Available online at www.sciencedirect.com
SCIENCE DIRECT®

Artificial Intelligence 169 (2005) 99–102

Artificial Intelligence

www.elsevier.com/locate/artint

Forthcoming Papers

B.W. Wah, Y. Chen, Constraint partitioning in penalty formulations for solving temporal planning problems

In this paper, we study the partitioning of constraints in temporal planning problems formulated as mixed-integer nonlinear programming (MINLP) problems. Constraint partitioning is attractive because it leads to much easier subproblems, where each is a significant relaxation of the original problem. Moreover, each subproblem is very similar to the original problem and can be solved by any existing solver with little or no modification. Constraint partitioning, however, introduces global constraints that may be violated when subproblems are evaluated independently. To reduce the overhead in resolving such global constraints, we develop in this paper new conditions and algorithms for limiting the search space to be backtracked in each subproblem. Using a penalty formulation of a MINLP where the constraint functions of the MINLP are transformed into non-negative functions, we present a necessary and sufficient extended saddle-point condition (ESPC) for constrained local minimization. When the penalties are larger than some thresholds, our theory shows a one-to-one correspondence between a constrained local minimum of the MINLP and an extended saddle point of the penalty function. Hence, one way to find a constrained local minimum is to increase gradually the penalties of those violated constraints and to look for a local minimum of the penalty function using any existing algorithm until a solution to the constrained model is found. Next, we extend the ESPC to constraint-partitioned MINLPs and propose a partition-and-resolve strategy for resolving violated global constraints across subproblems. Using the discrete-space ASPEN and the mixed-space MIPS planners to solve subproblems, we show significant improvements on some planning benchmarks, both in terms of the quality of the plans generated and the execution times to find them.

M. Fox, M. Ghallab, G. Infantes, D. Long, Robot introspection through learned hidden Markov models

In this paper we describe a machine learning approach for acquiring a model of a robot behaviour from raw sensor data. We are interested in automating the acquisition of behavioural models to provide a robot with an introspective capability. We assume that the behaviour of a robot in achieving a task can be modelled as a finite stochastic state transition system. Beginning with data recorded by a robot in the execution of a task, we use unsupervised learning techniques to estimate a hidden Markov model (HMM) that can be used both for predicting and explaining the behaviour of the

0004-3702/2005 Published by Elsevier B.V.
doi:10.1016/j.artint.2005.10.015

robot in subsequent executions of the task. We demonstrate that it is feasible to automate the entire process of learning a high quality HMM from the data recorded by the robot during execution of its task. The learned HMM can be used both for monitoring and controlling the behaviour of the robot. The ultimate purpose of our work is to learn models for the full set of tasks associated with a given problem domain, and to integrate these models with a generative task planner. We want to show that these models can be used successfully in controlling the execution of a plan. However, this paper does not develop the planning and control aspects of our work, focussing instead on the learning methodology and the evaluation of a learned model. The essential property of the models we seek to construct is that the most probable trajectory through a model, given the observations made by the robot, accurately diagnoses, or explains, the behaviour that the robot actually performed when making these observations. In the work reported here we consider a *navigation* task. We explain the learning process, the experimental setup and the structure of the resulting learned behavioural models. We then evaluate the extent to which explanations proposed by the learned models accord with a human observer's interpretation of the behaviour exhibited by the robot in its execution of the task.

P.M. Dung, R.A. Kowalski, F. Toni, Dialectic proof procedures for assumption-based, admissible argumentation

We present a family of dialectic proof procedures for the admissibility semantics of assumption-based argumentation. These proof procedures are defined for any conventional logic formulated as a collection of inference rules and show how any such logic can be extended to a dialectic argumentation system. The proof procedures find a set of assumptions, to defend a given belief, by starting from an initial set of assumptions that supports an argument for the belief and adding defending assumptions incrementally to counter-attack all attacks. The proof procedures share the same notion of winning strategy for a dispute and differ only in the search strategy they use for finding it. The novelty of our approach lies mainly in its use of backward reasoning to construct arguments and potential arguments, and the fact that the proponent and opponent can attack one another before an argument is completed. The definition of winning strategy can be implemented directly as a non-deterministic program, whose search strategy implements the search for defences.

S. Li, H. Wang, RCC8 binary constraint network can be consistently extended

The RCC8 constraint language developed by Randell et al. has been popularly adopted by the Qualitative Spatial Reasoning and GIS communities. The recent observation that RCC8 composition table describes only weak composition instead of composition raises questions about Renz and Nebel's maximality results about the computational complexity of reasoning with RCC8. This paper shows that any consistent RCC8 binary constraint network (RCC8 network for short) can be consistently extended. Given Θ, an RCC8 network, and z, a fresh variable, suppose $x\mathbf{T}y \in \Theta$ and $\mathbf{T}$ is contained in the weak composition of $\mathbf{R}$ and $\mathbf{S}$. This means that we can add two new constraints $x\mathbf{R}z$ and $z\mathbf{S}y$ to Θ without changing the consistency of the network. The result guarantees the applicability to RCC8 of one key technique, (Theorem 5) of [J. Renz, B. Nebel, On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the Region Connection Calculus. Artificial Intelligence 108 (1999) 69–123], which allows the transfer of tractability of a set of RCC8 relations to its closure under composition, intersection, and converse.

V. Vidal, H. Geffner, Branching and pruning: An optimal temporal POCL planner based on constraint programming

A key feature of modern optimal planners such as GRAPHPLAN and BLACKBOX is their ability to prune large parts of the search space. Previous Partial Order Causal Link (POCL) planners provide an alternative branching scheme but lacking comparable pruning mechanisms do not perform as well. In this paper, a domain-independent formulation of temporal planning based on Constraint Programming is introduced that successfully combines a POCL branching scheme with powerful and sound pruning rules. The key novelty in the formulation is the ability to reason about supports, precedences, and causal links involving actions that are not in the plan. Experiments over a wide range of benchmarks show that the resulting optimal temporal planner is much faster than current ones and is competitive with the best parallel planners in the special case in which actions have all the same duration.[1]

G. Lamperti, M. Zanella, Flexible diagnosis of discrete-event systems by similarity-based reasoning techniques

Diagnosis of discrete-event systems (DESs) may be improved by knowledge-compilation techniques, where a large amount of model-based reasoning is anticipated off-line, by simulating the behavior of the system and generating suitable data structures (compiled knowledge) embedding diagnostic information. This knowledge is exploited on-line, based on the observation of the system behavior, so as to generate the set of candidate diagnoses (problem solution). This paper makes a step forward: the solution of a diagnostic problem is supported by the solution of another problem, provided the two problems are somewhat similar. Reuse of model-based reasoning is thus achieved by exploiting the diagnostic knowledge yielded for solving previous problems. The technique still works when the available knowledge does not fit the extent of the system, but only a partition of it, that is, when solutions are available for subsystems only. In this case, the fragmented knowledge is exploited in a modular way, where redundant computation is avoided. Similarity-based diagnosis is meant for large-scale DESs, where the degree of similarity among subsystems is high and stringent time constraints on the diagnosis response is a first-class requirement.

R. Greiner, R. Hayward, M. Jankowska, M. Molloy, Finding optimal satisficing strategies for and-or trees

Many tasks require evaluating a specified Boolean expression φ over a set of probabilistic tests whose costs and success probabilities are each known. A *strategy* specifies when to perform which test, towards determining the overall outcome of φ. We are interested in finding the strategy with the minimum expected cost. As this task is typically NP-hard—for example, when tests can occur many times within φ, or when there are probabilistic correlations between the test outcomes—we consider those cases in which the tests are probabilistically independent and each appears only once in φ. In such cases, φ can be written as an *and-or tree*, where each internal node corresponds to

[1] This paper extends [V. Vidal, H. Geffner, Branching and pruning: An optimal temporal POCL planner based on constraint programming, in: Proceedings of AAAI-2004, San Jose, CA, 2004, pp. 570–577] by removing the canonicity restriction in the generation of plans. This is a restriction that forces every (ground) action in the domain to be done at most once in the plan. See the text for details.

either the "and" or "or" of its children, and each leaf node is a probabilistic test. In this paper we investigate "probabilistic and-or tree resolution" (PAOTR), namely the problem of finding optimal strategies for and-or trees. We first consider a depth-first approach: evaluate each penultimate rooted subtree in isolation, replace each such subtree with a single "mega-test", and recurse on the resulting reduced tree. We show that the strategies produced by this approach are optimal for and-or trees with depth at most two but can be arbitrarily sub-optimal for deeper trees. Each depth-first strategy can be described by giving the linear relative order in which tests are to be executed, with the understanding that any test whose outcome becomes irrelevant is skipped. The class of linear strategies is strictly larger than depth-first strategies. We show that even the best linear strategy can also be arbitrarily sub-optimal. We next prove that an optimal strategy honors a natural partial order among tests with a common parent node ("leaf-sibling tests"), and use this to produce a dynamic programming algorithm that finds the optimal strategy in time $O(d^2(r+1)^d)$, where r is the maximum number of leaf-siblings and d is the number of leaf-parents; hence, for trees with a bounded number of internal nodes, this run-time is polynomial in the tree size. We also present another special class of and-or trees for which this task takes polynomial time. We close by presenting a number of other plausible approaches to PAOTR, together with counterexamples to show their limitations.

J. Lee, F. Lin, Loop formulas for circumscription

Clark's completion is a simple nonmonotonic formalism and a special case of several nonmonotonic logics. Recently there has been work on extending completion with "loop formulas" so that general cases of nonmonotonic logics such as logic programs (under the answer set semantics) and McCain–Turner causal logic can be characterized by propositional logic in the form of "completion + loop formulas". In this paper, we show that the idea is applicable to McCarthy's circumscription in the propositional case, with Lifschitz's pointwise circumscription playing the role of completion. We also show how to embed propositional circumscription in logic programs and in causal logic, inspired by the uniform characterization of "completion + loop formulas".