HyperBench: A Benchmark and Tool for Hypergraphs and Empirical Findings*

Wolfgang Fischl[1], Georg Gottlob[1,2], Davide M. Longo[1], and Reinhard Pichler[1]

[1] TU Wien, *firstname.surname*@tuwien.ac.at
[2] University of Oxford, georg.gottlob@cs.ox.ac.uk

CEUR-WS.org/Vol-2369/short02.pdf

1 Introduction

Answering Conjunctive Queries (CQs) and solving Constraint Satisfaction Problems (CSPs) are classical NP-complete problems of high relevance in Computer Science. Consequently, there has been an intensive search for tractable fragments of these problems over the past decades. We are mainly interested here in tractable fragments defined via decomposition of the underlying hypergraph structure of a given CQ or CSP. The most important decomposition methods are hypertree decompositions (HDs), generalized hypertree decompositions (GHDs), and fractional hypertree decompositions (FHDs) alongside the corresponding width notions hypertree width (hw), generalized hypertree width (ghw), and fractional hypertree width (fhw), respectively.

It has been shown that CQ answering and CSP solving are tractable on every class of CQs/CSPs, if the underlying hypergraphs have bounded $hw(H)$, $ghw(H)$, or $fhw(H)$. Since $fhw(H) \leq ghw(H) \leq hw(H)$ holds for every hypergraph H, bounded fhw defines the biggest tractable class of CQ answering and CSP solving. On the other hand, only for hw, it is feasible in polynomial time to recognize if a given hypergraph has width $\leq k$ for fixed k. In contrast, for fhw and ghw, the problem of recognizing low width is NP-complete even for $k = 2$ [7].

In [7], the following properties of the underlying hypergraphs have been identified to ensure tractable computation of GHDs and FHDs of a given width (if they exist) or at least to allow for a good approximation thereof.

Definition 1. *The* intersection width $iwidth(H)$ *of a hypergraph* H *is the maximum cardinality of the intersection* $e_1 \cap e_2$ *of any two distinct edges* e_1, e_2 *of* H*. For positive integer* c*, the* c-multi-intersection width c-$miwidth(H)$ *of a hypergraph* H *is the maximum cardinality of any intersection* $e_1 \cap \cdots \cap e_c$ *of* c *distinct edges* $e_1, \ldots, e_c$ *of* H*. The* degree $\deg(H)$ *of a hypergraph* H *is the maximum number of edges a vertex of* H *occurs in, i.e.,* $\max_{v \in V(H)} |\{e \in E(H) \mid v \in e\}|$.

We say that a class $\mathcal{C}$ *of hypergraphs has the* bounded intersection property (BIP)*, the* bounded multi-intersection property (BMIP)*, or the* bounded degree property (BDP) *if there exist constants* i*,* c*, and* d*, such that every hypergraph in* $\mathcal{C}$ *satisfies* $iwidth(H) \leq i$*,* c-$miwidth(H) \leq i$*, or* $\deg(H) \leq d$*, respectively.*

Indeed, it has been shown in [7] that, if it exists, a GHD of low width can be computed in PTIME for hypergraphs enjoying the BDP, BIP, or BMIP. For

* This is an extended abstract of [6].

FHDs, an exact PTIME algorithm has been presented in case of the BDP; for the BIP, a polynomial time approximation scheme (PTAS) exists.

Despite the appealing theoretical results, little is known in practice about these properties and their interplay with the various notions of width. The goal of this work is to remedy this deficit. More concretely, we investigate questions such as the following: Do the hypergraphs underlying CQs and CSPs in practice, indeed have low degree and (multi-)intersection width? Are these properties non-trivial in the sense that, e.g., low intersection width does not immediately lead to low *hw*, *ghw*, and *fhw*? We also want to get a better understanding of the relationship between *ghw* and *hw*. In general, only the inequality $hw(H) \leq 3 \cdot ghw(H) + 1$ is known. But do these two notions of width indeed differ by factor 3 in practice? And do theoretical tractability results (such as tractable GHD computation in case of the BIP) indeed lead to practically feasible computation?

In order to provide answers to these questions, we have collected a vast amount of CQs and CSPs from concrete applications as well as randomly generated ones, and translated them into a uniform hypergraph format. We have successively performed a series of experiments on these hypergraphs – determining the *hw* and *ghw*, the degree and (multi-)intersection width as well as further metrics relating to the size such as number of vertices, number of edges, and maximum size of edges. All the hypergraphs thus collected together with the results of our experiments are publicly available at http://hyperbench.dbai.tuwien.ac.at/. Below, we give a summary of these results. Moreover, we note that this benchmark has already been profitably applied in the experimental evaluation of decomposition algorithms by other authors [5].

2 Basic Definitions

We assume the reader to be familiar with basic concepts such as CQs and CSPs. The *hypergraph corresponding to a CQ* ϕ is defined as $H = (V(H), E(H))$, where the set of vertices is $V(H) = \mathit{variables}(\phi)$ and the set of edges is $E(H) = \{e \mid \exists A \in \mathit{atoms}(\phi) : e = \mathit{variables}(A)\}$. Due to lack of space, we also have to assume familiarity with the various notions of decompositions and width.

We have already introduced the crucial properties BDP, BIP, and BMIP. By slight abuse of notation, we shall say in the sequel that a hypergraph H has BDP $= d$, BIP $= i$, or c-BMIP $= i$, if H satisfies the conditions $\mathit{iwidth}(H) = i$, c-$\mathit{miwidth}(H) = i$, or $\deg(H) = d$, respectively.

3 Results

We have collected 3070 CQs and CSPs from different sources and converted them into hypergraphs. Our collection of CQs comprises 535 queries used in practical applications and 500 randomly generated ones using the tool from [11]. The non-random CQs stem from various sources. Queries in [4] come from a huge SPARQL repository comprising over 26 million CQs, from which we have included only the hypergraphs with $hw \geq 2$. Queries in [9] come from a big collection of SQL queries from which we have extracted over 15,000 CQs (in particular, no nested SELECTs). Again, we have only included hypergraphs with $hw \geq 2$ into our benchmark. The remaining non-random CQs come from different benchmarks such as the Join Order Benchmark (JOB) [10] and TPC-H [12]. Our collection of 2035 CSPs is composed of 1953 instances from [2] and 82 instances used in previous analyses [3,8]. The instances from [2] are divided in two classes: 1090 of them come from applications and the remaining 863 are random instances.

Table 1. Percentage of instances having BDP, BIP, 3-BMIP, 4-BMIP ≤ 5.

	BDP (%)	BIP (%)	3-BMIP (%)	4-BMIP (%)
Application-CQs	81.68	100	100	100
Application-CSPs	53.67	99.91	100	100
Random	10.12	76.82	90.17	93.62

In our first experiment, we computed *BDP, BIP, 3-BMIP, 4-BMIP* for our collection of hypergraphs. The results are presented in Table 1. We group our instances in three classes: application-CQs, application-CSPs, and random instances (both CQs and CSPs). In the first place, we are interested in the percentage of hypergraphs whose values of BDP, BIP, 3-BMIP, and 4-BMIP are small. It turned out that all application-CQs have BIP ≤ 5 and yet smaller 3-BMIP and 4-BMIP. The BDP tends to be bigger, but there are still 81.68% of the application-CQs with BDP ≤ 5. As for the application-CSPs, the BIP and BMIP are also small in general, namely 99.91%, 100% and 100% have BIP ≤ 5, 3-BMIP ≤ 5, and 4-BMIP ≤ 5, respectively. Interestingly, the percentage of application-CSPs with BDP ≤ 5 is rather low (53.67%) compared with application-CQs. The random instances behave differently. Indeed, we have measured 76.82%, 90.17% and 93.62% of the random instances (with very similar behaviour of random CQs and random CSPs) to have small BIP, 3-BMIP, and 4-BMIP, respectively. The percentage of instances with BDP ≤ 5 even falls more dramatically to 10.12% for random instances. To conclude, BIP and BMIP indeed tend to be (very) small for both CQs and CSPs taken from applications and they are still reasonably small for random instances.

As a next step, we have systematically applied the computation of hw [8] to our benchmark. The aim of this experiment was to determine the hw or at least an upper bound thereof for each hypergraph. We have organized the computation in different rounds, each of which has a different value of k, which is initialized with $k = 1$. In each round, we check if $hw(H) \leq k$ holds and, if so, compute a concrete HD with this width. If the program ends with a yes-answer, we have an upper bound on hw. In case of a no-answer, we have a lower bound. No bound is obtained in case of a timeout (which we set at 3,600 seconds). For all the instances with no upper bound (i.e., either no-answer or timeout) for a value of k, we continue with $k \coloneqq k + 1$ in the next round. We were able to determine that for all application-CQs $hw \leq 3$ holds and to compute concrete HDs of this width. For 694 of all 1172 application-CSPs (59.22%) we have verified $hw \leq 5$. In total, considering also random instances, 1849 (60.23%) out of 3070 instances have $hw \leq 5$. For 1453 of them, we determined exact hw, for the others we only have an upper bound and the actual value of hw could be even less. We conclude that for the vast majority of CQs and CSPs (in particular those from applications), *hw* is small enough to allow for efficient CQ answering or CSP solving, respectively.

We have also analysed the correlation between all the hypergraph parameters studied here. BIP and BMIP are obviously highly correlated. More interestingly, we observe a high correlation between any two of the number of vertices, the arity (= maximum edge size), and *hw*. It is worth underlining that BDP, BIP, 3-BMIP, 4-BMIP have low correlation with hypertree width. That is, low values of these parameters are favourable for GHD computation [7] but they do not imply that also the *hw* and (as we will see below) the *ghw* are particularly small.

Finally, we have implemented several algorithms for GHD-computation, which exploit low BIP. For all hypergraphs with $hw \leq k$ and $k \in \{3, 4, 5, 6\}$, we checked whether $ghw \leq k - 1$ holds. To this end, we ran our algorithms with a timeout of 3,600 seconds. If the timeout does not occur, we say that the instance is "solved". We found out that in 98% of the solved cases and 57% of all instances with $hw \leq 6$, *hw* and *ghw* have identical values. Actually, we expect that the percentage in case of the solved case is more significant because the GHD computation algorithms usually take longer in case of a no-answers, i.e., we conjecture that most of the unsolved instances also have identical values of *hw* and *ghw*.

4 Conclusion

In this work, we have extensively experimented with a big collection of hypergraphs (from both CQs and CSPs). We have thus made several interesting observations, which have been summarized above. For instance, the discrepancy between *hw* and *ghw* seems to be much smaller in practice than the theoretical factor 3. Moreover, it has turned out that hypergraph parameters such as BIP, on the one hand, tend to be very small in practice and, on the other hand, low BIP is indeed very helpful for computing concrete decompositions – especially GHDs. This leads us to several directions for future work. Further improvements of our GHD algorithms and implementations are required to increase the number of "solved" instances. The development of algorithms exploiting low 3-BMIP or 4-BMIP seems to be a natural next step, since the latter parameters tend to be yet smaller than BIP. On the more theoretical side, it would be very interesting to settle the open question if bounded BIP also ensures tractable FHD-computation: so far, we only know that BIP allows for a PTAS for the *fhw*.

Acknowledgement

This work was supported by the Austrian Science Fund (FWF) project P30930-N35. Davide Mario Longo's work was supported by the Austrian Science Fund (FWF) project W1255-N23.

References

1. Arocena, P.C., Glavic, B., Ciucanu, R., Miller, R.J.: The ibench integration metadata generator. Proc. VLDB Endow. **9**(3), 108–119 (2015)

2. Audemard, G., Boussemart, F., Lecoutre, C., Piette, C.: XCSP3: an XML-based format designed to represent combinatorial constrained problems. http://xcsp.org
3. Berg, J., Lodha, N., Järvisalo, M., Szeider, S.: Maxsat benchmarks based on determining generalized hypertree-width. MaxSAT Evaluation 2017 p. 22 (2017)
4. Bonifati, A., Martens, W., Timm, T.: An analytical study of large SPARQL query logs. PVLDB **11**(2), 149–161 (2017)
5. Fichte, J., Hecher, M., Lodha, N., Szeider, S.: An SMT Approach to Fractional Hypertree Width. Proc. CP 2018, pp.109–127 (2018)
6. Fischl, W., Gottlob, G., Longo, D.M., Pichler, R.: Hyperbench: A benchmark and tool for hypergraphs and empirical findings. In: PODS 2019 (to appear) (2019)
7. Fischl, W., Gottlob, G., Pichler, R.: General and fractional hypertree decompositions: Hard and easy cases. In: Proc. PODS 2018. ACM (2018)
8. Gottlob, G., Samer, M.: A backtracking-based algorithm for hypertree decomposition. ACM Journal of Experimental Algorithmics **13** (2008)
9. Jain, S., Moritz, D., Halperin, D., Howe, B., Lazowska, E.: Sqlshare: Results from a multi-year sql-as-a-service experiment. In: Proc. SIGMOD 2016. ACM (2016)
10. Leis, V., Radke, B., Gubichev, A., Mirchev, A., Boncz, P., Kemper, A., Neumann, T.: Query optimization through the looking glass, and what we found running the join order benchmark. The VLDB Journal (2017).
11. Pottinger, R., Halevy, A.: MiniCon: A scalable algorithm for answering queries using views. The VLDB Journal **10**(2-3), 182–198 (Sep 2001)
12. Transaction Processing Performance Council (TPC): TPC-H decision support benchmark. http://www.tpc.org/tpch/default.asp (2014)