Australian Government
Department of Defence
Defence Science and
Technology Organisation

Automated Cyber Red Teaming

Joseph Yuen

Cyber and Electronic Warfare Division
Defence Science and Technology Organisation

DSTO-TN-1420

ABSTRACT

Cyber Red Teaming (CRT) is an important exercise to conduct for Defence agencies built on large technological infrastructures. Their size and relative importance may make them high priority targets for criminal organizations, issue-motivated groups and even foreign governments that are increasingly capable and willing to use technology for intelligence gathering. However, identifying a viable attack can be a time-consuming process for human analysts, and so Automated Planners are being considered as a viable method of discovering possible attack paths for CRT.

This report surveys the current state-of-the-art planning techniques, tools and frameworks, their performance at international competitions, and by comparing their performance against the operational requirements and limitations of CRT problems, recommend the most suitable ones for trialling.

RELEASE LIMITATION

Approved for public release

Published by

Cyber and Electronic Warfare Division
DSTO Defence Science and Technology Organisation
PO Box 1500
Edinburgh South Australia 5111 Australia

Telephone: 1300 333 363
Fax: (08) 7389 6567

AR-016-282
April 2015

APPROVED FOR PUBLIC RELEASE

Automated Cyber Red Teaming

Executive Summary

Cyber Red Teaming (CRT) is a common activity performed within large organisations to assess how susceptible their infrastructure, business processes and staff are to attacks from cyber-enabled adversaries. CRT involves drafting attack plans that could succeed on the current state of the organisation, optional attack execution/simulation, and impact analysis. The results are then used to develop mitigation strategies and countermeasures. As the attack plan drafting step can be a time-consuming process, the use of Automated Planners, Artificial Intelligence algorithms that generate problem-specific plans, is suggested to help reduce the cost of the overall exercise.

There are 3 major categories of Automated Planners: state-space planners, planning graph planners and hierarchical task network-based planners. State-space planning is in essence classical path-finding algorithms like breadth first search and A*, with added heuristics for more informed exploration. Planning graph planners converts a planning problem into planning graphs: data structure that compactly represents all "possible futures" for a given problem in stepped layers, and explores this graph to identify the earliest layer where a goal satisfying state is found. Hierarchical task network techniques use expert-designed plan templates to constrain the search to viable strategies, and focuses on searching for multiple plans within that range.

Other planning approaches exist, but they are not competitive with the three already mentioned in terms of efficiency and success. Additional techniques can also be applied on top of the basic planning algorithms, such as use of machine learning to guide the planning and factoring in uncertainty.

Using the benchmarking results from the International Planning Competition, we identified that algorithms and tools that use planning graphs are currently best suited to the Defence Cyber context, as they scale better computationally for larger scenarios and generating longer attack plans within reasonable time.

In particular, Portfolio-based Planning (PbP), a parallel computing framework which utilises a library of planning techniques and learns suitable portfolio configurations for each problem, has proven to be a promising off-the-shelf tool for planning. We conclude that future CRT exercises should consider trialling PbP.

This page is intentionally blank

Contents

Glossary

AI: Artificial Intelligence

CRT: Cyber Red Teaming

DDoS: Distributed Denial of Service

DFH: Delete-Free Heuristics

FD: Fast Downward

HTN: Hierarchical Task Network

ICAPS: International Conference on Automated Planning and Scheduling

IP: Internet Protocol

IPC: International Planning Competition

MDP: Markov Decision Process

NAT: Network Address Translation

NPC: Non-Player Character

PbP: Portfolio-based Planning

PDDL: Planning Domain Description Language

POMDP: Partially Observable Markov Decision Process

SAT: Satisfiability

SQL: Structured Query Language

STRIPS: Stanford Research Institute Problem Solver

TTP: Tactics, Techniques and Procedures

1. Introduction

Cyber Red Teaming (CRT) is an important exercise to conduct for Defence agencies built on large technological infrastructures [1]. Their size and relative importance may make them high priority targets for criminal organizations, issue-motivated groups and even foreign governments that are increasingly capable and willing to use technology for intelligence gathering.

Stated simply, CRT exercises determine the vulnerabilities that affect one's cyber system by searching for viable attack plans[1], and examining its effect on the system. It is a labour-intensive exercise as it typically involves specialist human analysts and operators to draft attack plans. Due to the dynamic nature of cyber environments, some findings from these exercises can quickly become invalid, which means they should be conducted frequently. Automation of the exercise would allow an organization to discover potential vulnerabilities more cost effectively, which in turn will permit more resources to be allocated towards mitigation and countermeasures.

This technical note introduces options available for automating CRT, specifically through the application of automated planning techniques. It is intended to provoke thoughts for people wanting to use automated planners for CRT. Readers of this note are expected to have some background in computer science, and exposure to general artificial intelligence concepts [2] will be beneficial. Familiarity with automated planning is not necessary.

The rest of the paper is organized as follows: we discuss what the CRT problem is, and provide guiding principles for modelling CRT scenarios into planning problems. We then introduce what automated planning is, discuss several modern approaches, and consider the applicability of automated planning to CRT problems. Finally, we recommend several state-of-the-art planning tools for trial and, more generally, when it is suitable to use automated planners in support of CRT exercises based on the current requirements.

Details of the implementation and complexity analysis of various planning algorithms, tools and frameworks will not be discussed here. It is best to consider this note as a pointer to other literature that may be more relevant in specific cases for which an extensive set of references is included (see Appendix A).

[1] For the purposes of this literature review, an attack plan is defined as a sequence of actions which, if taken by a person and/or a computer, could harm the target organization.

2. Cyber Red Teaming

CRT is a term often used interchangeably, though sometimes inaccurately, with penetration testing and vulnerability assessment. While CRT is an exercise in finding possible vectors for attack, penetration testing is an exercise in actually attacking the system. Vulnerability assessment on the other hand is about analysing software and exposing coding flaws which can be exploited.

Vulnerability assessment is conceptually similar to CRT but studies mostly individual software. It lacks the broader view of the system as a whole, focusing more on code flaws and less on system configuration and business processes [3]. And while the outcome of penetration testing has the same practical implications on a system as CRT, the attack vectors are very narrow and often doesn't say much about the system overall.

This section discusses modelling of an adversary's characteristics and behaviours (red teaming), modelling cyber infrastructure from a systemic perspective, attack plan construction via simulation, and the issues related to conducting CRT. This will help show how CRT involves aspects of both penetration testing and vulnerability assessment, but is able to draft attack plans that utilise multiple vulnerabilities across the system rather than isolated ones.

2.1 The World Model

In CRT terms, the overall system that is being red-teamed is commonly referred to as the World Model [4] [5]. This naming captures the idea that cyber systems are large, complex digital ecosystems with many intelligent entities sharing and consuming resources. It also alludes to the practice of modelling and simulating attacks in a test environment as opposed to running the exercise in a live, production environment. For our purposes, we divide the World Model into two parts: the adversary and the environment they target.

The Adversary

Not all adversaries are equal. Each adversary has a specific set of Tactics, Techniques and Procedures (TTPs); some are more resourceful and better resourced than others. Others may have very specific objectives when attacking an organization. Below is a non-exhaustive set of questions regarding an adversary one could and should ask in constructing a Red Teaming agent that represents them:

- **The adversary's target**: who or what are they after? What access do they have into various parts of the system?
- **The adversary's offensive capabilities**: this includes their TTPs, computational resources and domain knowledge.
- **The adversary's restrictions**: limited time windows for attacking, anonymity, visibility of the network etc.

- **The adversary's behaviour patterns**: have they attacked before? Previous attack patterns and targets may be indicative of future ones.

Accurately representing an adversary in a CRT exercise will make the proposed attack plans more relevant, and also affect the usefulness and reliability of the results when used to assess the actual system.

The Target Environment

A real computer network for an organization such as the US Department of Defence is generally very large, dynamic and complex [6] . Accurately modelling and simulating such a network for the purposes of CRT is the responsibility of the exercise creator, and may require applying abstractions or assumptions. Below are some guiding questions in support of building a problem-specific World Model:

- **What entities are there in the World Model?** This may include computers, users, software, routers, encrypted storage, network policies and more.
- **What are the relationships between entities?** Examples include "User A has an account on Computer B" and "Computer X has Software Y installed".
- **What are the World's dynamics?** Some system behaviour occurs independently from the adversary's actions.
- **Which parts of the World are relevant**? It is better to have concise system representation that pertains to the adversary's target to reduce unnecessary exploration [7].
- **Which parts of the World are visible**? Not all aspects of the system, even those that are relevant, may be visible to an adversary, even from another part of the system.

2.2 Attack Plan Generation

After describing a World Model, attack plans can then be drafted in accordance with an adversary's TTP set through simulated execution of the plan on the model. Each attack plan may include general coverage activities such as port scanning and IP ranging, or targeted actions such as sending a spear phishing email. Some attacks may also depend on specific responses from the target machine or user. Through simulation, damage assessment and mitigation planning based on the attack effects can be estimated.

There are often numerous possible attack plans conceived during a CRT exercise. The red team (exercise runners playing the role of an adversary) chooses which of these attacks to attempt first using one or more of the following factors:

- **Concerns-based**: prioritise attacks that are most concerning (to the organisation)
- **Success-based**: prioritise attacks that are most likely to succeed
- **Cost-based**: prioritise attacks that consume the least resources
- **Impact-based**: prioritise attacks that are most damaging if successful

- **Opportunity-based**: prioritise attacks that are relevant to certain situations
- **Verification-based**: prioritise attacks that have been dealt with before. This may be to verify that the protections/countermeasures already put in place are working as expected.

The level of plan abstraction is another consideration when preparing for a CRT exercise. If the exercise is only a thought experiment, a plan describing attack patterns may suffice, whereas an executable attack plan will require step-by-step details. Regardless, selecting the most suitable approach will help ensure that the red teaming exercises conducted meets the priorities of the organization they are conducted for.

2.3 Issues and Challenges

Planning and conducting a CRT exercise may face a variety of practical challenges, some of which cannot be remedied and may require changes to the exercise:

- **Limited Resources**: the time and computational cost to conduct certain exercises may be infeasible, such as the data-mining needed to conduct spear phishing, simulating a DDoS attack etc.
- **Asymmetric Threat**: adversaries may have TTPs that are beyond an organization's own capabilities, thus limiting its ability to detect or defend such attacks.
- **Reactivity**: As some forms of cyber-attacks occur and are completed in a matter of milliseconds, the situation assessment may need to be done in near real-time, online, continuously, within the production environment, and with no human-in-the-loop. This means the CRT exercise will have to employ computationally fast techniques which may entail loss of precision, if such techniques exist at all.
- **Model Complexity**: As computer network size grows, so does its complexity [8]. Because of this, modelling a large computer network realistically may prove to be a difficult or even impossible undertaking.
- **Model Incompleteness**: Having no known vulnerabilities doesn't mean there are no vulnerabilities. Modelling an adversary requires detailed and current intelligence on them, which may not always be available. In such cases CRT is only a best effort to replicate the real threat or the real environment.

These are challenges that limit what conclusions can be made from CRT exercises, and it is the objective of continued research and technology improvement to improve efficiency and automation of operating under practical limitations.

3. Automated Planning

Automated planning is a branch of Artificial Intelligence (AI) that is concerned with generation of plans [9]. The planner is tasked with answering one question: given a set of possible actions, an initial state, and some goals, can a sequence of these actions be found such that their execution will transition the system from the initial state into a goal state?

Automated planners are most popularly used in logistics [10] [11], scheduling [12], robotics [13] and computer game engines [14]. Most automated planners are general purpose and can be used to solve planning problems in a variety of domains [15]. However, the performance of different planning tools and techniques can vary depending on the planning problem itself [16].

The remainder of this section discusses these characteristics with respect to CRT, and how modern planning techniques can be used to solve CRT problems. Nau [9] provides more details regarding related theory and technique implementation.

3.1 The Planning Problem

A planning problem has an initial state of a system, and by performing a sequence of actions, a goal state can be reached. Each planning problem is encoded for a specific domain (e.g. airport logistics), which may have specific types of objects (e.g. flights) and propositions (e.g. passenger P is checked in on flight QF123) not present in other domains. Some planning techniques are able to leverage specific traits of specific domains.

The Stanford Research Institute Problem Solver (STRIPS) semantics [17] and the Planning Domain Description Language (PDDL) [18] are the two most popular input languages for defining the problem state as well as the available library of actions. STRIPS has been around for much longer than PDDL, and is more prevalent in current industrial planning tools, while PDDL is newer, and was created primarily for benchmarking purposes.

In both planning languages, a state is represented by a discrete set of observable, first-class entities, referred to as objects[2]. Facts about these objects are referred to as the propositions[3]. Examples of objects in CRT include host machines, users, software, websites and services, and a proposition may be something like "user X has an admin account on host machine Y".

An action is defined by the following:

- **Preconditions**: the propositions that must be true in order to perform this action
- **Add Effects**: objects and propositions that are introduced into the state by taking this action

[2] Some literature on planning also uses the term "instances"
[3] The term "predicates" is sometimes used as well

- **Delete Effects**: objects and prepositions that are removed from the state by taking this action
- **Costs**: non-boolean, qualitative values of the state affected by this action.

In terms of CRT, the planning problem is to draft an attack plan. All entities and relationships within the World Model, including where the adversary sits on the network, would form the initial state of the system. The goal state will contain the changes to the World Model that meets the adversary's objective. The actions are the adversary's TTPs.

3.2 Planning Domain Characteristics

Not all planning problems are the same. Planning the drive to work and planning a winning chess strategy not only requires different sets of actions, but the environments in which the problems reside are also different. It is important to understand these differences, as specific techniques may be more suited to specific domains.

According to Nau [9] and Russell [2], a planning domain is characterised by the following:

- **Observability**: a system is fully observable if every object and preposition of its current state is known to the planner. Otherwise it is considered only partially observable. In CRT, this observability relates to the visibility of a network environment from the perspective of the adversary.
- **Determinism**: are the effects of agent actions on the state predictable?
- **Dynamics**: does the state of the system change independent of agent action/plan execution?
- **Temporality**: does the time taken to complete actions matter?
- **Granularity**: are action decisions, effects and costs discrete or continuous values?

In some instances, there may also be problem-specific requirements:

- **Library**: what actions are available to the planner for a particular scenario?
- **Optimality**: do we want lowest cost plan (optimal) or just any valid plan (satisficing)?
- **Ordering**: are we generating totally ordered or partially ordered plans? Partially ordered plans allow for contingencies where actions are non-deterministic, or where the system is dynamic.
- **Preferences**: costs associated with a plan may be relevant, which affect the actions we prefer.
- **Extended goals and constraints**: are there certain states we don't want to pass through during plan execution? In other words, do we need to ensure parts of the system are unaffected by our attack?
- **Performance**: does the planning need to be done in real-time or is doing it offline acceptable?

It is critical to select a planning technique suitable to the planning problem's domain, but care should also be taken to avoid over-estimating these requirements. For instance, the

computational effort required to guarantee that a plan is optimal can exceed the effort to reach a sub-optimal alternative by several orders of magnitude [19]. If the optimal plan is not required, we don't need to select an algorithm with such a capability, as it may conflict with other requirements such as reactivity for real-time, operational use.

The most common type of domain tested in academia is a planning domain that is fully observable, deterministic and has a static system [20]. This is referred to as the classical planning problem. There is a view within the automated planning community that, with some model-mapping work, the currently more efficient and reliable classical planning techniques can be used to solve non-classical planning problems [21], but in many real-world domains performing such a mapping is hard. As such, some level of abstraction is needed when using these planners in practice.

3.3 State of the art Automated Planning

Historically, most planning techniques and algorithms were designed with the computing power of their time as a feasibility constraint [22]. As such they were seldom scalable solutions, and found little audience outside of academia. This is until DARPA ran a special workshop in 1990 [23], which pushed researchers away from worrying about computing power, and more towards developing scalable techniques and real-world applications.

CRT exercises are typically large scale, dynamic planning problems in partially observable environments [24] [25], which make many of the older planners unsuitable as they don't take advantage of the increased computational power available today.

This section explores a variety of approaches that are used by modern automated planning tools. The goal is to convey a basic understanding of the approaches and their associated strengths and weaknesses. Appendix A contains a table comparing implementations of the various planning techniques, and can be used as a catalogue for exploring planning tools beyond the ones recommended in this report.

3.3.1 State-Space Planning

State-Space Planners [9] model the planning problem as a directed graph where nodes represent possible states the system can be in, and arcs/edges are the actions that move the system from one state to another. Once the graph is constructed, it becomes a path-finding problem to generate the plan. Existing AI graph search algorithms such as Iterative Deepening A* Search [26] can be leveraged for this step of the planning. The final plan is represented by the path between the initial state node and the goal state node. Figure 1 shows part of a state-space graph.

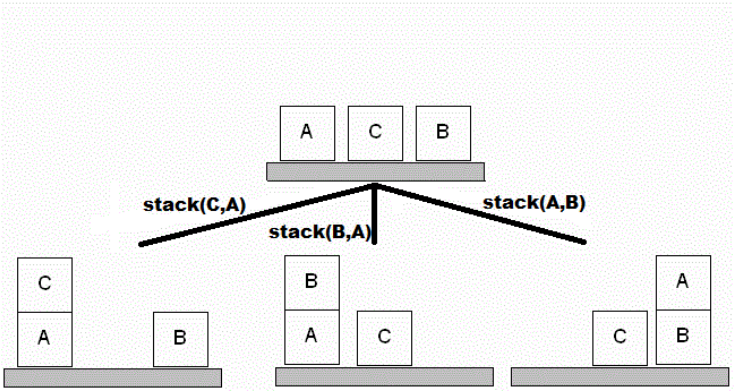

Figure 1 - Example of one expansion step of the State-Space for a block stacking problem.

The state-space planning approach is simple and has been shown to be very effective in planning for many domains [9]. As such, it is one of the most popular and enduring techniques for real-world planning problems. Many implementations of general purpose state-space planners exist [27] [28], and solutions engineered towards specific problem domains have also been developed [29] [10].

State-space planners are most effective for problems with fully observable, deterministic environments, and handle preferences and constraints well during plan generation [15]. Most have the option of providing completeness and optimality guarantees, and can be highly engineered by using domain-specific heuristics to guide the search. State-space graphs scale poorly for problems in dynamic systems and non-deterministic world models due to state-space explosion inherent in the branching factor, which is dependent on the number of valid actions, parameters and objects at any given step.

The majority of the current research in state-space planning is focused on pruning the search space for planning under uncertainty [30] [31] [32], and leveraging the relatively new concept of Delete-Free Heuristics (DFH) [33] [34]. When calculating the heuristic value in a DFH, the planner pretends that actions have no delete effects during the expansion phase. This speeds up the initial search, and deletions can be applied later for plan validity check.

Many state-space planners use DFH, but Katz and Hoffman points out in [35] that this form of relaxation actually slows the search when planning in an environment with non-replenishable resources. This may be a problem for CRT, as actions may involve disabling of network nodes or services as part of a larger or longer attack, which means use of DFH may not be recommended.

3.3.2 Planning Graphs

Planning Graphs are data structures that capture both the set of possible and impossible states a system can be in by keeping track of *mutexes*: pairs of observations that cannot be true at the same time [36]. For instance, an airline system can be in a state where two of its

planes are in the same airport, but it cannot be in a state where one plane is in two different airports at the same time.

Structured as a directed, layered graph, nodes in a Planning Graph are either an *action* or *propositional* node, belonging to an action or propositional layer respectively. The graph grows from the initial state, and expands into the next layer based on the action library.

The action layer represents the full set of possible actions that could be taken at a given step, while the propositional layer represents the set of states the system could be in after any of those actions are taken. Arcs represent either the action-effect or precondition-action relationships between action and propositional nodes.

The goal state is reachable when we arrive at a propositional layer where all the required propositions are true. Then the flow from the initial state to this layer will be the shortest partially-ordered plan.

Planning Graphs can be examined directly to generate the plan, and are often integrated into state-space planners such as Hoffmann's FastForward family [28]. Planning graphs are much smaller in size than equivalent state-space graphs; they only grow polynomial with respect to search depth. However, planning graphs are ultimately forward search planners [37]. As such, there is an unavoidable overhead in generating a planning graph for every new problem.

Many state-of-the-art planners make use of planning graphs [36] [38] [39] [40] [41] [42], because they allow for scalable plan generation with respect to plan length and action library size, and they provide implicit cycle detection of search paths since each layer of the graph compactly represents all possibilities after n steps. These graphs can also be deployed for dynamic systems to generate a complete representation that includes action effects on all the possible worlds [27]. However, the structure can be hard to comprehend by human inspection for larger problems, and it remains the responsibility of the algorithm to understand and extract the plan.

3.3.3 Hierarchical Task Networks

Hierarchical Task Networks are another data structure that has been popular in applied automated planning [43] [16]. Unlike planning graphs, which are driven by the relationship between states, HTNs are driven by the relationship between actions. Instead of trying to reach a goal state through graph search, HTNs view the planning problem as trying to perform a task which can be decomposed into smaller tasks, where the subtasks at the atomic level are called "primitive tasks" or "operators".

A primitive task has preconditions that are satisfied by *binding* instances, the objects in the system, to them. A precondition is a predicate of the state. Primitive tasks represent the actions that change the system, and a valid plan is generated when all primitive tasks preconditions inside a task network are satisfied through valid variable bindings. Some HTN planners also allow non-primitive tasks to have their own preconditions, which may reduce the number of precondition tests at the atomic level.

Unlike state-space planning and planning graphs, where only the basic operators need to be encoded, HTNs require a domain expert to determine and encode the task structures which the HTN planners can then use [44]. In doing so, many dead-ends plans can be avoided since every plan under the HTN is valid, and the main effort becomes finding a set of bindings that is relevant given the initial state. The bindings do not necessarily need to be grounded; as long as a particular set of variable types and relationships are true in the initial state of the planning problem, plan validity can be inferred.

Because of its efficiency compared to state-space planners, planning with HTN is by far the most popular technique in terms of industry application, having been used in robotics [13] [45], NPC scripting for computer games [14] [46], manufacturing, logistics and scheduling.

The effectiveness and comprehensiveness of HTNs depend on the human encoding the task structures. In the case of CRT, HTNs may only be applicable for validation-based and success-based exercises, where the attack vectors are already established. HTNs have been used for CRT in a Defence context [47].

3.3.4 Machine Learning

While Machine Learning theory and techniques have been around for decades, they were only adopted by the automated planning community in the 90s as a way to enhance or improve the planning process [48]. Machine learning has been applied to automated planning in several ways: policy learning, parameter tuning, discovering macro actions and portfolio construction.

- **Policy Learning** [49] is about identifying potentially conflicting propositions (like mutexes from Planning Graphs), and developing problem-specific search policies to reduce dead ends. Planning problems that benefit from policy learning the most are ones whose state-space contain many dead ends that stem from a small number of easily reached 'bad' states, where simply avoiding such states will make planning faster and more successful. An example in the airport scheduling domain would be to create a policy of not having any planes airborne for more than 24 hours consecutively, which will wear the hardware more slowly.

- **Parameter Tuning** is optimization of the weights and invariants associated with search heuristics for specific problems or problem classes. For example, some actions might be able to satisfy sub-goals that are difficult to reach, and achieving those sub-goals earlier may merit heavier weightings in the cost function. This in turn will allow the planner to favour those actions in the plan.

- **Macro Action Learning** [50] focuses on discovering and maintaining a library of useful plan structures (partial plans) that can help solve larger, harder planning problems in a specific domain. It is in essence automated construction of partial HTNs. Learning macro actions requires significant upfront training; a large sample of planning problems are needed to determine which partial structures are generally useful, and worth keeping in the macro action library.

- **Portfolio Construction** is a form of ensemble learning: the use of multiple algorithms to solve the current problems, and to predict future algorithm success for similar problems. Optimization occurs at the algorithm selection level, where an automated system will run one or more planners from a library of planners to solve a planning problem. The selection and configuration of planners depend on how well suited each technique is estimated to be by the system for the problem domain at hand. Like macro action learning, portfolio learning requires significant training to construct an accurate portfolio of each domain/problem set. Portfolio-based planners can however leverage distributed computing architectures by scheduling different planners on different processing units to run concurrently.

The main advantage of adding machine learning to automated planners is increased automation, reducing the work a human operator would need to do in planner reconfiguration, helping the planner adapt to the problems encountered in a domain at run time. However, machine learning carries some costs, most notably the training aspect. For maximum effectiveness, the training set must be representative of the planning problems that will be encountered. If this training is not available upfront, then the planner must be able to continue to learn over time.

3.3.5 Other Approaches

Many more planning techniques exist, however detailed discussion lies outside the scope of this note for a variety of reasons. Firstly, some have been superseded by one of the techniques discussed earlier. This is particularly true in terms of state-of-the-art performance, as will be shown later. Secondly, they are unsuited to the automation of CRT planning due to further abstractions needed to model and make these problems solvable. Such abstractions may render the generated attack plans meaningless or useless for vulnerability assessment and mitigation planning. Below is a selection of four of the more interesting of such techniques.

- **Plan-Space Planning** [51] is a graph traversal approach similar to state-space planning. However nodes represent partial plans instead of system state, and arcs represent plan refinement operations such as adding or removing actions from the plan. The algorithm generally starts from an initial plan that contains flaws, with the goal being plan refinement through flaw elimination. Compared to state-space planners, plan-space planners are computationally inefficient, and there is no domain-agnostic, systematic way to construct the initial plan.

- **Planning as Satisfiability** (or SAT planning) [52] encodes the planning problem as a Boolean satisfiability problem, then solves it using stochastic local search algorithms. SAT planning is popular for static systems such as electronic design automation, but is not applicable in dynamic systems like the problems faced in CRT.

- The **Markov Decision Process** (MDPs) [53] is a mathematical structure for representing actions in planning domains where the effects are non-deterministic.

The output of solving an MDP is a partial order plan containing conditional actions based on the observed effect from previous actions. There is current research into solving Partially Observable MDPs (POMDPs) for advanced applications in penetration testing [54], but solving MDPs has been shown to be intractable.

- **Model Checking** [55] is similar to SAT planning, but a custom planning model is created in order to determine whether a valid plan exists in theory before attempting to generate one. Majority of planners that do not follow the aforementioned approaches generally fall under this category.

Aside from planning techniques listed above, a wide variety of hybrid approaches exist, some of which are highly engineered to solve specific classes of planning problems.

The table below summarises the techniques are best suited for particular types of planning problems regardless of the domain they come from.

Problem trait	State-space planners	Planning Graphs	HTNs	Plan-space planners	SATPlan	MDPs
Action library size	Small	Medium	Large	Medium	Small	Small
Observability	Full	Full	Full	Full	Full	Partial
Determinism	Deterministic	Deterministic	Deterministic	Deterministic	Deterministic	Probabilistic
Dynamic states	Yes	Maybe	No	No	No	Yes
Plan Optimality	Any	Any	Any	Satisficing	Satisficing	Satisficing
Plan Ordering	Any	Any	Totally Ordered	Partially Ordered	Totally Ordered	Totally Ordered
Preference handling	Supported	Supported	Add-on	Supported	Supported	Supported
Constraint handling	Yes	Yes	Yes	Yes	Yes	No

3.4 Benchmarking via the International Planning Competition

While describing the various fundamental planning approaches above, a number of assertions were made with respect to their success rate in solving planning problems as well as how efficiently they did so. The quantitative comparisons were largely based on results from the International Planning Competitions (IPCs) [56] [57] [15] [58] [59] [60] [61] [62] [20] [63] [64].

The IPC, which is run in conjunction with the International Conference on Automated Planning and Scheduling (ICAPS), has been benchmarking automated planners since 1998 in an attempt to quantitatively measure the current state of the art.

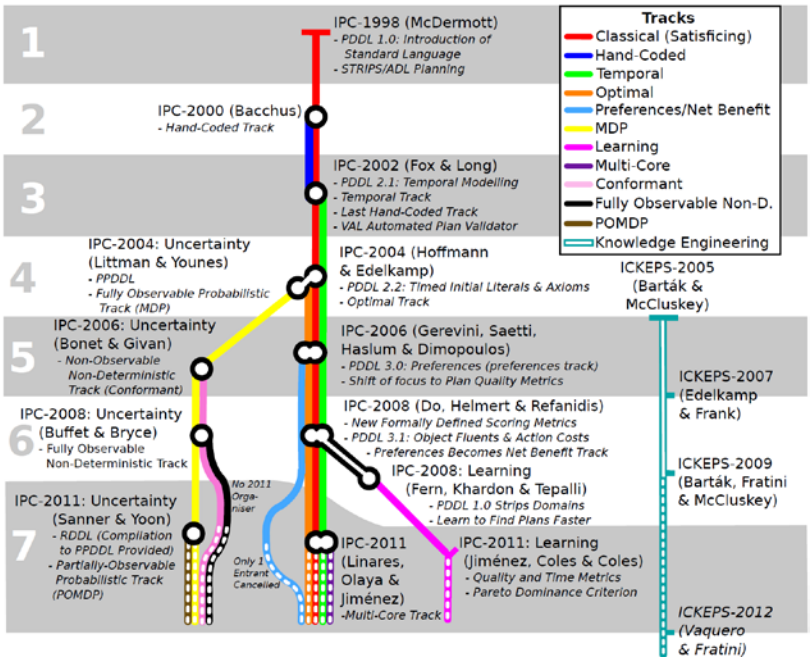

Figure 2: Historical Tracks of the IPC

As shown in Figure 2, a variety of competition tracks for different problem classes have emerged over time, but only the classical track has run for every competition. The classical track also consistently receives the most entries. The CRT domain contains problems that can belong to all these classes, but the largest subset does reside in the classical track. For this reason, the state-of-the-art will be more evident in this track.

Figure 3 shows the success rates of classical planners by year. The best performing implementation of each major planning approach was chosen to represent its respective sub-discipline. When interpreting the graph, it is important to note that each competition was different. For example, running time constraints varied between 10 minutes to 2 hours, the complexity of the problem sets changed drastically from year to year, and given the span of 13 years computational resources have also increased significantly.

This makes it difficult to quantify exactly how much better a planner with 90% coverage is compared to one with 85% coverage, given that those additional 5% could be from the hardest problems. However the competition organisers have described the problems set each competition to be pushing the field's state-of-the-art, thus consistent performance is a strong indicator for candidacy. For the more recent competitions, planners employing planning graphs appear to be the most successful in terms of problem coverage, followed by State-Space planners and HTN planners.

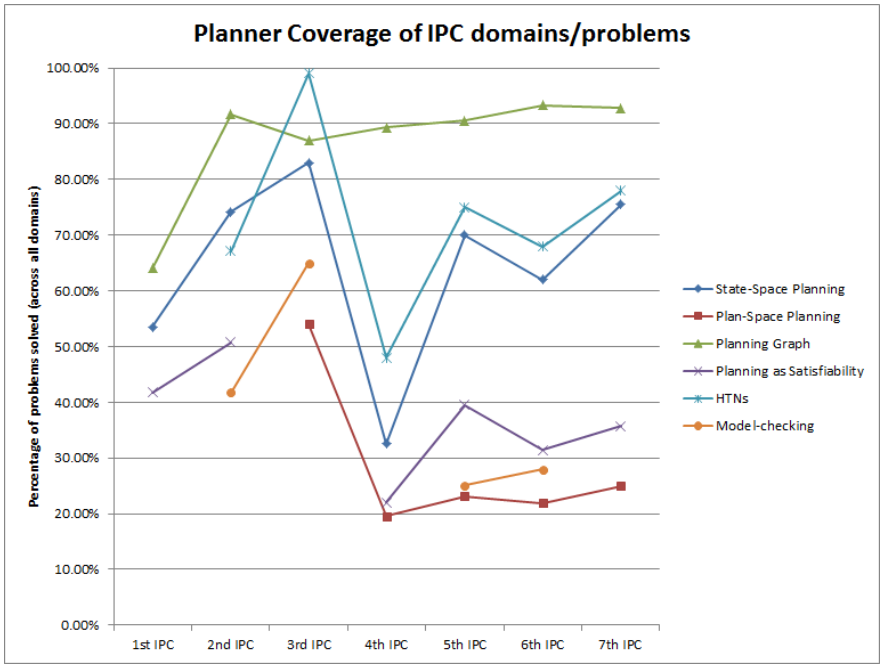

Figure 3 – Performance of various planning techniques based on problems solved

In a separate analysis of the learning track, it was shown that machine learning does improve the performance of planners [65]. However significant training was needed to achieve the improvement, and in general the training was domain-specific. This means that policies, macro actions and portfolios learned for one domain/problem class could not be reused in another domain. Machine learning can still benefit planning for CRT in the long run, provided that the training set includes sufficient examples of all types of cyber-attacks, and supplementary training is conducted when the CRT problem evolves.

The IPC benchmarks provide a strong indicator of general strength of planning techniques, and the test problems used are increasingly designed to mimic real problems faced by industry. However, there is limited participation by companies that have built proprietary planning software, therefore the IPC alone, and indeed the academic literature surveyed for this report, does not necessarily capture the complete state of the art.

The IPC also favours planners that fare well across multiple domains and problem classes, which promotes good general purpose tools rather than specialized solutions.

4. Automated Cyber Red Teaming

Now that we have discussed CRT issues as well as had an overview of automated planning, we discuss how best to model CRT as a planning problem and provide suggestions of tools and frameworks that may be suitable for automating the exercise.

4.1 Cyber Red Teaming as a Planning Problem

As mentioned in Section 3, automated planning has been used in computer game engines. Modern computer and video games often use automated planners for scripting Non-Player Character (NPC) behaviour [14], particularly when it plays an adversarial role. It has been shown that NPCs using automated planners are able to make tactical and strategic decision with better outcomes than expert human players could [66].

This ability to simulate intelligent autonomous adversaries like the NPCs in the game engine means that automated planners may lend itself well to aspects of CRT since the two problems are very similar. Other cyber security activities including, but not limited to, vulnerability risk assessment and mitigation planning could also be performed with automated planning tools as well. However for this report, the attack plan drafting in CRT is the primary focus.

CRT is also by nature a planning problem. The entities within the network environment are the objects, and the relationships between them as well as their attributes are the prepositions. The TTPs need to be encoded as plan actions with the appropriate dependencies and constraints, where the action effects reflect changes to the entities in the target network.

4.2 CRT Planning Domain characterisation

Using the list of planning problem traits from Section 3.2, here are the assumptions and abstractions suggested for conducting a CRT exercise.

4.2.1 Observability

Automated CRT is likely to be simpler if the domain is treated as a fully observable system, such that everything potentially needed for constructing attack plans is known and available. There are several reasons for approaching the problem this way:

- If the real network is only partially observable to all entities, valid attack plans may exist but what is known and available to the red team may not be sufficient to construct them. In such cases it would still be better to assume what the red team partially observes is the full system, so that the more efficient classical planners can be used. This is acceptable as the conclusion drawn regarding viable attack options would be the same from an adversary's perspective.

- From an efficiency perspective, removing irrelevant objects and prepositions from the planning process will insulate cognitive overload of the planner. If such inferences can be made with a good performance trade-off for planner performance, then do so. However, where such problem reductions are not possible, it is generally safer to overestimate the knowledge that an adversary has to construct a plan rather than underestimating it. This ensures that all possible attack vectors are considered at least once.

- Real networks use technologies such as proxies and firewalls to hide its architecture from outsiders, which some adversaries may find difficult to overcome. However, if we factor in alternate reconnaissance vectors such as social engineering, it is best to assume an adversary can still gain full network visibility.

CRT on fully observable systems not only lead to a more complete set of attack plans, but also decouples the planning work from the analysis of attack plan cost, risk and impact analysis depending on the adversary. The latter profile can then be used for a more efficient triage process. For these reasons, modelling the CRT exercise as a fully observable planning problem will suffice for the first cut.

4.2.2 Determinism

In practice, many attacks may fail or have unintended side effects. For example, a phishing email may not arrive at a target's inbox due to being blocked, or the target, upon receiving the email, chooses not to open the attachment containing the malicious payload and forwards the email to their network admin. A realistic model of CRT planning is non-deterministic, as repeat execution of the same attack plan may have different outcomes even if the system state remains the same.

However, modelling non-determinism is problematic, as the distribution of possible outcomes for each type of attack is generally not known. Learning each attack's failure rate is a separate and time-consuming process that blocks the main CRT exercise. It is also difficult due to the context sensitivity of these learned distributions, which may make a set of considered side effects in one setting inapplicable in another.

As such, we suggest that every action the adversary takes is assumed to be deterministic and is always successful. This assumption allows the use of more planning techniques, as only MDPs are currently effective at solving non-deterministic planning problems. Also, it is better to have false positives than false negatives in CRT.

4.2.3 State Dynamics

A given cyber environment is likely to contain both static aspects (system architecture, geography etc.) and dynamic aspects (staffing arrangements, network packet flows). Even if every action is always successful, state-changing events independent of the attacker's actions may affect the success of the rest of the attack in the new system state. For instance, if a user is updating a vulnerable version of the software that was intended to be an adversary's target, the malicious payload may arrive after the patch has already been applied, thus rendering it ineffective.

As it is hard to model non-determinism of attack outcomes [55], it is also hard to model the system dynamics. Because of this, in the context of a single planning run, assuming a static environment is recommended. This can be either in the form of a full snapshot of the current system, or removing actions that depend on dynamic elements prior to planning. This will simplify the planning, but may include attack plans that only work in limited cases, and also omit ones that could have succeeded.

To allow for and accommodate state dynamics in the planning process, there are several approaches one may consider. One is to favour plans that depend less on prepositions and objects that are tagged as dynamic, which can be machine learned through training or manually determined by a human domain expert. Another approach is to create robust attack plans with contingencies [67], which can be identified by frequent re-planning on the latest state of the system and seeing which attack vectors remains valid regardless of system dynamics. In both cases, it makes the planning problem larger, and the increased robustness of attack plans discovered may come at the cost of significantly added computation time.

4.2.4 Time and Resource Constraints

In the real network environment, both the adversary and their target have resource limitations; a small attack team with a dozen laptops will have difficulty orchestrating a successful Distributed Denial of Service attack on an organization like Google, but be sufficient to take down a small business's website with budget hosting. Some attacks may also require taking advantage of certain time windows such as between the announcement of a new software patch, and the system administrator installing the patch on the target machines. Moreover, such attacks would only be possible if this window of vulnerability [68] and the exploit method are known and accomplishable given the attacker's resources.

For encoding the CRT exercise as a planning problem, it is recommended that these resource constraints are not a factor in the validity of the attacker's plan. The quality and validity of attack plans under constraint can be quickly checked, archived and ranked once it is found.

4.2.5 Optimality and Ordering

Complex attacks that involve multiple entities in a Cyber environment can usually be conducted in various ways. For instance, you could attempt to steal a user's email account to get at their other email-verified accounts, or you could forge and send a password reset email to trick the owner into providing these details on a phishing site. There may also be multiple attack plans that can be used to achieve the same goal, and in reality the resources an adversary has is limited, so they would most likely try the most cost-effective attacks first.

For initial development, we are more concerned with attack possibility and impact than attack efficiency, so we do not need a planner to generate or guarantee an optimal plan. As long as it generates valid attack plans in reasonable time, and the plans are totally ordered, contingencies can be appended onto these plans if non-determinism is later introduced or modelled for the same problems.

4.2.6 Preference and Constraints

We assume that the adversary has resource limitations, and therefore cannot conduct attacks beyond their means. As such, action costs also need to be modelled and factored into the planning process. If records of past attacks by the adversary exist, their preferences in how they attack are also worth providing to the planner if available.

4.3 Candidate Tools

We have characterised and abstracted the CRT domain as a classical planning problem. Based on the information from the planner capability table, as well as the IPC results from Section 3, planning graph planners with parameter tuning or portfolio-based learning appear to be the most suitable tool for CRT exercise planning and execution.

Below is a list of planning tools/frameworks that use planning graphs, and either contain learning capabilities or can be easily extended to incorporate a learning step. They were selected from a larger pool of planning graph planners based on their individual performance in the more recent IPCs.

4.3.1 LAMA

Developed by Silvie Richter from NICTA, the LAMA system is part of the latest generation of heuristics-based forward searching techniques with planning graphs [69] [70]. LAMA identifies *landmarks*, states that valid plans must go through, to decompose the problem, and arrive at valid plans much faster than purely heuristic-driven end-to-end approaches. It was the best performing classical planner in both the 6th and 7th IPC, and is considered the state-of-the-art general purpose planner.

4.3.2 FD-Autotune

Developed by a team at University of Huddersfield [16], FD-Autotune is a machine learning variant and extension to Helmert's Fast Downward algorithm [71], which uses parameter tuning and macro action learning to supplement a hybrid planning engine. This engine uses planning graphs to reduce the search space, and features automated HTN construction in the pre-processing stage which can make plan drafting faster than traditional approaches.

The learning engine deployed is the automated algorithm configuration tool ParamILS [72] and the HAL experimentation environment [73], which profiles a particular problem using training examples, and optimizes parameters for various heuristic configurations used to guide the search. FD-Autotune also manages a library of heuristics that through learning, is able to automatically select the most suitable heuristic(s) for a given domain and problem type. FD-Autotune can optimize the planning for either planning speed (generate satisficing plans really fast) or plan quality (based on red team preferences).

The learning aspect of FD-Autotune is likely to be robust as it uses established tools for the parameter tuning. Combined with the FD algorithm, it has achieved solid benchmark

performance at the recent IPCs when planning for speed, though the speed increase when optimizing for plan quality makes it comparably slower to other speed-driven planners.

4.3.3 PbP2

Developed by a team at the University of Brescia [50], the Portfolio-based Planner (PbP) is an ensemble learning planner. It constructs a portfolio for each domain to determine which planners in its library are most suitable to complete the planning. Additionally, PbP constructs distinct macro actions for every planner in the portfolio, allows for automated parameter tuning, and generates a promising configuration.

The latest release of PbP (version 2) won the overall learning track of IPC-2011, and in terms of overall plan success, outperformed all planners in the classical track, including LAMA. It is however more resource-intensive due to the training component, but the newest stable release feature a distributed architecture, permitting concurrent scheduling of multiple planners for training or planning work.

4.3.4 Conclusion

With respect to CRT, LAMA appears to be the most suitable for quick deployment since it has the highest success rate of the classical planners, and doesn't require training for operational use. FD-Autotune and PbP2 may be more valuable in the long run if conducting CRT is part of the organization's business process, as continuous (offline) learning will improve the planner's understanding of what attacks are most relevant to the organization deploying it.

5. Recommendations and Future Work

This report discussed Cyber Red Teaming, what is involved with the modelling of each problem, attack plan generation and associated challenges. It has also introduced the fundamental and state-of-the-art theories and techniques in automated planning, analysed performance of various techniques from the IPC, and recommended the most suitable tools based on results as well as compatibility to the CRT problem.

Our recommendation is that planning graph planners are the most suitable approach for CRT, especially for real-time deployment on sensitive systems due to its scalable performance and measured success rate. Furthermore, if CRT needs to be conducted frequently, planner implementations that incorporate machine learning will be even better. The tools we recommend for trialling are LAMA, FD-Autotune and PbP2, as they have been shown to perform well above their peers.

From here, two steps are possible. The first would be to set up and conduct CRT on a large organisation's computing environment using one of these tools, in order to verify and more accurately measure the benefit they bring compared to traditional hands-on approaches to CRT. The other step would be to identify specialized planning tools for organizational CRT, and study their capabilities and performance.

Automated Planning is a well-established field of research, but its application on Cyber Red Teaming is relatively untouched, and deserves further exploration through collaboration between experts from both fields.

Finally, the CRT problem is evolving: organizations may switch to using cloud infrastructure, business policies may change allowing staff to bring in unvetted (thus unmodelled) personal electronics into the workplace, or major incidents may change the level of fidelity of mitigation plans required.

Therefore, on top of trialling planners for CRT, continuous monitoring of new research and other benchmarking results collected by academia and industry is highly recommended. This will help ensure the state-of-the-art is used in Automated CRT to get best performance and best results possible.

6. Works Cited

[1] H. Abbass and e. al., "Computational Red-Teaming: Past, Present and Future," *Computational Intelligence,* pp. 30-42, 1 6 2011.

[2] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995.

[3] B. Arkin, S. Stender and G. McGraw, "Software Penetration Testing," *Security and Privacy,* pp. 84-87, 1 3 2005.

[4] A. J. Holmgren, "A framework for vulnerability assessment of electric power systems," pp. 31-55, 2007.

[5] G. Taylor, R. Frederiksen and R. Vane, "Agent-based Simulation of Geo-Political Conflict," *Artificial Intelligence,* 2004.

[6] D. o. Defence, "Department of Defence IT Enterprise Strategy and Roadmap," 6 9 2011. [Online]. Available: http://dodcio.defense.gov/Portals/0/Documents/Announcement/Signed_ITESR_6SEP11.pdf. [Accessed 2013].

[7] P. Tague and R. Poovendran, "Modeling adaptive node capture attacks in multi-hop wireless networks," *Ad Hoc Networks,* pp. 801-814, 6 5 2007.

[8] I. Inc., "How Businesses can measure network complexity - and why they should," 2013. [Online]. Available: http://www.infoblox.com/sites/infobloxcom/files/resources/infoblox-whitepaper-network-complexity.pdf. [Accessed 2013].

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

[10] K. Tierney, R. M. Jensen, C. Kroer, A. Britt, A. Coles and A. Coles, "Automated Planning for Liner Shiping Fleet Repositioning," in *ICAPS 2013*, 2013.

[11] H. Munoz-Avila, D. W. Aha, L. Breslow and D. Nau, "HICAP: An Interactive case-based planning architecture and its applicationto noncombatant evacuation operations," AAAI, 1999.

[12] D. McDermott, "Estimated-Regression Planning for Interactions with Web Services," in *Proceedings of AIPS-02*, 2002.

[13] M. Ghallab, "An overview of planning technology in robotics," *Advances in Artificial Intelligence,* pp. 29-49, 2004.

[14] J. P. Kelly, A. Botea and S. Koenig, "Offline Planning with Hierarchical Task Networks in Video Games," in *AIIDE*, 2008.

[15] D. Long and M. Fox, "The 3rd International Planning Competition: Results and Analysis," *Journal of Artificial Intelligence Research,* vol. 20, no. 3, pp. 1-59, 2003.

[16] C. Fawcett, E. Karpas, M. Helmert, G. Roger, H. Hoos and J. Seipp, "FD-Autotune: Domain-Specific Configuration using Fast Downward," in *Proceedings of ICAPS 2011*, 2011.

[17] N. J. Nilsson and F. R. E., "STRIPS - A New Approach to the Application of Theorem Proving to Problem Solving," Elsevier, 1972.

[18] D. McDermott, "PDDL - The Planning Domain Definition Language," in *AIPS-98*, 1998.

[19] M. Likhachev, J. G. Geoffrey and S. Thrun, "ARA*: Anytime A* with Provable Bounds on Sub-optimality," *NIPS,* 2003.

[20] M. Helmert, "IPC-2008 - Deterministic Part," 20 10 2010. [Online]. Available: http://ipc.informatik.uni-freiburg.de/Planners. [Accessed 23 07 2013].

[21] H. Palacios and H. Geffner, "From Conformant into Classical Planning: Efficient Translations That May be complete too," in *ICAPS*, 2007.

[22] D.-Y. Yueng and G. A. Bekey, "A decentralized approach to the motion planning problem for multiple mobile robots," in *IEEE International Conference on Robotics and Automation*, 1987.

[23] K. P. Sycara, Innovative Approaches to Planning, Scheduling and Control, San Diego: Defence Advanced Research Projects Agency, 1990.

[24] A. Yang, H. Abbass and R. Sarker, "Characterizing warfare in red teaming," *IEEE Transactions on Systems, Man and Cybernetics,* pp. 268-285, 2006.

[25] L. P. Swiler, C. Phillips, D. Ellis and S. Chakerian, "Computer-attack graph generation tool," in *DARPA Information Survivability Conference*, Anaheim, 2001.

[26] K. R.E., "Iterative-Deepening A: An Optimal Admissible Tree Search," in *Proceedings of the 9th IJCAI*, 1985.

[27] J. Hoffman and R. I. Brafman, "Conformant planning via heuristic forward search: A new approach," *Artificial Intelligence,* vol. 170, no. 6-7, pp. 507-541, 2006.

[28] J. Hoffmann and B. Nebel, "The FF Planning System: Fast Plan Generation Through Heuristic Search," *Journal of Artificial Intelligence Research,* vol. 14, pp. 253-302, 2001.

[29] M. Boddy, J. Gohde, T. Haigh and S. Harp, "Course of Action Generation for Cyber Security Using Classical Planning," in *ICAPS*, 2005.

[30] A. Botea, E. Nikolova and M. Berlingerio, "Multi-Modal Journey Planning in the Presence of Uncertainty," in *ICAPS*, 2013.

[31] M. Wehrle and M. Helmert, "About Partial Order Reduction in Planning and Computer Aided Verification," in *ICAPS 2012*, 2012.

[32] A. Gefen and R. Brafman, "Pruning Methods for Optimal Delete-Free Planning," in *ICAPS 2012*, 2012.

[33] M. Crosby, M. Rovatsos and R. P. Petrick, "Automated Agent Decomposition for Classical Planning," in *ICAPS 2013*, 2013.

[34] E. Keyder, J. Hoffman and P. Haslum, "Semi-Relaxed Plan Heuristics," in *ICAPS 2012*, 2012.

[35] M. Katz, J. Hoffman and C. Domshlak, "Who Said We Need to Relax all Variables?," in *ICAPS 2013*, 2013.

[36] A. L. Blum and M. L. Furst, "Fast Planning through Planning Graph Analysis," *Artificial Intelligence,* vol. 90, pp. 281-300, 1997.

[37] B. Bonet and H. Geffner, "Planning as heuristic search," *Artificial Intelligence,* vol. 129, pp. 5-33, 2001.

[38] J. Hoffman, "The Metric-FF Planning System - Translating Ignoring Delete Lists to Numeric State Variables," *Journal of Artificial Intelligence,* vol. 20, pp. 291-341, 2003.

[39] S. Edelkamp and P. Kissmann, "GAMER: Bridging Planning and General Game Playing with Symbolic Search," in *Proceedings of IPC-6 Competition*, 2008.

[40] R. S. Nigenda, X. Nguyen and S. Kambhampati, "AltAlt: Combining the Advantages of Graphplan and Heuristic State Search," Arizona State University, Tempe, 2000.

[41] D. E. Smith and D. S. Weld, "Conformant Graphplan," in *AAAI-98 Proceedings*, 1998.

[42] A. Blum, "Probabilistic GraphPlan (PGP/TGP) Home Page," 12 2011. [Online]. Available: http://www.cs.cmu.edu/~avrim/pgp.html. [Accessed 22 7 2013].

[43] D. Nau, Y. Cao, A. Lotem and H. Munoz-Avila, "SHOP: Simple Hierarchical Ordered Planner," in *Proceedings of the 16th international joint conference on Artificial intelligence*, Stockholm, 1999.

[44] D. E. Wilkins and M. desJardins, "A Call for Knowledge-Based Planning," *AI Magazine,* 2001.

[45] L. P. Kaelbling and T. Lozano-Pérez, "Hierarchical Task and motion planning in the now," in *IEEE International Conference on Robotics and Automation*, 2011.

[46] H. Hoang, S. Lee-Urban and H. Muñoz-Avila, "Hierarchical Plan Representations for Encoding Strategic Game AI," in *AIIDE*, 2005.

[47] D. Grove, A. Murray, D. Gerhardy, B. Turnbull, T. Tobin and C. Moir, "An Overview of the Parallax Battlemind v1.5 for Computer Network Defence," DSTO, 2013.

[48] T. Zimmerman and S. Kambhampati, "Learning-Assisted Automated Planning: Looking Back, Taking Stock, Going Forward," *AI Magazine,* 2003.

[49] O. Buffet and D. Aberdeen, "The factored policy-gradient planner," *Artificial Intelligence,* vol. 173, pp. 722-747, 2009.

[50] A. E. Gerevini, A. Saetti and M. Vallati, "An Automatically Configurable Portfolio-based Planner with Macro-actions - PbP," in *ICAPS*, 2009.

[51] J. S. Penberthy and D. S. Weld, "UCPOP: A Sound, Complete, Partial Order Planner for ADL," in *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, 1992.

[52] H. Kautz, "blackbox - a SAT technology planning system," 15 1 2003. [Online]. Available: http://www.cs.rochester.edu/~kautz/satplan/blackbox/blackbox-download.html. [Accessed 22 7 2013].

[53] E. Karabaev and O. Skvortsova, "FCPlanner: A planning strategy for First-Order MDPs," Dresden University, Dresden, 2004.

[54] C. Sarraute, O. Buffet and J. Hoffmann, "POMDPs Make Better Hackers: Accounting for Uncertainty in Penetration Testing," in *PRoceedings of the 26th AAAI conference on Artificial Intelligence*, 2012.

[55] A. Cimatti, M. Roveri and P. Traverso, "Weak, strong, and strong cyclic planning via symbolic model checking," *AI,* vol. 147, pp. 35-84, 2003.

[56] D. McDermott, "The 1998 AI Planning Systems Competition," *AI Magazine,* vol. 21, pp. 35-56, 2000.

[57] F. Bacchus, "AIPS'00 Planning Competition," *AI Magazine,* vol. 22, no. 3, pp. 47-56, 2001.

[58] S. Edelkamp and J. Hoffman, "IPC-4 Results," University of Freiburg, 2004. [Online]. Available: http://www.tzi.de/~edelkamp/ipc-4/results.html. [Accessed 24 06 2013].

[59] B. Bonet and B. Givan, "Results of Conformant Track in the 5th International Planning Competition," 31 05 2006. [Online]. Available: http://ldc.usb.ve/~bonet/ipc5/docs/results-conformant.pdf. [Accessed 12 07 2013].

[60] B. Bonet and B. Givan, "Results of Probabilistic Track in the 5th International Planning Competition," 2 06 2006. [Online]. Available: http://ldc.usb.ve/~bonet/ipc5/docs/results-probabilistic.pdf. [Accessed 12 07 2013].

[61] A. E. Gerevini, P. Haslum, D. Long, A. Saetti and Y. Dimopoulos, "Deterministic Planning in the fifth international planning competition: PDDL3 and experimental evaluation of the planners," *Artificial Intelligence,* vol. 173, pp. 619-668, 2009.

[62] H. L. S. Younes, M. L. Littman, D. Weissman and J. Asmuth, "The First Probabilistic Track of the International Planning Competition," *Journal of Artificial Intelligence Research,* vol. 24, pp. 851-887, 2005.

[63] D. Bryce and O. Buffet, "International Planning Competition - Uncertainty Part - Benchmarks and Results," 27 06 2011. [Online]. Available: http://ippc-2008.loria.fr/wiki/index.php/Results.html. [Accessed 24 07 2013].

[64] A. Coles, A. Coles, A. G. Olaya, S. Jimenez, C. L. Lopez, S. Sanner and S. Yoon, "A survey of the seventh International Planning Competition," *AI Magazine,* vol. 33, no. 1, pp. 83-88, 2012.

[65] A. Fern, Khardon, Roni and P. Tadepalli, "The first learning track of the international planning competition," *Machine Learning,* vol. 84, pp. 81-107, 2011.

[66] P. H. M. Spronck, Adaptive Game AI, Universitaire Pers Maastricht, 2005.

[67] J. Hoffman, "Homepage of Conformant-FF and Contingent-FF and Probabilistic-FF," [Online]. Available: http://fai.cs.uni-saarland.de/hoffmann/cff.html. [Accessed 23 07 2013].

[68] W. A. Arbaugh, W. L. Fithen and J. McHugh, "Windows of Vulnerability: A Case Study Analysis," *Computer,* pp. 52-59, 2000.

[69] S. Richter and M. Westphal, "The LAMA Planner: Guiding Cost-based Anytimg Planning with Landmarks," *Journal of Artificial Intelligence Research,* vol. 39, pp. 127-177, 2010.

[70] S. Richter, M. Westphal and M. Helmart, "LAMA 2008 and 2011," in *Proceedings of the seventh international planning competition*, 2011.

[71] M. Helmert, "The Fast Downward Planning System," *Journal of Artificial Intelligence Research,* vol. 26, pp. 191-246, 2006.

[72] F. Hutter, H. Hoos, K. Leyton-Brown and T. Stutzle, "ParamILS: an automatic algorithm configuration framework," *Journal of Artificial Intelligence Research,* vol. 36, pp. 267-306, 2009.

[73] C. Nell, C. Fawcett, H. H. Hoos and K. Leyton-Brown, "HAL: a framework for the automated analysis and design of high-performance algorithms," *LION-5.*

[74] H. Kurniawati, D. Hsu and W. S. Lee, "SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces," in *Robotics: Science and Systems*, 2008.

[75] H. Kurniawati and D. Hsu, "APPL - Approximate POMDP Planning Toolkit," 18 5 2012. [Online]. Available: http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl/index.php. [Accessed 24 07 2013].

[76] R. Valenzano, H. Nakhost, M. Muller, J. Schaeffer and N. Sturtevant, "ArvandHerd: Parallel Planning with a Portfolio," in *Proceedings of ECAI*, 2012.

[77] S. Holldobler and H. P. Stoerr, "Solving the entailment problem in the Fluent Calculus using Binary Decision Diagrams," in *Computational Logic*, vol. 1861, Springer, 2000, pp. 747-761.

[78] H. Kautz and B. Selman, "BLACKBOX: A New Approach to the Application of Theorem Proving to Problem Solving," in *AIPS98 Workshop on Planning as Combinatorial Search*, 1998.

[79] H. Kautz and B. Selman, "Unifying SAT-based and Graph-based Planning," in *Proceedings of the 16th International Join Conference on Artificial Intellignece*, Stockholm, 1999.

[80] N. Lipovetzky and H. Geffner, "Inference and Decomposition in Planning using Causal Consistent Chains," in *ICAPS 2009*, 2009.

[81] J. Hoffman, "Fast-Forward," 2011. [Online]. Available: http://fai.cs.uni-saarland.de/hoffmann/ff.html. [Accessed 25 07 2013].

[82] V. Vidal and H. Geffner, "Branching and Pruning: An optimal temporal POCL planner based on constraint programming," *Artificial Intelligence,* vol. 170, 2006.

[83] J. Dreo, M. Schoenauer and P. Saveant, "Divide-and-evolve: the marriage of descartes and darwin," in *Proceedings of the 7th international planning competition*, 2011.

[84] M. Schoenauer, P. Saveant and V. Vidal, "Divide-and-Evolve: a New Memetic Scheme for Domain-Independent Temporal Planning," in *Evolutionary Computation in Combinatorial Optimization*, Springer Berlin Heidelberg, 2006, pp. 247-260.

[85] R. Huang, Y. Chen and W. Zhang, "DTG-Plan:Fast Planning by Search in Domain Transition Graphs," *AAAI,* 2008.

[86] M. Helmert, "Fast Downward: IPC Planners," 16 11 2011. [Online]. Available: http://www.fast-downward.org/IpcPlanners. [Accessed 15 08 2013].

[87] S. Grandcolas and C. Pain-barre, "Filtering, Decomposition and Search Space Reduction for Optimal Sequential Planning," in *Proc. National Conference on Artificial Intelligence*, 2007.

[88] M. Helmert, G. Roger and E. Karpas, "Fast Downward Stone Soup: A Baseline for Building Planner Portfolios," in *ICAPS 2011 Workshop on Planning and Learning*, 2011.

[89] E. Keyder and H. Geffner, "Heuristics for Planning with Action Costs Revisited," in *ECAI 2008*, 2008.

[90] S. Yoon, A. Fern and G. Robert, "FF-Replan: A Baseline for Probabilisitic Planning," Association for the Advancement of Artificial Intelligence, 2007.

[91] P. Kissman, "Download of the (GAMER) planner," 06 09 2012. [Online]. Available: http://fai.cs.uni-saarland.de/kissmann/planning/downloads/. [Accessed 22 08 2013].

[92] A. Kolobov, P. Dai, Mausam and D. S. Weld, "Reverse Iterative Deepening for Finite-Horizon MDPs with Large Branching Factors," in *ICAPS 2012*, 2012.

[93] A. Blum, "GraphPlan home page," 6 2001. [Online]. Available: http://www.cs.cmu.edu/~avrim/graphplan.html. [Accessed 22 7 2013].

[94] I. Refanidis and I. Vlahavas, "The GRT Planning System: Backward Heuristic Construction in Forward State-Space Planning," *Journal of Artificial Intelligence Research,* vol. 15, pp. 115-161, 2001.

[95] M. Ghallab and H. Laruelle, "Representation and Control in IxTeT, A Temporal Planner," in *Conference on Artificial Intelligence Planning and Scheduling Systems*, 1994.

[96] S. Richter, "Silvia Richter Software," 08 2011. [Online]. Available: http://www.informatik.uni-freiburg.de/~srichter/. [Accessed 23 07 2013].

[97] A. Gerevini and I. Serina, "LPG: A Planner Based on Local Search for Planning Graphs with Action Costs," in *Sixth International Conference on Artificial Intelligence Planning and Scheduling Systems*, 2002.

[98] A. Gerevini, A. Saetti, I. Serina and P. Toninelli, "Planning in PDDL 2.2 Domains with LPG-TD," in *International Planning Competition*, 2004.

[99] R. Nissim, J. Hoffman and M. Helmert, "Computing Perfect Heuristics in Polynomial Time: On Bisimulation and Merge-and-Shrink Abstraction in Optimal Planning," in *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*, 2011.

[100] A. Botea, M. Enzenberger, M. Muller and J. Schaeffer, "Macro-FF: improving AI planning with automatically learned macro-operators," *Journal of Artificial Intelligence Research,* vol. 24, pp. 581-621, 2005.

[101] A. Coles and A. Smith, "Marvin: A Heuristic Search Planner with Online Macro-Action Language," *Journal of Artificial Intelligence Research,* vol. 28, pp. 119-156, 2007.

[102] S. M. Majercik and M. L. Littman, "MaxPlan: A new approach to probabilistic planning," in *AIPS-98*, 1998.

[103] B. Bonet and H. Geffner, "mGPT: A Probabilistic Planner Based on Heuristic Search," *Journal of Artificial Intelligence Research,* vol. 24, pp. 933-944, 2005.

[104] S. Edelkamp and M. Helmert, "MIPS: The Model-checking Integrated Planning System," *AI Magazine,* vol. 22, no. 3, 2001.

[105] S. Thiebaux, C. Gretton, J. Slaney and D. Price, "Decision-Theoretic Planning with non-Markovian Rewards," *Journal of Artificial Intelligence Research,* vol. 25, pp. 17-74, 2006.

[106] M. v. d. Briel and S. Kambhampati, "Optiplan: Unifying IP-based and Graph-based Planning," in *International Planning Competition*, 2004.

[107] A. Gerevini, A. Saetti and M. Vallati, "PbP2 - Automatic Configuration of a Portfolio-based Multi-Planner," in *7th International Planning Competition*, 2011.

[108] A. E. Gerevini, A. Saetti and M. Vallati, "PbP," 21 06 2012. [Online]. Available: http://chronus.ing.unibs.it/pbp/. [Accessed 23 07 2013].

[109] J. L. Ambite and C. A. Knoblock, "Planning by Rewriting," *Artificial Intelligence,* vol. 118, no. 1-2, pp. 115-161, 2000.

[110] A. Blum and J. C. Langford, "Probabilistic Planning in the GraphPlan framework," *Recent Advances in AI Planning,* pp. 319-332, 2000.

[111] Y. Chen and Q. Lv, "Plan-A: A Cost Optimal Planner Based on SAT-Constrained Optimization," in *Proceedings of IPC-6*, 2008.

[112] D. Silver and J. Veness, "Monte-Carlo Planning in Large POMDPs," *Advances in Neural Information Processing Systems,* 2010.

[113] M. Fourman, "Propositional Planning," University of Edinburgh, Edinburgh, 2000.

[114] M. B. Do and S. Kambhampati, "Sapa: A Multi-objective Metric Temporal Planner," *Journal of Artificial Intelligence Research,* vol. 20, pp. 155-194, 2003.

[115] H. Kautz and B. Selman, "Planning as Satisfiability," in *Proceedings of ECAI*, 1992.

[116] H. Kauts, "SATPLAN04: Planning as Satisfiability," in *Working Notes in the Fourth International Planning Competition*, 2004.

[117] D. S. Weld, C. Anderson and D. E. Smith, "Extending Graphplan to handle uncertainty and sensing actions," American Association for Artificial Intelligence, 1998.

[118] Y. Chen, C.-W. Hsu and B. W. Wah, "SGPlan: Subgoal Partitioning and Resolution in Planning," Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois, 2004.

[119] Y. Chen, B. W. Wah and C.-W. Hsu, "Temporal Planning using Subgoal Partition and Resolution in SGPlan," *Journal of Artificial Intelligence Research,* vol. 26, pp. 323-369, 2006.

[120] C.-W. Hsu and B. W. Wah, "SGPlan," [Online]. Available: http://wah.cse.cuhk.edu.hk/wah/programs/SGPlan/index.html. [Accessed 23 07 2013].

[121] C.-W. Hsu, B. W. Wah, R. Huang and Y. Chen, "New Features in SGPlan for Handling Preferences and Constraints in PDDL 3.0," in *Proceedings of the Fifth International Planning Competition*, 2006.

[122] C.-W. Hsu, B. W. Wah, R. Huang and Y. Chen, "Handling Soft Constraints and Goals Preferences in SGPlan," in *Workshop on Preferences and Soft Constraints in Planning*, 2006.

[123] C.-W. Hsu and B. W. Wah, "The SGPlan Planning System in IPC-6," in *Proceedings of IPC 2008*, 2008.

[124] D. Nau, T.-C. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu and F. Yaman, "SHOP2: An HTN Planning System," *Journal of Artificial Intelligence Research,* pp. 379-404, 2003.

[125] D. Long and M. Fox, "Efficient Implementation of the Plan Graph in STAN," *Journal of Artificial Intelligence Research,* vol. 10, pp. 87-115, 1999.

[126] Z. Feng and E. A. Hansen, "Symbolic Heuristic Search for Factored Markov Decision Processes," American Association for Artificial Intelligence, 2002.

[127] M. T. Spaan and N. Vlassis, "Perseus: Randomised Point-based Value Iteration for POMDPs," *Journal of Artificial Intelligence Research,* vol. 24, pp. 195-220, 2005.

[128] P. Poupart, "Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes," University of Toronto, 2005.

[129] P. Poupart, "Pascal Poupart's software," 2011. [Online]. Available: https://cs.uwaterloo.ca/~ppoupart/software.html. [Accessed 25 07 2013].

[130] F. Lin, "A Planner Called R," *AI Magazine,* vol. 22, no. 3, pp. 73-76, 2001.

[131] H. Palacios, "Hector Palacios Software," 2009. [Online]. Available: http://ldc.usb.ve/~hlp/#software. [Accessed 22 7 2013].

[132] P. Doherty and J. Kvarnstrom, "TALplanner: A Temporal Logic-Based Planner," *AI Magazine,* vol. 22, no. 3, pp. 95-102, 2001.

[133] F. Maris and P. Regnier, "TLP-GP: a Planner to Solve Temporally-Expressive Problems," in *Proceedings of IPC-6*, 2008.

[134] F. Bacchus and F. Kabanza, "Using temporal logics to express search control knowledge for planning," *Artificial Intelligence ,* pp. 123-191, 2000.

[135] P. Haslum and H. Geffner, "Heuristic Planning with Time and Resources," *Proceedings of ECP,* 2001.

[136] H. L. S. Younes and R. G. Simmons, "VHPOP: Versatile Heuristic Partial Order Planner," *Journal of Artificial Intelligence Research,* vol. 20, pp. 405-430, 2003.

[137] H. Newton, J. Levine, M. Fox and D. Long, "Learning Macro-Actions for Arbitrary Planners and Domains," in *ICAPS*, 2007.

[138] H. Newton, J. Levine, M. Fox and D. Long, "Wizard's Home," 13 1 2009. [Online]. Available: http://www.buet.ac.bd/cse/users/faculty/newton/wizard.html. [Accessed 23 07 2013].

[139] V. Vidal, "YAHSP2: Keep It Simple, Stupid," in *Seventh International Planning Competition*, 2011.

[140] C. Baral and M. Gelfond, "Reasoning Agents in Dynamic Domains," *Linköping Electronic Articles in Computer and Information Science,* vol. 4, no. 40, pp. 1-19, 1999.

[141] B. Bonet and R. Givan, "5th International Planning Competition - Non-deterministic Track Call for Participation," in *Not in the Proceedings of the Fifth International Planning Competition*, 2005.

[142] B. Bonnet and H. Geffner, "Planning as Heuristic Search: New Results," in *Recent Advances in AI Planning: Proceedings of the 5th European Conference on Planning*, Berlin, 1999.

[143] A. Bouguerra and L. Karlsson, "PC-SHOP: a Probabilistic-Conditional Hierarchical Task Planner," in *Intelligenza Artificiale*, 2005.

[144] A. Boukhtouta, A. Bedrouni, J. Berger, F. Bouak and A. Guitouni, "A survey of military planning systems," Defence Research and Development Canada-Valcartier, 2004.

[145] T. Bylander, "The Computational Complexity of Propositional STRIPS Planning," University of Texas, San Antonio, 1994.

[146] G. Camilleri and J. Zalaket, "FAP: Forward Anticipating Planner," in *International Planning Competition*, 2004.

[147] T. M. Chen and J.-M. Robert, "Worm Epidemics in High-Speed Networks," *Computer,* vol. 37, no. 6, pp. 48-53, 2004.

[148] A. Coles, A. Coles, M. Fox and D. Long, "POPF2: A Forward Chaining Partial-Order Planner," in *The 2011 International Planning Competition*, 2011.

[149] C. Domshlak, E. Karpas and S. Markovitch, "To Max or not to Max: Online Learning for Speeding Up Optimal Planning," in *Proceedings of ICAPS 2011*, 2011.

[150] S. Edelkamp and S. Jabbar, "MIPS-XXL: Featuring External Shortest Path Search for Sequential Optimal Plans and External Branch-And-bound for Optimal Net Benefit," in *Proceedings of IPC-6*, 2008.

[151] E. Fink and J. Blythe, "A Complete Bidirectional Planner," AAAI, 1998.

[152] A. Garrido, E. Onaindia and F. Barber, "Time-optimal planning in temporal problems," in *Pre-proceedings of the Sixth European Conference on Planning* , Spain, 2001.

[153] A. Gerevini and D. Long, "Plan Constraints and Preferences in PDDL 3," Department of Electronics for Automation, University of Brescia, 2005.

[154] S. Grandcolas and C. Pain-Barre, "Filtering, Decomposition and Search Space Reduction for Optimal Sequential Planning," in *Proceedings of the National Conference on Artificial Intelligence*, 2007.

[155] C. Gretton, D. Price and S. Thiebaux, "NMRDPP: A System for Decision-Theoretic Planning with Non-Markovian Rewards," Australian National University, Canberra, 2003.

[156] M. Grzes and J. Hoey, "ICAPS 2014 International Probabilistic Planning Competition (IPPC) Discrete Track," 2013. [Online]. Available: https://cs.uwaterloo.ca/~mgrzes/IPPC_2014/. [Accessed 24 07 2013].

[157] K. Halsey, D. Long and M. Fox, "CRIKEY - A Temporal Planner Looking at the Integration of Scheduling and Planning," in *ICAPS-04 Workshop on Integrating Planning into Scheduling*, Whistler, ICAPS, 2004, pp. 46-52.

[158] S. Harp, J. Gohde, T. Haigh and M. Boddy, "Automated Vulnerability Analysis Using AI Planning," in *AAAI Spring Symposium*, 2005.

[159] P. Haslum, "TP4 '04 and HSPa," in *International Planning Competition*, 2004.

[160] P. Haslum, "Additive and Reversed Relaxed Reachability Heuristics Revisited," in *Proceedings of IPC-6*, 2008.

[161] H. Hautz, "SATPLAN04: Planning as Satisfiability," in *International Planning Competition*, 2004.

[162] M. Helmart and H. Lasinger, "The Scanalyzer Domain: Greenhouse Logistics as a Planning Problem," in *Proc. ICAPS*, 2010.

[163] M. Helmert, "IPC-2008 Deterministic Part results," 20 10 2010. [Online]. Available: http://ipc.informatik.uni-freiburg.de/Results. [Accessed 23 07 2013].

[164] H. Hoang, S. Lee-Urban and H. Munoz-Avila, "Hierarchical Plan Representations for Encoding Strategic Game AI," in *AAAI*, 2005.

[165] J. Hoffmann, N. Fates and H. Palacios, "Brothers in Arms? On AI Planning and Cellular Automata," in *Proc. ECAI 2010*, 2010.

[166] C.-W. Hsu, B. W. Wah, R. Huang and Y. Chen, "Constraint Partitioning for Solving Planning Problems with Trajectory Constraints and Goal Preferences," in *International Joint Conference on Artificial Intelligence*, 2007.

[167] C. International, "Common Vulnerabilties and Exposures," 25 07 2013. [Online]. Available: http://cve.mitre.org/. [Accessed 29 07 2013].

[168] L. Jin and K. Decker, "Ontology-Oriented Exploration of a HTN Planning Domain through Hypotheses and Diagnostic Execution," in *Proceedings of the Workshop on Knowledge Engineering for Planning and Scheduling*, 2010.

[169] L. P. Kaelbling, M. L. Littman and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence,* vol. 101, pp. 99-134, 1998.

[170] E. Karabaev and O. Skvortsova, "A Heuristic Search Algorithm for Solving First-Order MDPs," in *Conference on Uncertainty in Artificial Intelligence*, 2005.

[171] B. R. Kavuluri and S. U, "Tilsapa - Timed Initial Literals using SAPA," in *International Planning Competition*, 2004.

[172] T. Keller and P. Eyerich, "PROST: Probabilistic Planning based on UCT," in *ICAPS 2012*, 2012.

[173] J.-P. Kelly, A. Botea and S. Koenig, "Planning with Hierarchical Task Networks in Video Games," Australian National University, Canberra, 2007.

[174] L. Kocsis and C. Szepesvari, "Bandit based Monte-Carlo Planning," in *Proceedings of the 17th European Conference on Machine Learning*, 2006.

[175] J. Koehler, B. Nebel, J. Hoffmann and Y. Dimopoulos, "Extending Planning Graphs to an ADL subset," in *4th European Conference on Planning*, Toulouse, 1997.

[176] S. Koenig, "ICAPS," 2013. [Online]. Available: http://ipc.icaps-conference.org/. [Accessed 05 08 2013].

[177] J. E. Laird, "It Knows What You're Going to Do - Adding anticipation to a quakebot," in *Proc. of the 5th International Conference on Autonomous Agents*, 2001.

[178] I. Little and S. Thiebaux, "Probabilistic Planning vs Replanning," in *ICAPS Workshop on IPC: Past, Present and Future*, 2007.

[179] M. L. Littman and H. Younes, "The Probabilistic Planning Track of the 2004 International Planning Competition," Rutgers, State University of New Jersey, 2004. [Online]. Available: http://www.cs.rutgers.edu/~mlittman/topics/ipc04-pt/. [Accessed 24 06 2013].

[180] W. T. Lord, "USAF Cyberspace Command: To Fly and Fight in Cyberspace," USAF, 2008.

[181] Q. Lu, Y. Xu, R. Huang and Y. Chen, "The Roamer Planner: Random-Walk Assisted Best-First Search," in *Proceedings of the seventh international planning competition*, 2011.

[182] Y. Meiller and P. Fabiani, "TokenPlan - A Planner for Both Satisfaction and Optimization Problems," *AI Magazine,* vol. 22, no. 3, pp. 85-87, 2001.

[183] S. Milton, Machine Learning methods for planning, Morgan Kaufmann, 1994.

[184] M. Nance, A. Vogel and E. Amir, "Reasoning about Partially Observed Actions," in *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

[185] A. Nareyek, "AI in Computer Games," *Queue,* 2004.

[186] N. Nejati, P. Langley and T. Konik, "Learning Hierarchical Task Networks by Observation," in *Proc. of the 23rd International Conference on Machine Learning*, 2006.

[187] E. Onaindia, O. Sapena, L. Sebastia and E. Marzal, "SimPlanner - An Execution-Monitoring system for replanning in dynamic worlds," *Progress in Artificial Intelligence,* pp. 393-400, 2001.

[188] N. Onder, G. C. Whelan and L. Li, "Probapop: Probabilistic partial-order planning," in *Proceedings of the International Conference in Autommated Planning and Scheduling*, 2004.

[189] S. Ong, S. Png, D. Hsu and W. Lee, "Planning under uncertainty for robotic tasks with mixed observability," *International Journal of Robotics Research,* vol. 29, no. 8, pp. 1053-1068, 2010.

[190] E. Parker, "Making Graphplan Goal-directed," *Recent Advances in AI Planning,* 2000.

[191] P. Pederson, D. Lee, G. Shu, D. L. Z. Chen, N. Li and L. Sang, "Virtual Cyber-Security Testing Capability for Large Scale Distributed Information Infrastructure Protection," in *Technologies for Homeland Security*, Waltham, MA, USA, 2008.

[192] E. Pednault, "Formulating Multi-agent, Dynamic world problems in the classical planning framework," in *Reasoning about Actions and Plans*, San Mateo, Morgan Kaufmann, 1987, pp. 47-82.

[193] J. Pineau, G. Gordon and S. Thrun, "Point-based value iteration - An anytime algorithm for POMDPs," in *IJCAI*, 2003.

[194] M. Roberts, A. E. Howe, I. Ray and M. Urbanska, "Using Planning for a Personalized Security Agent," in *Working Notes of the 26th AAAI Conference on Artificial Intelligence*, 2012.

[195] J. Robertson, "Planning in Incomplete Domains," Utah State University, 2012.

[196] E. D. Sacerdoti, "A Structure for plans and behavior," SRI International Menlo Park CA Artificial Intelligence Center, 1975.

[197] E. D. Sacerdoti, "The nonlinear nature of plans," in *IJCAI*, 1975.

[198] J. Sanchez, M. Tang and A. D. Mali, "P-MEP: Parallel More Expressive Planner," in *International Planning Competition*, 2004.

[199] C. Sarraute, O. Buffet and J. Hoffman, "POMDPs Make Better Hackers: Accounting for Uncertainty in Penetration Testing," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[200] L. Sebastia, E. Onaindia and E. Marzal, "STeLLa: An Optimal Sequential and Parallel Planner," *Progress in Artificial Intelligence,* pp. 409-416, 2001.

[201] D. o. H. Security, "National Vulnerability Database," [Online]. Available: http://nvd.nist.gov/. [Accessed 01 07 2013].

[202] K. Staggs, "Security Solutions to Meet NERC-CIP Requirements," Honeywell Process Solutions, 2008.

[203] A. Tate, J. Levine, P. Jarvis and J. Dalton, "Using AI Planning Technology for Army Small Unit Operations," in *Proc. of AIPS 2000*, 2000.

[204] F. Teichteil-Konigsbuch, G. Infantes and U. Kuter, "RFF: A Robust, FF-based MDP Planning Algorithm for Generating Policies with Low Probability of Failure," in *Sixth International Planning Competition at ICAPS*, 2008.

[205] F. Teichteil-Konigsbuch and P. Fabiani, "Symbolic stochastic focused dynamic programming with decision diagrams," in *Proceedings of the Fifth International Planning Competition*, 2006.

[206] D.-V. Tran, H.-K. Nguyen, E. Pontelli and T. C. Son, "CPA(C)/(H) - Two Approximation-Based Conformant Planners," in *Proceedings of Sixth International Planning Competition*, 2008.

[207] V. Vidal, "The YAHSP Planning System: Forward Heuristic Search with lookahead plans analysis," in *International Planning Competition*, 2004.

[208] D. Weld and S. Penberthy, "The UCPOP Planner," [Online]. Available: http://www.cs.washington.edu/ai/ucpop.html. [Accessed 22 07 2013].

[209] B. J. Wood and R. A. Duggan, "Red Teaming of Advanced Information Assurance Concepts," in *DARPA Information Survivability Conference and Exposition*, 2000.

[210] S. Yoon, A. Fern and R. Givan, "Inductive Policy Selection for First-Order MPDs," in *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, 2002.

[211] R. Zhou and E. A. Hansen, "BFHSP: Breadth-First Heuristic Search Planner," in *International Planning Competition*, 2004.

[212] L. Zhu and R. Givan, "Heuristic Planning via Roadmap Deduction," in *4th International Planning Competition Booklet*, 2004.

[213] J. Pineau, G. Gordon and S. Thrun, "Point-based value iteration: An anytime algorithm for POMDPs," in *IJCAI*, 2003.

[214] V. Alcazar, D. Borrajo and C. L. Lopez, "Using Backwards Generated Goals for Heuristic Planning," in *ICAPS 2010*, 2010.

[215] E. W. Dijkstra, "A Note on Two Problems in Connection with Graphs," *Numerische mathematik,* 1959.

[216] B. Seegebarth, F. Muller, B. Schattenberg and S. Biundo, "Making Hybrid Plans More Clear to Human Users - A Formal Approach for generating sound explanations," in *ICAPS 2012*, 2012.

Appendix A: Planners' Capabilities

All planners mentioned in or studied as part of this report are listed below, ordered alphabetically. Several iterations of the same planner may appear if they each offer significantly different capabilities. Noteworthy planners have been highlighted in green.

Citation count refers to the total number of other books, papers and journal articles that have cited the main paper(s) describing the planner[4]. This provides some indication of the impact and influence each planner has within the academic community (ignoring time).

Planner	Source Code	Citation count	Plan Output	Planning Technique	Determinism	Observability	Preference and/or Quality-driven planning?	Handles Dynamic States (Contingent Planning)?	Hand-coded Control Knowledge?	Machine Learning	PDDL Compliance
AltAlt [40]		38	SP	PG	D	F	-	-	-	-	1.0
APPL [74]	[75]	196	ST	POMDP	C	P	-	-	-	-	**
ArvandHerd [76]		9					-	-	-	Yes	3.0
BDDPlan [77]		26	OT	MC	D	F	-	-	-	-	1.0
BLACKBOX [78] [79]	[52]	691	OT	SAT	D	F	-	-	-	-	1.0
C^3 [80]		9	ST	PG	D	F	-	-	-	-	3.0
Conformant-FF [27]	[81]	169	ST	PG	N	P	Yes	Yes	-	-	2.1
CPT [82]		162	OP	PS	D	F	-	-	-	-	2.2
DAEYAHSP [83] [84]		28	ST	PG	D	F	Yes	-	-	-	3.0
DTG [85]		13	ST	PG	D	F	Yes	-	-	-	
Fast Downward [71]	[71]	401	ST	PG/HTN	D	F	-	-	-	Yes	2.2
FCPlanner [53]		22									
FD-Autotune [16]	[86]	9	ST	PG/HTN	D	F	Yes	-	Yes	Yes	3.0
FDP [87]		18	OT	SAT/SS	D	F	-	-	-	-	3.0
FD-SS [88]		17									
FF [28]	[81]	1290	ST	PG	D	F	-	-	-	-	1.0
FF(H_a) [89]		58	ST	PG	D	F					
FF-rePlan [90]		134	ST	PG		F					
FPG [49]		60									
GAMER [39]	[91]	11	OT	PG	D	F	Yes	-	-	-	3.0
Glutton [92]		1									
GraphPlan [36]	[93]	2008	OT	PG	D	F	-	-	-	-	-
GRT [94]		61	OT	SS	D	F	-	-	-	-	1.0
HSP [37]		261	ST	SS	D	F	-	-	-	Yes	1.0
HSP2 [37]		657	ST	SS	D	F	-	-	-	Yes	1.0
IxTeT [95]		219	OP	HTN	D	F	-	Yes	-	-	2.1
LAMA [69] [70]	[96]	153		PG							2.2

[4] using Google Scholar citation database results on 30-Jul-2013

Planner	Source Code	Citation count	Plan Output	Planning Technique	Determinism	Observability	Preference and/or Quality-driven planning?	Handles Dynamic States (Plan Contingency)?	Hand-coded Control Knowledge?	Machine Learning	PDDL Compliance
LPG [97]		210	SP	PG	D	F	Yes	-	-	-	2.1
LPG-TD [98]		30	SP	PG	D	F	Yes	-	-	-	2.2
M&S [99]		24		SS							
Macro-FF [100]		100	ST	PG	D	F	-	-	-	Yes	2.2
Marvin [101]		60	ST	PG	D	F	-	-	-	Yes	2.2
MaxPlan [102]		102	O	SS							
Metric-FF [38]	[81]	293	ST	PG	D	F	Yes	-	-	-	2.1
mGPT [103]		54									
MIPS [104]		81	OT	MC	D	F	Yes	-	-	-	2.2
NMRDPP [105]		26	ST	MDP	P						
OptiPlan [106]		**25**	**OP**	**PG**	**D**	**F**	**-**	**-**	**-**	**-**	**2.2**
PbP [50] [107]	[108]	28	OT	Hybrid	D	F	Yes	-	-	Yes	3.0
PbR [109]		49	ST	SAT	D	F	Yes	-	Yes	Yes	1.0
PGP [110]	[42]	127		PG							
Plan-A [111]		9	ST	SAT	D	F	Yes	-	-	-	3.0
POMCP [112]		84		POMDP							
PropPlan [113]		38	OT	PG	D	F	-	-	-	-	1.0
Sapa [114]		159	SP	SS	D	F	Yes	-	-	-	2.1
SATPlan [115] [116]		914	OT	SAT							
SGP [117]		318	SP	PG	C	P	-	Yes	-	-	1.0
SGPlan [118]		53	ST	PG	D	F	-	-	-	-	2.1
SGPlan4 [119]	[120]	133	ST	PG	D	F	-	-	-	-	2.2
SGPlan5 [121] [122]	[120]	28	ST	PG	D	F	Yes	-	-	-	3.0
SGPlan6 [123]		26	ST	PG	D	F	Yes	-	-	-	3.0
SHOP [43]		391	OT	HTN	D	F	-	-	Yes	-	-
SHOP2 [124]		585	OT	HTN	D	F	-	-	Yes	-	-
STAN [125]		184	ST	PG	D	F	-	-	-	-	1.0
Symbolic Heuristic Search [126]		91	ST	MDP/MC	C	F	Yes	-	-	-	
Symbolic Perseus [127] [128]	[129]	391	ST	POMDP	C	P	Yes	-	-	-	
System R [130]		20	OT	SS	D	F	-	-	Yes	-	1.0
T_0 [21]	[131]	57									
TALPlanner [132]		124	ST	SS	D	F	-	-	-	-	1.0
TLP-GP [133]		10	ST	SAT	D	F	-	-	-	-	3.0
TLPlan [134]		469	OT	SS	D	F	-	-	-	-	2.1
TP4 [135]		164	OT	SS	D	F	Yes	-	-	-	2.1
UCPOP [51]		846	OP	PS							
VHPOP [136]		121	OP	PS	D	F	-	-	-	-	2.1
Wizard [137]	[138]	46									
YAHSP2-MT [139]		7	ST	SS							

{Plan output types: **OT** = Optimal, Total-Order | **OP** = Optimal, Partial-Order | **ST** = Satisficing, Total-Order | **SP** = Satisficing, Partial-Order}
{Planning technique used: **SS** = State-space | **PS** = Plan-space | **PG** = Planning Graph | **SAT** = Planning as Satisfiability | **HTN** = Hierarchical Task Networks | **MC** = Model Checking | **MDP** = Markov Decision Process solver | **POMDP** = Partial Observable MDP solver | **ML** = Machine Learning | Hybrid techniques will list all planner technique that is used }
{Determinism: **D** = Deterministic | **P** = Probabilistic | **C** = Conformant}
{Observability: **F** = Fully Observable | **P** = Partially Observable}

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA	1. DLM/CAVEAT (OF DOCUMENT)

2. TITLE	3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)
Automated Cyber Red Teaming	Document (U) Title (U) Abstract (U)

4. AUTHOR(S)	5. CORPORATE AUTHOR
Joseph Yuen	DSTO Defence Science and Technology Organisation PO Box 1500 Edinburgh South Australia 5111 Australia

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DSTO-TN-1420	AR-016-282	Technical Note	April 2015

8. FILE NUMBER	9. TASK NUMBER	10. TASK SPONSOR	11. NO. OF PAGES	12. NO. OF REFERENCES
2014/1173720/1			34	216

13. DSTO Publications Repository	14. RELEASE AUTHORITY
http://dspace.dsto.defence.gov.au/dspace/	Chief, Cyber and Electronic Warfare Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS Yes

18. DSTO RESEARCH LIBRARY THESAURUS

Automated Planning, Cyber Red Teaming, Algorithms

19. ABSTRACT

Cyber Red Teaming (CRT) is an important exercise to conduct for Defence agencies built on large technological infrastructures. Their size and relative importance may make them high priority targets for criminal organizations, issue-motivated groups and even foreign governments that are increasingly capable and willing to use technology for intelligence gathering. However identifying a viable attack can be a time-consuming process, and so Automated Planners are being considered as a viable method of discovering possible attack paths for Cyber Red Teaming. This report surveys the current state-of-the-art planning algorithms, tools and frameworks, and by observing its benchmark performance, recommends the most suitable ones for trialling.