A SAT Encoding for the AtMostSeqCard Constraint

Valentin Mayer-Eichberger

NICTA
University of New South Wales
valentin.mayer-eichberger@nicta.com.au

1 Introduction

Give description of the car sequencing problem and the straight forward encoding in IP/CNF.

The naive CNF and IP encoding of the car sequencing benchmark is far from optimal. We can do better.

2 Motivation

We are seeking an encoding that enforces GAC on the recently proposed AtMostSeqCard constraint. This constraint is not as expressive as the Sequence constraint but is more suited for some benchmark problems and has a linear filtering algorithm. Here we will show that there is a compact CNF encodings that shows good results in the benchmark set of the CSPLIB. Furthermore we will try to improve the bounds on the set of hard instance.

3 Encoding of one AtMostSeqCard

3.1 Encoding of counters

Fig. 1. A simple counter for option or class i that counts to two over a sequence of 10 sorted elements. The variables for U are set to false and the variables L are set to true. The rest proper variables of the problem $y_{i,j,k}$.

k/j	0	1	2	3	4	5	6	7	8	9	10
3			U	U	U	U	U	U	U	U	U
2		U	?	?	?	?	?	?	?	?	L
1	U	?	?	?	?	?	?	?	?	L	
0	L	L	L	L	L	L	L	L	L		

3.2 Extending to AtMostSeqCard

4 Encoding of Carsequencing

We need to relate the cars and options as in the problem specification. This can be done in two ways. First the straight forward way.

4.1 Relating Cars and Option directly

Let $c \in C$ be the index identifying a class and $o \in O$ be the index for options. The problem instance gives us a mapping $m : C \to 2^O$, relating to each class a set of options.

$$\bigwedge_{p \in P} \bigwedge_{\substack{c \in C \\ o \in m(c)}} \neg x_{c,p} \vee x_{o,p} \tag{1}$$

and the reverse

$$\bigwedge_{p \in P} (\neg x_{o,p} \vee \bigvee_{\substack{c \in C \\ o \in m(c)}} x_{c,p}) \tag{2}$$

Notive in an ASP encoding this would be modelled by one rule and the completion semantics is equivalent to this encoding.

4.2 The Purist's Way

Here we will show that there is an encoding of the car sequencing problem that does not use at all the variables $x_{i,j}$. The encoding builds entirely on the auxiliary variables and can consistently identify all solutions to this problem. This is rather surprising. Here the idea:

5 Evaluation

Best of results that can be robustly (standard heuristics) archived by current sat solvers. I compared newest version of minisat, lingeling, cryptominisat, glucose and clasp and they all consistently find solutions within 1h runtime.

Table 1.

	set1	**set2**	**set3**	**set4**
sat	70	4	0	7
unsat	0	0	4	13
unknown	0	0	1	10

This is by far better than most papers evaluating the car sequencing problem on some specialized algorithm (e.g. branch and bound) or special constraint (CP) or optimization (IP).

For the set 4 a more detailed view is interesting as the benchmark targets the optimization version of the car sequencing problem.

Table 2. Solutions to the proposed hard benchmark on the 2004 paper (IP) and solutions on the decision version on the SAT encoding (SAT).

Instance	**IP**	**SAT**
pb-200-01	0	SAT
pb-300-01	0	SAT
pb-400-01	1	UNKNOWN
pb-200-02	2	UNKNOWN
pb-300-02	12	UNKNOWN
pb-400-02	16	UNKNOWN
pb-200-03	4	UNSAT
pb-300-03	13	UNSAT
pb-400-03	9	UNSAT
pb-200-04	7	UNSAT
pb-300-04	7	UNSAT
pb-400-04	19	UNSAT
pb-200-05	6	UNSAT
pb-300-05	29	UNSAT
pb-400-05	0	SAT
pb-200-06	6	UNKNOWN
pb-300-06	2	UNKNOWN
pb-400-06	0	SAT
pb-200-07	0	SAT
pb-300-07	0	SAT
pb-400-07	4	UNKNOWN
pb-200-08	8	UNKNOWN
pb-300-08	8	UNSAT
pb-400-08	4	UNKNOWN
pb-200-09	10	UNSAT
pb-300-09	7	UNKNOWN
pb-400-09	5	UNSAT
pb-200-10	19	UNSAT
pb-300-10	21	UNSAT
pb-400-10	0	SAT

We see that already just treating the simple decision problem we can improve the bounds given in the 2004 paper:

6 Extensions

- Optimizations: there are two definitions of the cost function for the car sequencing problem. First is to allow arbirary cars without any options and minimize the number of cars with options. And second is to minimize the number of windows that exceed the capacity constraint on their options. It would be interesting to compare both definition and to evaluate against published results in the literature. There are still gaps between known upper and lower bounds.
- There is a natural extension of the AtMostSeqCard constraint that to a cyclic version and in the same and natural way we can extend the encoding given above. It would be interesting to find good benchmarks.
- The Sequence constraint consists of a sequence of among constraints and we should compare this encoding to the known CNF encodings and filtering algorithmsin the literature.