Motivating running instances $(\mathbf{K}-\mathbf{SAT})$

The ZH algorithm has been examined on a wide range of test cases, including random $2-\mathrm{MSP}$ instances and hard SAT instances generated using the Pigeonhole Formulas [TSZ10] and the RB model [XL00] (interestingly, it was claimed that even small instances can be hard near the phase-transition point chosen by the RB model [XBH+07]). Here, we illustrate a family of minimal unsatisfiable (abbr. MU) formulae—which was frequently mistaken by the audience as the "counter example". The formulae are:

$$
\begin{aligned}
F_n = &\left(\bigvee\nolimits_{1\le i\le n} x_i\right)\\
&\wedge \bigwedge\nolimits_{1\le i\le n}\left(\sim x_i \vee \bigvee\nolimits_{1\le j\le n, j\ne i} x_j\right)\\
&\wedge \bigwedge\nolimits_{1\le i<j\le n}\left((\sim x_i \vee \sim x_j)\vee \bigvee\nolimits_{1\le k\le n, k\ne i,j} x_k\right)\\
&\cdots\\
&\wedge\left(\bigvee\nolimits_{1\le i\le n} \sim x_i\right).
\end{aligned}
\tag{19}
$$

By the definition of $2-\mathrm{MSP}$, we just focus on F_2, F_3. The conversion from $3-\mathrm{SAT}$ to $2-\mathrm{MSP}$ is suggested in Appendix A, with auxiliary vertices "p_i","q_i","r_i". Although $2-\mathrm{SAT}\in \mathrm{P}$, we include the instance (Figure 1) might as well to obtain a more comprehensive perspective. Some extra processing is needed for $n>3$, which will make the instance become too large to be illustrated here. For example, when $n=3$, the corresponding $3-\mathrm{SAT}$ instance generated by Tseitin's Transformation [Tse68] will contain 64 clauses, not even including the extra stages of "virtual" vertices to be added by the reduction to $2-\mathrm{MSP}$.

Take F_3 (Figure 2) for example.

All edges of stage 16 can be pruned from each of $R(\langle p_5,\sim x_1,11\rangle)$, $R(\langle p_5,x_2,11\rangle)$, $R(\langle q_5,\sim x_1,11\rangle)$, $R(\langle q_5,\sim x_3,11\rangle)$, $R(\langle r_5,x_2,11\rangle)$, $R(\langle r_5,\sim x_3,11\rangle)$ by Operator 4. This results in $\chi^{D}_{R(E)}(\lambda(D))=\emptyset$ and thus a decision of unsatisfiability can be made. Note that the pruning of edges might even happen earlier than as depicted, while the latter makes it easier for illustration without changing the result.

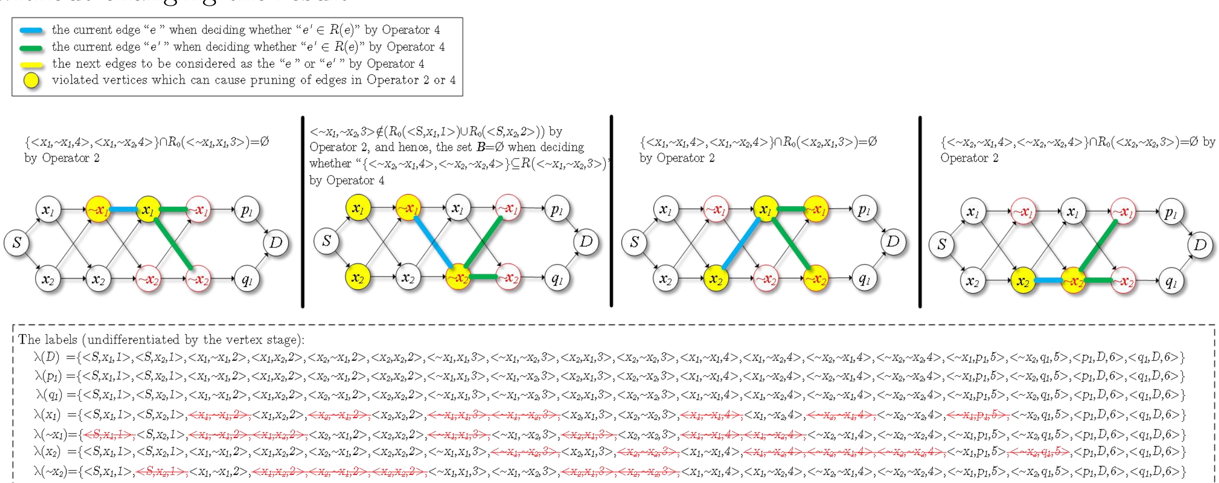

Figure 1: 2-SAT MU running instance

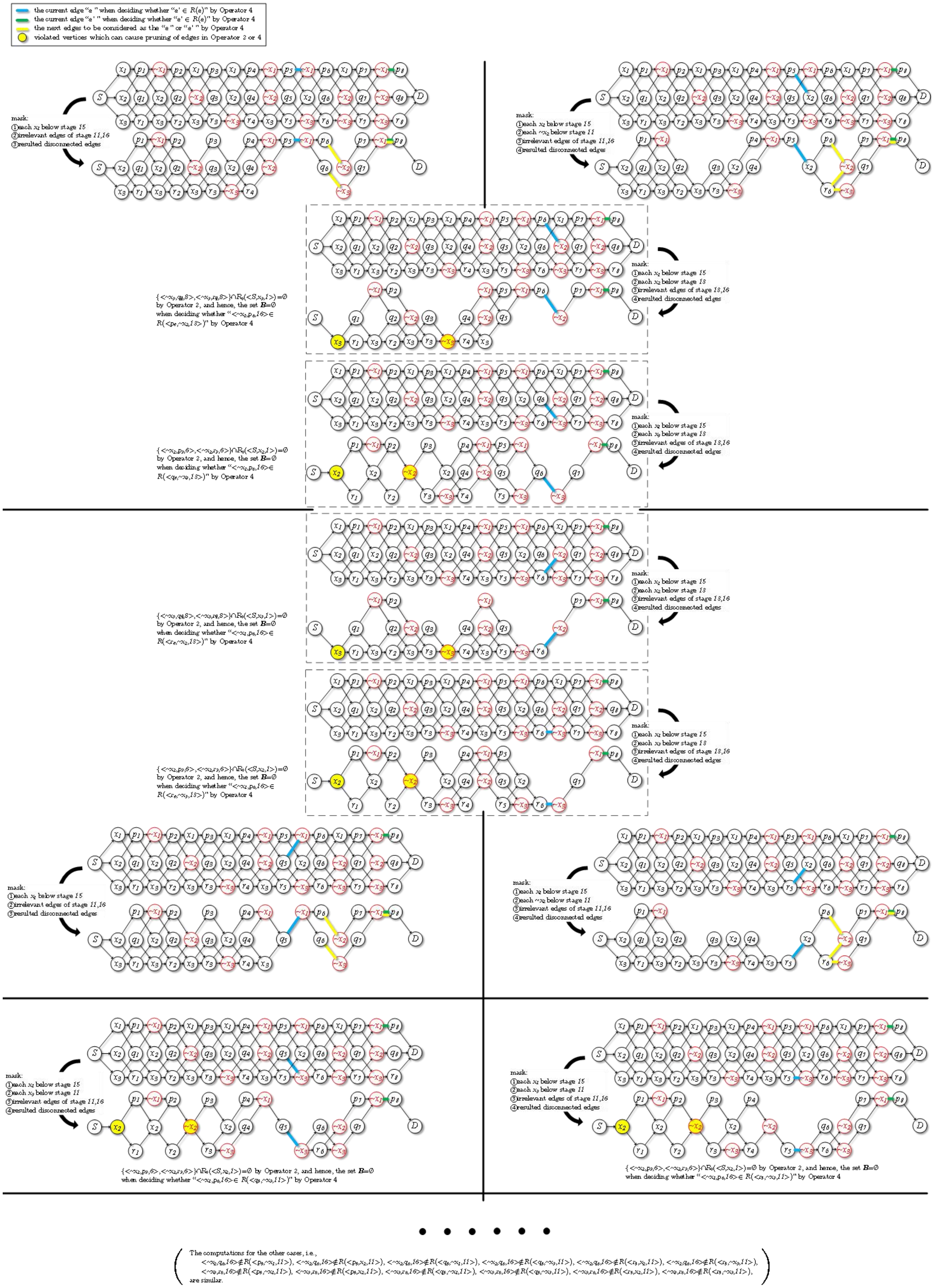

Figure 2: 3-SAT MU running instance

To better focus on the computation of set $\boldsymbol{A}$ and $\boldsymbol{B}$ in Operator 4, irrelevant edges are masked (we did not actually prune these edges from the labeled multi-stage graphs). For example, when deciding whether "$\langle \sim x_1, p_8, 16\rangle \in R(\langle p_5, x_2, 11\rangle)$" by Operator 4: (i) each edge below stage 15 which violates the label of the vertex $\sim x_1$ on $\langle \sim x_1, p_8, 16\rangle$ is masked (see the definition of set $\boldsymbol{A}$ when defining Operator 4); (ii) each edge below stage 11 which violates the label of the vertex x_2 on $\langle p_5, x_2, 11\rangle$ is masked (see the definition of set $\boldsymbol{B}$ when defining Operator 4); and (iii) each edge which deviates paths like $S - \cdots - p_5 - x_2 - \cdots - \sim x_1 - p_8 - D$ is masked (see the definition of set $\boldsymbol{C}$ when defining Operator 4).

Each literal and each clause is indispensable and responsible for the unsatisfiability of the MU formulas; with any missing of single piece of information, a polynomial-time algorithm might crush into a wrong decision. However, the ZH algorithm, which obviously is not specially designed to solve this very type of instances, is shown to be able to collect all necessary global information for a correct decision, by utilizing the computed stable set $R(E)$ which fulfills the constraint imposed by Operator 4. When deciding whether "$\langle \sim x_1, p_8, 16\rangle \in R(\langle p_5, \sim x_1, 11\rangle)$" by Operator 4, $\langle p_6, \sim x_2, 13\rangle$ can be masked during the computation of the set $\boldsymbol{B}$, since we have $\langle \sim x_1, p_8, 16\rangle \notin R(\langle p_6, \sim x_2, 13\rangle)$ by some earlier step of the ZH algorithm.

References

[Tse68] Grigori S. Tseitin. 1968. On the complexity of derivation in propositional calculus[J]. *Zapiski nauchnykh seminarov LOMI*, 8: 234-259.

[TSZ10] Olga Tveretina, Carsten Sinz, and Hans Zantema. 2010. Ordered binary decision diagrams, Pigeonhole Formulas and beyond[J]. *Journal on Satisfiability Boolean Modeling and Computation*, 7(1):35-58.

[XBH+07] Ke Xu, Frédéric Boussemart, Fred Hemery, and Christophe Lecoutre. 2007. Random constraint satisfaction: easy generation of hard (satisfiable) instances[J]. *Artificial Intelligence*, 171(8-9): 514-534.

[XL00] Ke Xu and Wei Li. 2000. Exact phase transitions in random constraint satisfaction problems[J]. *Journal of Artificial Intelligence Research*, 12(1): 93-103.