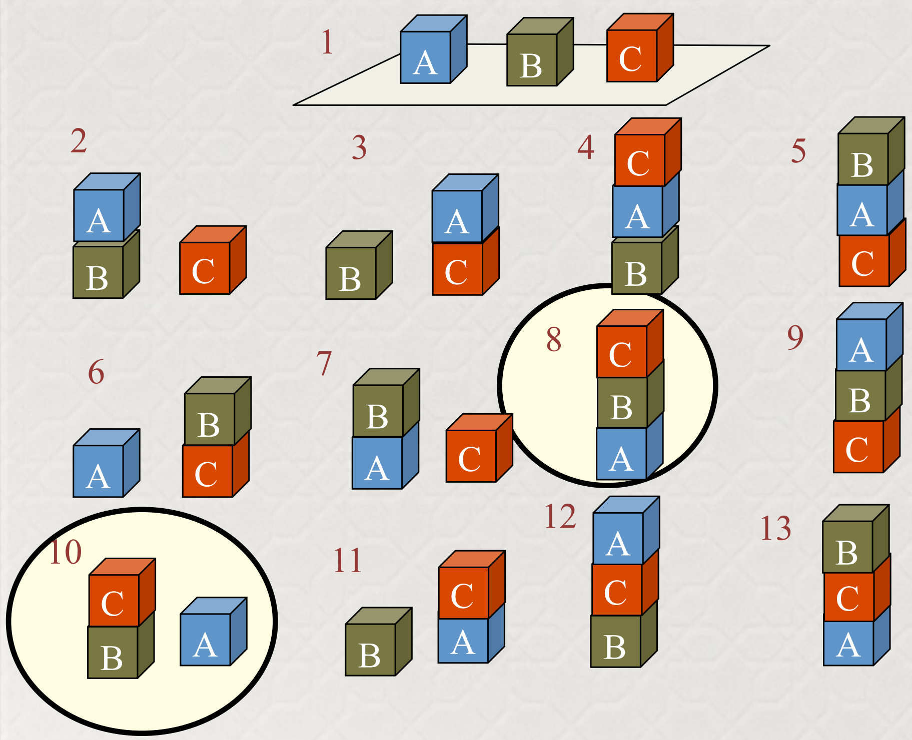
SIM

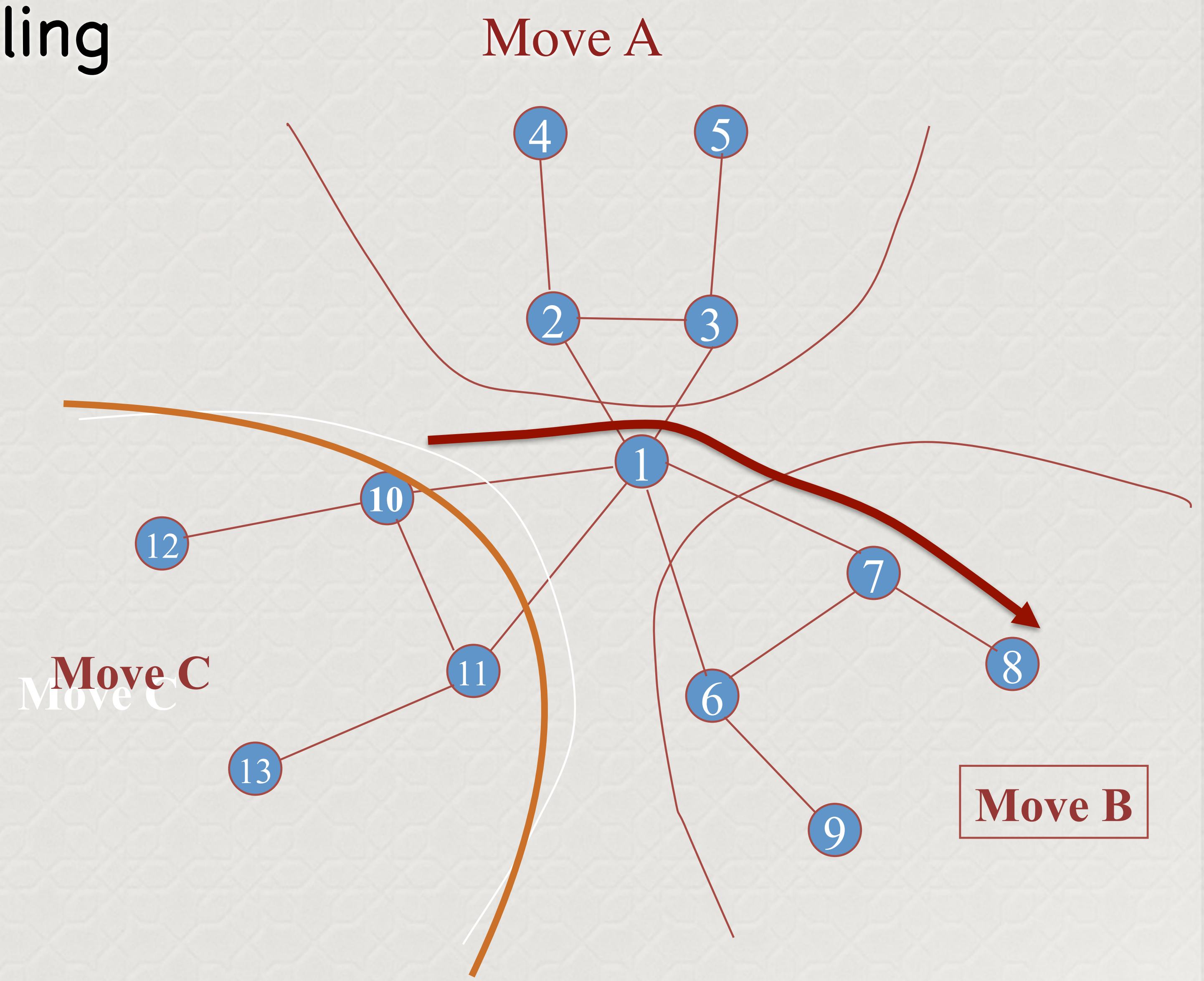


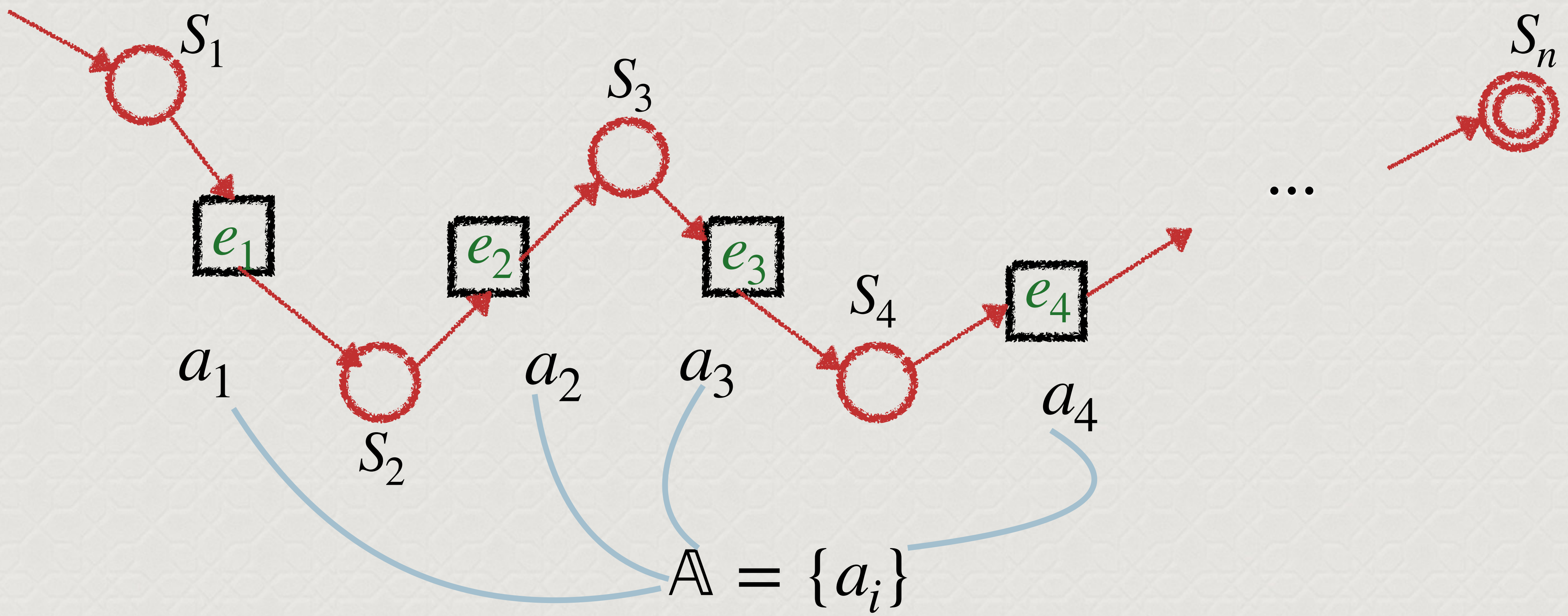
About a state-space planner (STRIPS)	
About actions, its admissibility and effects in the environment	About the domain, preferences and states

KEPS-Knowledge Engineering for Planning and Scheduling

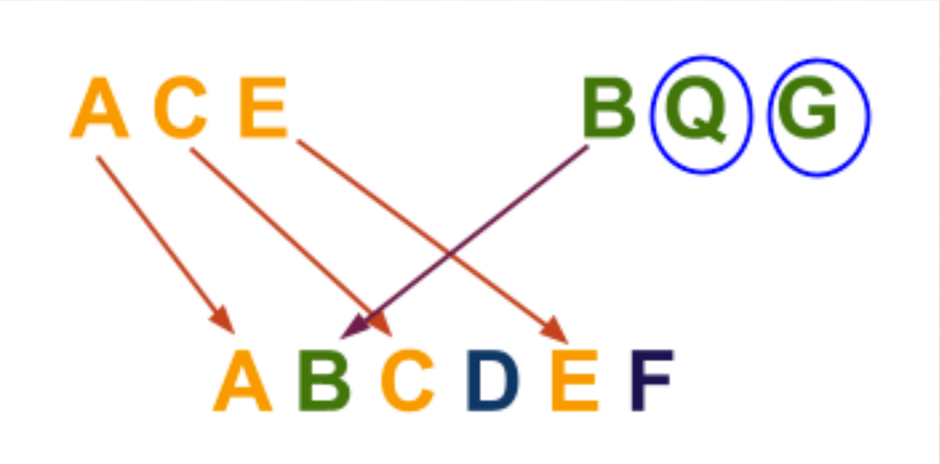
About a state-space planner (STRIPS)	
About actions, its admissibility and effects in the environment	About the domain, preferences and states







- Apesar de geral o algoritmo não é completo, e pode gerar a anomalia de Sussman;
- Não há nenhuma garantia de convergência, portanto, uma solução para o plano pode divergir;
- O algoritmo não é otimizante.



What Does Interleaving Mean?

Interleaving is a process or methodology to make a system more efficient, fast and reliable by arranging data in a noncontiguous manner. There are many uses for interleaving at the system level, including:

- Storage: As hard disks and other storage devices are used to store user and system data, there is always a need to arrange the stored data in an appropriate way.
- Error Correction: Errors in data communication and memory can be corrected through interleaving.
- Multi-Dimensional Data Structures

Interleaving is also known as sector interleave.

O problema do interleaving não é privilégio do problema modelo do mundo de blocos, mas uma limitação do algoritmo STRIPS. Realça também a existência da ordenação parcial entre as ações.

Online Learning of Robot Soccer Free Kick Plans Using a Bandit Approach

Juan Pablo Mendoza
Carnegie Mellon University

Reid Simmons
Carnegie Mellon University

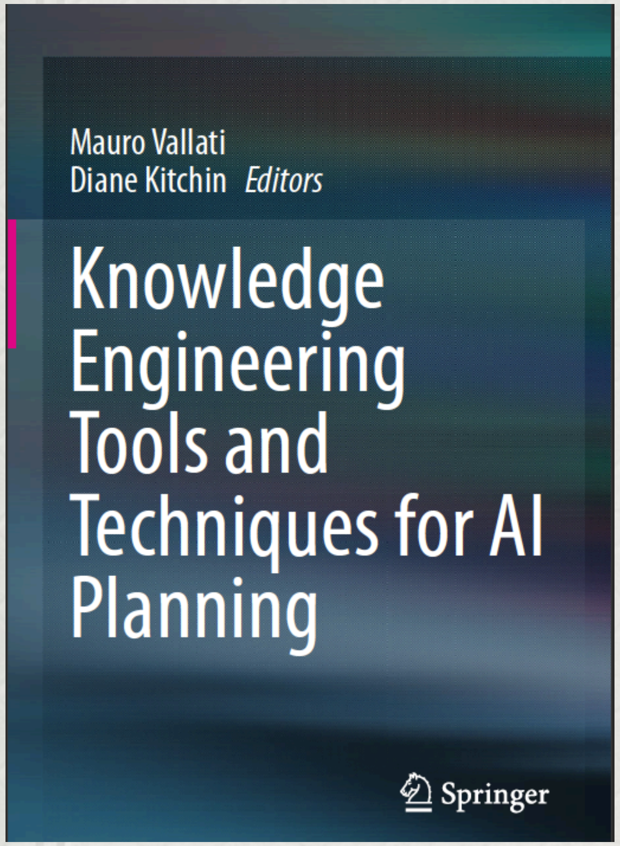
Manuela Veloso
Carnegie Mellon University

PDF

Published
2016-03-30

Abstract

This paper presents an online learning approach for teams of autonomous soccer robots to select free kick plans. In robot soccer, free kicks present an opportunity to execute plans with relatively controllable initial conditions. However, the effectiveness of each plan is highly dependent on the adversary, and there are few free kicks during each game, making it necessary to learn online from sparse observations. To achieve learning, we first greatly reduce the planning space by framing the problem as a contextual multi-armed bandit problem, in which the actions are a set of pre-computed plans, and the state is the position of the free kick on the field. During execution, we model the reward function for different free



Chapter 3
Formal Knowledge Engineering for Planning: Pre and Post-Design Analysis

Jose Reinaldo Silva, Javier Martinez Silva, and Tiago Stegun Vaquero

Abstract The interest and scope of the area of autonomous systems have been steadily growing in the last 20 years. Artificial intelligence planning and scheduling is a promising technology for enabling intelligent behavior in complex autonomous systems. To use planning technology, however, one has to create a knowledge base from which the input to the planner will be derived. This process requires advanced knowledge engineering tools, dedicated to the acquisition and formulation of the knowledge base, and its respective integration with planning algorithms that reason about the world to plan intelligently. In this chapter, we shortly review the existing knowledge engineering tools and methods that support the design of the problem and domain knowledge for AI planning and scheduling applications (AI P&S). We examine the state-of-the-art tools and methods of knowledge engineering for planning & scheduling (KEPS) in the context of an abstract design process for acquiring, formulating, and analyzing domain knowledge. Planning quality is associated with requirements knowledge (pre-design) which should match properties of plans (post-design). While examining the literature, we analyze the design phases that have not received much attention, and propose new approaches to that, based on theoretical analysis and also in practical experience in the implementation of the system itSIMPLE.

Keywords Planning design · Post-design analysis · Planning automation · Automation by planning

J. R. Silva (✉)
Escola Politécnica, Universidade de São Paulo, São Paulo, SP, Brazil
e-mail: reinaldo@usp.br; http://dlab.poli.usp.br

J. M. Silva
Centro Universitario da FEI, São Bernardo do Campo, SP, Brazil
e-mail: jmartinez@fei.edu.br

T. S. Vaquero
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA
e-mail: tiago.stegun.vaquero@jpl.nasa.gov

 47
M. Vallati, D. Kitchin (eds.), *Knowledge Engineering Tools and Techniques for AI Planning*, https://doi.org/10.1007/978-3-030-38561-3_3

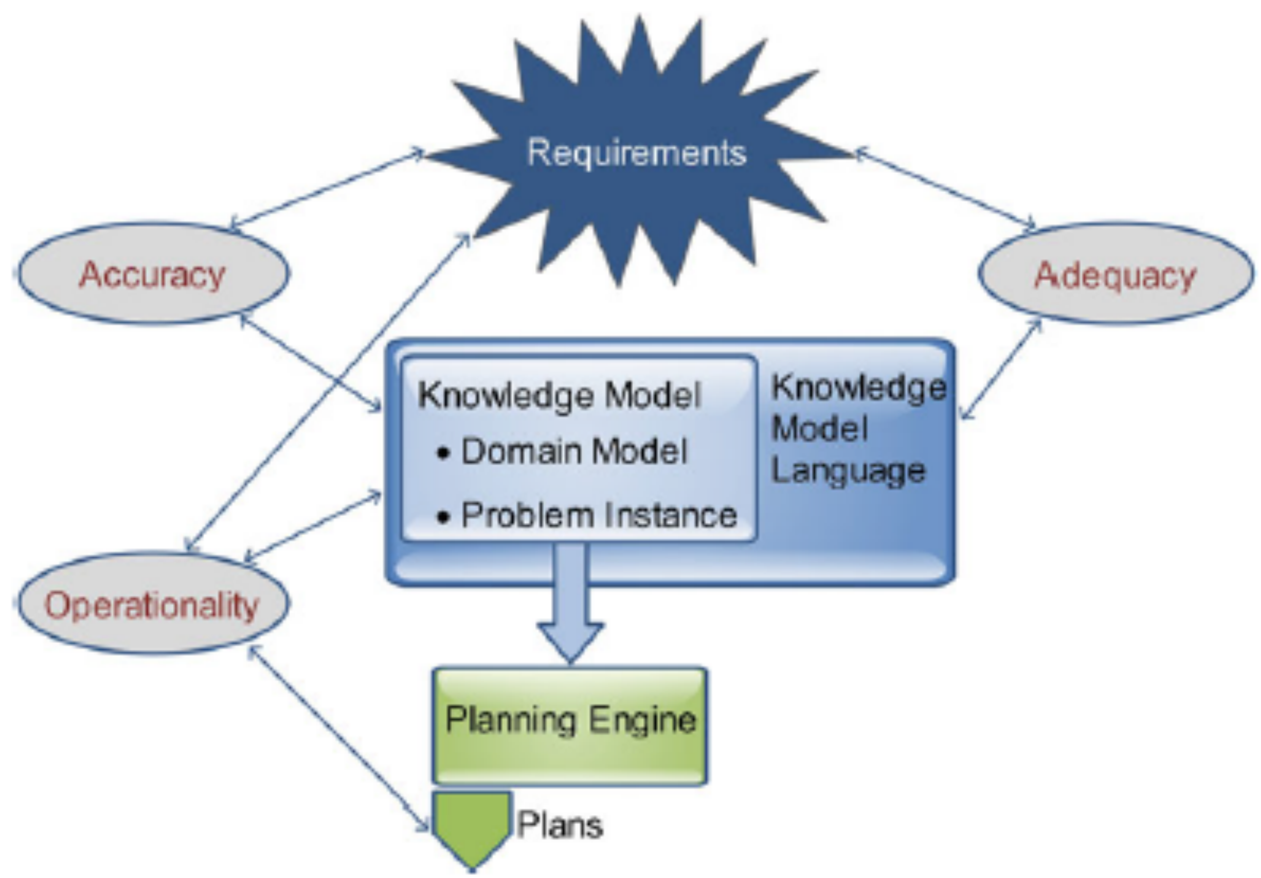


Fig. 3.1 PDM (planning domain modeling properties) basic properties as defined by LeeMc-Cluskey et al. and discussed later in McCluskey [27]

KEPS tools	Design Phases and Properties										
	Design Phases						Domain Independent	Planner Independent	Intended Expert	
	1	2	3	4	5	6			Domain	Planning
O-Plan		✓				≈	✓			✓
SIPE		✓					✓			✓
GIPO		✓	✓	✓	✓	✓	✓	≈	≈	✓
itSIMPLE	✓	✓	≈	✓	✓	✓	✓	✓	✓	✓
EUROPA		✓	✓		✓	✓	✓			✓
ModPlan		✓	≈		✓	✓	✓			✓
VIZ		✓		✓			✓		✓	
TIM			✓				✓			✓
DISCOPLAN			✓				✓	≈		✓
RSA			✓				✓			✓
RedOp			✓				✓	✓		✓
VAL						✓	✓	✓		✓
PlanWorks						✓	✓			✓
MrSPOCK						✓			✓	
JABBAH		✓		✓	✓	✓			✓	
PORSCE II				✓	✓			✓	✓	
CoastWatch						✓			✓	
FlowOpt		✓	✓		✓	✓			✓	
MARIO		✓		✓	✓				✓	
SLAF		✓					✓	✓		✓
LAMP		✓					✓	✓		✓
LOCM		✓					✓	≈		✓
ARMS		✓					✓	✓		✓
Bonasso & Boddy, 2010	✓			✓					✓	✓
Bouillet et al., 2007		✓					✓			✓
Fox & Long, 1999			✓				✓			✓
Crawford et al., 1996			✓				✓			✓
Fernández et al., 2009				✓				✓	✓	
Giuliano & Johnston, 2010						✓			✓	
Myers, 2006						✓	✓		✓	
Chrpa et al., 2012a						✓	✓	✓		✓
Chrpa et al., 2012b			✓				✓	✓		✓
Nakhost & Muller, 2010						✓	✓	✓		✓

Fig. 3.2 Summary of available tools and methods in the Knowledge Engineering For Planning and Scheduling literature, Design phases: (1) Requirements, (2) Knowledge Modeling, (3) Model Analysis, (4) Model Preparation, (5) Plan/Schedule Synthesis, (6) Plan/Schedule Analysis and Post-Design. *Checkmark* means that the feature is present in the tool, *approx* means that it is to some degree present, and *blank* means that it is not present

Existem vários blocos de algoritmos para planejamento inteligente:

- *os algoritmos baseados em busca (forward e backward) e no uso de heurísticas;*
- *Os algoritmos baseados em métodos não lineares;*
- *Os algoritmos associados a métodos de apoio a decisão.*

Forward and Backward planning

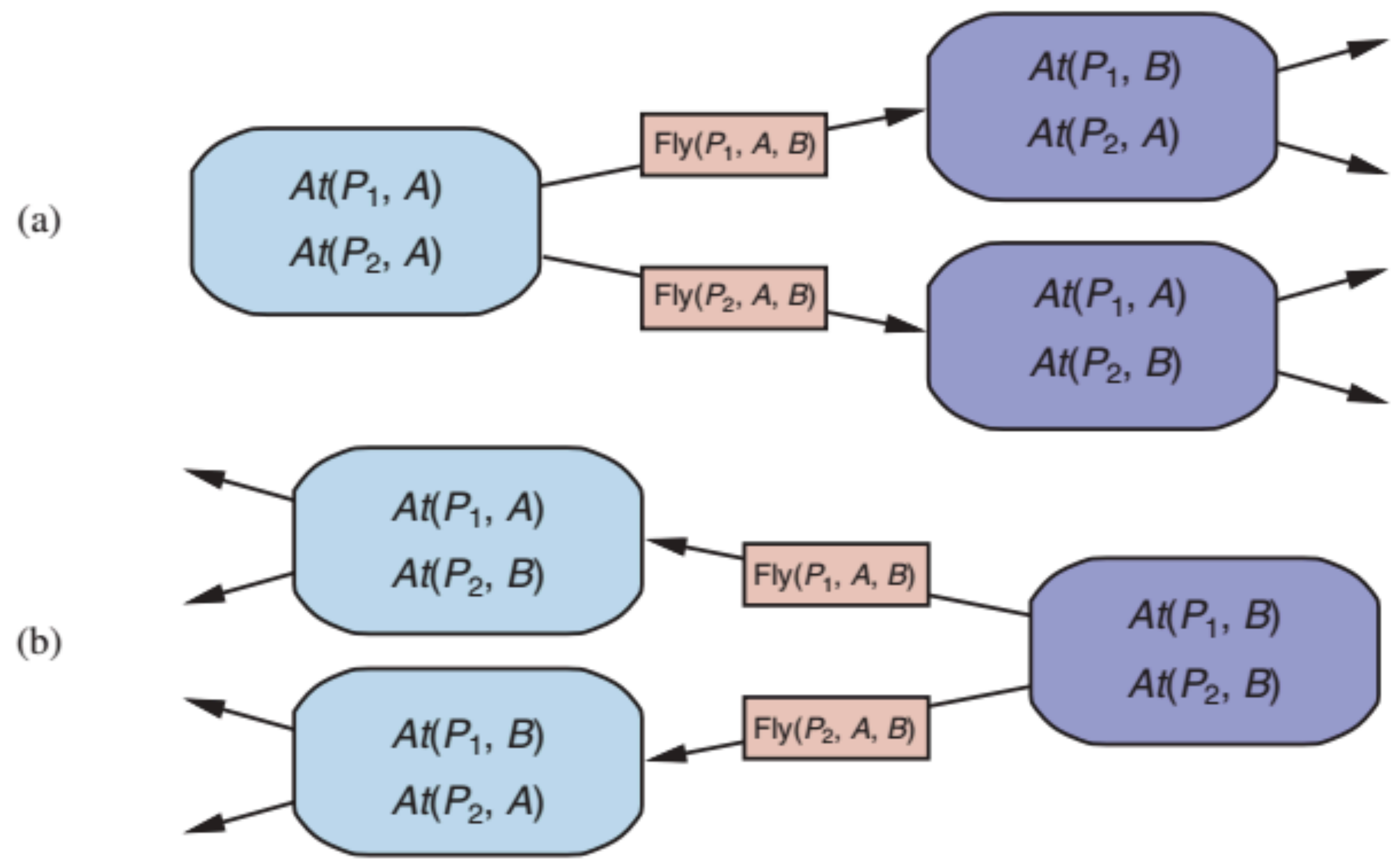
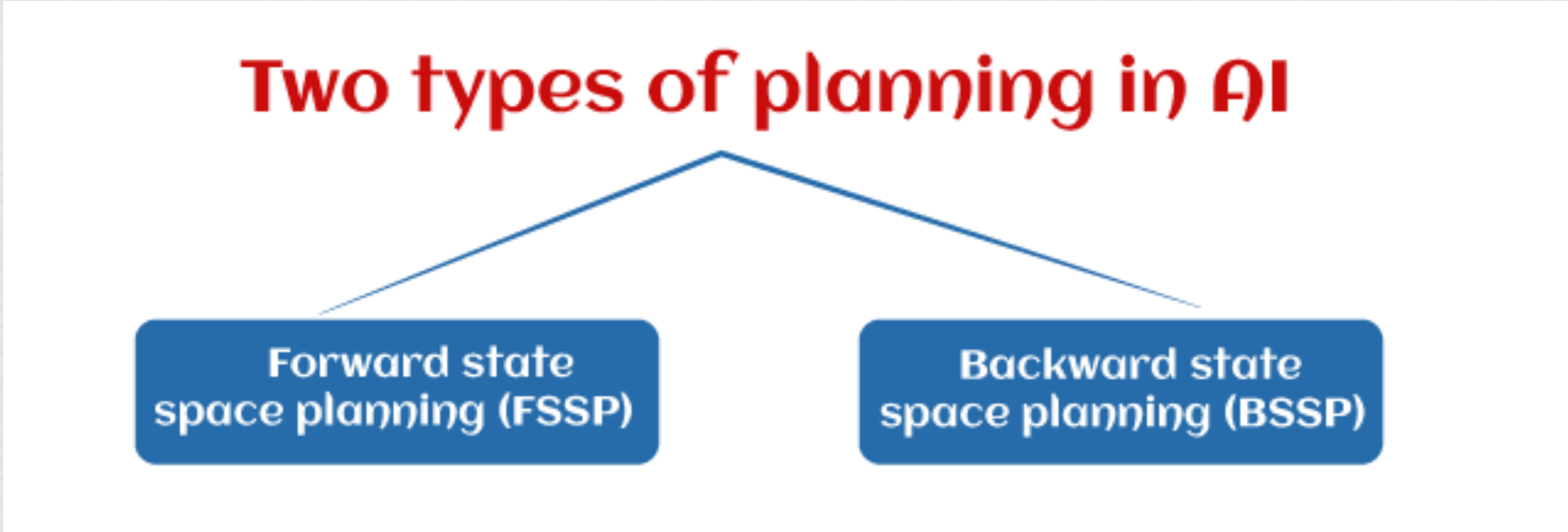
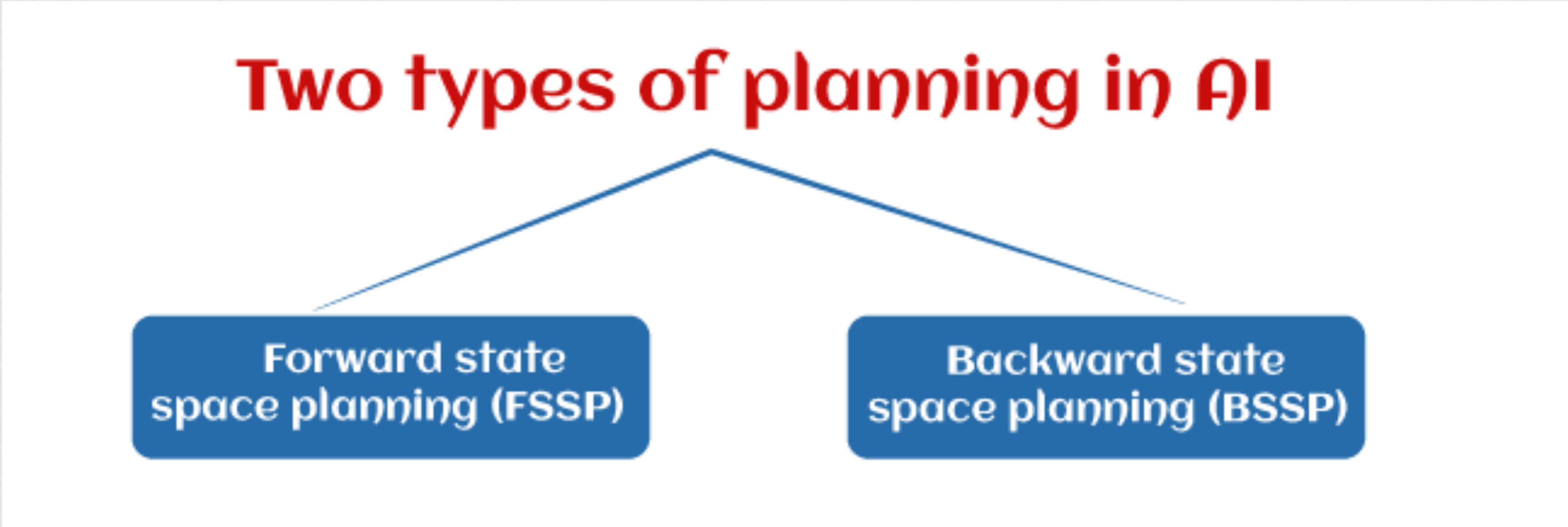


Figure 11.5 Two approaches to searching for a plan. (a) Forward (progression) search through the space of ground states, starting in the initial state and using the problem's actions to search forward for a member of the set of goal states. (b) Backward (regression) search through state descriptions, starting at the goal and using the inverse of the actions to search backward for the initial state.



1. Forward State Space Planning (FSSP)

- **Disadvantage**: Large branching factor
- **Advantage**: The algorithm is Sound



2. Backward State Space Planning (BSSP)

- **Disadvantages**: not sound algorithm (sometimes inconsistency can be found)
- **Advantage**: Small branching factor (much smaller than FSSP)

Existem vários blocos de algoritmos para planejamento inteligente:

- *os algoritmos baseados em busca (forward e backward) e no uso de heurísticas;*
- *Os algoritmos baseados em métodos não lineares;*
- *Os algoritmos associados a métodos de apoio a decisão.*

Fast Planning Through Planning Graph Analysis*

Avrim L. Blum
School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213-3891
avrim@cs.cmu.edu

Merrick L. Furst
School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213-3891
mxf@cs.cmu.edu

(Final version in *Artificial Intelligence*, 90:281–300, 1997)

Abstract

We introduce a new approach to planning in STRIPS-like domains based on constructing and analyzing a compact structure we call a Planning Graph. We describe a new planner, Graphplan, that uses this paradigm. Graphplan always returns a shortest-possible partial-order plan, or states that no valid plan exists.

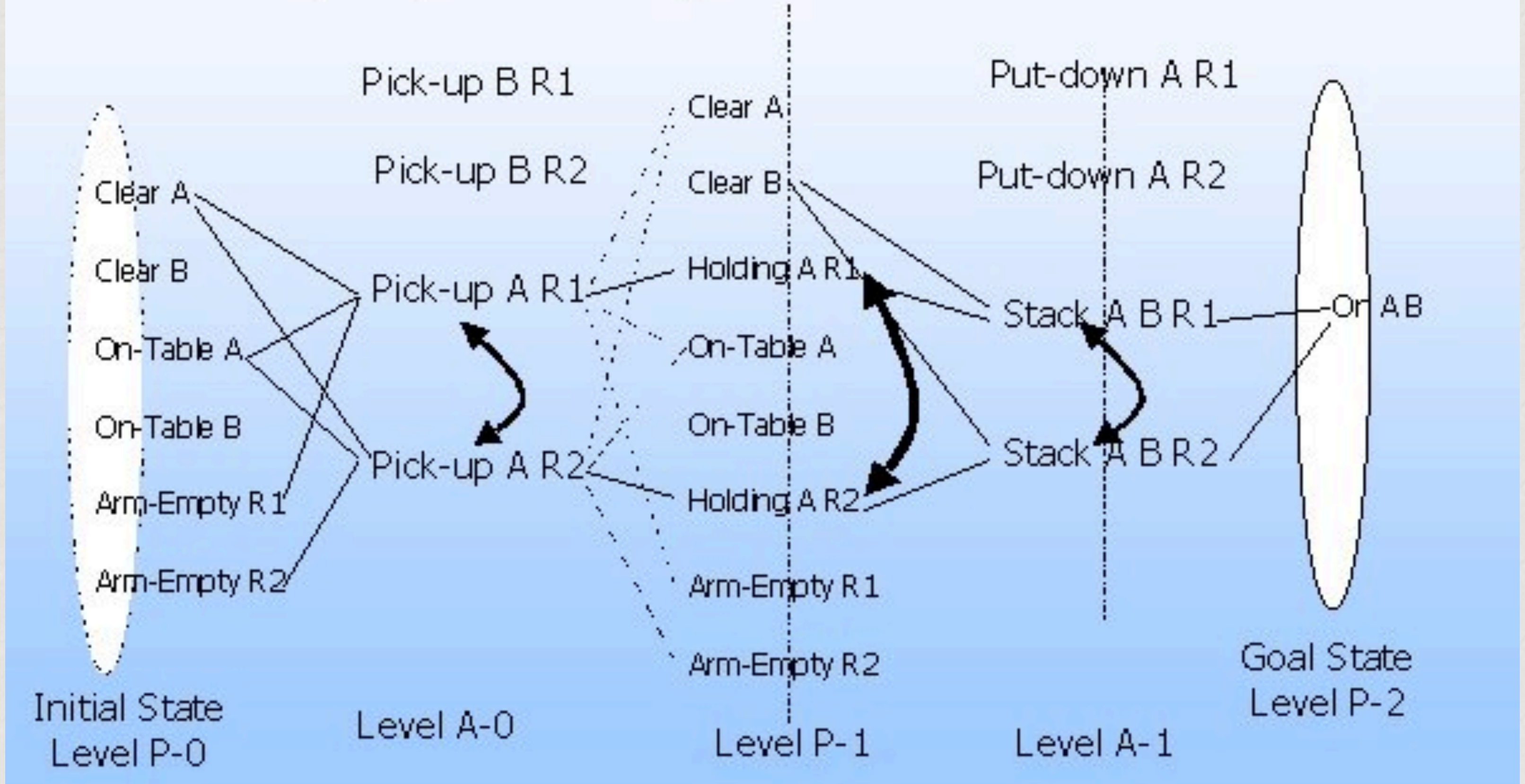
We provide empirical evidence in favor of this approach, showing that Graphplan outperforms the total-order planner, Prodigy, and the partial-order planner, UCPOP, on a variety of interesting natural and artificial planning problems. We also give empirical evidence that the plans produced by Graphplan are quite sensible. Since searches made by this approach are fundamentally different from the searches of other common planning methods, they provide a new perspective on the planning problem.

Keywords: General Purpose Planning, STRIPS Planning, Graph Algorithms, Planning Graph Analysis.

1 Introduction

In this paper we introduce a new planner, Graphplan, which plans in STRIPS-like domains. The algorithm is based on a paradigm we call Planning Graph Analysis. In this approach, rather than immediately embarking upon a search as in standard planning methods, the algorithm instead begins by explicitly constructing a compact structure we call a *Planning Graph*. A Planning Graph encodes the planning problem in such a way that many useful constraints inherent in the problem become explicitly available to reduce the amount of search needed. Furthermore, Planning Graphs can be constructed quickly: they have poly-

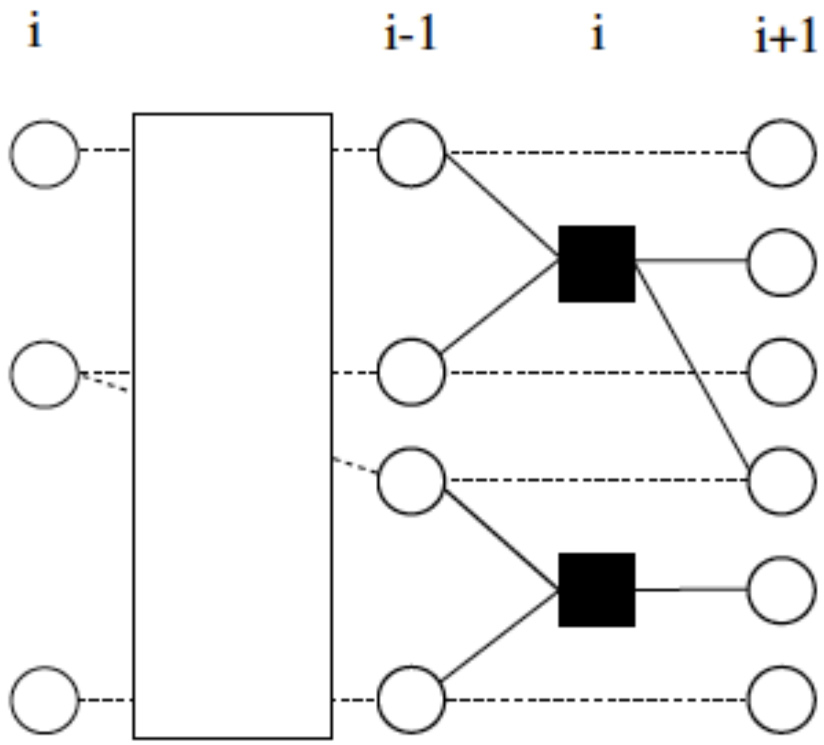
Graphplan Algorithm



Implementação de otimizações de performance no GraphPlan

Fernando Fuentes Giroleti e Ramon Fraga Pereira

O Grafo de Planejamento gerado pelo GraphPlan é estruturado em níveis, onde cada nível possui nodos de um determinado tipo. Estes nodos são de dois tipos: os que representam proposições e os que representam ações. Um nodo proposição representa uma parte do estado e um nodo ação representa modificações possíveis nas proposições. Os níveis do grafo estão divididos em níveis de proposições e de ações alternando de tipo a cada nível. Níveis pares são proposições, enquanto ímpares são as ações, sendo que o nível *0* é sempre composto pelas proposições que representam o estado inicial do problema[3].

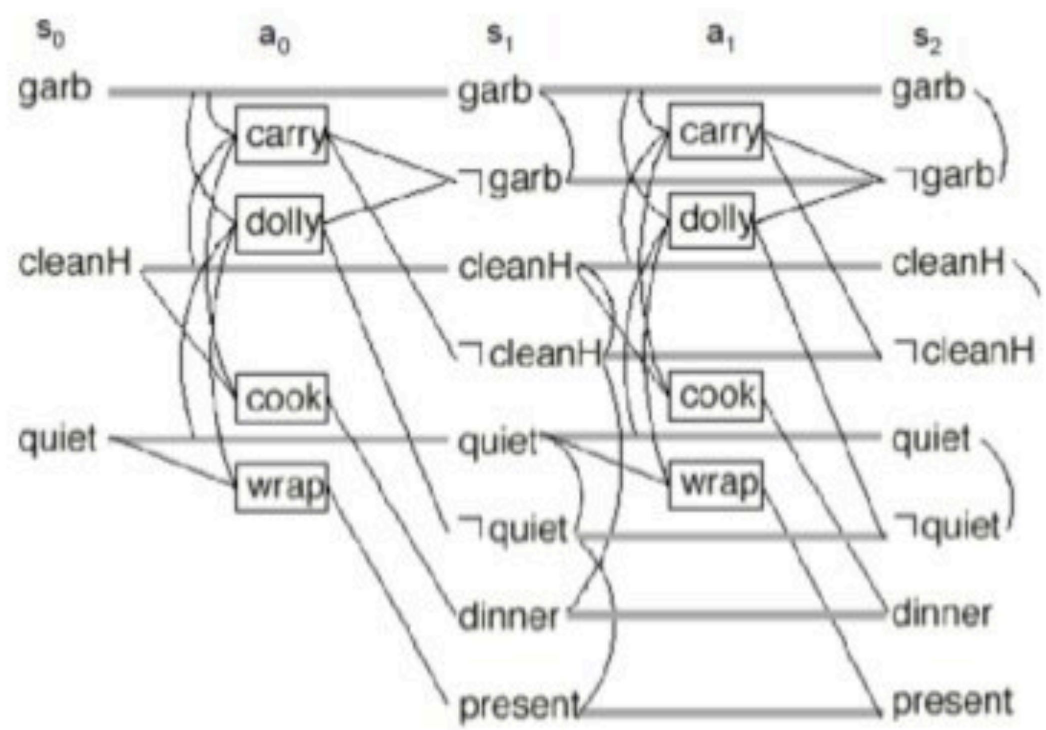


Basic idea

- Construct a graph that encodes constraints on possible plans
- Use this "planning graph" to constrain search for a valid plan:
  - If valid plan exists, it's a subgraph of the planning graph
- Planning graph can be built for each problem in polynomial time

GRAPHPLAN

- Extração do plano:
  G:{¬garb, dinner, present}



https://www.cs.cmu.edu/afs/cs.cmu.edu/usr/avrim/Planning/Graphplan/

Index of /afs/cs.cmu.edu/usr/avrim/Planning/Graphplan

Name	Last modified	Size	Description
Parent Directory		-	
BW.tar.gz	1995-12-11 10:30	5.5K	
README	1997-07-15 16:34	9.5K	
Source.tar.gz	1997-08-29 10:18	55K	
Source/	1997-06-11 16:38	-	
blocks_facts.suss	1994-07-22 11:19	215	
blocks_facts4	1994-07-26 14:14	257	
blocks_facts5	1998-05-13 14:45	314	
blocks_facts6	1998-01-20 15:34	370	
blocks_facts7	1998-05-13 14:45	426	
blocks_facts8	1998-01-20 15:24	482	
blocks_facts_impossible	1996-09-20 15:22	221	
blocks_facts_shuffle	1994-08-02 16:45	370	
blocks_ops	1995-01-23 11:04	1.0K	
blocks_ops.lisp	1994-07-26 14:09	1.1K	
d1s4/	1995-08-31 13:38	-	
fixit_facts1	1995-01-16 09:39	560	
fixit_facts2	1995-01-16 09:39	558	
fixit_ops	1995-08-31 16:28	2.1K	
foo	1998-05-13 14:47	3.2K	
fridge_facts1	1995-09-04 11:27	449	
fridge_facts2	1995-09-04 11:28	464	
fridge_ops	1995-09-04 11:30	2.0K	
gpoutput.ps	1999-07-21 12:52	55K	
graphplan	1995-08-31 16:41	168K	

GraphPlan: Story Generation by Planning with Event Graph

Hong Chen[1,3], Raphael Shu[1], Hiroya Takamura[2,3], Hideki Nakayama [1]
[1]The University of Tokyo, [2]Tokyo Institute of Technology,
[3]National Institute of Advanced Industrial Science and Technology, Japan
{chen, nakayama}@nlab.ci.i.u-tokyo.ac.jp, raphael@uaca.com, takamura.hiroya@aist.go.jp

Abstract

Story generation is a task that aims to automatically produce multiple sentences to make up a meaningful story. This task is challenging because it requires high-level understanding of semantic meaning of sentences and causality of story events. Naive sequence-to-sequence models generally fail to acquire such knowledge, as the logical correctness can hardly be guaranteed in a text generation model without the strategic planning. In this paper, we focus on planning a sequence of events assisted by event graphs, and use the events to guide the generator. Instead of using a sequence-to-sequence model to output a storyline as in some existing works, we propose to generate an event sequence by walking on an event graph. The event graphs are built automatically based on the corpus. To evaluate the proposed approach, we conduct human evaluation both on event planning and story generation. Based on large-scale human annotation results, our proposed approach is shown to produce more logically correct event sequences and stories.

1 Introduction

Narrative Intelligence (Mateas and Sengers 2003) is one form of Humanistic Artificial Intelligence that requires the system to organize, comprehend, and reason about narratives and produce meaningful responses. Story generation tasks can be considered as a test bed for examining whether a system develops a good understanding of the narratives.

Other than leaving the model to output random sentences,

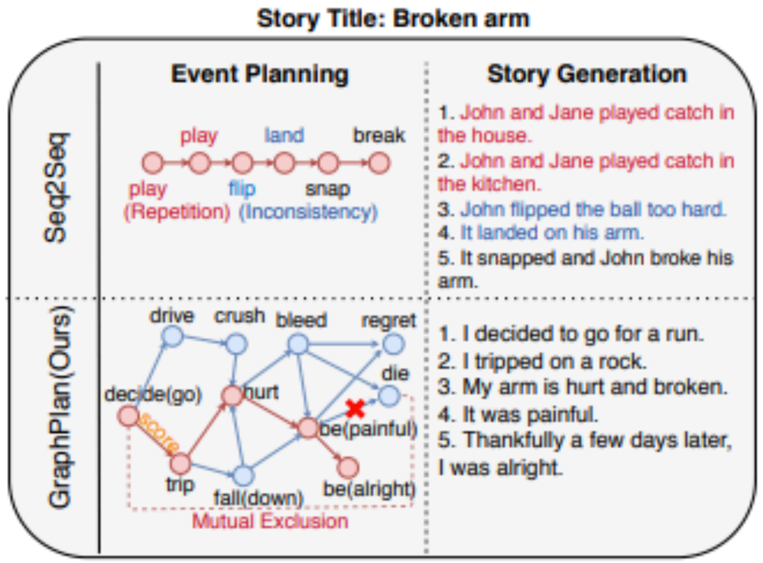


Figure 1: Comparison between sequence-to-sequence model and GraphPlan (ours). Two problems in the sequence-to-sequence model when generating events: Repetition and Logical Inconsistency. Repeated words (e.g., play) in the storyline result in the repeated sentences in the generated stories. Besides, The logic between "land" and "snap" lacks causality, thus generating incoherent stories. On the contrary, our GraphPlan method does not rely on any language model, it applies beam search on the event graph based on a well-designed score function. The mutually exclusive set further ensures the global logical consistency for the planned

applied sciences

MDPI

Review

A Survey of Planning and Learning in Games

Fernando Fradique Duarte [1,*], Nuno Lau [2], Artur Pereira [2] and Luis Paulo Reis [3]

[1] Institute of Electronics and Informatics Engineering of Aveiro (IEETA), University of Aveiro, 3810-193 Aveiro, Portugal

[2] Department of Electronics, Telecommunications and Informatics, University of Aveiro, 3810-193 Aveiro, Portugal; nunolau@ua.pt (N.L.); artur@ua.pt (A.P.)

[3] Faculty of Engineering, Department of Informatics Engineering, University of Porto, 4099-002 Porto, Portugal; lpreis@fe.up.pt

* Correspondence: fjosefradique@ua.pt

Received: 29 May 2020; Accepted: 27 June 2020; Published: 30 June 2020

check for updates

Abstract: In general, games pose interesting and complex problems for the implementation of intelligent agents and are a popular domain in the study of artificial intelligence. In fact, games have been at the center of some of the most well-known achievements in artificial intelligence. From classical board games such as chess, checkers, backgammon and Go, to video games such as Dota 2 and StarCraft II, artificial intelligence research has devised computer programs that can play at the level of a human master and even at a human world champion level. Planning and learning, two well-known and successful paradigms of artificial intelligence, have greatly contributed to these achievements. Although representing distinct approaches, planning and learning try to solve similar problems and share some similarities. They can even complement each other. This has led to research on methodologies to combine the strengths of both approaches to derive better solutions. This paper presents a survey of the multiple methodologies that have been proposed to integrate planning and learning in the context of games. In order to provide a richer contextualization, the paper also presents learning and planning techniques commonly used in games, both in terms of their theoretical foundations and applications.

Keywords: planning; learning; artificial intelligence; planning and learning; games

Perguntas?