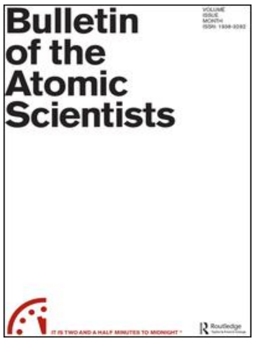

Bulletin
of the
Atomic
Scientists

Bulletin of the Atomic Scientists

ISSN: 0096-3402 (Print) 1938-3282 (Online) Journal homepage: http://www.tandfonline.com/loi/rbul20

(Automated) planning for tomorrow: Will artificial intelligence get smarter?

Edward Moore Geist

To cite this article: Edward Moore Geist (2017) (Automated) planning for tomorrow: Will artificial intelligence get smarter?, Bulletin of the Atomic Scientists, 73:2, 80-85, DOI: 10.1080/00963402.2017.1288435

To link to this article: https://doi.org/10.1080/00963402.2017.1288435

Published online: 17 Feb 2017.

BULLETIN OF THE ATOMIC SCIENTISTS, 2017
VOL. 73, NO. 2, 80–85
http://dx.doi.org/10.1080/00963402.2017.1288435

Routledge
Taylor & Francis Group

FEATURES

(Automated) planning for tomorrow: Will artificial intelligence get smarter?

Edward Moore Geist

ABSTRACT
Artificial-intelligence (AI) researchers have made very considerable advances in their theoretical knowledge of planning over the past few decades. But the impact of AI on society in the coming years will depend on how much these discoveries improve the real-world performance of automated planning, or AP, an AI subfield that seeks to create computer programs that can generate plans to achieve a particular goal. If practical applications of automated planning continue to stagnate, it could hold back all of AI, even as its other subfields continue to mature. Modest progress, meanwhile, would facilitate modest economic and military uses of artificial intelligence. And should AP experience the same kind of spectacular breakout as reinforcement learning, which is being used practically in a wide variety of fields, from robotics to finance, the peril and promise of artificial intelligence might be fully realized.

KEYWORDS
Artificial intelligence; automated planning

Why isn't artificial intelligence (AI) more intelligent? Dramatic progress in areas such as machine vision and the notoriously difficult game of Go have rendered the inadequacies of state-of-the-art AIs all the more glaring. Computers now humble the best human players at chess, Go, and even the classic Atari game Breakout, yet artificial intelligence continues to struggle at many other tasks that even human toddlers accomplish with ease. Unlike well-specified problems such as games, interacting with complex real-world environments continues to seriously challenge even the most advanced AI techniques. Does faster-than-anticipated progress on tasks such as game-playing suggest that artificial intelligence will soon excel at the economic and military applications anticipated or feared by so many?

Not necessarily. Artificial intelligence is best understood as a collection of different fields and approaches that aim to create machines capable of tasks that require cognition when humans do them. Many AI researchers are focused on finding ways to make computers perform very modest or utilitarian cognitive tasks, while others are devoted to theoretical or methodological investigations that might or might not lead to practical applications in the future. Spectacular progress in one area of AI research often doesn't do much to advance others. To assume otherwise is to repeat the mistake of the pioneers of the field, who became overconfident following the success of their earliest experiments in the 1950s and made predictions that led to immense embarrassment in subsequent decades (McCorduck 1979). For instance, our ability to make computer programs that hopelessly outclass even the best human chess masters has not taught us much about how to make a robot vacuum cleaner that won't get stuck on an unfamiliar bump in the carpet.

But the unpredictability and unevenness of progress in artificial intelligence also has a silver lining. AI breakthroughs sometimes emerge from lines of research that had previously been written off as disappointments. Take, for instance, deep learning and reinforcement learning, the technologies used by Google DeepMind to master Go and Atari 2600 games (Mnih, et al. 2015). As recently as the early 2000s, both of these technologies were curiosities studied by a handful of academic researchers that, for all practical intents and purposes, did not work. Within a few years, however, they have conquered problems many AI researchers expected would remain unsolved for the foreseeable future.

While some AI tasks, such as playing chess, are well-studied and well-understood, decades of intensive effort on others haven't paid off in the same way. Take, for instance, the field of automated planning (AP), which seeks to create computer programs that can generate plans to achieve a particular goal. These include domain-independent planners able to create plans for arbitrary problems, domain-configurable planners employing domain knowledge to solve a particular problem more efficiently, and domain-specific planners capable of solving only a narrow class of problems.

To carry out unfamiliar tasks that require planning, both humans and machines need to first create an

CONTACT Edward Moore Geist ✉ egeist@rand.org

appropriate model for reasoning about the problem and then reason with that model to generate a plan. Furthermore, in most cases it is necessary to modify the plan while carrying it out because of misconceptions included in the planning model, or changed circumstances.

For decades, artificial-intelligence researchers have attempted to program computers to perform the first two aspects of planning, and in recent years they have begun tackling the third. So far, their efforts have been met with only modest success, even though some of the spin-offs of their research have proved of immense practical significance. (Perhaps the most famous of these is the A* search algorithm, which you've probably used several times today without realizing it.) This is because planning turns out to be surprisingly difficult, for reasons that range from knowledge quality problems to computational complexity challenges.

In fiction, artificial intelligences effortlessly craft foolproof plans, but the understanding that has emerged from automated-planning research is that planning is devilishly hard. The lackluster performance of current automated-planning approaches poses a critical bottleneck to creating artificial-intelligence systems capable of many of the tasks eagerly sought by industry and the military.[1]

As long as progress in automated planning remains modest, the impact of artificial intelligence on both the workplace and the battlefield will necessarily be limited. Not only will robots be too stupid to effectively replace humans in many civilian jobs but they also won't be capable of navigating novel tactical situations, putting a brake on the utility of military robots. While campaigners urging for a ban on "killer robots" are absolutely right that placing decisions about taking human lives in the hands of present-day AP systems would be monstrously irresponsible, right now the temptation to employ this technology remains minimal because it isn't smart enough to provide a significant strategic advantage.

Due in part to the immaturity of automated planning, even the kind of autonomy essential for unarmed military robots remains elusive. Undersea submarine-hunting drones could remake nuclear strategy (Holmes 2016) – but they won't until they can successfully form and actualize plans in a dynamic contested environment without human assistance, and the most sophisticated AP systems aren't currently capable of this task. The current shortcomings of automated planning also stand in the way of extreme AI doomsday scenarios – after all, the "superintelligences" feared by some are essentially planning systems that are so powerful as to be apocalyptically dangerous.

But the same limitations forestalling pernicious applications of AI are also hindering its benefits to humanity. Many difficult and dangerous jobs that would be better left to robots still require much better planning capability than artificial intelligence currently provides; a space probe to explore the watery interiors of Jupiter's icy moons would demand the same autonomous capabilities as sub-hunting drones.

The disappointing practical utility of current AP systems, however, belies the fact that AI researchers have made very considerable advances in their theoretical knowledge of planning over the past few decades. The impact of artificial intelligence on society in coming years will depend on how much these discoveries improve the real-world performance of automated planning. If practical applications of automated planning continue to stagnate, it could hold back all of AI even as its other subfields continue to mature. Modest progress, meanwhile, would facilitate modest economic and military uses of artificial intelligence.

And should AP experience the same kind of spectacular breakout as reinforcement learning, both the peril and promise of artificial intelligence might be fully realized.

The history of automated planning

The history of automated planning is inextricably linked to another AI subfield – problem-solving. In the early decades of AI research, these two areas were essentially conflated with one another, and the exact boundary between them is still difficult to delineate (Russell and Norvig 1995). The first problem-solving program, the General Problem Solver, was developed by Herbert Simon, Allen Newell, and Cliff Shaw in the late 1950s (Newell, Shaw, and Simon 1959). As one of the first AI programs ever implemented on a physical computer, it inspired more excitement than it deserved in retrospect. While it handily cracked puzzles that strained human minds, it choked on more substantial challenges because it could not break a large problem into more digestible subtasks. In theory, the program could answer a huge range of problems; in practice, the exponential growth of the search space for all but the simplest ones rendered it a mere curiosity (McDermott 1976).

To address the shortcomings of the General Problem Solver, in the 1960s AI researchers developed planners. These programs aimed to make larger problems tractable by dividing them into easier subproblems and then combining the results. As it turned out, this approach was much more practical for most real-world tasks than that employed by the General Problem Solver.

By the end of the 1960s, planners had been incorporated in the Stanford Research Institute's pioneering robot, Shakey. At this time, problem-solving and planning were still conflated, and Shakey's planner was dubbed the Stanford Research Institute Problem Solver – Strips for short. Strips proved enormously influential, and the logical language it used to express problems is still used in planning research. Reflecting its shared heritage with problem-solving, Strips and most planning research of its era focused exclusively on so-called "classical planning" that only dealt in discrete, known actions in a perfectly known, deterministic environment (Fikes and Nilsson 1971). While practical robots would demand some way of planning under uncertainty, deterministic planning was adequate for Shakey's carefully engineered test environment and was more than challenging enough for state-of-the-art AI at the dawn of the 1970s.

For all its conceptual simplicity, however, breaking problems into smaller problems proved to be surprisingly tricky to do effectively. Early, naive approaches disregarded possible contradictions between the subplans, with the result that they couldn't solve certain problems, even if they were trivially simple. Take, for instance, "Sussman's anomaly," which is quite literally child's play. Identified by Gerald Sussman in his MIT doctoral dissertation, it consists of three blocks labeled A, B, and C. The goal is simple: to rearrange the blocks one at a time so that A is on top of B and B is on top of C. In the initial configuration, blocks A and B sit beside each other on a table, and C sits on top of A. The solution is to first move C to the table, then put B on top of C, and finally put A on top of B.

Early planning programs failed this task because they sought partial solutions such as "put B on C" rather than "put C on the table," which was part of the solution but not one of the explicit plan goals. To overcome this problem Sussman and his adviser, Marvin Minsky, created "interleaving" planners that aimed to coordinate the generation of the partial solutions to avoid these kinds of contradictions (Sussman 1975). While AI researchers demonstrated planners that could overcome Sussman's anomaly by the early 1970s, the fact that it took years to surmount it is a symptom of just how challenging automated planning is.

Also in the early 1970s, the maturation of computational complexity theory made apparent another obstacle to automated planning. While Alan Turing had proved that certain problems (most famously the halting problem) were simply undecidable (Turing 1936), when artificial-intelligence research began in the 1950s computer science still lacked the tools to classify the complexity of solving many decidable problems. Computer scientists, most prominently Richard Karp, introduced a system of complexity classes to describe the difficulty of computing these (Karp 1972).

The unavailability of computational complexity theory had been a factor in the embarrassingly overconfident predictions of pioneering AI researchers in the 1950s and 1960s, but the discovery that many AI problems belonged to forbiddingly difficult complexity classes poured cold water on hopes that human-level artificial intelligence might be just a decade or two away (Norvig 1992). Unfortunately, it turned out that even some relatively simple planning tasks fell into these classes. This finding did not prove that artificial intelligence could never produce satisfactory plans in a reasonable amount of time, but it called into question the premise that AI would readily generate optimal, formally correct plans free of cognitive biases.[2]

The 1990s saw considerable progress in both the theory and implementation of planning systems. At the beginning of that decade, researchers learned that classical planning problems could easily be converted to satisfaction problems, which consist of queries of the form "Does an answer satisfying this problem exist?" Traditional planners such as Strips operated on the basis of theorem proving, which is often difficult but seemed simpler than satisfaction problems. The introduction of a highly efficient algorithm for satisfaction problems suddenly turned this premise on its head (Kautz and Selman 1992). This enabled the creation of satisfaction planners, which could sometimes solve classical planning problems much faster than their predecessors.

Other advances emerged from efforts to combine classical planning with machine learning. While early planning systems had to be provided with both the set of available actions and a description of the environment, researchers sought to allow AP systems to learn these from training examples. By the mid-1990s, they had demonstrated systems that could learn the type of action models used by planners such as Strips (Jiménez, et al. 2012).

While even classical planning provided plenty of challenges for AI researchers, in the 1980s and 1990s they began to make some initial progress on planning algorithms that better reflected the uncertainties of the real world. Where classical planning assumes deterministic actions and a perfectly known environment, in actuality actions have probabilistic effects and our knowledge of the environment is necessarily imperfect.

Unfortunately, planners accounting for incomplete knowledge and probabilistic effects are even more computationally challenging than classical planners.

Just the same, AI researchers have made some progress combining these techniques with machine learning as well (Ghallab, Nau, and Traverso 2004). Planning the movement of robots, meanwhile, requires the use of continuous spaces even if the environment is treated as perfectly known and deterministic. Extensive study in the subset of AP addressing these problems, motion planning, has enabled the creation of domain-specific planners for some nontrivial problems (LaValle 2006).

By the end of the last century, practical applications of AP began to appear. Perhaps the most dramatic of these was NASA's incorporation of automated-planning software into some of its space probes, including its Mars rovers. The longtime delay before transmissions from Earth reached the planet demanded that these systems have some ability to plan and carry out tasks on their own without human guidance (Estlin, et al. 2003). Commercial applications of AP also emerged, but the ambitions of these tended to be relatively modest. Some well-specified industrial activities such as control of factory equipment leant themselves well to the limitations of available AP systems (Nau 2007).

Hidden potential?

At present, the limitations of automated planning act as a serious brake on what AI can accomplish. But the continuing challenge of applying AP to complex real-world problems belies the considerable theoretical progress that AP researchers have made over the last few decades. Might automated planning make the same kind of spectacular breakout as reinforcement learning?

Automated planning has traditionally been associated with the "symbolic" school of artificial intelligence that dominated the field before the 1980s "AI winter." Perhaps the most important trend in contemporary artificial intelligence, however, is the emergence of techniques that combine both "symbolic" reasoning capabilities with "connectionist" neural networks or statistical machine learning (Domingos 2006; Domingos and Lowd 2009). At present most of these systems remain too slow and computationally intensive for practical applications. But if they realize their promise they may surmount the obstacles that have limited real-world uses of automated planning up to this point.

Deep learning and reinforcement learning could contribute to the creation of more practical automated-planning systems. While conventional deep neural networks are not capable of forming plans on their own, one of the most active areas of AI research at the moment seeks ways to overcome this limitation. For example, despite its ability to play simpler Atari 2600 games at a superhuman level, DeepMind's initial Atari-playing deep learning system failed to play other games that demanded planning abilities.

In 2016 DeepMind Technologies announced its development of "differentiable neural computers" that learn to write information to an external memory using the same mechanism used to train the associated deep neural network. The DeepMind researchers' paper in *Nature* demonstrated that this system can play several of the Atari 2600 games their earlier Atari-playing AI failed at. While an extremely impressive technical result in many respects, it should be noted that the system's planning ability remained quite modest, and it only played these games as well as a novice human player (Graves, et al. 2016).

A different line of research that has attracted considerable enthusiasm in recent months is the value iteration network, which integrates a value-iteration planning algorithm into the architecture of a deep neural network (Tamar, et al. 2016). Another possibility is to train deep neural networks to output the kind of "symbolic" representations employed by classical planners. While this technique remains in its absolute infancy, and no one has demonstrated a planning system using it, preliminary results applying it to other problems suggest it might be possible to employ it to this end (Garnelo, Arulkumaran, and Shanahan 2016). Techniques such as these cannot alleviate the computational difficulty of planning itself, but if they mature they could make the application of existing planning techniques to real-world problems much easier by turning messy, real-world data into planning models.

The impact of artificial intelligence on human affairs will depend on how successful automated-planning researchers are at making AP systems more effective. In a pessimistic scenario, they would fail to make any meaningful improvements to current approaches. As projected advances in computing power are unlikely to overcome the inefficiencies of present-day AP methods, this failure would seriously constrain the possibilities of machine intelligence both for good and for ill. A likelier outcome is one of incremental improvements in automated-planning techniques. In this scenario, some of the theoretical advances made by AP researchers would translate into significant, albeit not spectacular, increases in the performance of AP systems evolved from current techniques. While this could enable AI to do many things it presently can't, it would probably still fail to reach a human level of competence in many tasks.

Automated-planning researchers increasingly believe that they will need to embrace a qualitatively different approach to their subject in order to make computers plan as well as people. Traditionally, forming plans has been treated as a distinct activity that can be treated separately from actualizing those plans. The poor performance of robots that try to separate planning from acting in this way suggests that it would be better to treat the two as aspects of a common whole.

Over the past decade AP researchers have been developing new intellectual frameworks that reconceptualize planning in this way, and the authors of the standard textbook on AP made their stance on this clear when they titled the updated edition *Automated Planning and Acting*. They argue that both planning and acting should be treated as "deliberate behavior." Defining "deliberation" as a "reasoning process" that "consists in deciding which actions to undertake and how to perform them to achieve an objective," they envision that AIs would need to deliberate constantly about both what actions to take and how to carry them out on the basis of uncertain, but constantly updated, knowledge (Ghallab, Nau, and Traverso 2016). Work in this area remains in its early stages, but the formalisms these researchers have proposed appear much more promising than either classical planning models or reinforcement learning models for guiding the development of sophisticated AIs.

For the time being, the limitations of automated planning see to it that there are certain things that computers can't do, constraining the uses of artificial intelligence both for good and for ill. Sixty years of research into automated planning give us little theoretical or empirical reason to believe that planning is easy for either humans or machines. But the spectacular breakout of reinforcement learning demonstrates that sometimes apparently moribund areas of AI research hold hidden potential. The ability of humans to form and carry out plans proves that planning is possible; it remains to be seen how successful we will be endowing machines with the same ability. As humans aren't very good at planning in a cosmic sense, machines might very well become better at it than we can ever be, even if it turns out that their plans can never be infallible.

Notes

1. Regarding types of planners: Generating plans to attain arbitrary goals would obviously be part of "general AI," and even the earliest AI researchers attempted to create domain-independent planning programs. Unfortunately, it turns out that it is very difficult to make such "domain-independent" planners practical. These programs are still useful, however, as a foundation for "domain-constrained" planners. These employ the same kind of "general" planning algorithms as domain-independent planners but exploit additional domain-specific knowledge to search for a solution more efficiently. "Domain-specific" planners take this approach to the extreme by using domain knowledge as the basis for algorithms that find plans for a specific domain as efficiently as possible. The trade-off is a total loss of generality – these programs can *only* solve the specific problems they were designed to.

   More practical AI approaches to planning could result from improved domain-impendent planning algorithms, which could then serve as the basis for faster domain-constrained planners; better strategies for exploiting domain knowledge in domain-constrained planners; or more effective machine learning techniques for acquiring the domain knowledge necessary for domain-constrained and domain-specific planners. AI researchers are investigating all of these.
2. Computational complexity matters more for planning than it does for some other areas of computer science. This is because the types of approaches that can often find satisfactory approximate answers for other complex problems tend to work less well for the categories to which domain-general planning problems belong. The way this is dealt with in AP is by using domain knowledge to simplify computation, but in many cases we don't know enough about the domain we want the planner for to do that.

Disclosure statement

No potential conflict of interest was reported by the author.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Notes on contributor

Edward Geist is an associate policy researcher at the RAND Corporation.

References

Domingos, P. 2006. "What's Missing in AI: The Interface Layer." In *Artificial Intelligence: The First Hundred Years*, edited by P. Cohen. Menlo Park, CA: AAAI Press.

Domingos, P., and D. Lowd. 2009. "Markov Logic: An Interface Layer for Artificial Intelligence." *Synthesis Lectures on Artificial Intelligence and Machine Learning* 3 (1): 1–155. doi:10.2200/S00206ED1V01Y200907AIM007.

Estlin, T., R. Castaño, R. Anderson, D. Gaines, F. Fisher, and M. Judd. 2003. "Learning and Planning for Mars Rover Science." In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, edited by Georg Gottlob and Toby Walsh. San Francisco: Morgan Kaufmann Publishers.

Fikes, R. E., and N. J. Nilsson. 1971. “STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving.” *Artificial Intelligence* 2 (3–4): 189–208. doi:10.1016/0004-3702(71)90010-5.

Garnelo, M., K. Arulkumaran, and M. Shanahan. 2016. “Towards Deep Symbolic Reinforcement Learning.” Arxiv Preprint. Arxiv:1609.05518. https://arxiv.org/abs/1609.05518.

Ghallab, M., D. Nau, and P. Traverso. 2004. *Automated Planning: Theory and Practice*. San Francisco: Elsevier.

Ghallab, M., D. Nau, and P. Traverso. 2016. *Automated Planning and Acting*. Cambridge: Cambridge University Press.

Graves, A., G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo et al. 2016. “Hybrid Computing Using a Neural Network with Dynamic External Memory.” *Nature* 538 (7626): 471–476. DOI:10.1038/nature20101.

Holmes, J. R. 2016. “Sea Changes: The Future of Nuclear Deterrence.” *Bulletin of the Atomic Scientists* 72 (4): 228–233. doi:10.1080/00963402.2016.1194060.

Jiménez, S., T. De La Rosa, S. Fernández, F. Fernández, and D. Borrajo. 2012. “A Review of Machine Learning for Automated Planning.” *The Knowledge Engineering Review* 27 (4): 433–467. doi:10.1017/S026988891200001X.

Karp, R. M. 1972. “Reducibility among Combinatorial Problems.” In *Complexity of Computer Computations*, edited by R. Miller. 85–103. Springer US.

Kautz, H. A., and B. Selman. 1992. “Planning as Satisfiability.” In *Ecai 92*, edited by B. Neumann. 359–363. New York: Wiley.

LaValle, S. M. 2006. *Planning Algorithms*. Cambridge: Cambridge University Press.

McCorduck, P. 1979. *Machines Who Think*. San Francisco: W. H. Freeman.

McDermott, D. 1976. “Artificial Intelligence Meets Natural Stupidity.” *ACM SIGART Bulletin* 57: 4–9. doi:10.1145/1045339.

Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves et al. 2015. “Human-Level Control through Deep Reinforcement Learning.” *Nature* 518 (7540): 529–533. DOI:10.1038/nature14236.

Nau, D. S. 2007. “Current Trends in Automated Planning.” *AI Magazine* 28 (4): 43–58.

Newell, A., J. C. Shaw, and H. A. Simon. 1959. “Report on a General Problem Solving Program.” *IFIP Congress*: 256–264.

Norvig, P. 1992. *Paradigms of Artificial Intelligence Programming: Case Studies in Common LISP*. San Francisco: Morgan Kaufmann.

Russell, S., and P. Norvig. 1995. *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall.

Sussman, G. J. 1975. *A Computer Model of Skill Acquisition*. New York: American Elsevier Publishing Company.

Tamar, A., Y. Wu, G. Thomas, S. Levine, and P. Abbeel. 2016. “Value Iteration Networks.” Arxiv Preprint. Arxiv:1602.02867. https://arxiv.org/abs/1602.02867.

Turing, A. M. 1936. “On Computable Numbers, with an Application to the Entscheidungsproblem.” *Journal of Math* 58: 345–363.