Performance of DLP on Random Modal Formulae

Ian Horrocks
University of Manchester
Manchester, England
horrocks@cs.man.ac.uk

Peter F. Patel-Schneider
Bell Labs Research
Murray Hill, New Jersey
pfps@research.bell-labs.com

A system that correctly reasons with information in an expressive description logic includes a decision procedure for an expressive propositional modal logic. This decision procedure must be heavily optimised if it is to be able to usefully reason with knowledge bases of any complexity. Description logic systems that incorporate a heavily-optimised propositional modal logic decision procedure include FACT [5], DLP [7], and HAM-ALC [4].

Therefore, a system that efficiently reasons with information in an expressive description logic can be used as a fast decision procedure for an expressive propositional modal logic. The current status is that not only are the fastest reasoners for expressive description logics fast reasoners for propositional modal logics, but for many classes of formulae they are the *fastest* such reasoners.

We have performed numerous experiments with DLP, showing that it is competitive with other reasoners for propositional modal logics, including comparisons presented at recent Tableaux conferences where DLP was the fastest system [1; 8]. In more recent tests we have compared DLP with KSAT [3], TA [6], and KSATC [2] on various collections of random formulae.

The two fastest of these systems by a considerable amount are DLP and KSATC. DLP is an experimental description logic system available from Bell Labs at `http://www.bell-labs.com/user/pfps/dlp`. It implements a very expressive description logic, including full regular expressions on roles. KSATC is a reasoner for $\mathbf{K}_{(\mathbf{m})}$, built on a fast Davis-Putnam-Logemann-Loveland decision procedure for propositional logic. KSATC is available at `ftp://ftp.mrg.dist.unige.it/pub/mrg-systems/KR98-sources/KSat-source/KSatC`.

Tests illustrating the differences between DLP and KSATC are reported in Figures 1, 2, and 3. The figures give results for randomly-generated 3CNF formulae in the modal logic $\mathbf{K}$ using a formula generator that we have devised, which is almost identical to the generator of Giunchiglia *et al* [2]. These two recent generators eliminate the problems with repeated sub-formulae that caused problems with earlier generators.

In a propositional setting, repeated subformulae reduce the effective size of the generated formulae, but in a modal setting such repetitions can make the entire generated formula inconsistent. Such inconsistencies can easily be detected by modern provers, and their presence can drastically change the run times for tests.

The tests were chosen because they demonstrate the relative changes in behaviour between DLP and KSATC. The first two tests are similar to tests performed by the authors of KSATC [2] and TA [6]. However, the formulae generated in these two tests are not at all close to the kinds of formulae expected to occur when classifying description logic knowledge bases. The third test explores a very different kind of formula—one that may be close to the kind of formulae that classifying description logic knowledge bases.

Two kinds of graphs are used to illustrate the results of the tests The first kind of graph shows run times for various numbers of propositional variables N. The lines show either the median, the 90th percentile, or the worst case run time on a log scale for the various ratio of clauses to number of propositional variables (L/N). The other kind of graph shows the fifth through tenth deciles of run time for a particular test and a fixed number of propositional variables N, as a three dimensional plot. Overlaid on the three-dimensional plots are dotted surfaces that show the ratio of satisfiable to total solvable formulae, as determined by DLP or KSATC, provided that the program could solve at least half of the formulae at that point.

Figure 1 gives the results for formulae with a maximum modal depth (role restriction depth) of 2. For a given number of propositional variables (N), ranging from 3 to 9, formulae were generated with a number of clauses ranging from the number of propositional variables up to 150 times this number (shown as L/N in the results). In each clause there was a 50% chance that an atom was a propositional variable or modal formulae at depths 0 and 1.

For large values of L/N in these tests KSATC often takes very little time (less than 0.01 seconds) whereas DLP takes considerably longer (about 0.1 seconds). These formulae are unsatisfiable even ignoring any modal subformulae. Both

DLP and KSAT can easily determine this sort of unsatisfiability. Because the DPLL algorithm in KSAT can perform single passes over a large set of clauses in a very short time, KSAT can process the trivially unsatisfiable formulae very quickly, but DLP's data structures and low-level algorithms are not heavily optimised so DLP takes a total time for such formulae of about 0.1 seconds.

For data points in the L/N range from about 20–50, many or most of the formulae are satisfiable. Here DLP is uniformly faster than KSATC. Much of this difference is due to KSATC investigating modal successors (role restrictions) at every choice point. Modal successors do not produce much unsatisfiability in these tests, so the investigation of modal successors does not cut off search.

Even outside the above ranges (of L/N values), DLP performs better than KSATC for *satisfiable* formulae. Although there may be some benefit in examining modal successors early for these formulae, this benefit would occur only when a large number of necessary modal formulae have been assigned true. Because of the repeated work performed by KSATC on such successors, it does not appear that the benefits are realized.

Figure 2 gives the results for formulae with a maximum modal depth (role restriction depth) of 1. Here all atoms at depth 0 are modal formulae. KSATC performs much better than DLP in this test. Because KSATC aggressively investigates the modal successors, it can eliminate search at the top level. DLP has to examine many modal successors, as it will generate many top-level assignments that contain nearly the same modal successors. Its caching of modal results does not help very much as there are so many different modal successors that can be generated.

In our third test we approximated formulae with unbounded modal depth by setting the maximum modal depth to 100. To generate formulae small enough to handle we had to increase the probability that an atom was a propositional variable to 0.7. The results of this testing are given in Figure 3.

It is obvious that these formulae are not nearly as difficult as some of the earlier formulae. DLP found all the formulae extremely easy—the processing required for the formulae that required the most search (for L/N from around 30 to 50) was less than the processing required for the largest formulae, which are trivially unsatisfiable and require almost no search.

KSATC found most of the formulae easy, although there is a small harder section for L/N around 10–30, resulting in behaviour qualitatively different from that of DLP. In this section there were even a few formulae that KSATC found very hard. These harder formulae are those where KSATC is investigating many modal successors multiple times. On the other hand, DLP found all these formula very easy, with the hardest of the hardest taking only around 0.1 seconds, except for the very large formulae where DLP's processing overhead pushed the processing time up to 0.5 seconds.

If the size of the formulae are taken into consideration, these formulae are much easier than the previous ones, even though KSATC finds some of them hard.

We have performed other tests using the same generator showing some areas where DLP dominates, some where KSATC dominates, and some where they perform equally well.

Even though we have performed many of these tests using randomly-generated formulae, we are not happy with the fact that we are restricted to such tests. We would much prefer to test the performance of DLP on real knowledge bases that use the more-powerful features of DLP's description logic. However, such knowledge bases do not yet exist. We hope that the performance of DLP, and of other expressive description logic systems, will encourage people to develop knowledge bases using these features. We are highly motivated to work with developers of such knowledge bases.

References

[1] P. Balsiger and A. Heuerding. Comparison of theorem provers for modal logics — introduction and summary. In H. de Swart, editor, *Automated Reasoning with Analytic Tableaux and Related Methods: International Conference Tableaux'98*, number 1397 in Lecture Notes in Artificial Intelligence, pages 25–26. Springer-Verlag, May 1998.

[2] E. Giunchiglia, F. Giunchiglia, R. Sebastiani, and A. Tacchella. More evaluation of decision procedures for modal logics. In A. G. Cohn, L. Schubert, and S. C. Shapiro, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference (KR'98)*, pages 626–635. Morgan Kaufmann Publishers, San Francisco, CA, June 1998.

[3] F. Giunchiglia and R. Sebastiani. A SAT-based decision procedure for $\mathcal{ALC}$. In L. C. Aiello, J. Doyle, and S. C. Shapiro, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference (KR'96)*, pages 304–314. Morgan Kaufmann Publishers, San Francisco, CA, November 1996.

[4] V. Haarslev, R. Möller, and A.-Y. Turhan. Implementing an $\mathcal{ALCRP}(\mathcal{D})$ abox reasoner – progress report. In E. Franconi, G. De Giacomo, R. M. MacGregor, W. Nutt, C. A. Welty, and F. Sebastiani, editors, *Collected Papers from the International Description Logics Workshop (DL'98)*, pages 82–86, 1998.

[5] I. Horrocks. *Optimising Tableaux Decision Procedures for Description Logics*. PhD thesis, University of Manchester, 1997.

[6] U. Hustadt and R. A. Schmidt. On evaluating decision procedures for modal logic. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, volume 1, pages 202–207, 1997.

[7] Peter F. Patel-Schneider. DLP system description. In E. Franconi, G. De Giacomo, R. M. MacGregor, W. Nutt, C. A. Welty, and F. Sebastiani, editors, *Collected Papers from the International Description Logics Workshop (DL'98)*, pages 87–89, 1998.

[8] *Automated Reasoning with Analytic Tableaux and Related Methods: International Conference Tableaux'99*, Lecture Notes in Artificial Intelligence. Springer-Verlag, June 1999.

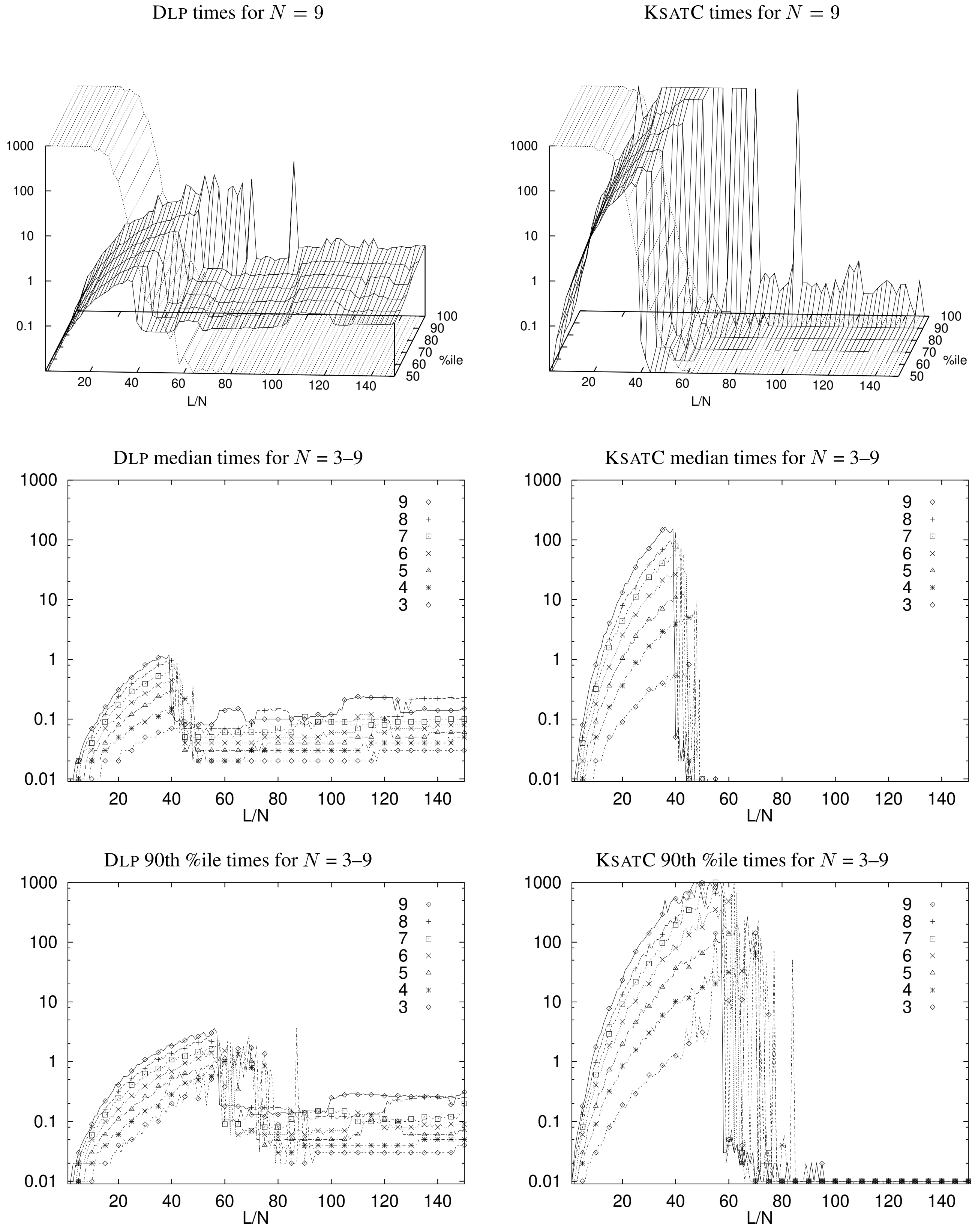

Figure 1: Results for test 1 (maximum modal depth 2)

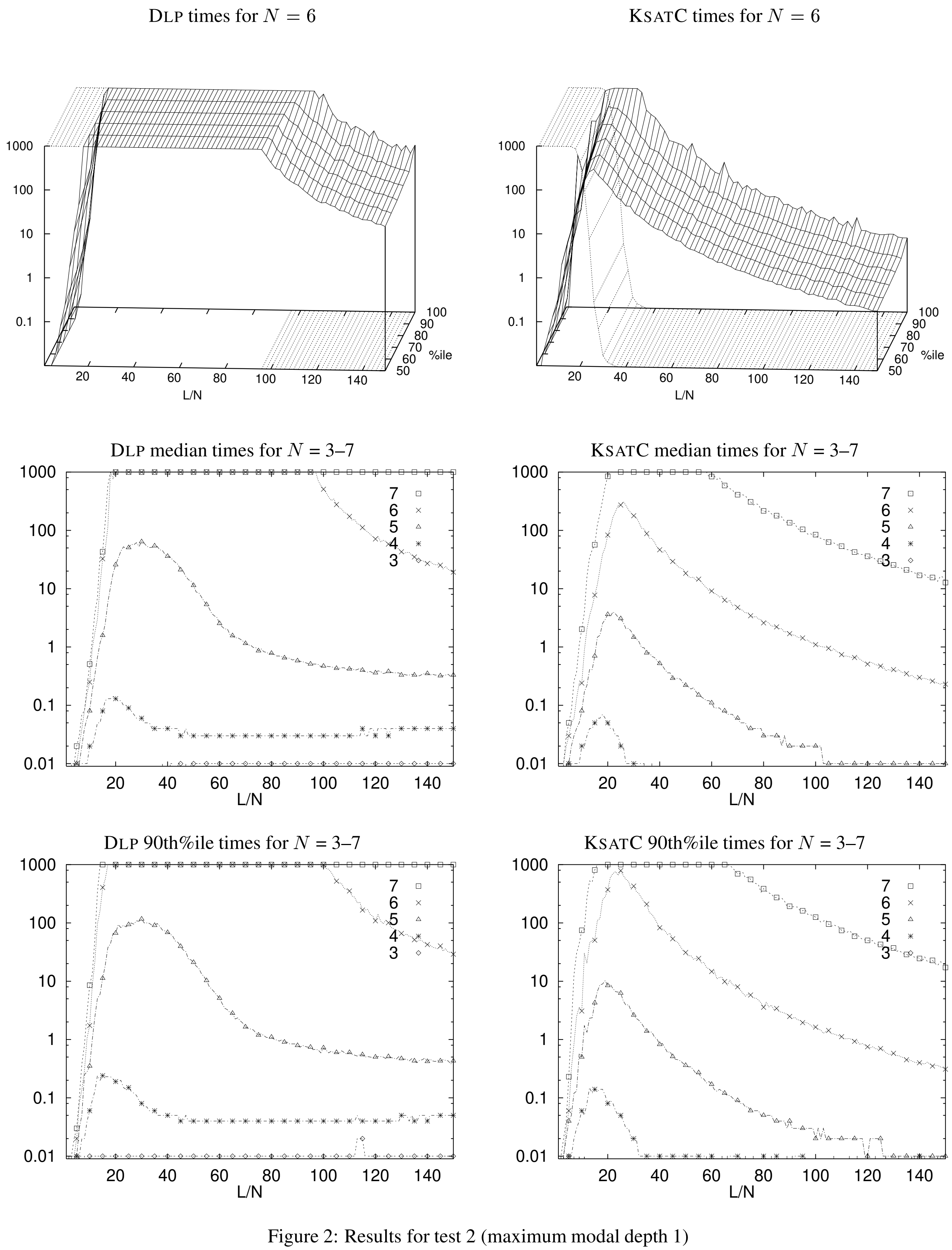

Figure 2: Results for test 2 (maximum modal depth 1)

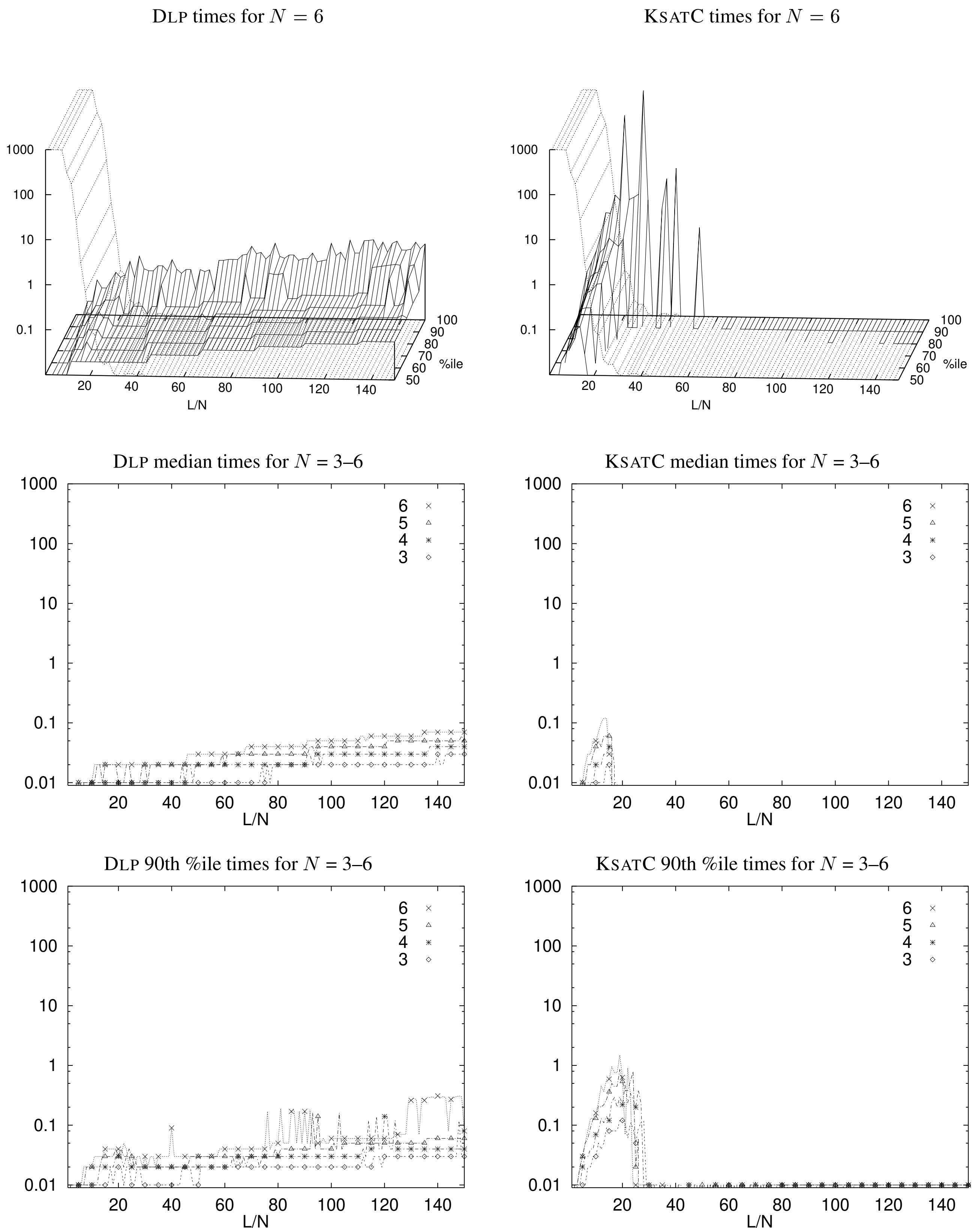

Figure 3: Results for test 3 (maximum modal depth 100)