

Planificación Automática

Grupo PLG

Universidad Carlos III de Madrid

IA. 2008-09

Índice

- 1 **Introducción**
- 2 Planificación clásica
- 3 Planificación neoclásica
- 4 Heurística
 - Planificación heurística
 - Redes de tareas jerárquicas
 - Conocimiento de control
 - Aprendizaje automático
- 5 Planificación en el mundo real

Índice

- 1 **Introducción**
- 2 **Planificación clásica**
- 3 Planificación neoclásica
- 4 Heurística
 - Planificación heurística
 - Redes de tareas jerárquicas
 - Conocimiento de control
 - Aprendizaje automático
- 5 Planificación en el mundo real

Índice

- 1 Introducción
- 2 Planificación clásica
- 3 Planificación neoclásica
- 4 Heurística
 - Planificación heurística
 - Redes de tareas jerárquicas
 - Conocimiento de control
 - Aprendizaje automático
- 5 Planificación en el mundo real

Índice

- 1 Introducción
- 2 Planificación clásica
- 3 Planificación neoclásica
- 4 Heurística
 - Planificación heurística
 - Redes de tareas jerárquicas
 - Conocimiento de control
 - Aprendizaje automático
- 5 Planificación en el mundo real

Índice

- 1 Introducción
- 2 Planificación clásica
- 3 Planificación neoclásica
- 4 Heurística
 - Planificación heurística
 - Redes de tareas jerárquicas
 - Conocimiento de control
 - Aprendizaje automático
- 5 Planificación en el mundo real

Índice

- 1 **Introducción**
- 2 Planificación clásica
- 3 Planificación neoclásica
- 4 Heurística
 - Planificación heurística
 - Redes de tareas jerárquicas
 - Conocimiento de control
 - Aprendizaje automático
- 5 Planificación en el mundo real

Índice

- 1 Introducción
- 2 Planificación clásica**
- 3 Planificación neoclásica
- 4 Heurística
 - Planificación heurística
 - Redes de tareas jerárquicas
 - Conocimiento de control
 - Aprendizaje automático
- 5 Planificación en el mundo real

Índice

- 1 Introducción
- 2 Planificación clásica
- 3 Planificación neoclásica**
- 4 Heurística
 - Planificación heurística
 - Redes de tareas jerárquicas
 - Conocimiento de control
 - Aprendizaje automático
- 5 Planificación en el mundo real

Índice

- 1 Introducción
- 2 Planificación clásica
- 3 Planificación neoclásica
- 4 Heurística**
 - Planificación heurística
 - Redes de tareas jerárquicas
 - Conocimiento de control
 - Aprendizaje automático
- 5 Planificación en el mundo real



Motivación

- Los algoritmos de planificación actuales permiten resolver muchos problemas eficientemente, utilizando:
 - heurísticas independientes del dominio con búsquedas eficientes
 - heurísticas dependientes del dominio creadas a mano
- Todavía hay sitio para mejorar
- Solución:
 - aprendizaje automático integrado con planificación para extraer conocimiento automáticamente de uno o varios problemas y así mejorar el rendimiento en futuros problemas

¿Por qué aprendizaje?

- Ingeniería del conocimiento en planificación
 - definición manual del conocimiento del dominio
 - especificar estrategias de control
 - definir conocimiento para producir planes mejores
- Pero ...
 - los planificadores dependen mucho de la representación
 - definir conocimiento de dominio o de control es difícil
 - las heurísticas actuales siempre resuelven igual el mismo problema
 - las heurísticas no siempre guían hacia las mejores soluciones

- Aprendizaje automático

Automatizar la interpretación de la experiencia de planificación en conocimiento generalizado y reusable: dominio, control, calidad

¿Por qué aprendizaje?

- Ingeniería del conocimiento en planificación
 - definición manual del conocimiento del dominio
 - especificar estrategias de control
 - definir conocimiento para producir planes mejores
- Pero ...
 - los planificadores dependen mucho de la representación
 - definir conocimiento de dominio o de control es difícil
 - las heurísticas actuales siempre resuelven igual el mismo problema
 - las heurísticas no siempre guían hacia las mejores soluciones
- Aprendizaje automático

*Automatizar la interpretación de la experiencia de planificación en conocimiento **generalizado** y **reusable**: dominio, control, calidad*

Tipos de técnicas de aprendizaje

- Métodos inductivos (neuronas, genéticos, ID3, ILP, ...)
 - intensivos en datos: extraen descripciones generales de un *concepto* a partir de muchos ejemplos
- Métodos deductivos (macro-operadores, EBL, ...)
 - intensivos en conocimiento: explican y analizan un solo ejemplo de un concepto
- Enfoques híbridos (EBL+ILP, analogy, Q-RRL, EVOCK, ...)
 - método deductivo para extraer información de un episodio de planificación
 - método inductivo para generalizar/especializar el conocimiento adquirido

Enfoques de aprendizaje en planificación

- Eficiencia: diferentes representaciones
 - Macro-operadores/Nuevos operadores: STRIPS, SOAR, EBL, SNLP+EBG, LEX2, OBSERVE, LOPE
 - Refinar teoría del dominio: ACM, EXPERIMENTER
 - Casos: CHEF, PRIAR, DÆDALUS, PRODIGY/ANALOGY
 - Conocimiento de control: muchos
 - Políticas generalizadas: L2ACT, DSPLAN
- Calidad: QUALITY, STEPPINGSTONE, HAMLET, EVOCK
- Adquisición de la teoría del dominio
 - por observación OBSERVE, SHOP
 - por experimentación EXPERIMENTER, LIVE, LOPE
 - jerarquías ALPINE, CAMEL
 - reglas de reescritura PBR

Algunos sistemas de aprendizaje en planificación

- **Lineal:** STRIPS [Fikes *et al.*, 1972], Rubik's cube [Korf, 1985], PRODIGY/EBL [Minton, 1988], STATIC [Etzioni, 1993], DYNAMIC [Pérez and Etzioni, 1992], ALPINE [Knoblock, 1991], GRASSHOPER [Leckie and Zukerman, 1998], LEX [Mitchell *et al.*, 1983], ACM [Langley, 1983], LEBL [Tadepalli, 1989], DOLPHIN [Zelle and Mooney, 1993], EXPERIMENTER [Carbonell and Gil, 1990], ...
- **No lineal “clásico”:** SOAR [Laird *et al.*, 1986], FAILSAFE [Bhatnagar, 1992], OBSERVE [Wang, 1994], COMPOSER [Gratch and DeJong, 1992], PRIAR [Kambhampati, 1989], SNLP+EBG [Kambhampati and Kedar, 1991], SNLP+EBL [Katukam and Kambhampati, 1994], UCPOP+EBL [Qu and Kambhampati, 1995], QUALITY [Pérez and Carbonell, 1994], STEPPINGSTONE [Ruby and Kibler, 1992], SCOPE [Estlin and Mooney, 1997], PIPP [Upal and Elio, 1998], PRODIGY/ANALOGY [Veloso, 1994], DERSNLP [Ihrig and Kambhampati, 1996], HAMLET [Borrajo and Veloso, 1997], EVOCK [Aler *et al.*, 2002], EXEL [Reddy and Tadepalli, 1999], ...

Más

- **No lineal “no clásico”:** rewrite rules [Ambite *et al.*, 2000, Upal and Elio, 2000], CAMEL [Ilghami *et al.*, 2002], HTN MODELS [Garland *et al.*, 2001], GRAPHPLAN+EBL [Kambhampati, 2000], SATPLAN+FOIL [Huang *et al.*, 2000], generalized policies [Khardon, 1999, Martín and Geffner, 2000], HAP [Vrakas *et al.*, 2003], ...
- **MDP models:** reinforcement learning [Kaelbling *et al.*, 1996], Q-LEARNING [Watkins and Dayan, 1992b], temporal differences [Sutton, 1988, Tesauro, 1992], LOPE [García-Martínez and Borrajo, 2000], ...
- Más en [Langley, 1996],[Minton and Zweben, 1993],[Mitchell, 1997],[Zimmerman and Kambhampati, 2003]

Aprendizaje para ser eficientes

- Macro-operadores
- EBL
- Analogía (casos)
- Métodos híbridos
- Aprendizaje por refuerzo MDP, Q-LEARNING
- Otros

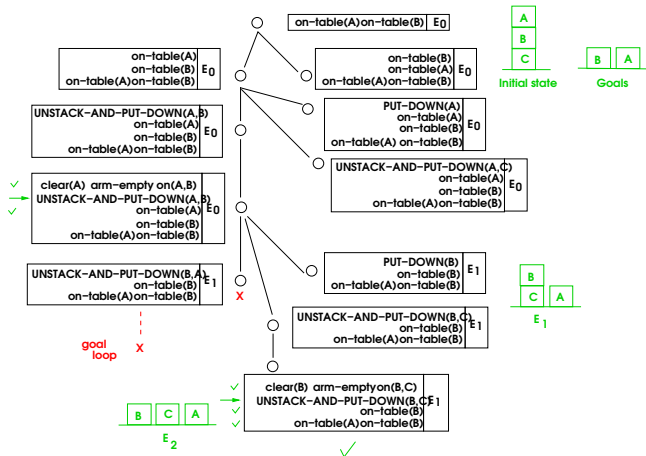
Macro-Operadores

- Una de las primeras ideas de aplicar aprendizaje a planificación
- Se utilizó para mejorar la eficiencia del planificador STRIPS [Fikes *et al.*, 1972]
- Concebido con un doble objetivo:
 - Aprendizaje de secuencias de acciones
 - Monitorización de ejecución de planes
- Idea clave: crear nuevos operadores juntando las descripciones de los operadores individuales que formaban un plan
- Se crean construyendo *tablas triangulares* y parametrizando
- Ejemplos: cubo de Rubik [Korf, 1985], ACT* [Anderson, 1983], MORRIS [Minton, 1985], MACRO-FF [Botea *et al.*, 2005] ...
- También hay macro-operadores iterativos y condicionales [Cheng and Carbonell, 1986, Shell and Carbonell, 1989]

Ejemplo de macro-operador

* encima(x,y) * libre(x) * brazo-libre	QUITAR(x,y)	
	* sujeto(x) libre(y)	DEJAR(x)
	libre(y)	brazo-libre en-mesa(x) libre(x)

Ejemplo de STRIPS



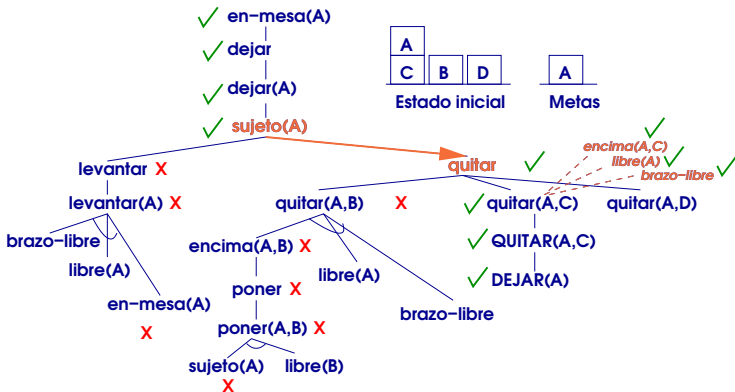
Discusión: Macro-Operadores

- Ventajas:
 - Reutilización de experiencias pasadas
 - Replanificación después de fallos
 - Menor profundidad de búsqueda
 - Menor tiempo de equiparación
 - Efecto secundario: aprendizaje de subsecuencias de operadores
- Desventajas:
 - Incremento del factor de ramificación
- Necesidad de considerar la utilidad

Aprendizaje Basado en la Generalización (EBG)

- Objetivo: re-expresar el conocimiento de un dominio de manera que pueda ser utilizada de forma más eficiente
- Utilizan una teoría del dominio
 - Es más fiable que utilizar múltiples ejemplos
 - Puede extraer más información de cada ejemplo de entrenamiento
 - Requiere, a cambio, que se posea un conocimiento correcto
- PRODIGY/EBL [Minton, 1988] utilizó EBL para planificación
- Muchos lo siguieron: STATIC, DYNAMIC, UCPOP+EBL, ...

Árbol de búsqueda de PRODIGY



Reglas aprendidas por EBL en logística

(control-rule seleccionar-descargar-avion

```
(if (and (current-goal (en <paquete> <lugar>))
         (true-in-state (en <paquete> <lugar1>))
         (type-of-object <paquete> PAQUETE)
         (type-of-object <lugar1> AIRPORT)))
(then select operator DESCARGAR-AVION))
```

(control-rule seleccionar-instanciacion-descargar-avion

```
(if (and (current-goal (en <paquete> <lugar>))
         (current-operator DESCARGAR-AVION)
         (true-in-state (en <paquete> <lugar1>))
         (true-in-state (en <avion> <lugar1>))
         (type-of-object <paquete> PAQUETE)
         (type-of-object <lugar1> AIRPORT)))
(then select bindings ((<package> . <paquete>) (<plane> . <avion>)
                      (<airport> . <lugar>))))
```

El problema de la *utilidad*

Cuantas más reglas aprende PRODIGY/EBL, más lento es

- También el *chunk caro* [Tambe and Rosenbloom, 1989]
- Posibles soluciones:
 - Construir un algoritmo de equiparación eficiente
 - Restringir el lenguaje
 - Análisis de utilidad y olvidarse de las reglas con menor utilidad [Minton, 1988]
 - Coste de equiparación (utilización)
 - Frecuencia de aplicación
 - Ahorro cada vez que se ejecuta
 - Utilizar heurísticas para aprender conocimiento eficiente
 - Buscar en el espacio de los conjuntos de reglas de control
 - Refinamiento incremental del conocimiento de control

El problema de la *utilidad*

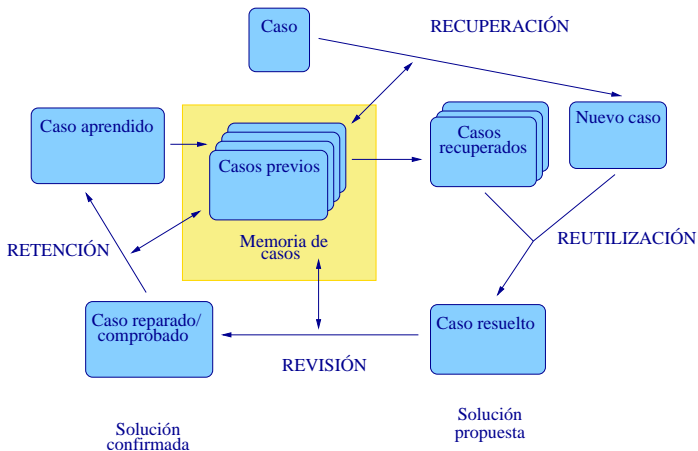
Cuantas más reglas aprende PRODIGY/EBL, más lento es

- También el *chunk caro* [Tambe and Rosenbloom, 1989]
- Posibles soluciones:
 - Construir un algoritmo de equiparación eficiente
 - Restringir el lenguaje
 - Análisis de utilidad y olvidarse de las reglas con menor utilidad [Minton, 1988]
 - Coste de equiparación (utilización)
 - Frecuencia de aplicación
 - Ahorro cada vez que se ejecuta
 - Utilizar heurísticas para aprender conocimiento eficiente
 - Buscar en el espacio de los conjuntos de reglas de control
 - Refinamiento incremental del conocimiento de control

Resolver problemas en función del pasado

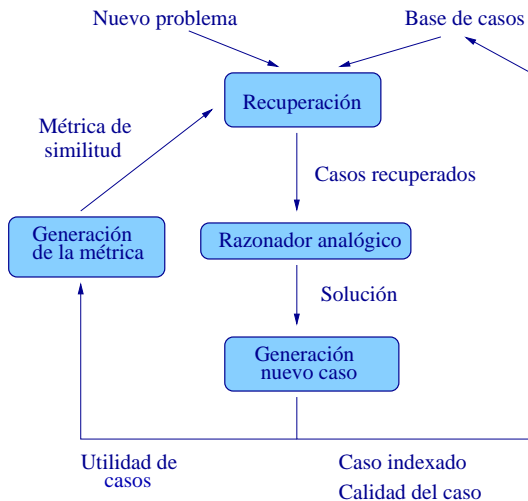
- Gran cantidad de problemas resueltos por los humanos se basan en la combinación de conocimiento de fondo y de casos pasados
- Una forma: las bases de datos, pero se pierde poder de generalización
- Si somos más flexibles permitimos recuperar y utilizar varios casos **parecidos** a los anteriores: razonamiento basado en casos (CBR)
- Modelos computacionales: Roger Shank [Schank, 1982] y sus discípulos (Jaime Carbonell [Carbonell, 1983], Janet Kolodner [Kolodner, 1993], Manuela Veloso [Veloso, 1994])

Arquitectura

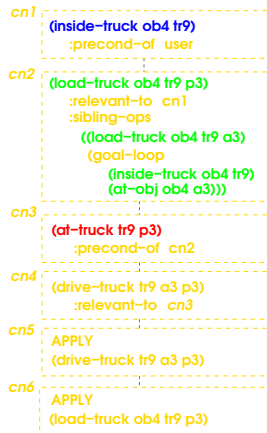
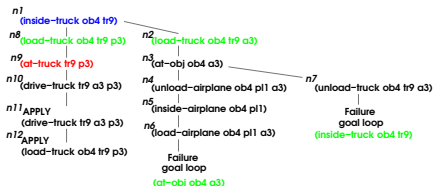


Fuente: [Aamodt and Plaza, 1994]

Prodigy/Analogy



Casos de plan

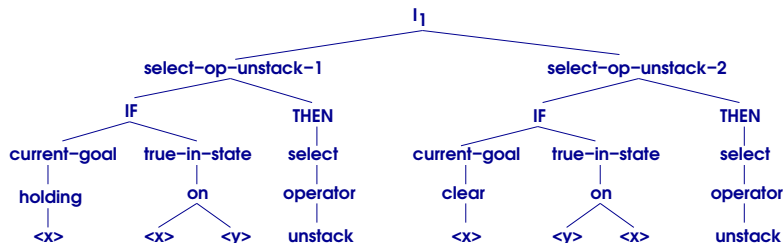


(state (and (at-obj ob4 p3) (at-obj ob7 a3) (at-truck tr9 a3)
 (at-airplane pl1 a3) (same-city a3 p3)))
 (goal (inside-truck ob4 tr9))

Otros enfoques de aprendizaje para eficiencia

- Híbridos: HAMLET [Borrajo and Veloso, 1997], EVOCK [Aler *et al.*, 2002], SCOPE [Estlin and Mooney, 1997], SATPLAN+FOIL [Huang *et al.*, 2000]
- Por refuerzo: Q-LEARNING [Kaelbling *et al.*, 1996],[Dzeroski *et al.*, 2001],[Watkins and Dayan, 1992a]
- Enfoques genéticos [Koza, 1992],[Muslea, 1997],[Spector, 1994],[Andre, 1994],[Handley, 1993]
- *Data mining* del mejor planificador [Vrakas *et al.*, 2003]
- Iniciativa mixta [Pérez and Carbonell, 1994, Aler and Borrajo, 2002]
- Aprendizaje de reglas de PBR [Ambite *et al.*, 2000, Upal and Elio, 2000]
- Inducción: GRASSHOPER [Leckie and Zukerman, 1998]
- Políticas generalizadas [Khardon, 1999],[Martín and Geffner, 2000],[Winner and Veloso, 2002],[Winner and Veloso, 2003]

Ejemplo de un individuo de EvoCK



Aprendizaje para mejorar la calidad de los planes

- En el pasado, casi todo el objetivo era planificación eficiente
- Ahora se presta más atención a la calidad de las soluciones (planes)
- Primero, es necesario representar funciones de coste en cada operador
- Pero, esto no implica que los planificadores sepan cómo obtener mejores soluciones eficientemente
- Es más difícil que *solamente* encontrar una solución
- El conocimiento de control puede ayudar, pero
 - es difícil de definir el conocimiento correcto
 - la definición de este conocimiento muchas veces necesita que se conozca muy bien al planificador
 - las métricas pueden variar con el tiempo, los usuarios y el dominio (mantenimiento)

Dos enfoques

- QUALITY [Pérez and Carbonell, 1994] aprende de la diferencia entre una solución *buena* y una *peor*
- HAMLET [Borrajo and Veloso, 1997] aprende a seleccionar alternativas que llevan a soluciones óptimas

Aprendizaje de conocimiento de dominio

- EXPERIMENT [Carbonell and Gil, 1990]: refina operadores por experimentación
- LIFE [Shen, 1993]: descubre operadores de planificación
- OBSERVE [Wang, 1994]: aprende operadores observando a un agente y refina por resolución de nuevos problemas
- ALPINE [Knoblock, 1990]: adquiere jerarquías de planificación
- Aprendizaje de reglas de decomposición [Reddy and Tadepalli, 1999]
- CAMEL [Ilghami *et al.*, 2002]: aprende precondiciones de métodos SHOP
- ARMS [Yang *et al.*, 2005]: aprende modelos de acciones a partir de planes
- Otros: [McCluskey *et al.*, 2002], [Grant, 1996], [García-Martínez and Borrajo, 2000]

Índice

- 1 Introducción
- 2 Planificación clásica
- 3 Planificación neoclásica
- 4 Heurística
 - Planificación heurística
 - Redes de tareas jerárquicas
 - Conocimiento de control
 - Aprendizaje automático
- 5 Planificación en el mundo real

Referencias



Agnar Aamodt and Enric Plaza.

Case based reasoning: Foundational issues, methodological variations and system approaches.

AI Communications, 7(1):39–59, 1994.



Ricardo Aler and Daniel Borrajo.

On control knowledge acquisition by exploiting human-computer interaction.

In Malik Ghallab, Joachim Hertzberg, and Paolo Traverso, editors, *Proceedings of the Sixth International Conference on Artificial Intelligence Planning Systems (AIPS-02)*, pages 112–120, Toulouse (France), April 2002. AAAI Press.

Ponencia.



Ricardo Aler, Daniel Borrajo, and Pedro Isasi.

Using genetic programming to learn and improve control knowledge.

Artificial Intelligence Journal, 141(1-2):29–56, October 2002.



José Luis Ambite, Craig A. Knoblock, and Steven Minton.

Learning plan rewriting rules.

In Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling, pages 14–17, Breckenbridge, CO (USA), April 2000.



John R. Anderson.

The Architecture of Cognition.

Harvard University Press, Cambridge, Mass, 1983.



David Andre.

Evolution of mapmaking ability: Strategies for the evolution of learning, planning, and memory using genetic programming.

In Proceedings of the 1994 IEEE World Congress on Computational Intelligence, volume 1, pages 250–255, Orlando, Florida, USA, 27-29 June 1994. IEEE Press.



Neeraj Bhatnagar.

On-line learning from search failures.

PhD thesis, Rutgers University, 1992.



Daniel Borrajo and Manuela Veloso.

Lazy incremental learning of control knowledge for efficiently obtaining quality plans.

AI Review Journal. Special Issue on Lazy Learning,
11(1-5):371–405, February 1997.

Also in the book "Lazy Learning", David Aha (ed.), Kluwer Academic Publishers, May 1997, ISBN 0-7923-4584-3.



Adi Botea, Martin Mueller, and Jonathan Schaeffer.

Learning partial-order macros from solutions.

In *Proceedings of ICAPS'05*, Monterrey (USA), June 2005.



Jaime G. Carbonell and Yolanda Gil.

Learning by experimentation: The operator refinement method.

In R. S. Michalski and Y. Kodratoff, editors, *Machine Learning: An Artificial Intelligence Approach, Volume III*, pages 191–213. Morgan Kaufmann, Palo Alto, CA, 1990.



Jaime G. Carbonell.

Learning by analogy: Formulating and generalizing plans from past experience.

In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning, An Artificial Intelligence Approach*, pages 137–162, Palo Alto, CA, 1983. Tioga Press.



Pat W. Cheng and Jaime G. Carbonell.

The FERMI system: Inducing iterative rules from experience.

In *Proceedings of AAAI-86*, pages 490–495, Philadelphia, PA, 1986.



Saso Dzeroski, Luc De Raedt, and Kurt Driessens.

Relational reinforcement learning.

Machine Learning, 43:7–52, 2001.



Tara A. Estlin and Raymond J. Mooney.

Learning to improve both efficiency and quality of planning.

In Martha Pollack, editor, *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 1227–1232. Morgan Kaufmann, 1997.



Oren Etzioni.

Acquiring search-control knowledge via static analysis.

Artificial Intelligence, 62(2):255–301, 1993.



Richard E. Fikes, P. E. Hart, and Nils J. Nilsson.

Learning and executing generalized robot plans.

Artificial Intelligence, 3:251–288, 1972.



Ramón García-Martínez and Daniel Borrajo.

An integrated approach of learning, planning, and execution.

Journal of Intelligent and Robotic Systems, 29(1):47–78,
September 2000.



A. Garland, K. Ryall, and C. Rich.

Learning hierarchical task models by defining and refining examples.

In *In First International Conference on Knowledge Capture*, 2001.



Tim Grant.

Inductive Learning of Knowledge-Based Planning Operators.

PhD thesis, Rijksuniversiteit Limburg te Maastricht, 1996.



Jonathan Gratch and Gerald DeJong.

COMPOSER: A probabilistic solution to the utility problem in speed-up learning.

In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 235–240, 1992.



S. Handley.

The genetic planner: The automatic generation of plans for a mobile robot via genetic programming.

In Proceedings of the Eighth IEEE International Symposium on Intelligent Control, 1993.



Yi-Cheng Huang, Bart Selman, and Henry Kautz.

Learning declarative control rules for constraint-based planning.

In Pat Langley, editor, Proceedings of the Seventeenth International Conference on Machine Learning, ICML'00, Stanford, CA (USA), June-July 2000.



Laurie H. Ihrig and Subbarao Kambhampati.

Design and implementation of a replay framework based on a partial order planner.

In Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96), pages 849–854, Portland, Oregon, USA, 1996. AAAI Press / The MIT Press.

Learning action strategies for planning domains.

Artificial Intelligence, 113(1-2):125–148, 1999.



Craig A. Knoblock.

Learning abstraction hierarchies for problem solving.

In *Proceedings of Eighth National Conference on Artificial Intelligence*, pages 923–928, Cambridge, MA, 1990. AAAI Press/The MIT Press.



Craig A. Knoblock.

Automatically Generating Abstractions for Problem Solving.

PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1991.

Available as technical report CMU-CS-91-120.



Janet Kolodner.

Case-Based Reasoning.

Morgan Kaufmann, 1993.



Richard E. Korf.

Macro-operators: A weak method for learning.

Artificial Intelligence, 26:35–77, 1985.



J. R. Koza.

Genetic Programming: On the Programming of Computers by Means of Natural Selection.

MIT Press, Cambridge, MA, 1992.



John E. Laird, Paul S. Rosenbloom, and Allen Newell.

Chunking in SOAR: The anatomy of a general learning mechanism.

Machine Learning, 1:11–46, 1986.



Pat Langley.

Learning effective search heuristics.

In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 419–421, Los Altos, CA, 1983.
Morgan Kaufmann.



Pat Langley.

Elements of Machine Learning.

Morgan Kaufmann, 1996.



Christopher Leckie and Ingrid Zukerman.

Inductive learning of search control rules for planning.

Artificial Intelligence, 101(1–2):63–98, May 1998.



Mario Martín and Héctor Geffner.

Learning generalized policies in planning using concept languages.

In *Proceedings of the 7th Int. Conf. on Knowledge Representation and Reasoning (KR 2000)*, Colorado, 2000.

Morgan Kaufmann.



T. L. McCluskey, N. E. Richardson, and R. M. Simpson.

An Interactive Method for Inducing Operator Descriptions.

In Proc. of AIPS'02 Workshop on Planning for Temporal Domains, pages 151–159, 2002.



Steven Minton and Monte Zweben.

Machine Learning Methods for Planning.

Morgan Kaufmann, 1993.



Steven Minton.

Selectively generalizing plans for problem solving.

In Proceedings of AAAI-85, pages 596–599, 1985.



Steven Minton.

Learning Effective Search Control Knowledge: An Explanation-Based Approach.

Kluwer Academic Publishers, Boston, MA, 1988.



Tom M. Mitchell, Paul E. Utgoff, and R. B. Banerji.

Learning by experimentation: Acquiring and refining problem-solving heuristics.

In *Machine Learning, An Artificial Intelligence Approach*. Tioga Press, Palo Alto, CA, 1983.



Tom M. Mitchell.

Machine Learning.

McGraw-Hill, 1997.



Ion Muslea.

A general-purpose AI planning system based on the genetic programming paradigm.

In John R. Koza, editor, *Late Breaking Papers at the 1997 Genetic Programming Conference*, pages 157–164, Stanford University, CA, USA, 13–16 July 1997. Stanford Bookstore.



M. Alicia Pérez and Jaime G. Carbonell.

Control knowledge to improve plan quality.

In Proceedings of the Second International Conference on AI Planning Systems, pages 323–328, Chicago, IL, 1994. AAAI Press, CA.



M. Alicia Pérez and Oren Etzioni.

DYNAMIC: A new role for training problems in EBL.

In Proceedings of the Ninth International Conference on Machine Learning, pages 367–372. Morgan Kaufmann, Aberdeen, Scotland, 1992.



Yong Qu and Subbarao Kambhampati.

Learning search control rules for plan-space planners: Factors affecting the performance.

Technical report, Arizona State University, February 1995.



Chandra Reddy and Prasad Tadepalli.

Learning horn definitions: Theory and an application to planning.

New Generation Computing, 17:77–98, 1999.



David Ruby and Dennis Kibler.

Learning episodes for optimization.

In *Proceedings of the Ninth International Conference on Machine Learning*, pages 379–384, Aberdeen, Scotland, 1992. Morgan Kaufmann.



Roger C. Schank.

Dynamic Memory.

Cambridge University Press, 1982.



Peter Shell and Jaime G. Carbonell.

Towards a general framework for composing disjunctive and iterative macro-operators.

In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 1989.



W. Shen.

Discovery as autonomous learning from environment.

Machine Learning, 12:143–165, 1993.



L. Spector.

Genetic programming and AI planning systems.

In Proceedings of Twelfth National Conference on Artificial Intelligence, Seattle, Washington, USA, 1994. AAAI Press/MIT Press.



Richard Sutton.

Learning to predict by the methods of temporal differences.

Machine Learning, 3(1):9–44, August 1988.



Prasad Tadepalli.

Lazy explanation-based learning: A solution to the intractable theory problem.

In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, pages 694–700, San Mateo, CA, 1989.

Morgan Kaufmann.



Milind Tambe and Paul Rosenbloom.

Eliminating expensive chunks by restricting expressiveness.

In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, pages 731–737, San Mateo, CA, 1989. Morgan Kaufmann.



Gerald Tesauro.

Practical issues in temporal difference learning.

Machine Learning, 8(3/4):257–277, May 1992.



Muhammad Afzal Upal and René Elio.

Learning to improve quality of the solutions produced by partial-order planners.

In In Notes of the AIPS-98 Workshop on Knowledge Engineering and Acquisition for Planning: Bridging Theory and Practice, pages 94–104, 1998.



M. Afzal Upal and René Elio.

In *Proceedings of the Second International Conference on AI Planning Systems, AIPS-94*, pages 335–340, Chicago, IL, June 1994. AAAI Press, CA.



C. J. C. H. Watkins and P. Dayan.

Q-Learning.

Machine Learning, 8:279–292, 1992.



C. J. C. H. Watkins and P. Dayan.

Technical note: Q-learning.

Machine Learning, 8(3/4):279–292, May 1992.



Elly Winner and Manuela Veloso.

Analyzing plans with conditional effects.

In *Proceedings of the Sixth International Conference on Artificial Intelligence Planning Systems*, Toulouse, France, April 2002.



Elly Winner and Manuela Veloso.

DISTILL: Towards learning domain-specific planners by example.

In *Proceedings of ICML'03*, Washington, DC, August 2003.



Qiang Yang, K. Wu, and Y. Jiang.

Learning action models from plan examples with incomplete knowledge.

In *Proceedings of ICAPS'05*, Monterrey (USA), June 2005.



J. Zelle and R. Mooney.

Combining FOIL and EBG to speed-up logic programs.

In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1106–1113, Chambéry, France, 1993. Morgan Kaufmann.



Terry Zimmerman and Subbarao Kambhampati.

Learning-assisted automated planning: Looking back, taking stock, going forward.

AI Magazine, 24(2):73–96, Summer 2003.