Randomized Variants of Johnson's Algorithm for MAX SAT *†

Matthias Poloczek Georg Schnitger

Abstract

We give a randomized variant of Johnson's algorithm for MAX SAT [12] and show that its expected approximation ratio is $\frac{3}{4}$. Our solution also works in an online setting where variables are revealed one by one together with the clauses they appear in. Our simple algorithm does not use the power of linear programming and, to the best of our knowledge, is the first such algorithm to reach approximation ratio $\frac{3}{4}$. We also investigate a variant of Johnson's algorithm proposed in [5] that processes variables in random order. Here we show that the expected approximation ratio is worse than $\frac{3}{4}$, thus providing a partial answer to a question of [5].

1 Introduction

In the maximum satisfiability problem (MAX SAT) we are given a collection of clauses and their (nonnegative) weights. Our goal is to find an assignment that satisfies clauses of maximum total weight.

In his fundamental work [12] Johnson presented a greedy algorithm that processes variables in some arbitrary order. The modified weight μ is introduced, where $\mu(c) = w_c \cdot 2^{-|c|}$ for each clause c with weight w_c and length $|c|$. Observe that $\mu(c)$ favors shorter clauses which are in greater danger of being falsified. If x is the currently processed variable, then x is set to one iff $\mu_x = \sum_{c, x \in c} \mu(c)$, the support for $x = 1$, is at least as large as $\mu_{\overline{x}} = \sum_{c, \overline{x} \in c} \mu(c)$, the support for $x = 0$. Yannakakis [16] remarked that Johnson's algorithm can be interpreted as the derandomization of setting $x = 1$ with probability $\frac{1}{2}$.

The formula $(x_1 \vee \overline{x}_2) \wedge (\overline{x}_1 \vee x_2) \wedge \overline{x}_2$, with all three clauses of weight one, shows that the approximation ratio is at least $\frac{2}{3}$: the algorithm begins by assigning $x_1 = 1$, then $x_2 = 1$ and hence satisfies clauses of total weight two (out of three). Chen, Friesen and Zheng [5] proved that $\frac{2}{3}$ is indeed the exact approximation ratio; they also asked whether there is a variant of Johnson's algorithm performing better than $\frac{2}{3}$. A simpler analysis was provided by Engebretsen [7].

In this proceedings Costello, Shafira, and Tetali [6] show that a randomly chosen order actually improves the approximation ratio to $\frac{2}{3} + c$ for some constant $c > 0$. We show in Theorem 3.2 that the expected approximation ratio for 2CNF formulae is at most $2\sqrt{15} - 7 \approx 0.746 < \frac{3}{4}$.

Using network flow techniques and linear programming to construct an appropriate probability distribution, Yannakakis [16] gave a $\frac{3}{4}$ approximation algorithm. Goemans and Williamson [9] achieved the same ratio by a best–of–two approach comparing the result of Johnson's algorithm with the result obtained by a linear program relaxation. It is natural to ask whether there are algorithms achieving an approximation ratio of $\frac{3}{4}$ without applying the power of linear programming, a problem posed by Williamson in his lecture notes on approximation algorithms.

MAX EkSAT and MAX kSAT are the MAX SAT versions where every clause has length *exactly*, resp. *at most* k. In a seminal paper Goemans and Williamson [10] used semidefinite programming for MAX 2SAT. Concluding a series of papers [8, 15], the (currently) best algorithm for MAX 2SAT is due to Lewin, Livnat and Zwick [14]. Their approximation ratio of matches the bound of Austrin [3] obtained by assuming the Unique Games Conjecture. Karloff and Zwick [13] gave a $\frac{7}{8}$ approximative algorithm for MAX 3SAT, a computer assisted analysis appeared in [17]. This is the best possible approximation ratio, since Håstad [11] showed that it is NP–hard to approximate MAX E2SAT within a factor of $\frac{21}{22}+\varepsilon$ and MAX E3SAT within $\frac{7}{8} + \varepsilon$ for any $\varepsilon > 0$.

The best approximation ratio of 0.797 for MAX SAT is achieved by a hybrid algorithm due to Avidor, Berkovitch and Zwick [4]. Moreover, they give an additional algorithm with a conjectured performance of 0.843. The hybrid algorithm runs several MAX SAT algorithms in parallel and picks the best solution. This approach was introduced by Asano and Williamson [2] and refined in [1].

1.1 Our Contributions.

We first study the canonical randomization (CR) of Johnson's algorithm, where the current variable x is set to one with probability $\frac{\mu_x}{\mu_x + \mu_{\overline{x}}}$, and show (Theorem 2.1 in the appendix)

*partially supported by DFG SCHN 503/5-1

†Institute of Computer Science, University of Frankfurt, Robert–Mayer–Str. 11–15, Germany. Email: {matthias, georg}@thi.cs.uni-frankfurt.de

that its expected approximation ratio is smaller than $\frac{3}{4}$, even when restricted to 2CNF formulae.

A closer analysis shows that CR produces weak results only if the slack $|\mu_x - \mu_{\overline{x}}|$ is rather small, i.e. belongs to the open interval $(0, w_x + w_{\overline{x}})$. ($w_x$ is the combined weight of all clauses which are equivalent to the unit clause x after the previous assignments.) We utilize this observation to design a new randomized variant which we call the Slack–Algorithm. The Slack–Algorithm adjusts probabilities carefully, in particular the "majority probability" is increased by a suitably small amount.

Theorem 1.1. *Denote the total weight of all clauses by W. If an optimal assignment satisfies clauses of total weight* Opt*, then*

$$\mathbb{E}[\mathrm{Sat}] \geq \frac{2\mathrm{Opt} + W}{4} \geq \frac{3}{4}\mathrm{Opt},$$

where $\mathbb{E}[\mathrm{Sat}]$ is the expected clause weight satisfied by the Slack–Algorithm. The analysis is tight: no approximation ratio $c > \frac{3}{4}$ can be achieved.

How important is the order in which variables are processed? We show in Theorem 3.1 that any 2CNF formula ϕ has an *optimal* order: call an order optimal for ϕ, if Johnson's algorithm provides an optimal solution for ϕ and does so for any tie-breaking scheme.

Determining an optimal order is certainly an intractable task. However, what is the expected approximation ratio for a randomly chosen order? We show in Theorem 3.2 that the expected approximation ratio for 2CNF formulae is at most $2\sqrt{15} - 7 \approx 0.746 < \frac{3}{4}$. This partially answers a question of [5].

Instead of using the modified weight $\mu(c) = w_c \cdot 2^{-|c|}$, we double the weight of unit clauses and keep the weight of longer clauses unchanged. In the case of a 2CNF formula both strategies are equivalent. Also, whenever we show that an algorithm is deficient, we show that it is already deficient for 2CNF formulae.

1.2 Organisation of the Paper.

In the next section we state and analyze the Slack–Algorithm. A single step of the CR algorithm is examined in Sect. 2.1. We present an overview of the analysis in Sect. 2.2 and show in Sect. 2.3 how to adjust the assignment probabilities. We present the algorithm and prove Theorem 1.1 in Sect. 2.4. The role of the variable order is investigated in Sect. 3. We verify the existence of an optimal order for 2CNF formulae (Theorem 3.1) in Sect. 3.1 and analyze the worst-case behavior of Johnson's algorithm on a random order (Theorem 3.2) in Sect. 3.2. We show in the appendix that the approximation ratio of the CR algorithm is at most $\frac{17}{23} < \frac{3}{4}$.

2 A $\frac{3}{4}$ Approximation

We begin with the canonical randomization (CR) of Johnson's algorithm, where the current variable x is set to one with probability $\frac{\mu_x}{\mu_x + \mu_{\overline{x}}}$. We construct a hard family of 2CNF formulae and analyze its expected approximation ratio.

Theorem 2.1. *The approximation ratio of the CR algorithm is at most*

$$\frac{17}{23} < \frac{3}{4}.$$

Canceling the weight of contradictory unit clauses $x, \overline{x}$ before computing the assignment probabilities does not improve the approximation ratio.

The analysis is moved to the appendix.

2.1 A Single Step of CR.

As a consequence of Theorem 2.1, to obtain a $\frac{3}{4}$ approximation we have to come up with a new strategy. We start by analyzing a single step of CR. Let π be a fixed optimal assignment. We use π as reference throughout our analysis.

Assume that the variable x is to be decided for a given CNF formula. Let fanin (fanout) be the weight of all clauses of length at least two that contain the literal x (resp. $\overline{x}$). For the weights $w_x, w_{\overline{x}}$ of the unit clauses $x, \overline{x}$ resp., and $\Delta = \mathrm{fanout} + 2w_{\overline{x}} + \mathrm{fanin} + 2w_x$, we define

$$q_0 := \mathrm{prob}[x = 0] = \frac{2w_{\overline{x}} + \mathrm{fanout}}{\Delta} \quad \text{and}$$
$$q_1 := \mathrm{prob}[x = 1] = \frac{2w_x + \mathrm{fanin}}{\Delta}$$

as assignment probabilities. Notice that we double the unit clauses but keep the weight of longer clauses unchanged when determining assignment probabilites.

Assume w.l.o.g. $q_0 \leq q_1$ and hence Johnson's algorithm assigns $x = 1$. Our approach will be to increase the majority probability slightly, i.e., replace q_1 by $p_1 = q_1 + \varepsilon$ and q_0 by $p_0 = q_0 - \varepsilon$. Such an increase is certainly a good idea, if $x = 1$ is an optimal assignment (i.e., $\pi_x = 1$), however the increase is dangerous in case $x = 1$ is not optimal (i.e., $\pi_x = 0$).

How does CR behave if we use the new assignment probabilities p_0 and p_1 instead of q_0 and q_1? For p_0, p_1 we introduce the random variable "slack" as the difference between the support in favor of the current decision and the support against it:

(2.1)
$$\mathrm{slack} = \begin{cases} 2w_{\overline{x}} + \mathrm{fanout} - (2w_x + \mathrm{fanin}) & \text{if } x = 0, \\ 2w_x + \mathrm{fanin} - (2w_{\overline{x}} + \mathrm{fanout}) & \text{otherwise.} \end{cases}$$

Let "sat" and "unsat" be the random variables which equal the total weight of all satisfied resp. (terminally) falsified clauses when fixing x. Then obviously

$$\begin{aligned}&\mathbb{E}[\text{sat}] = p_0(w_{\overline{x}} + \text{fanout}) + p_1(w_x + \text{fanin}) \text{ and}\\ &\mathbb{E}[\text{unsat}] = p_0 w_x + p_1 w_{\overline{x}}.\end{aligned} \tag{2.2}$$

Call a clause c containing x or $\overline{x}$ *alive*, iff c, after fixing x, contains at least one unassigned literal and c has not been satisfied yet. Now define the random variable "wounded" as the total weight of all alive clauses. We obtain

$$\mathbb{E}[\text{wounded}] = p_0 \cdot \text{fanin} + p_1 \cdot \text{fanout}. \tag{2.3}$$

Lemma 2.1. *For any assignment probabilities*

$$\mathbb{E}[\text{sat} - 3\text{unsat}] = \mathbb{E}[\text{slack} + \text{wounded}] - (w_x + w_{\overline{x}})$$

holds.

Proof. $\mathbb{E}[\text{sat} - 3\text{unsat}]$

$$\begin{aligned}
&\overset{(2.2)}{=} && p_0(w_{\overline{x}} + \text{fanout}) + p_1(w_x + \text{fanin})\\
& && - 3 \cdot (p_0 \cdot w_x + p_1 \cdot w_{\overline{x}})\\
&= && p_0(w_{\overline{x}} + \text{fanout} - 3w_x - \text{fanin})\\
& && + p_1(w_x + \text{fanin} - 3w_{\overline{x}} - \text{fanout})\\
& && + p_0 \cdot \text{fanin} + p_1 \cdot \text{fanout}\\
&\overset{(2.1)}{=} && \mathbb{E}[\text{slack}] - p_0 \cdot (w_{\overline{x}} + w_x) - p_1 \cdot (w_x + w_{\overline{x}})\\
& && + p_0 \cdot \text{fanin} + p_1 \cdot \text{fanout}\\
&\overset{(2.3)}{=} && \mathbb{E}[\text{slack}] - (w_{\overline{x}} + w_x) + \mathbb{E}[\text{wounded}]
\end{aligned}$$

and this was to be shown. □

2.2 An Overview of the Analysis. Remember that π is the fixed optimal assignment. We study the impact of fixing variable x on the amount of "contradiction" with respect to π. In particular, the random variable c is the weight of all alive clauses not satisfied by π *before* fixing x and c' is the same quantity *after* fixing x. Obviously, as a consequence of Lemma 2.1, we have

$$\begin{aligned}&\mathbb{E}[\text{sat} - 3 \cdot \text{unsat} - 2(c' - c)]\\ &\quad = \mathbb{E}[\text{slack} + \text{wounded} - 2(c' - c)] - (w_{\overline{x}} + w_x).\end{aligned} \tag{2.4}$$

Assume that we have determined probabilities p_0, p_1 such that the right hand side of (2.4) is always nonnegative. Let Sat, Unsat be the random variables indicating the total weight of satisfied and unsatisfied clauses after fixing all variables. Opt is the weight of all clauses satisfied by π and W is the total clause weight. Then, after summing up the left hand side of (2.4), we have

$$\begin{aligned}
\mathbb{E}[\text{Sat}] &\geq 3\mathbb{E}[\text{Unsat}] - 2 \cdot (W - \text{Opt})\\
&= 3 \cdot (W - \mathbb{E}[\text{Sat}]) - 2 \cdot (W - \text{Opt})\\
&= W - 3\mathbb{E}[\text{Sat}] + 2\text{Opt}
\end{aligned}$$

and the first part of Theorem 1.1 follows.

In Lemma 2.3 we show that

$$\mathbb{E}\left[\text{wounded} - 2(c' - c)\right] - (w_{\overline{x}} + w_x)$$

is relatively large and, in particular, we relate this quantity to the slack of the Johnson decision for x and the probability adjustment ε. As a first step we bound $\mathbb{E}[c' - c]$, the expected net contradiction introduced, resp. resolved when fixing x.

Lemma 2.2. *Assume that the assignment probabilities are given by* $p_0 = \text{prob}[x = 0]$*,* $p_1 = \text{prob}[x = 1]$*.*
If $x = 1$ *is optimal, then*

$$\mathbb{E}[c' - c] \leq p_0 \cdot \text{fanin} - w_{\overline{x}}$$

holds, and if $x = 0$ *is optimal,*

$$\mathbb{E}[c' - c] \leq p_1 \cdot \text{fanout} - w_x.$$

Proof. First assume that $x = 1$ is optimal. If we set $x = 0$, i.e., if we fix x non-optimally, then we may create new contradictions. Observe that fanin is the total weight of all alive clauses in this case. Let fanin^c denote the weight of all alive clauses which become contradictory after setting $x = 0$: any such clause is not satisfied if the remaining variables are set according to π. Clearly, $\text{fanin}^c \leq \text{fanin}$ holds.

If we assign $x = 1$, then fanout is the total weight of all alive clauses. Since π remains our reference when measuring contradictions, any clause c considered in fanout remains contradictory, resp. remains non-contradictory after fixing x. Thus, no new contradictions are created.

Irrespective of the particular assignment, the contradictory unit clause $\overline{x}$ is removed from the set of alive clauses: the contradiction due to $\overline{x}$ is resolved. Thus we get

$$\mathbb{E}[c' - c] = p_0 \cdot (\text{fanin}^c - w_{\overline{x}}) - p_1 \cdot w_{\overline{x}} \leq p_0 \cdot \text{fanin} - w_{\overline{x}}.$$

The remaining case of an optimal assignment $x = 0$ is treated analogously. □

We set $\text{Slack} = |2w_x + \text{fanin} - (2w_{\overline{x}} + \text{fanout})|$ and can now make a first important step in bounding the right-hand side of (2.4). Remember that q_0, q_1 are the original assignment probabilities of CR and observe that $q_1 - q_0 = \frac{\text{Slack}}{\Delta}$ provided $q_0 \leq q_1$.

Lemma 2.3. *(a)* $q_1\text{fanout} - q_0\text{fanin} = 2q_0 w_x - 2q_1 w_{\overline{x}}$*.*

(b) Assume that $q_0 \leq q_1$*. For assignment probabilities* $p_0 = q_0 - \varepsilon, p_1 = q_1 + \varepsilon$*, if* $x = 1$ *is optimal,*

$$\begin{aligned}&\mathbb{E}[\text{wounded} - 2(c' - c)] - (w_x + w_{\overline{x}})\\ &\qquad \geq -\frac{\text{Slack}}{\Delta}(w_x + w_{\overline{x}}) + \varepsilon(\text{fanout} + \text{fanin})\end{aligned}$$

and if $x = 0$ is optimal,

$$\mathbb{E}[\text{wounded} - 2(c' - c)] - (w_x + w_{\overline{x}}) \geq \frac{\text{Slack}}{\Delta}(w_x + w_{\overline{x}}) - \varepsilon(\text{fanout} + \text{fanin})$$

Before giving the proof we discuss the consequences of Lemma 2.3. We may assume w.l.o.g. that $q_0 \leq q_1$ holds. An algorithm can easily verify $q_0 \leq q_1$, but cannot determine efficiently if, say, $x = 1$ is optimal, i.e., if $\pi_x = 1$ holds. If $x = 1$ is optimal and hence if CR agrees, in its majority decision, with π, then the right hand side of (2.4) may be negative for $\varepsilon = 0$, in which case ε has to be increased to enforce $\mathbb{E}[\text{slack}] - \frac{\text{Slack}}{\Delta}(w_x + w_{\overline{x}}) + \varepsilon(\text{fanout} + \text{fanin}) \geq 0$. Let ε^* be a smallest such ε. Observe that the algorithm has the required knowledge to determine ε^*.

If however $x = 0$ is optimal and hence if CR behaves non-optimally with its majority decision, then the right side of (2.4) will be nonnegative. Thus, if we can make sure, that the right hand side of (2.4) stays nonnegative for $\varepsilon = \varepsilon^*$ in case $x = 0$ is optimal, then the right hand side is nonnegative in either case.

Proof. (a) Observe that (a) is equivalent with

$$q_1 \cdot (2w_{\overline{x}} + \text{fanout}) = q_0 \cdot (2w_x + \text{fanin})$$

and this equation is true, since both sides coincide with $q_0 \cdot q_1 \cdot \Delta$.
(b) First assume that $x = 1$ is optimal. We observe

$$\begin{aligned} &\mathbb{E}[\text{wounded} - 2(c' - c)] - (w_x + w_{\overline{x}}) \\ &\geq p_0\text{fanin} + p_1\text{fanout} - 2\left(p_0\text{fanin} - w_{\overline{x}}\right) - (w_x + w_{\overline{x}}) \\ &= p_1\text{fanout} - p_0\text{fanin} + w_{\overline{x}} - w_x =: \Gamma, \end{aligned}$$

where the inequality follows from (2.3) and Lemma 2.2. But

$$\begin{aligned} &p_1 \cdot \text{fanout} - p_0 \cdot \text{fanin} + w_{\overline{x}} - w_x \\ &= q_1 \cdot \text{fanout} - q_0 \cdot \text{fanin} + w_{\overline{x}} - w_x \\ &\quad + \varepsilon \cdot (\text{fanout} + \text{fanin}) \\ &\overset{(a)}{=} 2q_0 w_x - 2q_1 w_{\overline{x}} + w_{\overline{x}} - w_x \\ &\quad + \varepsilon \cdot (\text{fanout} + \text{fanin}) \\ &= (q_0 - q_1)(w_x + w_{\overline{x}}) + \varepsilon \cdot (\text{fanout} + \text{fanin}) \end{aligned}$$

$$(2.5) \quad = -\frac{\text{Slack}}{\Delta}(w_x + w_{\overline{x}}) + \varepsilon \cdot (\text{fanout} + \text{fanin}).$$

The last equation follows, since $q_0 \leq q_1$ holds, by assumption, and hence $q_0 - q_1 = -\frac{\text{Slack}}{\Delta}$. If $x = 0$ is optimal, then again with Lemma 2.2

$$\begin{aligned} &\mathbb{E}[\text{wounded} - 2(c' - c)] - (w_x + w_{\overline{x}}) \\ &\geq p_0\text{fanin} + p_1\text{fanout} - 2p_1\text{fanout} + 2w_x - (w_x + w_{\overline{x}}) \\ &= p_0\text{fanin} - p_1\text{fanout} + w_x - w_{\overline{x}} = -\Gamma. \end{aligned}$$

We apply (2.5) and the claim follows. □

2.3 How to Adjust the Canonical Probabilities?

Assume that we process variable x. We may also assume w.l.o.g. that $q_0 \leq q_1$ holds. For assignment probabilities $p_0 = q_0 - \varepsilon, p_1 = q_1 + \varepsilon$ we determine $\mathbb{E}[\text{slack}]$, the expected slack when fixing x. Then

$$\begin{aligned} \mathbb{E}[\text{slack}] &= (p_1 - p_0) \cdot (2w_x + \text{fanin} - (2w_{\overline{x}} + \text{fanout})) \\ &= (q_1 - q_0 + 2\varepsilon) \cdot (q_1 - q_0) \cdot \Delta \\ &= (q_1 - q_0)^2 \cdot \Delta + 2\varepsilon \cdot (q_1 - q_0) \cdot \Delta \end{aligned}$$

$$(2.6) \quad = \frac{\text{Slack}^2}{\Delta} + 2\varepsilon \cdot \text{Slack},$$

where we utilize that $\text{Slack} = (q_1 - q_0) \cdot \Delta$ follows, since $q_0 \leq q_1$ holds.

Our goal is to determine ε (with $\varepsilon \leq q_0$) such that the right hand side of invariant (2.4) is nonnegative. We first additionally assume that $x = 1$ is optimal. We apply (2.6)

$$\begin{aligned} &\mathbb{E}[\text{slack} + \text{wounded} - 2(c' - c)] - (w_{\overline{x}} + w_x) \\ \overset{(2.6)}{=}\; &\frac{\text{Slack}^2}{\Delta} + 2\varepsilon \cdot \text{Slack} + \mathbb{E}[\text{wounded} - 2(c' - c)] \\ &- (w_{\overline{x}} + w_x) \\ \geq\; &\frac{\text{Slack}^2}{\Delta} + 2\varepsilon \cdot \text{Slack} - \frac{\text{Slack}}{\Delta}(w_x + w_{\overline{x}}) \\ &+ \varepsilon(\text{fanout} + \text{fanin}). \end{aligned}$$

and the last inequality follows from Lemma 2.3. The right hand side is negative for $\varepsilon = 0$ only if Slack belongs to the interval $(0, w_x + w_{\overline{x}})$. Hence we have to adjust the assignment probabilites only if $\text{Slack} \in (0, w_x + w_{\overline{x}})$, which we assume from now on. The right hand side vanishes for

$$(2.7) \qquad \varepsilon_1 = \frac{-\frac{\text{Slack}^2}{\Delta} + \frac{\text{Slack}}{\Delta}(w_x + w_{\overline{x}})}{2\text{Slack} + \text{fanout} + \text{fanin}}.$$

If however $x = 0$ is optimal, then we obtain with (2.6)

$$\begin{aligned} &\mathbb{E}[\text{slack} + \text{wounded} - 2(c' - c)] - (w_{\overline{x}} + w_x) \\ \overset{(2.6)}{=}\; &\frac{\text{Slack}^2}{\Delta} + 2\varepsilon \cdot \text{Slack} + \mathbb{E}[\text{wounded} - 2(c' - c)] \\ &- (w_{\overline{x}} + w_x) \\ \geq\; &\frac{\text{Slack}^2}{\Delta} + 2\varepsilon \cdot \text{Slack} + \frac{\text{Slack}}{\Delta}(w_x + w_{\overline{x}}) \\ &- \varepsilon(\text{fanout} + \text{fanin}) \end{aligned}$$

and the last inequality again follows from Lemma 2.3. The right hand side is nonnegative for $\varepsilon = 0$. It is also nonnegative for all $\varepsilon \geq 0$, provided $\text{fanout} + \text{fanin} \leq 2\text{Slack}$ holds. We therefore assume the opposite: as a consequence the right side is monotone decreasing and vanishes for

$$\varepsilon_2 = \frac{\frac{\text{Slack}^2}{\Delta} + \frac{\text{Slack}}{\Delta}(w_x + w_{\overline{x}})}{-2\text{Slack} + \text{fanout} + \text{fanin}}.$$

The numerator of ε_1 is at most as large as the numerator of ε_2 and its denominator is at least as large as the denominator of ε_2. Hence $\varepsilon_1 \leq \varepsilon_2$ and

$$\mathbb{E}[\text{slack} + \text{wounded} - 2(c' - c)] - (w_{\overline{x}} + w_x) \geq 0$$

for $p_0 = q_0 - \varepsilon_1$ and $p_1 = q_1 + \varepsilon_1$, irrespective of the optimality of $x = 0$ or $x = 1$.

LEMMA 2.4. *Assume that $q_0 \leq q_1$. Then p_0, p_1 are well defined, i.e., $0 \leq \varepsilon_1 \leq q_0$.*

Proof. We first observe that ε_1 is nonnegative, since Slack $\in (0, w_x + w_{\overline{x}})$. It remains to show that

$$\text{(2.8)} \qquad \begin{aligned} \varepsilon_1 &= \frac{-\frac{\text{Slack}^2}{\Delta} + \frac{\text{Slack}}{\Delta}(w_x + w_{\overline{x}})}{2\text{Slack} + \text{fanout} + \text{fanin}} \\ &\overset{!}{\leq} q_0 = \frac{\text{fanout} + 2w_{\overline{x}}}{\Delta} \end{aligned}$$

holds. We increase $w_x, w_{\overline{x}}$ and decrease fanin, fanout in such a way that fanin$\,+\,2w_x$, fanout$\,+\,2w_{\overline{x}}$ remain fixed: observe that Slack and Δ remain fixed and if (2.8) is false before this modification, then it is false afterwards. Hence we may assume fanin $=$ fanout $= 0$ and (2.8) is equivalent with

$$\text{(2.9)} \qquad -\text{Slack} + w_x + w_{\overline{x}} \overset{!}{\leq} 4w_{\overline{x}}.$$

But $q_0 \leq q_1$ and hence Slack $= 2(w_x - w_{\overline{x}})$ follows. Thus (2.9) is equivalent with

$$-2(w_x - w_{\overline{x}}) + w_x + w_{\overline{x}} = 3w_{\overline{x}} - w_x \overset{!}{\leq} 4w_{\overline{x}}$$

and this inequality is correct. □

2.4 The Slack–Algorithm and Its Performance. We present the Slack–Algorithm in Fig. 1 and prove Theorem 1.1.

Proof. We have shown in Sect. 2.2 that an approximation ratio of at least $\frac{3}{4}$ follows if local invariant (2.4) holds.

To show that our analysis is tight, we work with the $2n$ variables $x_1, \ldots x_n$ and $y_1, \ldots, y_n$, where n is sufficiently large. Our hard formula ϕ consists of all equivalences $(x_i \vee \overline{y_j}) \wedge (\overline{x}_i \vee y_j)$, where each clause receives weight one. Observe that the all-ones assignment satisfies ϕ and hence Opt $= 2n^2$.

We first process all x-variables in some arbitrary order and subsequently all y-variables in some arbitrary order as well. When fixing x-variables we get prob$[x_i = 1] = \frac{1}{2}$, since Slack $= 0$. If all x-variables are fixed, then even a best assignment for the y-variables does not help much: the expected weight of falsified implications is at least

$$\sum_{k=0}^{n} \binom{n}{k} \cdot 2^{-n} \cdot \min\{k, n-k\} \cdot n = \frac{n^2}{2} - o(n^2).$$

(This follows, since $\frac{n}{2} \pm o(n)$ x–variables are set to one whp.) Thus the weight of falsified clauses is at least $\frac{1}{4} - o(1)$ of the total weight and no approximation ratio larger than $\frac{3}{4}$ can be reached. □

3 How to Choose the Order?

We first show the existence of an optimal order for 2CNF formula and then study Johnson's algorithm for a random order.

3.1 Optimal Orders for 2CNF Formulae. Call an order optimal for a 2CNF formula ϕ, if Johnson's algorithm provides an optimal solution for ϕ and does so for any tie-breaking scheme.

THEOREM 3.1. *Every 2CNF formula has an optimal order.*

Proof. We construct an optimal order by simulating Johnson's algorithm. As long as there is a variable such that the decision of Johnson's algorithm –for each tie-breaking scheme– is consistent with an optimal assignment, select such a variable as the next variable to be processed.

Thus, we may assume that all decisions for the set V of remaining variables are inconsistent with an optimal assignment, given some worst-case tie-breaking. We may also assume w.l.o.g that Johnson's algorithm assigns zero to all variables in V. Then the all–ones assignment is the only optimal assignment, since otherwise there is a zero–decision which is consistent with an optimal solution, in contradiction to our assumption. Thus we get, after removing "mixed" clauses,

$$\text{(3.10)} \quad \sum_{x \in V} w_x + \sum_{\substack{\{x,y\} \subseteq V \\ x \neq y}} w_{x \vee y} > \sum_{x \in V} w_{\overline{x}} + \sum_{\substack{\{x,y\} \subseteq V \\ x \neq y}} w_{\overline{x} \vee \overline{y}}.$$

On the other hand, the algorithm decides non–optimally for all $x \in V$ and hence

$$2w_x + \sum_{\substack{y \in V \\ x \neq y}} (w_{x \vee \overline{y}} + w_{x \vee y}) \leq 2w_{\overline{x}} + \sum_{\substack{y \in V \\ x \neq y}} (w_{\overline{x} \vee y} + w_{\overline{x} \vee \overline{y}})$$

---

procedure SLACK–ALGORITHM(Formula ϕ) ▷ the formula ϕ has variable set V
  for all $x \in V$ **do** ▷ process variables in the given order
    $q_0 \leftarrow \frac{2w_{\overline{x}}+\text{fanout}}{\Delta}$, $q_1 \leftarrow \frac{2w_x+\text{fanin}}{\Delta}$
    Slack $\leftarrow |q_0 - q_1| \cdot \Delta$
    $\varepsilon_1 \leftarrow$ the probability adjustment as defined in (2.7)
    if Slack $= 0$ or Slack $\geq w_x + w_{\overline{x}}$ **then**
      $p_0 \leftarrow q_0$, $p_1 \leftarrow q_1$
    else ▷ the slack is in the critical range
      if $q_0 < q_1$ **then**
        $p_0 \leftarrow q_0 - \varepsilon_1$, $p_1 \leftarrow q_1 + \varepsilon_1$
      else
        $p_0 \leftarrow q_0 + \varepsilon_1$, $p_1 \leftarrow q_1 - \varepsilon_1$
    assign $x = 0$ with probability p_0 and $x = 1$ with probability p_1

Figure 1: The Slack–Algorithm

for any $x \in V$. Summing all inequalities yields

$$\sum_{x\in V} 2w_x + \sum_{\substack{x,y\in V\\ x\neq y}} w_{x\vee y} \leq \sum_{x\in V} 2w_{\overline{x}} + \sum_{\substack{x,y\in V\\ x\neq y}} w_{\overline{x}\vee\overline{y}}.$$

(Observe that clauses consisting of one positive and one negative literal appear on both sides and therefore can be removed.) But each clause $x \vee y$, resp. $\overline{x} \vee \overline{y}$ appears twice and we have obtained the desired contradiction with (3.10). □

Remark: Theorem 3.1 does not hold for 3CNF formulae. Remember that Johnson's algorithm uses the modified weight $\mu(c) = w_c \cdot 2^{-|c|}$ for each clause c with weight w_c and length $|c|$. For a counterexample consider a formula ϕ with variables x, y and z. ϕ consists of all seven 3-clauses (with weight 1 each) containing at least one positive literal and of the unit clauses $\overline{x}, \overline{y}, \overline{z}$ (with weight 4/15 each). The all–ones assignment satisfies clauses with weight 7, whereas any other assignment achieves a weight less than 7. But Johnson's algorithm assigns zero in its first step regardless which variable is picked.

3.2 Random Order. We investigate a variant of Johnson's algorithm that processes the variables in random order, as proposed in [5], and construct a hard 2CNF formula. Our formula consists of variables $x_1, \ldots, x_n, y_1, \ldots, y_n$ and the following clauses:

- unit clauses $\overline{x}_i$ with (small) weight $S = \delta \cdot n$ and y_i with (large) weight $L = (1-\varepsilon)\cdot n$ for $i = 1, \ldots, n$,
- all equivalences $x_i \leftrightarrow x_j$ with weight 1 for $1 \leq i \neq j \leq n$
- and all equivalences $x_i \leftrightarrow y_j$ with weight 1 for $1 \leq i, j \leq n$.

We choose δ to be a sufficiently small positive constant and carefully select ε to belong to the interval $[0, \frac{1}{2}]$. Thus the weight S of the x-variables is indeed rather small compared with the weight L of the y-variables. Now consider the all-ones assignment. All equivalences and all y-units are satisfied and hence a total weight of

$$W = 2 \cdot \binom{n}{2} + 2 \cdot n^2 + (1-\varepsilon)\cdot n^2 = (4-\varepsilon)\cdot n^2 - n$$

is satisfied. Now assume that π is a random order of the variables. How will Johnson's algorithm behave on π? Initially x-variables will be set to zero and y-variables set to one. When setting the ith x-variable x_j we expect that roughly as many x-variables as y-variables are set to zero and one respectively. The influences of the already fixed x- and y-variables on x_j almost cancel: the already fixed x-variables support $x_j = 0$ and the already fixed y-variables support $x_j = 1$. Thus we expect the unit weight S for $\overline{x}_j$ to tip the scales and the algorithm to set $x_j = 0$.

To make this argument precise we consider a randomly permuted sequence of n zeroes (i.e., the x-variables) and n ones (i.e., the y-variables). What is the probability p that there is a position k with at least $S = \delta \cdot n$ more ones than zeroes in positions at most k? –Observe that Johnson's algorithm sets some x-variable to one with probability at most p.– A standard argument shows

$$p \leq 2n \cdot e^{-\Theta(\delta^2 \cdot n)}.$$

Hence all x-variables are set to zero with probability at least $1 - p$.

Now consider the ith y-variable y_k. Assume that f_i x-variables are already fixed to zero. They push y_k towards zero, whereas the unit weight L pushes towards one: the difference $L - f_i$ decides. We utilize that, for any time k, the number of x- and y-variables differs by at most S with probability at least $1-2p$. Thus the number of y-variables set to one belongs to the interval $[L-S, L+S]$ with probability at least $1-2p$.

We now determine the expected score of the Johnson algorithm up to lower order terms. We may assume that all x-variables are set to zero and that the number of y-variables set to one belongs to the interval $[L-S, L+S]$. All equivalences between x-variables, all x-units and all implications $x_i \to y_j$ (with a combined weight of $2\cdot\binom{n}{2}+n\cdot S+n^2 = 2\cdot n^2-n+n\cdot S$) are satisfied. Moreover, the combined weight of satisfied y-units and implications $y_j \to x_i$ is at most $(L+S)\cdot L+(n-L+S)\cdot n$. Observe that the algorithm does not satisfy clauses with expected total weight of $L\cdot(n-L-S)+n\cdot(L-S)$, a loss which turns out to be larger than $\frac{W}{4}$, where W is the weight of the optimal assignment.

Hence the expected satisfied weight is at most

$$\begin{aligned} &2\cdot n^2+L^2+(n-L)\cdot n+O(S\cdot n) \\ &\qquad = (2+(1-\varepsilon)^2+\varepsilon)\cdot n^2+O(S\cdot n). \end{aligned}$$

We choose δ to be sufficiently small and hence the approximation ratio approaches

$$\alpha = \frac{2+(1-\varepsilon)^2+\varepsilon}{4-\varepsilon} = \frac{3-\varepsilon+\varepsilon^2}{4-\varepsilon}.$$

A simple calculation shows that the quotient is minimized for $\varepsilon = 4-\sqrt{15}$. We obtain

$$\begin{aligned} \alpha &= \frac{3-4+\sqrt{15}+(16-8\sqrt{15}+15)}{\sqrt{15}} \\ &= \frac{30-7\sqrt{15}}{\sqrt{15}} = 2\sqrt{15}-7. \end{aligned} \tag{3.11}$$

THEOREM 3.2. *The approximation ratio of Johnson's algorithm, when applied to a random order, converges to at most*

$$2\sqrt{15}-7 \approx 0.746.$$

We have shown in Sect. 3.1 that any 2CNF formula has an optimal order. Our analysis, however, shows that picking an order with approximation ratio $\frac{3}{4}$ or better is extremely unlikely.

References

[1] T. Asano. An improved analysis of Goemans and Williamson's LP-relaxation for MAX SAT. *Theor. Comput. Sci.*, 354(3):339–353, 2006.

[2] T. Asano and D. P. Williamson. Improved approximation algorithms for MAX SAT. *J. Algorithms*, 42(1):173–202, 2002.

[3] P. Austrin. Balanced MAX 2–SAT might not be the hardest. In *STOC*, pages 189–197, 2007.

[4] A. Avidor, I. Berkovitch, and U. Zwick. Improved approximation algorithms for MAX NAE-SAT and MAX SAT. In *WAOA*, pages 27–40, 2005.

[5] J. Chen, D. K. Friesen, and H. Zheng. Tight bound on Johnson's algorithm for maximum satisfiability. *J. Comput. Syst. Sci.*, 58(3):622–640, 1999.

[6] K. Costello, A. Shapira, and P. Tetali. Randomizing two derandomized greedy algorithms. In *SODA*, 2011.

[7] L. Engebretsen. Simplified tight analysis of Johnson's algorithm. *Inf. Process. Lett.*, 92(4):207–210, 2004.

[8] U. Feige and M. X. Goemans. Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT. In *ISTCS*, pages 182–189, 1995.

[9] M. X. Goemans and D. P. Williamson. New 3/4-approximation algorithms for the maximum satisfiability problem. *SIAM J. Discrete Math.*, 7(4):656–666, 1994.

[10] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995.

[11] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[12] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci.*, 9(3):256–278, 1974.

[13] H. J. Karloff and U. Zwick. A 7/8-approximation algorithm for MAX 3SAT? In *FOCS*, pages 406–415, 1997.

[14] M. Lewin, D. Livnat, and U. Zwick. Improved rounding techniques for the MAX 2-SAT and MAX DI-CUT problems. In *IPCO*, pages 67–82, 2002.

[15] S. Matuura and T. Matsui. 0.935-approximation randomized algorithm for MAX–2SAT and its derandomization. Technical Report METR 2001-03, Department of Mathematical Engineering and Physics, the University of Tokyo, Japan, 2001.

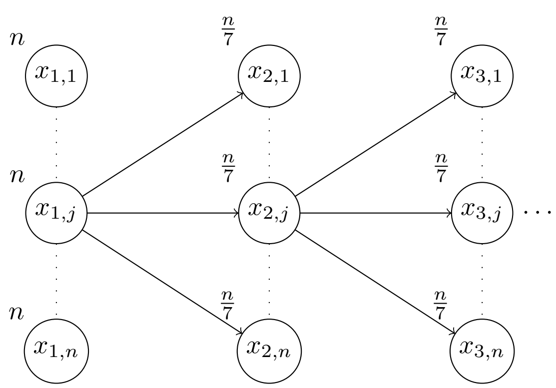
[16] M. Yannakakis. On the approximation of maximum satisfiability. *J. Algorithms*, 17(3):475–502, 1994.

[17] U. Zwick. Computer assisted proof of optimal approximability results. In *SODA*, pages 496–505, 2002.

A Proof of Theorem 2.1

In this section we investigate the CR algorithm, the canonical randomization of Johnson's Algorithm. Variables are processed in an arbitrary order. Remember that, when deciding x, we assign one with probability $\frac{\mu_x}{\mu_x+\mu_{\overline{x}}}$ and zero otherwise. The support μ_x for the decision $x = 1$ is defined as the combined weight of all clauses containing the literal x, adjusted by their length. There is a natural variant of the canonical randomization, which cancels contradictory unit clauses $x, \overline{x}$ before computing the assignment probabilities. Cancelation seems to be of help whenever the weight of contradictory unit clauses is significant.

We construct a family $(\phi_n | n \in \mathbb{N})$ of difficult 2CNF formulae, which will be treated identically by both variants. Let c be a sufficiently large constant. The formula ϕ has $c \cdot n$ variables $x_{i,j}$ for $1 \le i \le c, 1 \le j \le n$. The unit clauses $x_{1,j}$ receive a weight of n, all unit clauses $x_{i,j}$ for $1 < i < c$ receive a weight of $\frac{n}{7}$. Finally ϕ contains all implications $\overline{x}_{i,j} \vee x_{i+1,k}$ for $i < c$ and $1 \le j, k \le n$, where each implication has weight $\frac{3}{2}$. The formula is depicted below: in column i we show the n variables $x_{i,j}$ of group i together with the weight of their unit clause.



Observe that the all-ones assignment satisfies ϕ and hence satisfies the weight $W = n^2 + (c-2) \cdot \frac{n^2}{7} + (c-1) \cdot \frac{3n^2}{2}$. CR first processes all variables $x_{1,j}$ in some arbitrary order, followed by all variables $x_{2,j}$ in some arbitrary order and so forth. All variables $x_{1,j}$ are set to one with probability $p = \frac{2n}{2n+3n/2} = \frac{4}{7}$. We now argue that all remaining variables $x_{i,j}$ for $2 \le i < c$, i.e., all remaining variables except for the variables in group c, are set to one with probability $\frac{4}{7} \pm o(1)$. Consider variable $x_{2,j}$. If exactly $K = \frac{4n}{7} \pm o(n)$ variables $x_{1,k}$ are set to one, then

$$\begin{aligned}
\text{prob}\left[x_{2,j} = 1 \mid \sum_{k=1}^{n} x_{1,k} = K\right] &= \frac{2(\frac{n}{7} + \frac{3}{2}K)}{2(\frac{n}{7} + \frac{3}{2}K) + \frac{3n}{2}} \\
&= \frac{2(\frac{1}{7} + \frac{3}{2}(\frac{4}{7} \pm o(1)))}{2(\frac{1}{7} + \frac{3}{2}(\frac{4}{7} \pm o(1))) + \frac{3}{2}} \\
&= \frac{2 \pm o(1)}{\frac{7}{2} \pm o(1)} = \frac{4}{7} \pm o(1).
\end{aligned}$$

Conditioned on $\sum_{k=1}^{n} x_{1,k} = K$, large deviations of $\sum_{k=1}^{n} x_{2,k}$ from its expected value are extremely unlikely, since we are dealing with n independent experiments. The condition $K = \sum_{k=1}^{n} x_{1,k} = \frac{4n}{7} \pm o(n)$ however can be dropped, since large deviations of $\sum_{k=1}^{n} x_{1,k}$ from its expected value $\frac{4n}{7}$ are also extremely unlikely.

Observe that this argument also holds for variables $x_{i,j}$ for $2 \le i < c$, since we may assume that large deviations of $\sum_{k=1}^{n} x_{i-1,k}$ from its expected value $\frac{4n}{7} \pm o(n)$ are unlikely. As a consequence only an approximate $\frac{4}{7}$-fraction of the entire unit weight is satisfied. Now consider an implication $\overline{x}_{i,j} \vee x_{i+1,k}$, which is falsified with probability

$$\begin{aligned}
&\text{prob}[x_{i,j} = 1, x_{i+1,k} = 0] \\
&\qquad = \text{prob}[x_{i+1,k} = 0 \mid x_{i,j} = 1] \cdot \text{prob}[x_{i,j} = 1] \\
&\qquad = \text{prob}[x_{i+1,k} = 0] \cdot \text{prob}[x_{i,j} = 1] \pm o(1) \\
&\qquad = \frac{3}{7} \cdot \frac{4}{7} \pm o(1).
\end{aligned}$$

Hence a slightly larger fraction than $\frac{3}{4}$ of the entire implication weight is satisfied, but this is insufficient to make up for the loss of unit weight, since the expected weight of satisfied clauses is at most

$$W' = \frac{4}{7} \cdot (n^2 + (c-2)\frac{n^2}{7}) + \frac{37}{49} \cdot (c-2) \cdot \frac{3n^2}{2} + \frac{3n^2}{2} \pm o(c \cdot n^2).$$

(Observe that all implications from group $c-1$ to group c are counted in W'.) If c is sufficiently large, then we get

$$\begin{aligned}
\frac{W'}{W} &\approx \frac{\frac{4}{7} \cdot c \cdot \frac{n^2}{7} + \frac{37}{49} \cdot c \cdot \frac{3}{2}n^2}{c \cdot \frac{n^2}{7} + c \cdot \frac{3}{2}n^2} \\
&= \frac{\frac{4}{7} \cdot \frac{1}{7} + \frac{37}{49} \cdot \frac{3}{2}}{\frac{1}{7} + \frac{3}{2}} \\
&= \frac{\frac{119}{2 \cdot 49}}{\frac{23}{2 \cdot 7}} = \frac{119}{7 \cdot 23} \\
&= \frac{17}{23} < \frac{3}{4}.
\end{aligned}$$

This completes the proof of Theorem 2.1.