Towards Adversarial Geometric Planning

Stefan Edelkamp, Pavel Rytíř, Martin Loebl, Lukáš Chrpa, Rotislav Horčík

Department of Computer Science, Faculty of Electrical Engineering
Czech Technical University in Prague
edelkste@fel.cvut.cz, rytirpav@fel.cvut.cz, loebl@kam.mff.cuni.cz, chrpaluk@fel.cvut.cz, xhorcik@fel.cvut.cz

Abstract

One of the main goals of AI is the construction of intelligent agents such as an opponent in a computer game or an unmanned aerial vehicle delivering a parcel to a customer. In this paper we propose an adversarial planner capable of geometric state progression and reasoning. A planning domain definition language extension is planned and theoretical analyses should study feasibility, correctness and performance of a unique and joint state representation as well as properties of convergence. The framework should be implemented on top of existing planners and a framework for double-oracle.

Introduction

One of the main goals of AI is the construction of intelligent agents such as an opponent in a computer game or an unmanned aerial vehicle delivering a parcel to a customer. These intelligent agents perceive and act in various environments to achieve their goals. For example, in case of a computer game, the goal is to defeat the player. In case of parcel delivery drone, the goal is to deliver the parcel in time to the customer.

The agent perceives the state of an environment and needs to decide what to do next. One possible approach is reinforcement learning [36], where an agent learns from the interactions with the environment. This approach was successful for several domains, and achieved superhuman performance in Go [60], Starcraft [66], or Atari games [41]. Another approach how an agent could act in an environment is to create a plan of actions in advance. For a given goal the agent computes a sequence of actions that leads to it. Automated planning has been successful in many areas, as in the Deep Space 1 [4] or in the Mars rover mission [1]. One drawback of automated planning is that when an environment is unexpectedly changed, the agent usually cannot longer proceed to the goal. This happens either randomly or by actions of other adversary agents. To explicitly reason about other agents and find a robust plan, game-theoretic methods [59] like double-oracle (DO, see Fig.1) have to be used. There are several successful applications of game-theoretic algorithms in practice, e.g., in domains of physical security [64] or protecting wildlife [19]. Further cases we are concerned about are combat situations, like defending a nuclear plant with drones against aggressors.

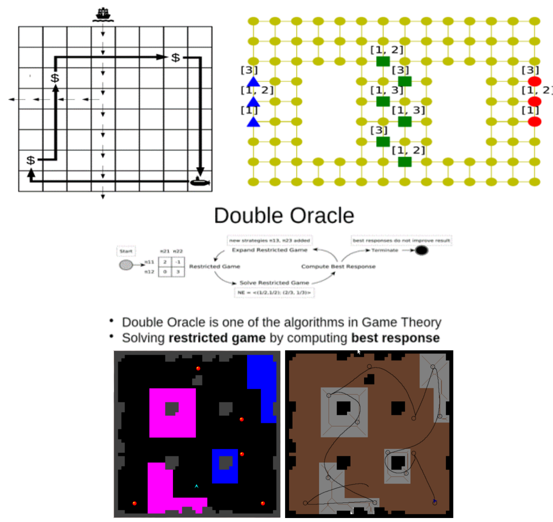

Figure 1: Adversarial planning, resource allocation, double-oracle algorithm, geometric navigation (left to right).

The main goal of this work is to advance algorithms for automated adversarial planning by enhancing geometric reasoning. Although the planning domain definition language (PDDL) [39] is an expressive modelling tool, a significant limitation is imposed on the structure of actions: the parameters of actions are restricted to values from finite (in fact, explicitly enumerated) domains. A motivation for this limitation is that it ensures that the set of grounded actions is finite and, ignoring duration, the branching factor of action choices at a state is therefore finite. Although duration parameters can make this choice infinite, very few planners support this possibility, but restrict themselves to fixed durations. Problems like a jeep crossing a desert of unknown width cannot be solved [32].

We propose extensions to PDDL to enrich actions with geometry. We implement planners capable to lift reasoning to spatial domains and apply them in adversarial settings. We illustrate that these approaches scale to solve interesting problems and apply this work to a task and motion planning scenarios (Fig. 2) to show that our work has a great potential of re-inventing the way task planners are used in

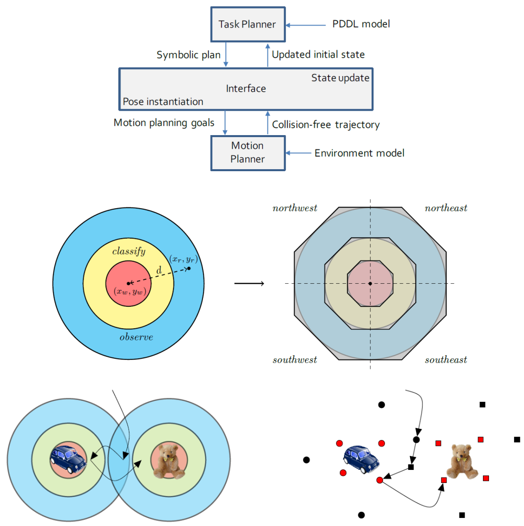

Figure 2: Geometric task-motion planning: loop, linear approximations, motion planning for inspection (left to right).

robotics. Geometry is effective even without adversaries, but the planners are called many times for best-responses in the DO algorithm, so as a multiplier we have that if the adversarial planning domains were geometric, solubility and scaling would become much better.

State-of-the-Art

Classical domain-independent planning is among the key areas of artificial intelligence. The focus of research is often on plan generation methods that allow scaling up and solving real-world problems [65].

Many classical planners use heuristics to guide the search through the state space of the planning problem [42]. One of the very successful heuristics is potential heuristic [51]. This heuristic, for a given planning problem, solves a linear program that assigns real values to the facts of the problem. That creates an embedding of states into the real space. This can be viewed as a geometric representation of the planning problem. Another type of potential heuristic [50] assigns real values to the tuples of facts creating high-dimensional embedding of the states.

In this project we would like to leverage our experience with geometric representations of combinatorial structures. In Rytíř [54, 53, 38] we showed that every linear code can be represented as a two-dimensional simplicial complex. We say that a linear code $\mathcal{C}$ over a field $\mathbb{F}$ is triangular representable if there exists a two dimensional simplicial complex Δ such that $\mathcal{C}$ is a punctured code of the kernel $\ker \Delta$ of the incidence matrix of Δ over $\mathbb{F}$ and there is a linear mapping between $\mathcal{C}$ and $\ker \Delta$ which is a bijection and maps minimal codewords to minimal codewords. We showed that the linear codes over rationals and over GF(p), where p is a prime, are triangular representable. In the case of finite fields, we showed that this representation determines the weight enumerator of $\mathcal{C}$.

Geometric Planning

Over the last decade, significant progress in ensuring the geometric feasibility of symbolic plans in highly confined workspaces has been made by various researchers [63, 34]. Among the other pioneering approaches, aSyMov [6] is a highly recognized task-motion planner (TAMP) that integrates the Metric-FF task planner [24] with a motion planner. It extends Metric-FF to check, whether the symbolic action to be applied results in a geometrically feasible state (based on the feedback from motion planner). Although Metric-FF is a domain-independent planner, extending it to check the state geometric reachability makes the approach domain-specific.

A parallel research line to this approach has been recently studied by Garrett, Lozano-Perez and Kaebeling [20, 21], and their system is called FFRob. In their work, they identify that aSyMov suffers from the lack of heuristic guidance, thus they propose an extension to the RPG heuristic in the FF planner that takes geometric information into account, Erdogan and Stilman [18, 17] encode geometrical constraints in action descriptions (in PDDL-like representation) and propagate the constraints while finding a symbolic plan for a structural design problem. Although the approach reasons with action level constraints, the task planner is not domain-independent. Erdem et al. [16], similar to the approach of Dornhege et al. [14, 15], use a fully symbolic task planner. In their approach, the task and motion planner guides each other in case any of these planners deem unsolvability. For instance, in case the task planner cannot find a symbolic plan, the motion planner updates the problem file for re-planning. If the motion planner fails finding a trajectory, the task planner suggests a new symbolic route to the motion planner.

We compute symbolic plans that are aware of the geometrical constraints encoded in a domain-independent PDDL environment, so that any domain-independent task planner that can reason with the full semantics of the PDDL2.1 language and our proposed language extension can be used. Unlike aSymov and FFRob, the geometric constraints, which model the locations as continuous regions, are directly encoded in the action schemas.

One domain-independent planner that can deal with geometric reasoning is Popcorn [57]. While the control parameters allow for inequalities on numerical fluents in general, it has been applied in geometric planning as found in his PhD *Temporal-Numeric Planning with Control Parameters* [58] in chapter 5 *Case Study: Spatial Reasoning in Task Planning with Control Parameters*. No subsequent work on this planner aspect was found. A related approach of general-purpose planning with control parameters has been followed in the context of reasoning of key performance indicators in a multi-agent setting [45].

Adversarial Planning

Many of the domains where planning algorithms can be applied are not single-agent. Examples include planning defense measures, information collection in an adversarial environment or planning a robust mission where nature acts as the other agent.

Using a plan that ignores the presence of other agents can have severe consequences and the actual quality of the plan can be arbitrarily worse compared to the expectations computed by the planning algorithm. In order to explicit reason about other agents and calculate a provably robust plan, game-theoretic methods have to be used.

The concept of planning in adversarial environment is not new [2]. Succinct symbolic representations of state sets helped generating optimistic and strong cyclic adversarial plans [26, 11], a setting conceptually related to FOND planning [31]. Such a setting, however, has to explore most if not all alternatives (in analogy to traditional game-tree methods such as minimax). Monte-Carlo Tree Search (MCTS) and Online Evolutionary Planning have been applied in adversarial environments such as the Hero Academy game [27], or Starcraft [28]. Deep Reinforcement Learning (DRL) has shown impressive results in Starcraft [66] and other (adversarial) domains such as the games of Chess or Go [61]. MCTS and DRL approaches work "online": they select the most promising action (or move) in the current state of the environment and they continue to do so until the terminal state is reached.

From the planning side, Speicher et al. [62] used the game-theoretic framework of *Stackelberg games* for generating robust plans against actions of the adversary. In a similar spirit, *Plan Interdiction Games* have been proposed to describe the problem of attackers and defenders, where the former plans to intrude a computer network, while the latter tries to prohibit attackers' actions [35, 67]. A recent work about counter-planning goes in a similar direction as one agent tries to invalidate landmarks required by the opposite agent [52]. Planning-based techniques work offline, i.e., they generate plans upfront, which are then executed (as they are).

In many scenarios, one agent must randomize between actions to reach the best expected reward. By randomizing between different plans or actions, the other agents are uncertain about the plan and, thus, it can be impossible for them to exploit such strategy. Current planning algorithms, however, are not able to find such randomized plans. In recent years, many new game-theoretic algorithms have been proposed and the scalability improved, s.t. real-world applications are getting into scope.

One of the best-known game-theoretic algorithms is the incremental strategy generation method called double-oracle [40]. The algorithm tackles one common problem of games – the exponential number of possibilities to choose from. The number of plans needed to achieve certain goals is usually exponential with respect to the number of agent's actions (while omitting plans with loops, i.e., during the plan execution none of the states is visited more than once). Recently, an effort to combine domain-independent planning and double-oracle has been made [55]. In particular, the (best) response in double oracle is encoded as a planning task in which "critical actions" are associated with cost such that the cost reflects probability of applying these actions before those of the adversary.

There are similar approaches that uses reinforcement learning as best-response oracle [33, 43, 68].

Methodology

We develop new adversarial planning algorithms for large-scale so-called *spatial* domains in order to improve security and efficiency. The size of the domain is measured by the number of agents and the number goals that an agent wants to achieve, and by the map complexity. As objectives we have

- Identification of suitable benchmarks and domains. We could use for example UAV and Taxi domains from [9], and the benchmarks like the Squirrl domain from [58]. We also expect to extract problem domains from the framework of Erion Plaku [47], and from virtual robotics competitions like SubT (`https://subtchallenge.world`). In case of grid worlds, we will take the Grid competition problems from Nathan Sturtevant's Moving AI Lab (`https://movingai.com/benchmarks/grids.html`) that include game maps like Baldurs' game and Starcraft, and from the CIG PTSP problem suite [46]. We will compare our planners also on standard and extended benchmarks from the international planning competition [65], or against our results in [55, 10].
- Developing an efficient representation of large-scale domains. Depending on the granularity of discretization the grounded PDDL representation of even simpler domains models like mazes, usually exceeds several MBs or even GBs of memory. Therefore, we will resort to a representation of states as meshes and unions of polytopes, by exploiting CGAL (`https://www.cgal.org/`). In case of complex regions like circles and spheres, we approximate them linearly with bounding boxes and kd-trees [12], or with sphere-trees [44].
- Applying state-space factorization alias decomposition methods that have been shown to be effective in the discrete setting (current conference submission *Effective Planning in Resource-Competition Problems by Task Decomposition*), should induce better scaling also for planning in a more compact world of geometric objects.
- Calling new oracles for iterative algorithms (best responses for the double-oracle algorithm), e.g., by including
  - waypoint-based inspection algorithms, with waypoints generated from the opponent mixed strategy.
  - sampling-based motion planning algorithms like RRT or RRT* together with planning heuristics.
  - swarm-intelligence oracles, where the opponent mixed strategy is encoded into the risk function.

We aim to develop a framework to create and test various geometric representations of planning and game-playing problems. These geometric representations will be embeddings of states into a metric space. The framework will connect the existing classical planning framework fast-downward [22], the probabilistic Monte-Carlo tree-searching framework PROST [29], the motion-planning framework by Plaku et al. [48, 49], and our own framework as used in [55, 10] that combines the double-oracle algorithm and domain-independent planners. The combined framework will facilitate easy experimentation with various planning engines, geometric representations, and heuristics. For a given planning or game-playing problem, the framework will create a geometric representation leveraging either linear programming, global optimization methods (e.g., simulated annealing [30]), or unsupervised learning (e.g., k-means clustering [37]). The created geometric representation will be used for the computation of a heuristics that exploits the geometric properties of the representation, such as the distance between the current and the goal state in some metric space. As shortest paths define a metric, we expect handling obstacles. The goals are

- to develop a framework for efficient testing of geometric representations in planning and game playing domains.
- to construct geometric representations of planning problems or game problems from textual descriptions.
- to develop novel planning algorithms that exploit the spatial representations using geometry-based heuristics.
- to investigate theoretical properties of the geometric representations in terms of scalability in time and space.

Geometric Adversarial Planning Framework

Towards the workbench of geometric adversarial planning we see the following steps.

Formalization and Transformation into Various Geometric Representations

We generally assume that the world is known in the initial state and the conditions to hold in the goal state. This may change dynamically, due to the application of actions in the real space. For each geometric representation we will provide an algorithm that takes a planning or game problem in this representation as the input. The output will be a state space embedding into a particular metric space. Each approach will be specific to the given kind of geometric representation, e.g., there is an inspiring related canonical representation for Presburger arithmetics [5], where state sets, operators and goal are represented as minimized automata.

Modeling 2D and 3D Environments Path finding and inspection in a *polygonial world* (2D) or in *meshes* (3D) has been a topic for a long time in computational geometry [13], with different computational results, wrt. the structure of the environment model. In the case of a grid or voxel representation, specialized discretizations apply. In a first step, point clouds may have to be moved into bounding boxes and matched with other sensor information. Starting with the work of Savas [58] we will set up benchmarks in a PDDL2.1-type language with the feature of specifying a geometric environment and actions that consist of geometric precondition and effects.

From Linear Programming to Priced Timed Automata In order to construct geometric representations, we will leverage the following methods: (a) linear programming (b) global optimization methods (e.g. simulated annealing [30]) (c) unsupervised learning (e.g. k-means clustering [37]). (d) reinforcement learning [36]. There is an additional link of geometric planning to (priced) timed automata like in UPAAL (Tiga) [8] that reasons about reachability in polytopes, with symbolic states. Shortest-path reduction canonicalize the symbolic state representation as simple temporal networks, also used in Optic [3]. In terms of adversaries in input models we think of a network of timed game automata, where edges are marked either controllable or uncontrollable. This defines a two-player game with on one side the controller (mastering the controllable edges) and on the other side the environment (mastering the uncontrollable edges). Winning conditions of the game are specified through temporal formulae.

Geometric Planning Heuristics Inspired by potential heuristics [51, 50] and our experience from developing geometric representation [54, 54, 38], we will develop various geometric representation of planning heuristics. The key idea is abstraction on the geometric objects which corresponds to coarsen the representation accuracy. We expect to be able to unify the heuristic parts from different part of the state representation, logic (proposition) as in FF [25], numerical (fluents) as in Metric-FF [24] with geometry abstractions.

Framework Implementation

In this work package we will design and implement our planning software framework *GeoPlanBench* that will act as the *workhorse*, and connect the existing planning framework fast-downward [22], the metric framework Optic [3], and the control parameter extension in Popcorn [58, 57], probabilistic Monte Carlo Tree search planning framework PROST [29], and the framework used in [55, 10] that combines the double-oracle algorithm with domain-independent planners. The planning framework will consist of multiple parts, and should align with the unified planning framework currently proposed in AIPlan4EU.

Preprocessing and Partial Grounding

This part takes the planning or game problem described using either PDDL [39] language or RDDL [56]. Then, it will extract a geometric (symbolic state) representation, and combine it into a mixed logical, numerical and geometric representation of the problem. Alternatively we use lifted geometric inputs in the problem description, which are either fully or partially grounded.

Heuristic Search Algorithms. This part will be closely integrated within a planner. The heuristic search algorithms like A*, IDA*, enforced-hill climbing and friends, will need to be provided a heuristic value of a given state using the geometric properties of the mixed logical, numerical and geometric representation of the problem. We will separate the heuristics from these planning heuristic search algorithms as done in the fast-downward planner.

Benchmarks and Simulation.

We will execute experiments with IPC benchmarks, slighly modified adversarial versions, and automatically generated simulation problems.

Tests in IPC Domains and related Benchmarks

Each candidate for a good geometric representation will be evaluated in the experiment management framework on IPC domains [65] against state-of-the-art planners (e.g., as found in the fast-downward repository such as planning with abstractions or potential heuristics [51, 50]).

Tests on Adversarial Planning Domains

We will also test it on zero-sum game domains (e.g., unmanned aerial vehicle domain, taxi company competition domain from [55, 10]). We will write a specialized GUI for displaying the algorithms. In this part, we will aim towards multi-goal task-motion planning with multiple agents.

Test within Motion-Planning Framework

We will test the framework implementation in a simulated world of polygons and polytopes as provided in the framework of Erion Plaku [47], where the environment is specified in a textual input, the robot model as a generic simulated model and intersection via physics-based engines, preprocessed and some multi-step and possibly multi-goal plans are found and shown in a GUI.

Theoretical Studies

In this work package we will study theoretical properties of geometric representations that performed well in our empirical evaluations. For example, we will identify the critical geometric features, that contributes to the reduction of the computation time of the planners.

Correctness Considerations As with planning with mixed logical and geometrical representation, we are effectively dealing with infinite state spaces, so there always is the need to align the representation with correctness and convergence considerations. As with automata and canonical representation we are quickly entering language theory, one important question is: What basic operation like union, negation and intersection of geometric objects are closed in which formalism? What abstraction mechanism preserves admissibility?

Time Complexity Analyses Planning with atomic propositions is PSPACE-hard, with numbers even undecidable. It is also known from real-time model checking that stopwatch automata are undecidable, while rectangular and regular automata are not (see Tom Henzinger et al.'s paper: *What is decidable about hybrid automata?* [23]).

Space Complexity Analyses The research with state set representations in BDD representation has shown, that the amount of memory taken by the exploration is even more important than time. We assume that the geometric representations are succinct and save computational time, so the compression ratios are of interest.

Field Experiments

While most of the efforts included in developing algorithms is done in the simulation environment, in this WP we will execute plans on real robots. Our group has continuously growing experience in planning for robots. While ROS-Plan [7] includes many expressive plan formalisms, there are first work on robot planning in a motion planning framework. CTU have a fleet of robots, which are adequate for this research. In this WP we will look at some further robots and do some feasibility study towards a complete integration.

Digital-Twin Experiments Before we operate in real world, we require interfaces as in ROS-Plan for virtual environments. In this task we will work on robot-simulation software like Gazebo or, our current favorite, Unity, a game playing design studio that supports complex 2D and 3D models, collision detection, visibility considerations, as well as the automated generation of environnment models and the inclusion of rigid-body dynamics.

Experiments on Quadcopters For this task we will use the 30 quadcopters that we want to buy for conducting a nuclear plant mixed in- and outdoor combat scenario, with 15 attackers and 15 defenders on each side maybe teamed up with some ground robots. Beside geometric setting imposed by the building to defend, limits in perception as well as safety and energy consumption are additional side constraints to be considered.

Figure 3: Robotic Lab robots (top) and UAVs to be bought (bottom, e.g., DJI Mavic mini) for the experimental validation of the proposed solutions.

Experiments on Lab Robots Last, but not least we will execute field studies on real robots available to us via the robotics labs in larger geometric environments like subterranean or logistics scenarios. If we get an opportunity, we like to work with one of the robotic dogs *Spot* the robotics lab has bought from Boston Dynamics, but we will also be happy with the wheeled or legged robots that we have access to in our labs.

Our team has access to the FEE's computing infrastructure and the national computing infrastructure. CTU are experienced with complex robotic systems with the current highlight of the CTU-CRAS team in the *DARPA SubT Challenge*. (`http://robotics.fel.cvut.cz/cras/darpa-subt`)

For the adversarial planning field study, we will experiment with a fleet of 30 (mini-sized) quadcopters.

Conclusion

Integrated geometrical and task planning reasoning is essential for robotics, as robot motion planning alone does not serve long-term goals, and task planning alone cannot deal with the intrinsic challenges of robot geometry and motion. Motion planning is inherently geometrical, e.g., free-space decomposition methods like triangulations or trapezoidal maps, Minkowski sums, nearest neighbor search, randomized road maps, RRT-type and rubber-band algorithms, visibility polygons, sweep-lines algorithms for segment intersections, kd-trees for localization, or Voronoi diagrams. Moreover, the world is going to be populated with robots, and our solutions will have a tremendous impact on their abilities to plan. While modeling an adversarial is always a daunting aspect of competing resources, it got an unfortunate relevance with respect to recent changes to the political and military powers.

Adversarial geometric planning is important not only for AI and robotics, but can prove highly influential in the other billion-dollar industry of computer games by modeling non-playing characters. While some combat situations as in Starcraft and similar real-time strategy games might be better dealt with domain-dependent and deep learning approaches, the generality, potential, relevance, and impact of domain-independent symbolic planning joint up with geometric constraints should not be underestimated. It is essential for prototypes and reducing time to-market.

The scientific contribution of geometric planning aims to provide optimized solution of spatial routing problems with continuous model-free robotic applications. The primary motivation is planning, where the current solutions rely on sampling continuous domains into a finite set of values being addressed as variants of geometric route planning. Solving inspection and observation and delivery problems are holy grail in robotics applications like cleaning up a room in a cluttered environment, in warehousing, or finding oil leakages in underwater pipes. The ultimate motivation, however, is to lift the software towards the next generation of autonomous robots, and contribute surplus towards long-term autonomy in exploration and routing, but also logistics applications.

The optimal solution of the discretized problem does not guarantee an optimal solution to the original problem. Furthermore, heuristics usually provide better solutions, however, without any solution quality estimates. We aim to develop the fundamental blocks to assess the solution and provide quality guarantee. The model-free solvers should be general enough to open a wide range of possible applications. The established algorithmics will support further research in challenging optimization problems, and novel computational techniques will improve scalability of nowadays and future algorithms to solve large instances. The proposed research will fertilize the deployments of robotic systems in various fields. Finally, stability analysis, identifying stability regions, and finding robust solutions under perturbations are important steps towards applications with dynamic and on-demand changes.

References

[1] Mitchell Ai-Chang, John L. Bresina, Leonard Charest, Adam Chase, Jennifer Cheng-jung Hsu, Ari K. Jónsson, Bob Kanefsky, Paul H. Morris, Kanna Rajan, Jeffrey Yglesias, Brian G. Chafin, William C. Dias, and Pierre F. Maldague. MAPGEN: mixed-initiative planning and scheduling for the mars exploration rover mission. *IEEE Intelligent Systems*, 19(1):8–12, 2004.

[2] Carol Applegate, Christopher Elsaesser, and James C. Sanborn. An architecture for adversarial planning. *IEEE Trans. Systems, Man, and Cybernetics*, 20(1):186–194, 1990.

[3] J. Benton, Amanda Jane Coles, and Andrew Coles. Temporal planning with preferences and time-dependent continuous costs. In Lee McCluskey, Brian Charles Williams, José Reinaldo Silva, and Blai Bonet, editors, *ICAPS*. AAAI, 2012.

[4] D. E. Bernard, E. B. Gamble, N. F. Rouquette, B. Smith, Y. W. Tung, N. Muscettola, G. A. Dorias, B. Kanefsky, J. Kurien, W. Millar, P. Nayal, K. Rajan, and W. Taylor. Remote agent experiment ds1 technology validation report. Technical report, Ames Research Center and JPL, 2000.

[5] Björn Ulrich Borowsky and Stefan Edelkamp. Optimal metric planning with state sets in automata representation. In Dieter Fox and Carla P. Gomes, editors, *AAAI*, pages 874–879. AAAI Press, 2008.

[6] Stéphane Cambon, Rachid Alami, and Fabien Gravot. A hybrid approach to intricate motion, manipulation and task planning. *Int. J. Robotics Res.*, 28(1):104–126, 2009.

[7] Michael Cashmore, Maria Fox, Derek Long, Daniele Magazzeni, Bram Ridder, Arnau Carrera, Narcís Palomeras, Natàlia Hurtós, and Marc Carreras. Rosplan: Planning in the robot operating system. In *ICAPS*, pages 333–341. AAAI Press, 2015.

[8] Franck Cassez, Alexandre David, Emmanuel Fleury, Kim Guldstrand Larsen, and Didier Lime. Efficient on-the-fly algorithms for the analysis of timed games. In Martín Abadi and Luca de Alfaro, editors, *CONCUR*, volume 3653 of *LNCS*, pages 66–80. Springer, 2005.

[9] Lukas Chrpa, Pavel Rytíř, and Rostislav Horcik. Planning against adversary in zero-sum games: Heuristics for selecting and ordering critical actions. In *Proceedings of the 13th Annual Symposium on Combinatorial Search (SoCS 2020)*, pages 1–1, 2020.

[10] Lukáš Chrpa, Pavel Rytíř, and Rostislav Horčík. Planning against adversary in zero-sum games: Heuristics for selecting and ordering critical actions. In *the 13th International Symposium on Combinatorial Search (SoCS 2020)*, 2020.

[11] Alessandro Cimatti, Marco Pistore, Marco Roveri, and Paolo Traverso. Weak, strong, and strong cyclic planning via symbolic model checking. *Artif. Intell.*, 147(1-2):35–84, 2003.

[12] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, second edition, 2000.

[13] M. de Berga, J. Gudmundssonb, M. J. Katzc, C. Levcopoulosd, M. H. Overmarse, and A. F. van der Stappen. TSP with neighborhoods of varying size. *Journal of Algorithms*, 57(1):22–36, 2005.

[14] Christian Dornhege, Patrick Eyerich, Thomas Keller, Sebastian Trüg, Michael Brenner, and Bernhard Nebel. Semantic attachments for domain-independent planning systems. In *ICAPS*. AAAI, 2009.

[15] Christian Dornhege, Patrick Eyerich, Thomas Keller, Sebastian Trüg, Michael Brenner, and Bernhard Nebel. Semantic attachments for domain-independent planning systems. In *Towards Service Robots for Everyday Environments*, volume 76 of *Springer Tracts in Advanced Robotics*, pages 99–115. Springer, 2012.

[16] Esra Erdem, Kadir Haspalamutgil, Can Palaz, Volkan Patoglu, and Tansel Uras. Combining high-level causal reasoning with low-level geometric reasoning and motion planning for robotic manipulation. In *ICRA*, pages 4575–4581. IEEE, 2011.

[17] C. Erdogan. *Planning in constraint space for multi-body manipulation tasks*. PhD thesis, Georgia Institute of Technology, 2016.

[18] Can Erdogan and Mike Stilman. Planning in constraint space: Automated design of functional structures. In *ICRA*, 2013.

[19] Fei Fang, Peter Stone, and Milind Tambe. When Security Games Go Green: Designing Defender Strategies to Prevent Poaching and Illegal Fishing. In *In Proceedings of 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[20] Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. FFRob: an efficient heuristic for task and motion planning. In H. Levent Akin, Nancy M. Amato, Volkan Isler, and A. Frank van der Stappen, editors, *WAFR*, volume 107 of *Springer Tracts in Advanced Robotics*, pages 179–195. Springer, 2014.

[21] Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. FFRob: leveraging symbolic planning for efficient task and motion planning. *Int. J. Robotics Res.*, 37(1):104–136, 2018.

[22] Malte Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006.

[23] Thomas A. Henzinger, Peter W. Kopke, Anuj Puri, and Pravin Varaiya. What's decidable about hybrid automata? *Journal of Computer and System Sciences*, 57(1):94–124, 1998.

[24] Jörg Hoffmann. The Metric-FF planning system: Translating "ignoring delete lists" to numeric state variables. *CoRR*, abs/1106.5271, 2011.

[25] Jörg Hoffmann and Bernhard Nebel. The FF planning system: Fast plan generation through heuristic search. *J. Artif. Intell. Res.*, 14:253–302, 2001.

[26] R.M. Jensen, M.M. Veloso, and M.H. Bowling. Obdd-based optimistic and strong cyclic adversarial planning. In *ECP*, pages 265–276, 2001.

[27] Niels Justesen, Tobias Mahlmann, Sebastian Risi, and Julian Togelius. Playing multiaction adversarial games: Online evolutionary planning versus tree search. *IEEE Trans. Games*, 10(3):281–291, 2018.

[28] Niels Justesen and Sebastian Risi. Continual online evolutionary planning for in-game build order adaptation in starcraft. In *The Genetic and Evolutionary Computation Conference, GECCO 2017*, pages 187–194, 2017.

[29] Thomas Keller and Patrick Eyerich. Prost: Probabilistic planning based on uct. In *Proceedings of the Twenty-Second International Conference on International Conference on Automated Planning and Scheduling*, ICAPS'12, page 119–127. AAAI Press, 2012.

[30] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[31] Peter Kissmann and Stefan Edelkamp. Solving fully-observable non-deterministic planning problems via translation into a general game. In *KI*, volume 5803, pages 1–8. Springer, 2009.

[32] Richard Korf. A jeep crossing a desert of unknown width. *American Mathematical Monthly*, (Accepted), 2022.

[33] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 4190–4203, 2017.

[34] Duong Le and Erion Plaku. Multi-robot motion planning with dynamics via coordinated sampling-based expansion guided by multi-agent search. *IEEE Robotics Autom. Lett.*, 4(2):1868–1875, 2019.

[35] Joshua Letchford and Yevgeniy Vorobeychik. Optimal interdiction of attack plans. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13*, pages 199–206. IFAAMAS, 2013.

[36] Michael L. Littman. Reinforcement learning improves behaviour from evaluative feedback. *Nature*, 521(7553):445–451, May 2015.

[37] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[38] Martin Loebl. Binary linear codes, dimers and hypermatrices. *Electronic Notes in Discrete Mathematics*, 59:19 – 35, 2017.

[39] Drew McDermott, Malik Ghallab, Craig Howe, Adele; Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. PDDL–the planning domain definition language. Technical Report TR98003/DCS TR1165, CT: Yale Center for Computational Vision and Control, 1998.

[40] H. Brendan McMahan, Geoffrey J. Gordon, and Avrim Blum. Planning in the Presence of Cost Functions Controlled by an Adversary. In *ICML*, pages 536–543, 2003.

[41] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015.

[42] Dana Nau, Malik Ghallab, and Paolo Traverso. *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[43] F. A. Oliehoek, R. Savani, J. Gallego-Posada, E. van der Pol, E. D. de Jong, and R. Gross. GANGs: Generative Adversarial Network Games. *ArXiv e-prints*, 2017.

[44] Stephen M. Omohundro. Five balltree construction algorithms. Technical report, ICSI, Berkeley, 1989.

[45] Florian Pantke, Stefan Edelkamp, and Otthein Herzog. Planning with numeric key performance indicators over dynamic organizations of intelligent agents. In Jörg P. Müller, Michael Weyrich, and Ana L. C. Bazzan, editors, *MATES*, volume 8732 of *LNAI/LNCS*, pages 138–155. Springer, 2014.

[46] Diego Perez, Philipp Rohlfshagen, and Simon M. Lucas. The physical travelling salesman problem: Wcci 2012 competition. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, 2012.

[47] E. Plaku, L.E. Kavraki, and M.Y. Vardi. Motion planning with dynamics by a synergistic combination of layers of planning. *IEEE Transactions on Robotics*, 26(3):469–482, 2010.

[48] Erion Plaku. Region-guided and sampling-based tree search for motion planning with dynamics. *IEEE Transactions on Robotics*, 31(3):723–735, 2015.

[49] Erion Plaku, Sara Rashidian, and Stefan Edelkamp. Multi-group motion planning in virtual environments. *Computer Animation and Virtual Worlds*, 29(6):e1688, 2018.

[50] Florian Pommerening, Malte Helmert, and Blai Bonet. Higher-dimensional potential heuristics for optimal classical planning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3636–3643. AAAI Press, 2017.

[51] Florian Pommerening, Malte Helmert, Gabriele Röger, and Jendrik Seipp. From non-negative to general operator cost partitioning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 3335–3341. AAAI Press, 2015.

[52] Alberto Pozanco, Yolanda E-Martín, Susana Fernández, and Daniel Borrajo. Counterplanning using goal recognition and landmarks. In *The Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 4808–4814, 2018.

[53] Pavel Rytíř. Geometric representations of binary codes and computation of weight enumerators. *Advances in Applied Mathematics*, 45(2):290 – 301, 2010.

[54] Pavel Rytíř. Geometric representations of linear codes. *Advances in Mathematics*, 282:1 – 22, 2015.

[55] Pavel Rytíř, Lukáš Chrpa, and Branislav Bošanský. Using classical planning in adversarial problems. In *Proceedings of the 31st IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1327–1332, 2019.

[56] Scott Sanner. Relational dynamic influence diagram language (RDDL): Language description, 2010.

[57] Emre Savas, Maria Fox, Derek Long, and Daniele Magazzeni. Planning using actions with control parameters. In *ECAI*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 1185–1193. IOS Press, 2016.

[58] O. E. Savas. *Temporal-Numeric Planning wit Control Parameters*. PhD thesis, King's College London, 2018.

[59] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, February 2009.

[60] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.

[61] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[62] Patrick Speicher, Marcel Steinmetz, Michael Backes, Jörg Hoffmann, and Robert Künnemann. Stackelberg planning: Towards effective leader-follower state space search. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[63] Siddharth Srivastava, Eugene Fang, Lorenzo Riano, Rohan Chitnis, Stuart J. Russell, and Pieter Abbeel. Combined task and motion planning through an extensible planner-independent interface layer. In *ICRA*, pages 639–646. IEEE, 2014.

[64] Milind Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.

[65] Mauro Vallati, Lukás Chrpa, and Thomas Leo McCluskey. What you always wanted to know about the deterministic part of the international planning competition (IPC) 2014 (but were too afraid to ask). *Knowledge Eng. Review*, 33:e3, 2018.

[66] Oriol Vinyals, Igor Babuschkin, Wojciech Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John Agapiou, Max Jaderberg, and David Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575, 11 2019.

[67] Yevgeniy Vorobeychik and Michael Pritchard. Plan interdiction games. In *Adaptive Autonomous Secure Cyber Systems*, pages 159–182. Springer, 2020.

[68] Yufei Wang, Zheyuan Ryan Shi, Lantao Yu, Yi Wu, Rohit Singh, Lucas Joppa, and Fei Fang. Deep reinforcement learning for green security games with real-time information. In *AAAI Conference on Artificial Intelligence*, 2019.