AI Planning

21. Planning Systems and the IPC

After All this Theory, A Little Bit of Practice
Or: "Eine kleine Gutenachtgeschichte"

Álvaro Torralba, Cosmina Croitoru

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Winter Term 2018/2019

Thanks to Prof. Jörg Hoffmann for slide sources

Agenda

Motivation

Why bother considering the IPC?

- The International Planning Competition (IPC) has been a major driving force (during some periods, the *main* driving force) in the field since about 1997.
- In particular, the efficient implementation of planning systems was and is mostly done by people hoping to win an IPC award.

Why bother making a lecture on it?

- To perform research in planning, it is essential to have at least some basic knowledge about IPC history.
- While we have covered all the basics of heuristic search, many additional techniques are used by implemented systems, and of course the field is broader than this. Here I briefly outline some of the more important things that were out of our scope.
- I thought you'd appreciate, after all these results, to hear a bit about how and in which order they were discovered, and how they are being used.

Disclaimer

It's complicated!

→ The IPC has grown into an incredibly complicated beast, explaining which comprehensively would take an entire course, rather than a 90-minute lecture.

It's controversial!

→ What follows is, to a large extent, my personal perspective. In particular, when I say "What people remember it for" I basically mean "What *I* remember it for".

Our Agenda for This Chapter

Chronological.

I'll basically try to summarize the main events and trends, spiking it up with a variety of fun/odd/embarrassing facts from IPC history.

IPC'98 Quick Facts

Organizer: Drew McDermott.

Number of planning systems: 5.

Language: STRIPS and ADL.

What people remember it for:

- The advent of PDDL.
- The excitement of a competition.
- Somebody did neither Graphplan nor planning-as-SAT. (Aka: The stone age of heuristic search planning.)

Planning Systems Competing in IPC'98

STAN [Long and Fox (1999)]

- **Language**: STRIPS.
- **Approach**: Graphplan, i.e., regression with h^2 (cf. **Chapters 6 and 8**).
- **Plan quality**: Time-step optimal, i.e., smallest number of mutually non-interfering action sets.
- **Specifics**: Effective bitvector implementation of bottleneck routines.

SGP

- **Language**: STRIPS, ADL.
- **Approach**: Graphplan.
- **Plan quality**: Time-step optimal.
- **Specifics**: Pre-processor compiles ADL into STRIPS.

Planning Systems Competing in IPC'98

IPP [Koehler *et al.* (1997)]

- **Language**: STRIPS, ADL.
- **Approach**: Graphplan.
- **Plan quality**: Time-step optimal.
- **Specifics**: Pre-processor compiles ADL into STRIPS with conditional effects, Graphplan extended to deal with this more general input.

BlackBox [Kautz and Selman (1996)]

- **Language**: STRIPS.
- **Approach**: Planning as SAT.
- **Plan quality**: Time-step optimal.
- **Specifics**: Encodes plan existence in a length-d planning graph (cf. Chapter 8) into SAT. Runs $d = 0, 1, 2, \dots$ until first satisfiable d.

Planning Systems Competing in IPC'98

HSP [Bonet and Geffner (2001)] $\rightarrow$ **Chapter 9**

- **Language**: STRIPS.
- **Approach**: Heuristic search.
- **Plan quality**: No guarantees, plans returned are sequential.
- **Specifics**: Forward greedy best-first search with h^{add}.

IPC'98 Benchmarks

Mostly traditional benchmarks, sometimes modified to fool planners/their inventors:

- Assembly: ADL; a complex object must be assembled out of its parts,obeying ordering constraints etc.
- Grid: A robot moves in a grid world and needs to transport keys; grid positions may be locked and must then be opened with a matching key.
- Gripper: A robot with 2 hands must transport n balls from room A into room B. $\rightarrow$ *Intended to point out issues with symmetries.*
- Logistics: As seen, but with airplanes for transport between "cities".
- Movie. Buy some snacks and start a movie. $\rightarrow$ *Intended to point out issues when uselessly scaling the number of snacks available.*
- Mystery: Similar to Logistics, but trucks use one fuel unit in every move; no refuelling possible. $\rightarrow$ *Predicate names disguised to prevent developers from understanding the domain.*
- Mprime. Like Mystery, but with extra operator to transfer fuel between locations. $\rightarrow$ *Predicate names: same as in Mystery.*

IPC'98 Results

Punchline: Not very conclusive.

$\rightarrow$ So what? Generally, all these planners were "in the same league". Which *was* exciting because one of them – HSP – did *not* inherit its efficiency from Graphplan. Several people started to work on this kind of approach.

IPC'00 Quick Facts

Organizer: Fahiem Bacchus.

Number of planning systems: 12 fully automatic, 8 hand-tailored.

Language: STRIPS and ADL.

What people remember it for:

- The advent of FF and the relaxed plan heuristic.
- Dramatic performance boost of automatic planning, brought about by heuristic search planners.
- Dramatic performance of hand-tailored planning with TALplanner.

The "Hand-Tailored" Track

The Credo of Automatic Planning

"Physics, not advice."

... which is to say, the planning system should solve problems just based on a description of what the problem is, without any information as to how the problem should be solved.

- This notwithstanding, PDDL *can* be used like a programming language [Rintanen (2000)]
- If there is knowledge about how to solve the problem, why should we not allow the user to provide it ...??

→ In the hand-tailored track, the developer is allowed to supply the planning system with an additional (arbitrary) input file for each planning task.

Selected Planning Systems Competing in IPC'00

FF [Hoffmann and Nebel (2001)] → Chapter 9

- **Language**: STRIPS, ADL.
- **Approach**: Heuristic search.
- **Plan quality**: No guarantees, plans returned are sequential.
- **Specifics**: Forward search with h^{FF}, using enforced hill-climbing with helpful actions pruning, if that fails switching to greedy best-first search.

TALplanner [Bacchus and Kabanza (2000); Doherty and Kvarnström (2001)]

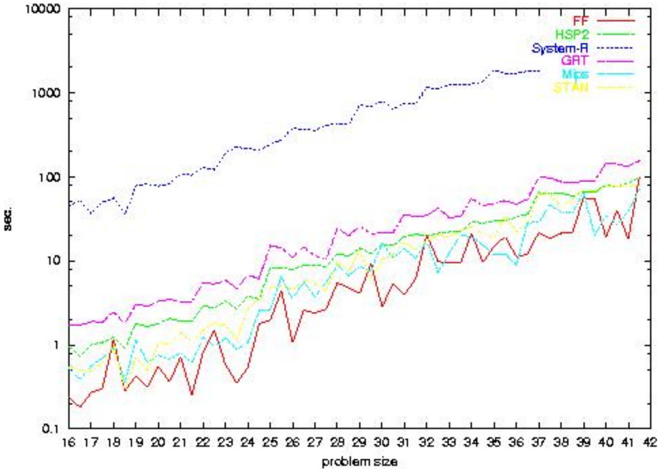
- **Language**: STRIPS, ADL.
- **Approach**: Depth-first search with user-provided pruning.
- **Plan quality**: No guarantees, plans returned are sequential.
- **Specifics**: Forward depth-first search, pruning a search branch whenever a control rule, specified by the user in a temporal logic, applies.

IPC'00 Benchmarks

Traditional benchmarks, a game, and an application:

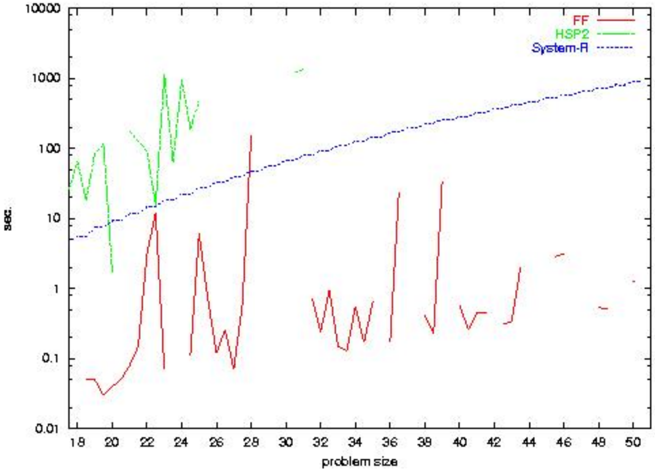
- Logistics: As in 1998.
- Blocksworld: Move around blocks on a table (yeah, I know).
- Freecell: The card game.
- Miconic-ADL: A complex elevator-control problem.
- Schedule: A simple scheduling problem where objects must be processed with various machines.

IPC'00 Results, Fully Automatic Track



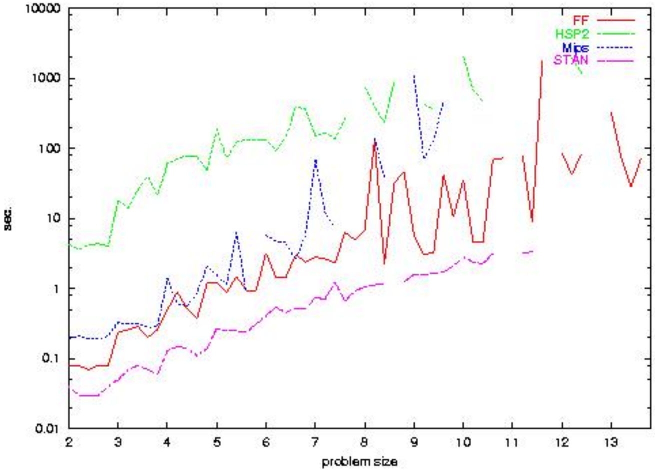
Logistics

IPC'00 Results, Fully Automatic Track



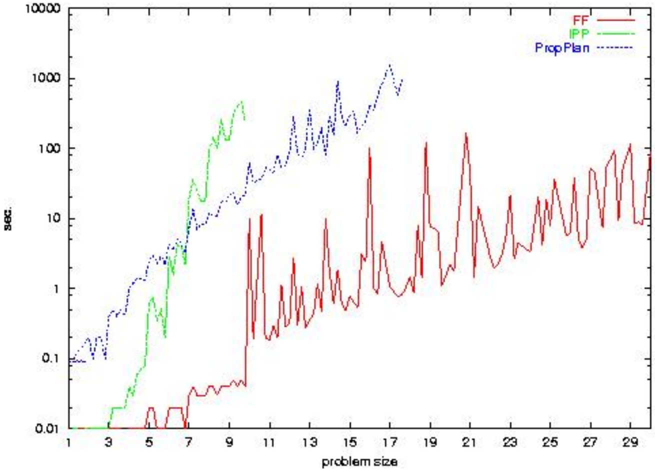
Blocksworld

IPC'00 Results, Fully Automatic Track



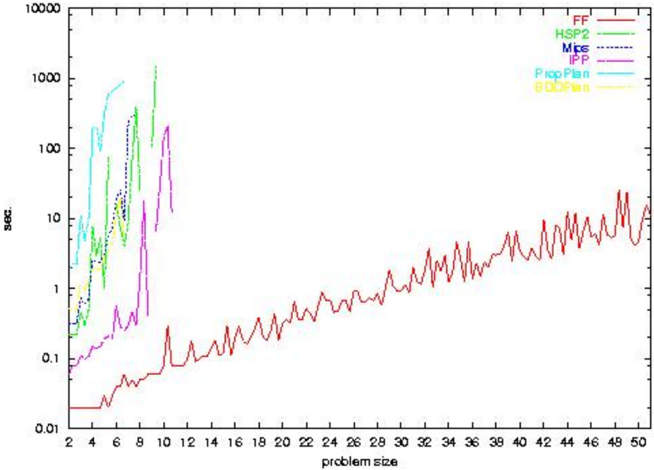
Freecell

IPC'00 Results, Fully Automatic Track



Miconic

IPC'00 Results, Fully Automatic Track



Schedule

IPC'00 Results, Fully Automatic Track: So What?

This was *dramatic!*

→ Systems using delete relaxation heuristics, in particular FF, dramatically outperformed the others, solving tasks that were previously completely infeasible.

Three main consequences:

- Lots of people taking up research along these lines, eventually leading to the rise of heuristic search from "uh yeah, I think Hector Geffner does that, right?" to the uncontested ruler of the house.
- People outside the planning community realizing (slowly but gradually) that they can now use planning systems to solve their problems.
- People inside the planning community "setting their aims higher", to numeric and temporal planning.

IPC'02 Quick Facts

Organizers: Maria Fox and Derek Long.

Number of planning systems: 11 fully automatic, 3 hand-tailored.

Language: STRIPS, numeric, temporal.

What people remember it for:

- The advent of temporal and numeric planning in the IPC.
- LPG combines stochastic local search with relaxed plan heuristics, performing very well across all language categories.
- Post-IPC disenchantment with the hand-tailored track: *"What are we evaluating here?"*

Selected Planning Systems Competing in IPC'00

Metric-FF [Hoffmann (2002, 2003)]

- **Language**: STRIPS, ADL, numeric.
- **Approach**: Heuristic search.
- **Plan quality**: No guarantees, plans returned are sequential.
- **Specifics**: Identical to FF, but with an extension of h^{FF} to numeric planning (transformation to a positive normal form, then ignoring effects decreasing the values of numeric variables).

Selected Planning Systems Competing in IPC'00

LPG [Gerevini *et al.* (2003)]

- **Language**: STRIPS, ADL, numeric, temporal.
- **Approach**: Stochastic local search.
- **Plan quality**: No guarantees, plans returned are sequential.
- **Specifics**: Search nodes are partial plans embedded into a 2-planning graph. Search starts from the empty plan. In each search step, the possible successor nodes modify the current partial plan by adding or removing an action, keeping track of conflicts in the planning graph. Node quality is evaluated by a delete relaxation heuristic adding penalties for conflicts.

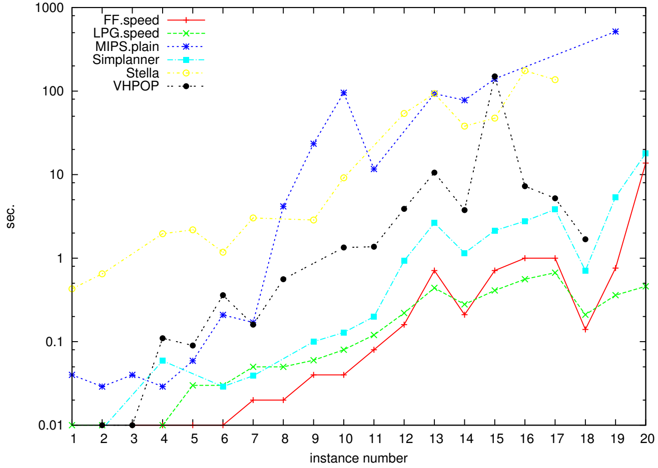
  → This search has many parameters, choosing "good" values for which is important to obtain good performance ...

IPC'02 Benchmarks

Modified traditional benchmarks and space applications, most with a STRIPS, a numeric, a temporal, and a numeric-temporal version:

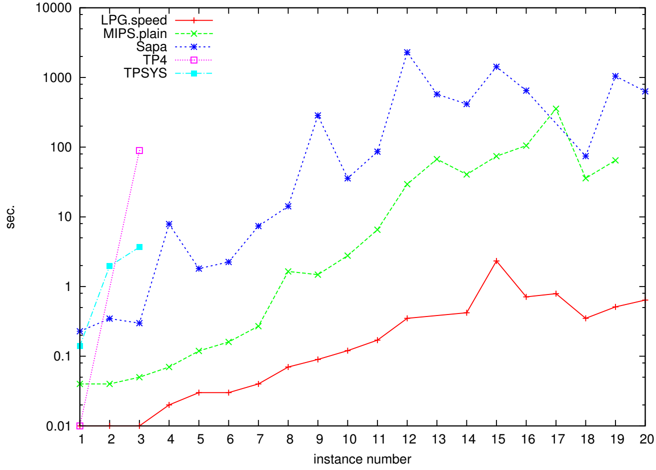
- Depots: A combination of Blocksworld and Logistics.
- Driverlog: Trucks need drivers in order to move on a road map without one-way streets.
- Freecell: As in 2000. $\rightarrow$ *Just STRIPS, not temporal/numeric.*
- Rovers: Rovers must navigate along a road map, and take soil/rock samples as well as images.
- Satellite: Satellites must take images.
- Settlers: Build up an infrastructure of trains, housing, etc. $\rightarrow$ *Relies almost exclusively on numeric variables, no plain STRIPS and temporal versions.*
- Zenotravel: Airplanes use fuel, refuelling possible.

IPC'02 Results, Fully Automatic Track: Satellite



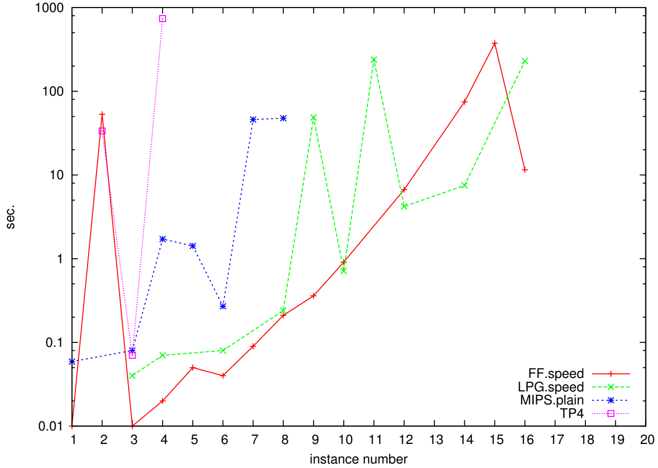
STRIPS

IPC'02 Results, Fully Automatic Track: Satellite



Temporal

IPC'02 Results, Fully Automatic Track: Satellite



Numeric

IPC'02 Results, Fully Automatic Track: So What?

Punchlines:

- STRIPS planning: Delete relaxation heuristics, in particular within FF and LPG, still dominant.
- Temporal planning: Effectiveness of delete relaxation heuristics, in particular within LPG, can often be retained from the STRIPS version.

  $\rightarrow$ Note though: These temporal benchmarks were obtained by extending STRIPS benchmarks, introducing an unintentional bias. It was later found that they are all "without required concurrency", i.e., sequential plans exist [Cushing *et al.* (2007)].
- Numeric planning: Here these heuristics – and all other competing techniques – are *not* as effective.

  $\rightarrow$ Plausible explanation: The numeric variables here express resource constraints, which are **NP**-hard to deal with. Delete relaxation heuristics, in particular, act as if resources are not consumed at all. (See [Coles *et al.* (2008); Nakhost *et al.* (2012)] for recent work on this issue.)

What About the Hand-Tailored Track?

Winner this time: SHOP2 [Nau *et al.* (2003)].

But so what??? *What does it mean to be "more effective" in the hand-tailored track?*

→ Is it because the approach is good? Or is it because the domain knowledge supplied is good? . . . as Rao Kambhampati put it:

"Download SHOP2 here, download a Dana Nau here."

→ Hand-tailored planning is a form of programming. What we would need to evaluate is how effectively one can program in that language. Doing so in an IPC setting seems quite difficult.

IPC'04 Quick Facts

Organizers: Jörg Hoffmann and Stefan Edelkamp (deterministic); Michael Littman and Hakan Younes (probabilistic).

Number of planning systems: 13 satisficing, 7 optimal, 8 probabilistic.

Language: STRIPS, ADL, derived predicates, numeric, temporal, probabilistic.

What people remember it for:

- SGPlan exhibits great performance, clearly winning the satisficing track. But then later on (2008), SGPlan is identified to be cheating (invoking domain-specific code based on finding parts of domain names etc).
- Satisficing planners separated from optimal ones. SATPLAN wins the optimal track.
- The probabilistic guys "beat themselves with FF".

IPC'04 Tracks: Optimal vs. Satisficing

Some completely obvious facts:

- A satisficing planner makes a claim about *one* plan ("This action sequence is a plan."). An optimal planner makes a claim about *all* plans ("Every other plan has higher cost.").
- They are solving different problems: PlanEx vs. PlanOpt.
- While both are **PSPACE**-complete in general, for many benchmark domains their domain-specific complexity is different.
- Optimal planning typically is *much* slower than satisficing planning in practice.
- In most applications, it makes a huge difference whether or not the solver provides us with a quality guarantee.

→ So we introduced separate tracks in IPC'04.

→ You wouldn't believe how controversial this was, at the time!

IPC'04 Tracks: Probabilistic Planning

Pioneering a competition for planning under uncertainty:

- Probabilistic PDDL (PDDL) extends PDDL with probability distributions over action effects:

```
(probabilistic 0.166 (dice-1)
               0.166 (dice-2) ... 0.17 (dice-6))
```

  → **Chapters 2, 3** (partly skipped over in the lecture)
- Evaluation through online plan *execution*.
- Competitor connects to server. Repeat until goal or give-up: Server provides current state, planner provides action to take.
- Score: Number of tasks solved within a given amount of time.

Selected Planning Systems Competing in IPC'04

Probabilistic track: "First-order value iteration", "NMRDPP augmented with control knowledge", "Offline policy iteration by reduction to classification", ... and:

FF-Replan [Yoon *et al.* (2007)]

- **Language**: Probabilistic STRIPS.
- **Approach**: Re-planning using FF.
- **Plan quality**: No guarantees, plans returned are sequential.
- **Specifics**: For every state s obtained from the server for task Π, obtain deterministic Π' by acting as if we could choose the outcome of each action, run FF to obtain a plan, give that plan's first action back to the server.

  In other words: If the "wrong" outcome occured, just re-try.

Selected Planning Systems Competing in IPC'04

Fast Downward [Helmert (2006)]

- **Language**: STRIPS, ADL, derived predicates.
- **Approach**: Heuristic search.
- **Plan quality**: No guarantees, plans returned are sequential.
- **Specifics**: Forward greedy best-first search with a new heuristic function (the "causal graph heuristic" Helmert (2004), later on subsumed by the "context-enhanced additive heuristic" Helmert and Geffner (2008)).

  Fast Downward also introduced dual-queue best-first search, and new pre-processing methods for translating the PDDL input into FDR [Helmert (2009)].

Selected Planning Systems Competing in IPC'04

SGPlan [Yixin Chen, Benjamin Wah, Chih-Wei Hsu; citation omitted]

- **Language**: STRIPS, ADL, derived predicates, temporal, numeric.
- **Approach**: Supposedly, problem decomposition and heuristic search.
- **Plan quality**: No guarantees, plans returned are sequential.

→ In 2008, when all competitors were asked to publish their source code, Malte Helmert looked into the SGPlan code and discovered that it was recognizing particular benchmark domains and invoking domain-specific planners.
The domain recognition was designed stealthily in awareness that this was against the rules ("if domain name starts with "d" and last action has 5 arguments, then . . . ").

→ This really is how to NOT do it. To be fair: It is not clear who had which part in this, and Yixin Chen at least has been doing good work since then.

IPC'04 Deterministic Benchmarks

Creating "good" benchmarks was our main endeavor: *All of them were modeled based on applications, and we took care to cover different complexity classes and regions of* h^+ *topology [Hoffmann* et al. *(2006)].*

- Airport: Ground traffic on airports [Trüg *et al.* (2004)].
- Pipesworld: Liquid flow through a pipeline system [Milidiu *et al.* (2003)].
- Promela: Deadlock detection in communicating automata expressed in the "Promela" language [Edelkamp (2003a)].
- PSR: Power supply restoration in faulty electricity networks [Thiébaux and Cordier (2001)].
- Satellite: The IPC'02 domain, enhanced with time windows for sending data to earth.
- Settlers: As in IPC'02.
- UMTS: UMTS (Universal Mobile Telecommunications System) call setup (time optimization problem) [Englert (2005)].

IPC'04 Results

Punchlines:

- Satisficing planning: SGPlan was the most effective, or among the most effective, planners for each and every domain. So it got the 1st prize.
- Optimal planning: SATPLAN (BlackBox updated with more recent SAT solvers) was most efficient by far, so it got the 1st prize.
- Probabilistic planning: FF-Replan worked best by far . . . !

→ Quote presentation at ICAPS: "Just as a sanity check, we made this simple planner using FF . . . and then we actually beat ourselves with it!"

IPC'06 Quick Facts

Organizers: Alfonso Gerevini, Alessandro Saetti, Patrick Haslum, Yannis Dimopoulos (deterministic); Blai Bonet and Bob Givan (uncertainty).

Number of planning systems: 6 satisficing, 6 optimal, 7 uncertainty.

Language: STRIPS, ADL, derived predicates, numeric, temporal, soft goals, trajectory constraints, non-deterministic, probabilistic.

What people remember it for:

- Extremely complex language for deterministic planning, now also with soft goals and trajectory constraints; most (all?) competing systems compile these away in a pre-process to planning.
- SGPlan wins again. (Two more years til we realize they are cheating . . .)
- SATPLAN wins the optimal track again.
- The probabilistic guys "beat themselves with FF" yet again.
- The best non-deterministic planner compiles into ADL and uses FF.

Selected Planning Systems Competing in IPC'06

MIPS [Edelkamp (2003b, 2006)]

- **Language**: STRIPS, ADL, derived predicates, numeric, temporal, soft goals, trajectory constraints.
- **Approach**: Heuristic search.
- **Plan quality**: No guarantees, plans returned are sequential.
- **Specifics**: Compilations! Forward greedy best-first search with Metric-FF relaxed plan heuristic. Temporality is ignored during planning, then put back in in a plan-scheduling post-process. Soft goals and trajectory constraints are encoded as temporal logic, which is encoded into automata using methods from Verification, and these automata are then encoded into artificial facts and actions.

Selected Planning Systems Competing in IPC'06

T0 [Palacios and Geffner (2007)]

- **Language**: Non-deterministic STRIPS.
- **Approach**: Compilation to ADL, using FF.
- **Plan quality**: No guarantees, plans returned are sequential.
- **Specifics**: Deals with conformant planning under initial state uncertainty, where the initial state is a CNF formula and the same plan should work for every state satisfying that formula. Compilation to ADL using artificial "what-if" facts, like "If I was at A initially, then I am now at B".

→ **Chapter 2** (covered very briefly only)

IPC'06 Deterministic Benchmarks

New application benchmarks, some modified previous ones:

- TPP: Traveling and buying goods at selected markets, minimizing costs (from OR with variants, **NP**-hard).
- Openstacks: Combinatorial optimization problem in production scheduling (from CSP benchmarks, **NP**-hard).
- Storage: Moving and storing crates of goods by hoists from containers to depots, with spatial maps.
- Pathways: Finding a sequence of biochemical (pathways) reactions in an organism producing certain substances.
- Trucks: Moving packages between locations by trucks under certain spatial constraints and delivery deadlines.
- Rovers: From IPC'02, with new variants involving new PDDL features.
- PipesWorld: From IPC'04, with new variants involving new PDDL features.

IPC'06 Results

Punchlines:

- Non-deterministic planning: T0 (compilation to deterministic planning with FF) won very clearly.

  → Funny part: One of the competitors was Conformant-FF [Hoffmann and Brafman (2006)], my extension of FF to conformant planning. Palacios & Geffner beat me with my own system!

  → Palacios & Geffner later showed that there are compilations exponential only in a "width" parameter that is small in most current benchmarks [Palacios and Geffner (2009)]. Their work spawned a whole research line successfully compiling various forms of planning under uncertainty into classical planning. For many variants, this is currently the state of the art.

IPC'06 Results, ctd.

Punchlines:

- Deterministic planning: Compilation is the most popular approach taken to deal with the complicated language.
- Probabilistic planning: Same as last time ... and Bob Givan's presentation *really* pissed off people working on probabilistic planning.

(The slides are not online, but essentially he said "it seems that probabilistic planning is not useful").

Questionnaire

Question!

Is probabilistic planning useless?

(A): Yes (B): No

→ No! FF-Replan won because of the benchmark design!

Example 1: If all you need to do is throw a dice and wait for a 6, then of course there is no point in probabilistic reasoning.

Example 2: (used in the competition!) If you have no chance to avoid a possibly deadly event ("block explodes"), then as well there is no point in probabilistic reasoning.

→ Probabilistically interesting problems: Ones where several alternative choices exist and their risk/benefit trade-off needs to be evaluated [Little and Thiebaux (2007)]. Example? "Which train to take to the airport?"

IPC'08 Quick Facts

Organizers: M. Do, M. Helmert, I. Refanidis (deterministic); O. Buffet, D. Bryce (uncertainty); A. Fern, R. Khardon, P. Tepalli (learning).

Number of planning systems: 9 sequential satisficing, 8 sequential optimal, 5 temporal satisficing, 3 net-benefit optimal, 7 uncertainty, 14 learning.

Language: STRIPS + action costs, STRIPS + durative actions, STRIPS + action costs + soft goals, non-deterministic, probabilistic.

What people remember it for:

- Tightening the rules: Clearly defined separate tracks and optimization objectives, blind planner submission, precise "winning" criterion.
- The advent of LAMA (winner sequential satisficing).
- The baselines are hard to beat: FF would have 2nd place in sequential satisficing, simplistic temporal FF would have 1st place in temporal satisficing, blind search would have 1st place in sequential optimal.
- FF-Replan stops winning the probabilistic competition; now it's RFF.
- The best learner is a portfolio that performs better *without* learning.

IPC'08 Tracks: Deterministic

6 completely separate tracks with clearly defined objectives:

- Sequential: STRIPS + action costs; objective is to minimize the summed-up cost of actions in the plan.
- Temporal: STRIPS + durative actions; objective is to minimize the makespan: the difference between end- and start-time of the plan.
- Net benefit: STRIPS + action costs + soft goals; objective is to maximize the difference between the utility of achieved goals and cost of the plan.

- Within each of these 3 categories, a satisficing track and an optimal track. One winner, one runner-up for each track.

IPC'08 Deterministic Track: Rules

Blind planner submission:

- The participants submit their final planners *before* seeing any of the competition benchmarks.
- (Previously, they ran their planners themselves and were allowed to make modifications provided all results were obtained with the same planner.)

Precise "winning" criterion:

- Satisficing: For each task, the planner gets score $\frac{\text{cost of best known plan}}{\text{cost of returned plan}}$.
  $\rightarrow$ Runtime does not matter at all, full 30 minutes can/should be taken to improve plan quality.
- Optimal: For each task solved optimally, the planner gets score 1; if a task is solved non-optimally, then 0 points for the entire domain.

IPC'08 Tracks: Learning

Pioneering a competition for planning with learning:

- In applications, typically a planner keeps solving closely related instances (from the same domain).
- Leverage this for more effective planning!
- Competition format: Participants submit a "learner" and a "planner"; the learner runs on "training" instances and outputs a knowledge file; the knowledge file is added to the input of the planner when run on the competition (the "test") instances.
- Evaluation by scores as in the deterministic track, for runtime and plan quality.

Selected Planning Systems Competing in IPC'08

PbP.s [Gerevini *et al.* (2009b)]

- **Language**: STRIPS, ADL, derived predicates, action costs.
- **Approach**: Portfolio, learning time shares for a set of component planners.
- **Plan quality**: No guarantees, plans returned are sequential.
- **Specifics**: Component planners are, amongst others, Fast Downward, Metric-FF, LPG, SGPlan.

Selected Planning Systems Competing in IPC'08

LAMA [Richter and Westphal (2010)]

- **Language**: STRIPS, ADL, derived predicates, action costs.
- **Approach**: Heuristic search.
- **Plan quality**: No guarantees, plans returned are sequential.
- **Specifics**: To find a first plan quickly, runs forward multiple-queue greedy best-first search, with h^{FF} and an inadmissible landmarks heuristic (counting unachieved landmarks, cf. Chapter 14), each with two queues one for all actions one for only helpful actions.

  To find incrementally improved plans, runs weighted A* with iteratively decreasing weight, where the best known plan is used for pruning.

→ Baseline sequential satisficing: Throw away costs and run FF.

→ Baseline temporal satisficing: Throw away durations and run FF, schedule plan in post-process.

→ Baseline sequential optimal: A^* with $h = 0$.

Selected Planning Systems Competing in IPC'08

RFF [Teichteil-Königsbuch *et al.* (2010)]

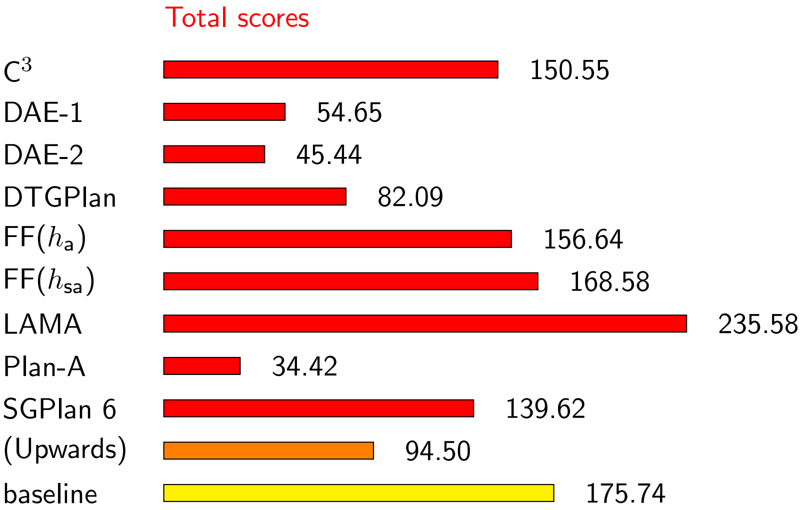
- **Language**: STRIPS, probabilistic.
- **Approach**: Incremental policy build-up using FF.
- **Plan quality**: No guarantees, plans returned are partial policies, mapping states to actions.
- **Specifics**: Starting with empty policy, generate a deterministic plan like FF-Replan does, and integrate the corresponding states and actions into the policy. Execute. If a state not in the policy is encountered, make a deterministic plan from that state to some state already in the policy, and integrate the plan into the policy.

IPC'08 Deterministic Benchmarks

New application benchmarks, games, extended previous benchmarks:

- Crew planning: Application planning the activities of spaceflight crews [Barreiro *et al.* (2009)].
- Cyber security: Attack planning [Boddy *et al.* (2005)] (cf. **Chapter 4**).
- Elevators: Control a system of elevators with capacity constraints.
- Model train: Control train movements and switches (German: "Weichen").
- Openstacks: As in IPC'06.
- PARC printer: Printer control [Ruml *et al.* (2011)] (cf. **Chapter 4**).
- Peg solitaire: The game.
- Scanalyzer: Greenhouse application [Helmert and Lasinger (2010)].
- Sokoban: The game.
- Transport: Logistics-type problem with capacity constraints and road maps.
- Woodworking: Scheduling domain involving operations on wood.

IPC'08 Results: Sequential Satisficing



IPC'08 Results

Punchlines:

- Sequential satisficing: LAMA wins very clearly. Closest competitor is the baseline (FF ignoring costs).
- Sequential optimal: Baseline ($\mathrm{A}^*$with $h = 0$) is 1 point ahead of the best competitor.

  $\rightarrow$ Note: This was *before* the invention of admissible landmark heuristics; PDBs and merge-and-shrink did not participate.
- Temporal satisficing: Baseline (FF ignoring durations and scheduling plan in post-process) clearly outperforms all competitors.
- Probabilistic: New domains chosen to be probabilistically interesting as per [Little and Thiebaux (2007)]; RFF wins clearly, FF-Replan lags far behind (the single domain it wins is an old, not probabilistically interesting, one).
- Learning: PbP.s wins very clearly, but performs slightly worse *with* learning than *without* learning. Other planners *do* profit from their learning phases.

IPC'11 Quick Facts

Organizers: Angel Garcia Olaya, Carlos Linares Lopez, Sergio Jimenez (deterministic); Scott Sanner, Sungwook Yoon (uncertainty); Sergio Jimenez, Amanda Coles, Andrew Coles (learning).

Number of planning systems: 27 sequential satisficing, 12 sequential optimal, 8 temporal satisficing, 8 sequential multi-core satisficing, 11 uncertainty, 8 learning.

Language: STRIPS + action costs, temporal STRIPS, non-deterministic, probabilistic.

What people remember it for:

- Deterministic language + rules constant from IPC'08.
- LAMA wins again.
- Heuristic search (first 12 [9] places in sequential satisficing [optimal] track) and portfolios (3 out of first 4 places, in both tracks) rules the house.
- Fast Downward is the definite code base for deterministic non-temporal planning: 20 out of 47 competing planners, 6 out of 7 awards.
- PPDDL turns into RDDL, and FF-Replan disappears.

IPC'11 Tracks

Deterministic:

- Sequential, Temporal, Net benefit: As in IPC'08.
- Multi-Core: As sequential but with multiple processors; only satisficing.

Learning: As at IPC'08.

Uncertainty:

- MDPs and POMDPs; in particular, no non-deterministic track (without probabilities).
- Language change to Relational Dynamic Influence Diagram Language (RDDL), based on dynamic Bayesian networks. More expressive than PPDDL, closer to languages traditionally used in MDP community. focuses on "external events" (e.g. traffic distribution in traffic light control).

Selected Planning Systems Competing in IPC'11

Fast Downward Stone Soup (Satisficing) [Helmert *et al.* (2011)]

- **Language**: STRIPS, ADL, derived predicates, action costs.
- **Approach**: Portfolio over heuristic search technique combinations.
- **Plan quality**: No guarantees, plans returned are sequential.
- **Specifics**: Portfolio assembled from a set of component planners by assigning each a time share. The time shares were assigned by hill-climbing in the space of time shares, maximizing performance on the IPC'98 – IPC'08 benchmarks.

  Component planners are any possible combination of: greedy search vs. weighted A^*; "eager" search vs. "lazy" search; helpful actions or not; any subset of h^{add}, h^{FF}, causal graph heuristic, and context-enhanced additive heuristic [Helmert and Geffner (2008)].

  Outcome of hill-climbing difficult to describe, a mixture of lots of things.

Selected Planning Systems Competing in IPC'11

Fast Downward Stone Soup (Optimal) [Helmert *et al.* (2011)]

- **Language**: STRIPS, action costs.
- **Approach**: Portfolio over heuristic search planners.
- **Plan quality**: Sequential cost-optimal.
- **Specifics**: As on previous slide, but with cost-optimal component planners.

Further optimal competitors: (all using A^*)

- LM-cut alone.
- Portfolio of 2 bisimulation-based merge-and-shrink heuristics.
- Selective max [Domshlak *et al.* (2012)] which learns when to use an accurate but expensive heuristic (LM-cut) vs. a less accurate but fast heuristic (fact LMs induced LMs heuristic).

Selected Planning Systems Competing in IPC'11

Madagascar, Madagascar-p [Rintanen (2010)]

- **Language**: STRIPS, ADL.
- **Approach**: Planning as SAT.
- **Plan quality**: No guarantees, plans returned are non-temporal parallel.
- **Specifics**: A combination of techniques developed during the last decade, speeding up SATPLAN at the cost of foresaking optimality guarantees.

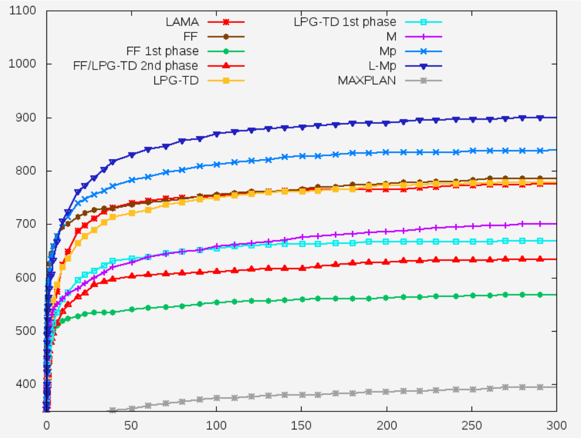
  (A) Instead of Graphplan's sets of pairwise non-interfering actions, use sets for which *at least one* legal sequentialization exists ($\Longrightarrow$ more parallelity, fewer time steps, less SAT calls). (B) Instead of increasing bound d from 0 incrementally, run some d for a limited amount of time then run some other d. (C) [only in Madagascar-p] Instead of using generic SAT solver, use SAT solver with planning-specific branching heuristic in DPLL.

IPC'11 Deterministic Non-Temporal Benchmarks

ALL IPC'08 benchmarks, plus new ones of which many are intended to be challenging for relaxed plan heuristics:

- Barman: Robot barman manipulates drink dispensers, glasses and a shaker.
- Floortile: Robots need to paint a floor, and once a tile is painted they cannot traverse it any longer. → *Need to take care to not "paint yourself into a corner", something relaxed plan heuristics are completely unaware of.*
- NoMystery: Transportation with fuel consumption and no refuelling; initial fuel supplies calculated to be close to minimum needed. → *Difficult for relaxed plan heuristics because they are unaware of fuel consumption.*
- Tidybot: Robots move in a grid world, having to displace objects using grippers/carts.
- VisitAll: Need to visit all locations in a grid, starting from the middle. → *Difficult for relaxed plan heuristics because advancing towards one goal (e.g., moving to the left) increases the distance to other goals (e.g., on the right), leading to huge plateaus in the search space.*

Non-IPC'11 Results: Coverage of Madagascar



X-axis: time-out (sec.), Y-axis: coverage (number of instances solved).

M, Mp: Madagascar, Madagascar-p; L-Mp: + LAMA.

IPC'11 Results

Punchlines:

- Sequential satisficing: LAMA wins, followed closely by two versions of Fast Downward Stone Soup. Heuristic search on the first 12 places. M and Mp not competitive in IPC'11 score, due to very small coverage in 9 domains.
- Sequential optimal: Two versions of Fast Downward Stone Soup win very clearly, followed by merge-and-shrink portfolio and selective max with identical score (169), followed by LM-cut with score 167 (all these are implemented in Fast Downward). Heuristic search on the first 9 places.
- Learning: PbP.s (new variant) wins again. This time, for all but one of these planners, the learned knowledge helped (at least to improve coverage).
- Uncertainty: Both tracks (MDP/POMDP) are won by systems relying on UCT [Kocsis and Szepesvári (2006)]. FF-Replan is not run (to the dismay of the organizers, the most frequent question during the results presentation at ICAPS'11 was "where is FF-Replan?").

IPC'14

Disclaimer: There were deterministic, learning, and probabilistic tracks.

(A) I'll do only the deterministic one (too lazy for the other two right now).

(B) As these slides clearly won't scale anyway (1 more section every 3 years), let's just say it with the organizers' words:

`http://helios.hud.ac.uk/scommv/IPC-14/repository/slides.pdf`

IPC'14 Deterministic Track Punchlines

- Prediction-based portfolios are on the rise!

  $\rightarrow$ Idea: Collect a set C of component planners; define features F on planning task; offline, learn predictor P which $c \in C$ works best given which values of F; online, when receiving a new input planning task Π, measure the values of F and apply P to select $c \in C$.

  Originally conceived in the SAT community: SATzilla [Xu *et al.* (2008)]. Realized for planning in IBACOP [Cenamor *et al.* (2014)].

- BDDs + reachability analysis beats heuristic search in optimal planning!

- SAT-based satisficing planning close to the top in fast ("agile") planning!

IPC'18 Quick Facts

Organizers: Florian Pommerening and Álvaro Torralba (classical); Moisés Martinez, Amanda Coles, Andrew Coles (temporal); Thomas Keller (probabilistic).

Here, I focus on the clasical track. Four subtracks:

- Satisficing: find plans of as good quality as possible
- Optimal: find optimal plans
- Agile: find plans as quickly as possible
- Cost-Bounded: find plans that have a cost lower than a given bound

`https://ipc2018-classical.bitbucket.io/`

So, what techniques are right now the "state of the art"? (disclaimer: with this particular setting, on this benchmark set, etc.)

	A^*	Symbolic	PDB	M&S	CEGAR	LM-cut	h^m	Symmetry	POR	# best	Score
Delfi1	✓	✓	✓	✓		✓		✓	✓	1	126
Complementary2	✓		✓							1	124
Complementary1	✓		✓							2	124
Planning-PDBs	✓		✓							1	122
symb. Bi-dir.		✓								2	118
Scorpion	✓		✓		✓					5	109
Delfi2	✓	✓	✓							0	105
FDMS2	✓			✓						0	104
FDMS1	✓			✓						0	101
DecStar	✓					✓		✓	✓	2	99
Metis1	✓					✓		✓	✓	1	93
MSP	✓	✓				✓				0	91
Metis2	✓					✓		✓	✓	2	87
Blind	✓									0	84
Symple-2		✓								0	58
Symple-1		✓								0	57
maplan-2	✓						✓			1	46
maplan-1	✓						✓			0	43

Sat score	**GBFS**	**EHC**	**SAT**	**hFF**	**hRB**	**hCFF**	**Lm**	**Nov**	**# best**	**Score**
FF	✓	✓		✓						
LPG				✓						
Fast Downward	✓			✓						
LAMA 2011	✓			✓			✓		1	107
IbaCop 2014	✓		✓	✓			✓			
Stone Soup	✓			✓			✓		1	123
Remix	✓			✓			✓		1	120
DUAL-BFWS	✓			✓			✓	✓	2	119
Saarplan	✓	✓		✓	✓	✓	✓		1	116
DecStar	✓			✓			✓		0	111
Cerberus | -gl	✓				✓		✓	✓	1 | 2	108 | 106
BFWS-Pref.	✓			✓			✓	✓	1	106
OLCFF	✓	✓		✓		✓	✓	✓	0	92
POLY-BFWS	✓							✓	0	90
IBaCoP 1–2	✓		✓	✓	✓		✓		2	73 | 66
MERWIN	✓				✓		✓	✓	1	62
mercury	✓				✓		✓		0	61
DFS+	✓							✓	0	60
fs-sim	✓							✓	0	53
fs-blind	✓							✓	0	50
freelunch-dr | -ma			✓						0	22
Symple-1 | 2									0	11 | 11
alien	✓			✓					0	9

Summary

- "IPC" is for International Planning Competition, which was run in '98, '00, '02, '04, '06, '08, and '11. Optimal planners are separated from satisficing ones since '04.
- PDDL was initially conceived for IPC'98, was extended until IPC'06, and has been stable (back to a very small subset, in IPC'08 and IPC'11) since then.
- Apart from the large deterministic track, there are many interesting developments in the uncertainty and learning tracks.
- Heuristic search revolutionized satisficing planning in '00, and optimal planning in '11. Most award-winning systems rely on this paradigm.

  → But beware! This is only on IPC benchmarks and according to IPC evaluation criteria! (Cf. **Christmas Surprise Lecture**)

  → In particular, SAT-based planning has caught up a lot recently [Rintanen *et al.* (2006); Rintanen (2010)], even though it didn't get high scores at IPC'11.

IPC Links and Publications

Official ICAPS IPC web page: (contains links to all IPCs)

`http://www.icaps-conference.org/index.php/Main/Competitions`

Papers on the IPC:

- IPC'98: [Long *et al.* (2000)].
- IPC'00: [Bacchus (2001)].
- IPC'02 PDDL language: [Fox and Long (2003)]. IPC'02 results: [Long and Fox (2003)].
- IPC'04 PDDL language and results: [Hoffmann and Edelkamp (2005)]. IPC'04 benchmarks: [Hoffmann *et al.* (2006)]. IPC'04 probabilistic track: [Younes *et al.* (2005)].
- IPC'06 PDDL language and results: [Gerevini *et al.* (2009a)].
- IPC'11: [Coles *et al.* (2012)].

References I

Fahiem Bacchus and Froduald Kabanza. Using temporal logics to express search control knowledge for planning. *Artificial Intelligence*, 116(1–2):123–191, 2000.

Fahiem Bacchus. The AIPS'00 planning competition. *The AI Magazine*, 22(3):47–56, 2001.

J. Barreiro, G. Jones, and S. Schaffer. Peer-to-peer planning for space mission control. In *IEEE Aerospace conference 2009*, pages 1 –9, 2009.

Mark Boddy, Jonathan Gohde, Tom Haigh, and Steven Harp. Course of action generation for cyber security using classical planning. In Susanne Biundo, Karen Myers, and Kanna Rajan, editors, *Proceedings of the 15th International Conference on Automated Planning and Scheduling (ICAPS-05)*, pages 12–21, Monterey, CA, USA, 2005. Morgan Kaufmann.

Blai Bonet and Héctor Geffner. Planning as heuristic search. *Artificial Intelligence*, 129(1–2):5–33, 2001.

Isabel Cenamor, Tomás de la Rosa, and Fernando Fernández. IBACOP and IBACOP2 planner. In *IPC 2014 planner abstracts*, pages 35–38, 2014.

References II

Andrew Coles, Maria Fox, Derek Long, and Amanda Smith. A hybrid relaxed planning graph'lp heuristic for numeric planning domains. In Jussi Rintanen, Bernhard Nebel, J. Christopher Beck, and Eric Hansen, editors, *Proceedings of the 18th International Conference on Automated Planning and Scheduling (ICAPS'08)*, pages 52–59. AAAI Press, 2008.

Amanda Jane Coles, Andrew Coles, Angel García Olaya, Sergio Jiménez, Carlos Linares López, Scott Sanner, and Sungwook Yoon. A survey of the seventh international planning competition. *The AI Magazine*, 33(1), 2012.

William Cushing, Subbarao Kambhampati, Mausam, and Daniel Weld. When is temporal planning *really* temporal? In Manuela Veloso, editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 1852–1859, Hyderabad, India, January 2007. Morgan Kaufmann.

Patrick Doherty and Jonas Kvarnström. TALplanner: A temporal logic based planner. *The AI Magazine*, 22(3):95–102, 2001.

Carmel Domshlak, Erez Karpas, and Shaul Markovitch. Online speedup learning for optimal planning. *Journal of Artificial Intelligence Research*, 44:709–755, 2012.

References III

Stefan Edelkamp. Promela planning. In Thomas Ball and Sriram Rajamani, editors, *Proceedings of the 10th International SPIN Workshop on Model Checking of Software (SPIN-03)*, pages 197–212, Portland, OR, May 2003. Springer-Verlag.

Stefan Edelkamp. Taming numbers and durations in the model checking integrated planning system. *Journal of Artificial Intelligence Research*, 20:195–238, 2003.

Stefan Edelkamp. On the compilation of plan constraints and preferences. In Derek Long and Stephen Smith, editors, *Proceedings of the 16th International Conference on Automated Planning and Scheduling (ICAPS'06)*, pages 374–377, Ambleside, UK, 2006. Morgan Kaufmann.

Roman Englert. Planning to optimize the umts call setup for the execution of mobile applications. *Applied Artificial Intelligence*, 19(2):99–117, 2005.

Maria Fox and Derek Long. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research*, 20:61–124, 2003.

Alfonso Gerevini, Alessandro Saetti, and Ivan Serina. Planning through stochastic local search and temporal action graphs. *Journal of Artificial Intelligence Research*, 20:239–290, 2003.

References IV

Alfonso Gerevini, Patrik Haslum, Derek Long, Alessandro Saetti, and Yannis Dimopoulos. Deterministic planning in the fifth international planning competition: PDDL3 and experimental evaluation of the planners. *Artificial Intelligence*, 173(5-6):619–668, 2009.

Alfonso Gerevini, Alessandro Saetti, and Mauro Vallati. An automatically configurable portfolio-based planner with macro-actions: PbP. In Alfonso Gerevini, Adele Howe, Amedeo Cesta, and Ioannis Refanidis, editors, *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS'09)*, pages 350–353. AAAI Press, 2009.

Malte Helmert and Hector Geffner. Unifying the causal graph and additive heuristics. In Jussi Rintanen, Bernhard Nebel, J. Christopher Beck, and Eric Hansen, editors, *Proceedings of the 18th International Conference on Automated Planning and Scheduling (ICAPS'08)*, pages 140–147. AAAI Press, 2008.

Malte Helmert and Hauke Lasinger. The Scanalyzer domain: Greenhouse logistics as a planning problem. In Ronen I. Brafman, Hector Geffner, Jörg Hoffmann, and Henry A. Kautz, editors, *Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS'10)*, pages 234–237. AAAI Press, 2010.

References V

Malte Helmert, Gabriele Röger, Jendrik Seipp, Erez Karpas, Jörg Hoffmann, Emil Keyder, Raz Nissim, Silvia Richter, and Matthias Westphal. Fast Downward Stone Soup. In *IPC 2011 planner abstracts*, pages 38–45, 2011.

Malte Helmert. A planning heuristic based on causal graph analysis. In Sven Koenig, Shlomo Zilberstein, and Jana Koehler, editors, *Proceedings of the 14th International Conference on Automated Planning and Scheduling (ICAPS'04)*, pages 161–170, Whistler, Canada, 2004. Morgan Kaufmann.

Malte Helmert. The Fast Downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006.

Malte Helmert. Concise finite-domain representations for PDDL planning tasks. *Artificial Intelligence*, 173:503–535, 2009.

Jörg Hoffmann and Ronen Brafman. Conformant planning via heuristic forward search: A new approach. *Artificial Intelligence*, 170(6–7):507–541, 2006.

Jörg Hoffmann and Stefan Edelkamp. The deterministic part of ipc-4: An overview. *Journal of Artificial Intelligence Research*, 24:519–579, 2005.

References VI

Jörg Hoffmann and Bernhard Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302, 2001.

Jörg Hoffmann, Stefan Edelkamp, Sylvie Thiébaux, Roman Englert, Frederico dos Santos Liporace, and Sebastian Trüg. Engineering benchmarks for planning: the domains used in the deterministic part of IPC-4. *Journal of Artificial Intelligence Research*, 26:453–541, 2006.

Jörg Hoffmann. Extending FF to numerical state variables. In F. Van Harmelen, editor, *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI'02)*, pages 571–575, Lyon, France, July 2002. Wiley.

Jörg Hoffmann. The Metric-FF planning system: Translating "ignoring delete lists" to numeric state variables. *Journal of Artificial Intelligence Research*, 20:291–341, 2003.

Henry A. Kautz and Bart Selman. Pushing the envelope: Planning, propositional logic, and stochastic search. In William J. Clancey and Daniel Weld, editors, *Proceedings of the 13th National Conference of the American Association for Artificial Intelligence (AAAI'96)*, pages 1194–1201, Portland, OR, July 1996. MIT Press.

References VII

Levente Kocsis and Csaba Szepesvári. Bandit based Monte-Carlo planning. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Proceedings of the 17th European Conference on Machine Learning (ECML 2006)*, volume 4212 of *Lecture Notes in Computer Science*, pages 282–293. Springer-Verlag, 2006.

Jana Koehler, Bernhard Nebel, Jörg Hoffmann, and Yannis Dimopoulos. Extending planning graphs to an ADL subset. In S. Steel and R. Alami, editors, *Proceedings of the 4th European Conference on Planning (ECP'97)*, pages 273–285. Springer-Verlag, 1997.

Ian Little and Sylvie Thiebaux. Probabilistic planning vs replanning. In *ICAPS Workshop on the International Planning Competition: Past, Present and Future*, 2007.

Derek Long and Maria Fox. Efficient implementation of the plan graph in stan. *Journal of Artificial Intelligence Research*, 10:87–115, 1999.

Derek Long and Maria Fox. The 3rd international planning competition: Results and analysis. *Journal of Artificial Intelligence Research*, 20:1–59, 2003.

References VIII

Derek Long, Henry Kautz, Bart Selman, Blai Bonet, Héctor Geffner, Jana Koehler, Michael Brenner, Jörg Hoffmann, Frank Rittinger, Corin R. Anderson, Daniel S. Weld, David E. Smith, and Maria Fox. The AIPS-98 planning competition. *The AI Magazine*, 21(2):13–33, 2000.

Ruy L. Milidiu, Frederico dos Santos Liporace, and Carlos J. de Lucena. Pipesworld: Planning pipeline transportation of petroleum derivatives. In Stefan Edelkamp and Jörg Hoffmann, editors, *Proceedings of Workshop on the Competition: Impact, Organization, Evaluation, Benchmarks, at ICAPS'03*, Trento, Italy, Jun 2003. AAAI Press.

Hootan Nakhost, Jörg Hoffmann, and Martin Müller. Resource-constrained planning: A Monte Carlo random walk approach. In Blai Bonet, Lee McCluskey, José Reinaldo Silva, and Brian Williams, editors, *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS'12)*, pages 181–189. AAAI Press, 2012.

Dana S. Nau, Tsz-Chiu Au, Okhtay Ilghami, Ugur Kuter, J. William Murdock, Dan Wu, and Fusun Yaman. Shop2: An htn planning system. *Journal of Artificial Intelligence Research*, 20:379–404, 2003.

References IX

Hector Palacios and Hector Geffner. From conformant into classical planning: Efficient translations that may be complete too. In Mark Boddy, Maria Fox, and Sylvie Thiebaux, editors, *Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS'07)*, Providence, Rhode Island, USA, 2007. Morgan Kaufmann.

Hector Palacios and Hector Geffner. Compiling uncertainty away in conformant planning problems with bounded width. *Journal of Artificial Intelligence Research*, 35:623–675, 2009.

Silvia Richter and Matthias Westphal. The LAMA planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research*, 39:127–177, 2010.

Jussi Rintanen, Keijo Heljanko, and Ilkka Niemelä. Planning as satisfiability: parallel plans and algorithms for plan search. *Artificial Intelligence*, 170(12-13):1031–1080, 2006.

Jussi Rintanen. Incorporation of temporal logic control into plan operators. In W. Horn, editor, *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI'00)*, pages 526–530, Berlin, Germany, August 2000. Wiley.

References X

Jussi Rintanen. Heuristics for planning with SAT. In *Proceeedings of the 16th International Conference on Principles and Practice of Constraint Programming*, pages 414–428, 2010.

Wheeler Ruml, Minh Binh Do, Rong Zhou, and Markus P. J. Fromherz. On-line planning and scheduling: An application to controlling modular printers. *Journal of Artificial Intelligence Research*, 40:415–468, 2011.

Florent Teichteil-Königsbuch, Ugur Kuter, and Guillaume Infantes. Incremental plan aggregation for generating policies in MDPs. In Wiebe van der Hoek, Gal A. Kaminka, Yves Lespérance, Michael Luck, and Sandip Sen, editors, *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'10)*, pages 1231–1238. IFAAMAS, 2010.

S. Thiébaux and M.-O. Cordier. Supply restoration in power distribution systems — a benchmark for planning under uncertainty. In A. Cesta and D. Borrajo, editors, *Proceedings of the 6th European Conference on Planning (ECP'01)*, pages 85–95. Springer-Verlag, 2001.

References XI

Sebastian Trüg, Jörg Hoffmann, and Bernhard Nebel. Applying automatic planning systems to airport ground-traffic control — a feasibility study. In S. Biundo, T. Frühwirth, and Günther Palm, editors, *KI-04: Advances in Artificial Intelligence*, pages 183–197, Ulm, Germany, 2004. Springer-Verlag.

Lin Xu, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Satzilla: Portfolio-based algorithm selection for sat. *Journal of Artificial Intelligence Research*, 32:565–606, 2008.

Sung Wook Yoon, Alan Fern, and Robert Givan. FF-Replan: a baseline for probabilistic planning. In Mark Boddy, Maria Fox, and Sylvie Thiebaux, editors, *Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS'07)*, pages 352–359, Providence, Rhode Island, USA, 2007. Morgan Kaufmann.

Håkan L. S. Younes, Michael L. Littman, David Weissman, and John Asmuth. The first probabilistic track of the international planning competition. *Journal of Artificial Intelligence Research*, 24:851–887, 2005.