Proceedings of the Ninth International
Symposium on Combinatorial Search (SoCS 2016)

Optimal Solitaire Game Solutions Using A* Search and Deadlock Analysis

Gerald Paul
Boston University
Boston, Massachusetts, USA
gerryp@bu.edu

Malte Helmert
University of Basel
Basel, Switzerland
malte.helmert@unibas.ch

Abstract

We propose an efficient method for determining optimal solutions to such skill-based solitaire card games as Freecell. We use A* search with an admissible heuristic function based on analyzing a directed graph whose cycles represent deadlock situations in the game state. To the best of our knowledge, ours is the first algorithm that efficiently determines optimal solutions for Freecell games. We believe that the underlying ideas should be applicable not only to games but also to other classical planning problems which manifest deadlocks.

Introduction

Games have always been a fertile ground for advances in computer science, operations research and AI. Solitaire card games, and Freecell in particular, have been the subject of study in both the academic and popular literature. Our work applies to *skill-based* solitaire games in which all cards are dealt face up. For these games, there is no element of chance involved after the initial deal, and hence they are *classical planning* problems (Ghallab, Nau, and Traverso 2004). We use Freecell because it is the most widely played and analyzed skill solitaire card game. It is NP-hard (Helmert 2003) and thus provides a demanding test for heuristic search approaches. While a number of computer solvers for Freecell are available, we know of no work which provides provably *optimal* solutions to solitaire games.

One of the fundamental properties of skill-based games like Freecell is that there are *deadlocks* where actions contributing towards the goal cyclically depend on each other. In order to resolve the deadlocks, actions that do not directly contribute to the goal are required. Deadlocks have long been known to make optimal planning hard (Gupta and Nau 1992). A key insight of this work is that very strong admissible heuristic functions for Freecell can be constructed by analyzing such deadlocks.

In the following, we describe our approach, present results of our solver implementation and discuss related work and future research directions. For space reasons, all these aspects are discussed very briefly, and few references to related work are provided. We refer to a longer version of this paper for details, including a description of the rules of Freecell (Paul and Helmert 2016).

Freecell Heuristics for A*

We propose optimally solving Freecell deals using the A* algorithm with an admissible heuristic. The simplest nontrivial heuristic we consider is the *card counting* heuristic, which uses the number of cards that still need to be moved to the foundation piles as its heuristic estimate.

Card counting is admissible but very optimistic, because some cards are usually blocked from movement to the foundations. The simplest example of this is a tableau pile where a card of a given suit and lower rank is buried underneath a card of the same suit and higher rank. Until the higher-rank card is moved to a temporary location, the lower-rank card cannot be moved to the foundations, and hence the higher-rank card must be moved at least twice. This is an example of a deadlock involving two cards.

More complex deadlocks involve more than two cards. In general, we define a directed graph with one vertex for each card that has not yet been moved to the foundations and a directed edge from card c to card c' whenever c must wait for c' to be moved before c can be moved to the foundations. There are two scenarios that cause such edges. Firstly, if c is located underneath c' in a tableau pile, we obtain a *blocking edge*, as c' must be moved out of the way to access c. Secondly, if c and c' belong to the same suit and c is of higher rank than c', we obtain a *foundations edge*, as c' must be moved to the foundations before c. For every cycle of edges in this directed graph, at least one card involved in the cycle must be moved twice.

Usually many such cycles exist, and because they can overlap, merely counting the number of cycles leads to an inadmissible heuristic. For any given set of cycles C, an admissible heuristic can be obtained by computing a *minimum hitting set* H for C, i.e., a set of cards of minimum cardinality that includes at least one card from every cycle. Adding $|H|$ to the card counting heuristic is admissible.

However, in general minimum hitting sets are expensive to compute. The computational effort can be reduced, at some loss in heuristic accuracy, by limiting attention to a restricted set of cycles. For $p \in \{0, \ldots, 4\}$, let h_p be the heuristic that only considers cycles involving at most p different suits in foundations edges. (Hence h_0 is the card counting heuristic.) Then h_p can be computed in a time that scales exponentially (only) in p, and p controls the trade-off between heuristic accuracy and evaluation speed.

game	$h_2(I)$	$h^*(I)$	time	states
#1	73	82	30.8s	567699
#2	68	73	1.9s	101186
#3	70	70	0.6s	20499
#4	72	79	31.0s	1220026
#5	78	85	122.0s	3687136
#6	73	75	1.1s	31912
#7	72	76	1.7s	74369
#8	70	74	13.2s	367784
#9	77	81	2.0s	77990
#10	73	80	7.7s	315643

Table 1: Results for the Freecell games 1–10. Columns, in order: game number, h_2 value for initial state, optimal solution length, runtime, evaluated states.

Experimental Evaluation

We evaluated the deadlock-based heuristics for Freecell within an A* implementation with standard efficiency enhancements such as bucket-based priority queues and duplicate detection via Zobrist hashing.

Preliminary investigations showed that deadlock reasoning is clearly necessary: h_0 is too weak to solve Freecell deals. The best overall performance was obtained with h_2, and hence this heuristic was used for the main experiment.

The main experiment was run on an Intel core i3 4160 processor running at 3.60 GHz with 8 GB of memory. Our test cases were games 1–5000 of Microsoft Freecell. Optimal solutions were found for all test cases. Runtime varied between 0.4 and 6579 seconds, with an average of 39.9 seconds. Optimal solution lengths varied in the range 64–93, with an average of 77. Detailed results for games 1–10 are shown in Table 1.

Related Work

The use of deadlocks as a central concept in our heuristic is reminiscent of the classic blocks world domain, where computing optimal solutions is NP-hard due to the existence of deadlocks in ordering dependencies of essentially the same kind as those which arise in Freecell (Gupta and Nau 1992; Slaney and Thiébaux 2001). Indeed, the Freecell heuristics described in this paper can be understood as a two-stage relaxation:

- Firstly, relax the Freecell game into a blocks world task.
- Secondly, compute an admissible heuristic for this blocks world task.

Indeed, the case where we consider *all* deadlocks (i.e., the h_4 heuristic) is equivalent to mapping the Freecell task to a blocks world task and solving the resulting blocks world task *optimally*, and this is in turn equivalent to solving a relaxation of the original Freecell task with unlimited free cells.

In other words, the h_4 heuristic completely solves the "move ordering" aspect of the problem while ignoring the "space contention" aspect. We discuss these relationships in more detail in an extended version of this paper (Paul and Helmert 2016).

Future Work

Looking beyond solitaire games and blocks world, are there wider implications of these results? We believe that this is the case: that deadlocks are a phenomenon that occurs in a much wider range of domains than Freecell games or blocks world tasks, and that heuristic functions based on covering deadlocks are a promising direction for a wide range of search problems.

For example, deadlocks of essentially the same form as in the blocks world domain are the major source of hardness in a number of standard planning benchmarks, including Logistics, Miconic-STRIPS and Miconic-SimpleADL (Helmert 2001). Many other planning domains with a "transportation" component share this problem aspect, though often mixed with other aspects. Deadlock covering problems also occur at the computational core of many optimization problems outside of planning, such as many of the *implicit hitting set* problems identified by Chandrasekaran et al. (2011). Slaney (2014) describes deeper connections between blocks world, implicit hitting sets, and combinatorial optimization in general.

Finally, a similar form of deadlocks (actions cyclically supporting each other's preconditions without being ultimately supported by effect/precondition links from the current state) is the major source of inaccuracy in *flow heuristics* that have recently attracted much attention (van den Briel et al. 2007; Bonet and van den Briel 2014; Pommerening et al. 2014). A better understanding of the general role of dependency deadlocks could go a long way towards overcoming the limitations of these heuristics.

References

Bonet, B., and van den Briel, M. 2014. Flow-based heuristics for optimal planning: Landmarks and merges. In *Proc. ICAPS 2014*, 47–55.

Chandrasekaran, K.; Karp, R.; Moreno-Centeno, E.; and Vempala, S. 2011. Algorithms for implicit hitting set problems. In *Proc. SODA 2011*, 614–629.

Ghallab, M.; Nau, D.; and Traverso, P. 2004. *Automated Planning: Theory and Practice*. Morgan Kaufmann.

Gupta, N., and Nau, D. S. 1992. On the complexity of blocks-world planning. *AIJ* 56(2–3):223–254.

Helmert, M. 2001. On the complexity of planning in transportation domains. In *Proc. ECP 2001*, 120–126.

Helmert, M. 2003. Complexity results for standard benchmark domains in planning. *AIJ* 143(2):219–262.

Paul, G., and Helmert, M. 2016. Optimal solitaire game solutions using A* search and deadlock analysis. In *ICAPS 2016 Workshop on Heuristics and Search for Domain-independent Planning*.

Pommerening, F.; Röger, G.; Helmert, M.; and Bonet, B. 2014. LP-based heuristics for cost-optimal planning. In *Proc. ICAPS 2014*, 226–234.

Slaney, J., and Thiébaux, S. 2001. Blocks World revisited. *AIJ* 125(1–2):119–153.

Slaney, J. 2014. Set-theoretic duality: A fundamental feature of combinatorial optimisation. In *Proc. ECAI 2014*, 843–848.

van den Briel, M.; Benton, J.; Kambhampati, S.; and Vossen, T. 2007. An LP-based heuristic for optimal planning. In *Proc. CP 2007*, 651–665.