AI Planning for Autonomy

1. Short Introduction to Automated (AI) Planning

Chris Ewin

University of Melbourne

Semester 1, 2021

With slides by Nir Lipovetzky

Intro:Chris Ewin

I completed my PhD in 2018 at the University of Melbourne

My main research interests are in the area of Automated planning, specifically:

- Knowledge representation
- Planning in rich domains
- Situation Calculus

You can find me on Zoom / Piazza / Email

Contact: cewin@unimelb.edu.au

Intro: Tim Miller

I completed my PhD in 2005 at the University of Queensland.

My main research interests are in the area of artificial intelligence, specifically:

- Decision making in multi-agent environments
- Reasoning about action and knowledge
- Automated planning
- Human-agent interaction and collaboration

Contact: tmiller@unimelb.edu.au

Intro: Tutors

Great Knowledge on COMP90054 topics.

Guang Hu, Chengyuan Zhang, Anam Ahmad Khan, Anubhav Singh, Shima Rashidi

- few former top-students of COMP90054, Phd and Research Students.

Head Tutor: Guang Hu

Contact: ghu1@student.unimelb.edu.au

Course Structure

Weekly Lectures:

- Short videos providing background knowledge (instead of Thursday's lecture)
- Live Zoom lecture (Friday): Q&A, polls, worked examples

Online Review Quiz (starting Week 2)

Assignments:

- Individual early project: Search
- Small individual project: Modeling a planning problem
- Assessed Quiz: Planning
- Final group project: Game competition

Workshops

- Workshops will run from Week 2.
- Mostly online workshops, with the following in-person workshops:
  - Mon, 13:15-14:15. Location: PAR-Elec. Engineering-123
  - Tue, 11:00-12:00. Location: PAR-Elec. Engineering-123
  - Thu, 14:15-15:15. Location: PAR-Elec. Engineering-123
  - Fri, 14:15-15:15, Location: PAR-Elec. Engineering-123
- All online workshops will have a link in the 'Zoom' section on the LMS.

Outline of the Course

1. Introduction to AI and (AI) Planning
2. Classical Planning as Heuristic Search and Width-Based Search
3. Beyond Classical Planning:
   - *Factored-Model-Free, Non Determinism, Uncertainty, Soft goals, Plan Recognition, Epistemic (social) Planning, Path-Planning, Control*
4. Reinforcement Learning: Learning through Experience
5. Multi agent Planning
6. Hot/Latest exciting discussions on AI Ethics

Acknowledgment:

$\rightarrow$ Slides based on earlier courses by Hector Geffner, Joerg Hoffmann, and Carmel Domshlak

Things you Should Know to Enjoy the Course

1. Algorithms such as Dynamic Programming
2. Basic Set Theory and Propositional Logic
3. Probabilistic Theory such as Conditional Probabilities
4. Python, start this week the recommended Tutorial (LMS->Resources)

and importantly, you need to stay up to date reviewing and understanding slides, as most lectures build up on previous knowledge

If you don't understand...have a question...

Origins of AI: Darmouth 1956

Articles

A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence

August 31, 1955

John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon

■ The 1956 Dartmouth summer research project on artificial intelligence was initiated by this August 31, 1955 proposal, authored by John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon. The original typescript consisted of 17 pages plus a title page. Copies of the typescript are housed in the archives at Dartmouth College and Stanford University. The first 5 papers state the proposal, and the remaining pages give qualifications and interests of the four who proposed the study. In the interest of brevity, this article reproduces only the proposal itself, along with the short autobiographical statements of the proposers.

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

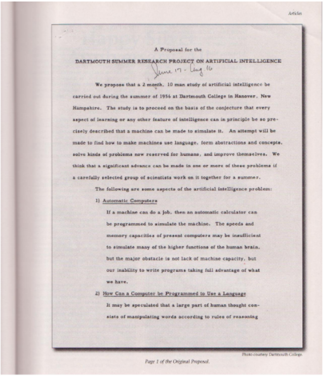
The following are some aspects of the artificial intelligence problem:

1. Automatic Computers

If a machine can do a job, then an automatic calculator can be programmed to simulate the machine. The speeds and memory capacities of present computers may be insufficient to simulate many of the higher functions of the human brain, but the major obstacle is not lack of machine capacity, but our inability to write programs taking full advantage of what we have.

2. How Can a Computer be Programmed to Use a Language

It may be speculated that a large part of human thought consists of manipulating words according to rules of reasoning and rules of conjecture. From this point of view, forming a generalization consists of admitting a new



Articles

A Proposal for the

DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE

June 17 - Aug 16

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

The following are some aspects of the artificial intelligence problem:

1) Automatic Computers

If a machine can do a job, then an automatic calculator can be programmed to simulate the machine. The speeds and memory capacities of present computers may be insufficient to simulate many of the higher functions of the human brain, but the major obstacle is not lack of machine capacity, but our inability to write programs taking full advantage of what we have.

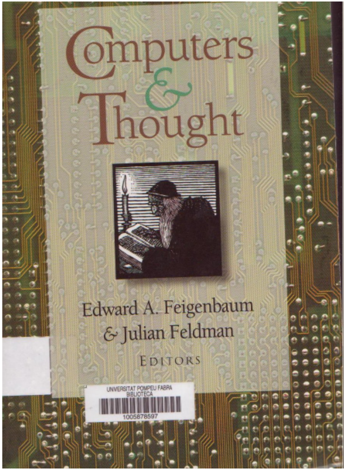
2) How Can a Computer be Programmed to Use a Language

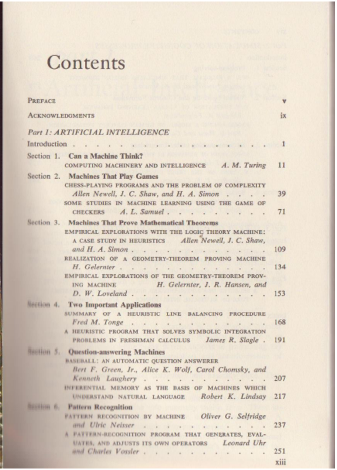
It may be speculated that a large part of human thought consists of manipulating words according to rules of reasoning

Photo courtesy Dartmouth College.

Page 1 of the Original Proposal.

The proposal (for the meeting) is to proceed on the basis of the conjecture that every aspect of . . . intelligence can in principle be so precisely described that a machine can be made to simulate it





Contents

xiii

An early collection of AI papers and programs for playing chess and checkers, proving theorems in logic and geometry, planning, etc.

Many of the key AI contributions in 60's, 70's, and early 80's had to to with programming and the representation of knowledge in programs:

- Lisp (Functional Programming)
- Prolog (Logic Programming)
- Rule-based Programming
- 'Expert Systems' Shells and Architectures

AI Methodology: Theories as Programs

For writing an AI dissertation in the 60's, 70's and 80's, it was common to:

- pick up a task and domain X
- analyze/introspect/find out how task is solved
- capture this reasoning in a program

The dissertation was then

- a theory about X (scientific discovery, circuit analysis, computational humor, story understanding, etc), and
- a program implementing the theory, tested over a few examples.

Many great ideas came out of this work . . . but there was a problem . . .

Methodology Problem

→ Theories expressed as programs cannot be proved wrong: when a program fails, it can always be blamed on 'missing knowledge'

Three approaches to this problem

- narrow the domain (expert systems)
    - problem: lack of generality
- accept the program is just an illustration, a demo
    - problem: limited scientific value
- fill up the missing knowledge (intuition, commonsense)
    - problem: not successful so far

AI Winter: the 80's

$\rightarrow$ The knowledge-based approach reached an **impasse** in the 80's, a time also of debates and controversies:

- **Good Old Fashioned AI** is 'rule application' but intelligence is not (Haugeland)

Many criticisms of mainstream AI partially valid then; less valid now.

AI 90's - 2021

Formalization of AI techniques and increased use of mathematics. Recent issues of AIJ, JAIR, AAAI or IJCAI shows papers on:

1. SAT and Constraints
2. Search and Planning
3. Probabilistic Reasoning
4. Probabilistic Planning
5. Inference in First-Order Logic
6. Machine Learning
7. Natural Language
8. Vision and Robotics
9. Multi-Agent Systems

$\rightarrow$ Areas 1 to 4 often deemed about techniques, but more accurate to regard them as models and solvers.

Solver Example

$$\textit{Problem} \Longrightarrow \boxed{\textit{Solver}} \Longrightarrow \textit{Solution}$$

Example:

- **Problem:** The age of John is 3 times the age of Peter. In 10 years, it will be only 2 times. How old are John and Peter?
- **Expressed as:** J = 3P ; J + 10 = 2(P + 10)
- **Solver:** Gauss-Jordan (Variable Elimination)
- **Solution:** P = 10 ; J = 30

Solver is **general** as deals with any problem expressed as an instance of **model** Linear Equations Model, however, is **tractable**, AI models are not . . .

AI Solver

$$\textit{Problem} \Longrightarrow \boxed{\textit{Solver}} \Longrightarrow \textit{Solution}$$

- The basic models and tasks include
  - **Constraint Satisfaction/SAT:** find state that satisfies constraints
  - **Planning Problems:** find action sequence that produces desired state
  - **Planning with Feedback:** find strategy for producing desired state
- Solvers for these models are **general**; not tailored to specific instances
- All of these models are **intractable**, and some extremely powerful (POMDPs)
- The challenge is mainly computational: **how to scale up**
- For this, solvers must **recognize and exploit structure** of the problems
- Methodology is **empirical**: benchmarks and competitions

SAT and CSPs

- **SAT**: determine if there is a **truth assignment** that satisfies a set of clauses

$$x \vee \neg y \vee z \vee \neg w \vee \ldots \tag{1}$$

- Problem is NP-Complete, which in practice means worst-case behavior of SAT algorithms is **exponential** in number of variables ($2^{100} = 10^{30}$)
- Yet current SAT solvers manage to solve problems with **thousands of variables** and clauses, and used widely (circuit design, verification, planning, etc)
- **Constraint Satisfaction Problems (CSPs)** generalize SAT by accommodating non-boolean variables as well, and constraints that are not clauses
- Key is **efficient (poly-time) inference** in every node of search tree: **unit resolution, conflict-based learning**, ...
- Many other ideas **logically possible**, but **do not work** (don't scale up): pure search, pure inference, etc.

Classical Planning Model

- Planning is the **model-based approach** to autonomous behavior,
- A system can be in one of many **states**
- States assign **values** to a set of **variables**
- **Actions** change the values of certain variables
- **Basic task**: find **action sequence** to drive **initial** state into **goal** state

$$\mathit{Model} \Longrightarrow \boxed{\mathit{Planner}} \Longrightarrow \mathit{Action\ Sequence}$$

- **Complexity**: NP-hard; i.e., exponential in number of vars in **worst case**
- **Planner** is generic; it should work on any domain no matter what variables are about

Why do we need such an AI?

- **Chess**: 2 player zero-sum game
- **Music/Speech Recognition**
- **Recommender systems**
- **Medical Diagnosis**: decission support systems
- **Self-driven car**
- **Playing Atari Games** Deep Learning
- ...


AAAI
Association for the Advancement of Artificial Intelligence
Poster development supported in part by
Research
YAHOO!
AI LAB
MEDICAL CENTER
BOOKS & COFFEE
Optimizing Paths & Flows
AI in Art
AI in Music
AI & Creative Expression
Planning
Plan Recognition
Autonomous Vehicles & Safety
Robot Guides & Assistants
Augmenting Cognition
Humanoid Robots
Intelligent Tutoring
Ubiquitous Computing
Social Computing
Gesture Recognition
Multimodal Interfaces
Mixed-Initiative Collaboration
Autonomous Space Exploration
AI and Preferences, Media & Entertainment
Home Robotics
Robots for Education
Robotic Surgery
Diagnosis
Drug Design
Scientific Discovery
Assistive Technology
Search & Retrieval
Security & Privacy
User Modeling
Communications Triage
Machine Translation
Vehicle Navigation
Ecocomputing
Handwriting & Sketch Recognition
Recommender Systems & Question Answering
Descartes
Aristotle
Leibniz
Whitehead
Lovelace
Russell
Turing
See the AI timeline and more at
www.aaai.org/AILandscape
The AI Landscape
David Leake, Indiana University, Poster Development Committee Chair
Poster Design: Giacomo Marchesi; www.GiacomoMarchesi.com

Why do we need such AI Planning?

Settings where greater autonomy required:

- **Space Exploration**: (RAX) first artificial intelligence control system to control a spacecraft without human supervision (1998)
- **Business Process Management**
- **First Person Shooters & Games:** classical planners playing Atari Games
- **Interactive Storytelling**
- **Network Security**
- **Logistics/Transportation/Manufacturing**: Multi-model Transportation, forest fire fighting, PARC printer
- ...

Find out more at Special Interest Group for Applications of AI Planning and Scheduling

Summary: AI and Automated Problem Solving

- A **research agenda** that has emerged in last 20 years: **solvers** for a range of **intractable models**
- **Solvers** unlike other programs are **general** as they do not target individual problems but families of problems (**models**)
- The challenge is **computational**: how to scale up
- Sheer **size of problem** shouldn't be impediment to meaningful solution
- **Structure** of given problem must recognized and **exploited**
- Lots of room for **ideas** but methodology **empirical**
- Consistent **progress**
  - effective inference methods (derivation of h, conflict-learning)
  - islands of tractability (treewidth methods and relaxations)
  - transformations (compiling away incomplete info, extended goals, . . .)