Department of Applied Mathematics
Faculty of EEMCS

University of Twente
The Netherlands

P.O. Box 217
7500 AE Enschede
The Netherlands

Phone: +31-53-4893400
Fax: +31-53-4893114

Email: memo@math.utwente.nl
www.math.utwente.nl/publications

Memorandum No. 1709

An improved local search algorithm for 3-SAT

T. Brueggemann and W. Kern

January, 2004

ISSN 0169-2690

An improved local search algorithm for 3-SAT

Tobias Brueggemann [1], Walter Kern [*]

Department of Applied Mathematics, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

Abstract

We slightly improve the pruning technique presented in Dantsin et. al. (2002) to obtain an $\mathcal{O}^*(1.473^n)$ algorithm for 3-SAT.

Key words: exact algorithm, local search, 3-SAT
MSC2000: 68Q25

1 Introduction

An instance of 3-SAT is a boolean formula φ in n variables $x_1, \ldots, x_n$, defined as the conjunction of a set $\mathcal{C}$ of disjunctive clauses of length at most 3. Satisfiability of φ can be tested in a straightforward manner in time

$$\mathcal{O}\left(2^n \cdot n^3\right) = \mathcal{O}^*(2^n).$$

Here, as usual, we use the $\mathcal{O}^*$-notation to indicate that polynomial factors are suppressed.

During the last years so-called *exact algorithms* have been designed solving 3-SAT in time $\mathcal{O}^*(\alpha^n)$ with $\alpha < 2$, see Schoening [3] for an overview. The currently fastest randomized algorithms run in time $\mathcal{O}^*(1.3302^n)$ (see Hofmeister, Schoening, Schuler and Watanabe [2]) and the fastest deterministic algorithm (see Dantsin et. al. [1]) takes $\mathcal{O}^*(1.481^n)$. We slightly improve the pruning technique used in Dantsin et. al. [1] to obtain a running time of $\mathcal{O}^*(1.473^n)$.

* Corresponding author.
Email addresses: `t.brueggemann@math.utwente.nl` (Tobias Brueggemann), `kern@math.utwente.nl` (Walter Kern).

1 supported by the Netherlands Organization for Scientific Research (NWO) grant 613.000.225 (Local Search with Exponential Neighborhoods)

2 Local search

Let φ be an instance of 3-SAT given by a set $\mathcal{C}$ of clauses in variables $x_1, \ldots, x_n$. For $a \in \{0,1\}^n$ let $B_r(a) \subseteq \{0,1\}^n$ denote the set of 0-1 vectors with Hamming distance at most r from a. The currently fastest algorithms for 3-SAT are based on *local search*: First, a *covering code* of suitable *radius* $r \leq n$ is constructed, i.e. a set $A \subseteq \{0,1\}^n$ such that

$$\{0,1\}^n = \bigcup_{a \in A} B_r(a)$$

holds. Next we search for a truth assignment for φ in each $B_r(a)$, $a \in A$, separately. To make our paper self-contained, we briefly describe the basic idea for constructing a covering code and (to some extent) the local search within a given $B_r(a)$ as presented in Dantsin et. al. [1].

Covering codes. As $B_r := B_r(0)$ contains exactly

$$V(n,r) = \sum_{i=1}^{r} \binom{n}{i}$$

elements, a covering code $A \subseteq \{0,1\}^n$ of radius $r \leq n$ must necessarily satisfy

$$|A| \geq \frac{2^n}{V(n,r)}.$$

Covering codes of approximately this size indeed exist and can be constructed randomly: Choose

$$t = \frac{n 2^n}{V(n,r)}$$

elements from $\{0,1\}^n$ uniformly at random, resulting in a set $A \subseteq \{0,1\}^n$ of size $|A| \leq t$. The probability that a particular $a^* \in \{0,1\}^n$ is *not* covered by any $B_r(a)$, $a \in A$ is at most

$$P[a^* \text{ not covered}] = \left(1 - \frac{V(n,r)}{2^n}\right)^t \leq \mathrm{e}^{-n},$$

using $1 + x \leq \mathrm{e}^x$ for $x \in \mathbb{R}$. So the probability that A is *not* a covering code is at most $2^n \mathrm{e}^{-n}$, which tends to 0 as $n \to \infty$.

This procedure can be de-randomized by taking in each step a new code word $a \in \{0,1\}^n$ that is best possible in the sense that it covers as many as possible of the yet uncovered elements in $\{0,1\}^n$. Note, however, that this *greedy construction* takes $\mathcal{O}^*(2^n)$ per step and thus almost $\mathcal{O}(2^{2n}) = \mathcal{O}^*(4^n)$ in total (which is far too slow). Dantsin et. al. [1] therefore propose the following. Let

$K \in \mathbb{N}$ be a constant and assume w.l.o.g. that $n = Kn_0$ and $r = Kr$. Then construct a covering code $A_0 \subseteq \{0,1\}^{n_0}$ in time $\mathcal{O}(4^{n_0}) = \mathcal{O}^*\left(\sqrt[K]{4}^n\right)$ and take

$$A = \underbrace{A_0 \times \ldots \times A_0}_{K \text{ times}}$$

as a covering code for $\{0,1\}^n$. Proceeding this way, the time needed for constructing the covering code becomes negligible.

Local search. Assume we want to search for a truth assignment for φ in $B_r(a) \subseteq \{0,1\}^n$. We may assume w.l.o.g. that $a = 0$, i.e., we search in $B_r = B_r(0)$. (Interchange x_i with $\overline{x}_i$ if necessary.) If $a = 0$ is not a truth assignment for φ, there must exist a *false clause*, i.e. a clause $C \in \mathcal{C}$ that is false under $a = 0$, say $C = (x_i \vee x_{i'} \vee x_{i''})$. It then suffices to search for a truth assignment in $B_{r-1} \subseteq \{0,1\}^{n-1}$ w.r.t. each of the formulae

$$\varphi_1 = \varphi[x_i = 1], \ \varphi_2 = \varphi[x_{i'} = 1] \ \text{ and } \ \varphi_3 = \varphi[x_{i''} = 1],$$

obtained by fixing a variable as indicated in brackets. If necessary, we may even fix in addition some variables to zero, e.g., define $\varphi_1 := \varphi[x_i = 1], \varphi_2 := \varphi[x_{i'} = 1, x_i = 0]$ and $\varphi_3 := \varphi[x_{i''} = 1, x_i = 0, x_{i'} = 0]$.

Continuing this way, our search can be described by a *search tree* T_r, constructed by *branching on false clauses* (one false clause per node), as indicated in figure 1.

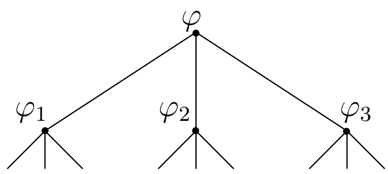

Fig. 1. The search tree T_r

Needless to say that we never branch to formulas $\varphi' = \varphi[x_i = 1, \ldots]$ that are obviously non-satisfiable because they contain an empty (non-satisfiable) clause. (For example, if $(\overline{x}_i) \in \mathcal{C}$, we would only branch to φ_2 and φ_3 in figure 1.) We denote the number of leaves of T_r by $|T_r|$ and refer to it as the *size* of T_r. Clearly,

$$|T_r| \leq 3^r \tag{1}$$

holds, an immediate consequence of the recursion $|T_r| \leq 3|T_{r-1}|$ (see figure 1). In case φ contains a false 2-clause $C \in \mathcal{C}$, then branching on C would yield $|T_r| \leq 2|T_{r-1}|$.

As pointed out in Dantsin et. al. [1], this simple argument already gives an $\mathcal{O}^*\left(\sqrt[2]{3}^n\right) \approx \mathcal{O}^*(1.7321^n)$ algorithm: Take $r = \frac{n}{2}$ and search $B_r(0)$ and $B_r(1)$

separately in time $\mathcal{O}^*(3^r) = \mathcal{O}^*\left(\sqrt[2]{3}^n\right)$ each.

Smaller search trees. The trivial bound (1) on the size of the search tree can be improved by a clever branching technique, as shown in Dantsin et. al. [1]: Assume that φ contains three pairwise disjoint false clauses $C = (x_i \vee x_{i'} \vee x_{i''}), C_1 = (x_j \vee x_{j'} \vee x_{j''})$ and $C_1' = (x_k \vee x_{k'} \vee x_{k''})$ and a (true) clause $(\overline{x}_i \vee \overline{x}_j \vee \overline{x}_k)$. We may then *branch along* $(\overline{x}_i \vee \overline{x}_j \vee \overline{x}_k)$, i.e. first branch on C at the root node φ, then branch on C_1 at $\varphi_1 = \varphi[x_i = 1]$ and finally branch on C_1' at $\varphi_1' = \varphi_1[x_j = 1] = \varphi[x_i = 1, x_j = 1]$. The resulting search tree is indicated in figure 2.

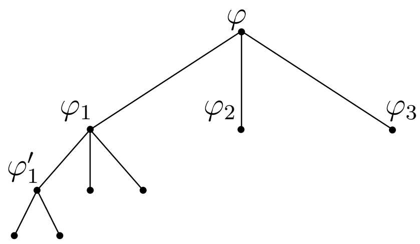

Fig. 2. Branching along $(\overline{x}_i \vee \overline{x}_j \vee \overline{x}_k)$

Note that the node corresponding to φ_1' has only two descendants because $\varphi[x_i = 1, x_j = 1, x_k = 1]$ is ruled out by the clause $(\overline{x}_i \vee \overline{x}_j \vee \overline{x}_k)$.

If a similar branching was possible also at φ_2 and φ_3, we would get a search tree satisfying a recursion

$$|T_r| \leq 6|T_{r-2}| + 6|T_{r-3}|. \tag{2}$$

Indeed, this is what Dantsin et. al. [1] show. Assuming inductively that $|T_k| \leq c\alpha^k$ holds for some constant $c > 0$, (2) implies that

$$|T_r| \leq \mathcal{O}(\alpha^r), \tag{3}$$

where $\alpha = \sqrt[3]{4} + \sqrt[3]{2} \approx 2.848$ is the largest root of $\alpha^3 - 6\alpha - 6 = 0$.

The main result of our paper slightly improves this bound as follows.

Theorem 1 *By branching on false clauses we can ensure that*

$$|T_r| \leq c\beta^r,$$

where $\beta = \frac{1+\sqrt{21}}{2} \approx 2.792$ *is the largest root of* $\beta^3 - 6\beta - 5 = 0$.

Running time. Let $\varrho < \frac{1}{2}$ and $r = \varrho n$. By Stirling's formula, the size of a covering code we construct is (up to a polynomial factor) bounded by

$$|A| = \mathcal{O}^*\left(\left[2\varrho^{\varrho}\left(1-\varrho\right)^{1-\varrho}\right]^n\right).$$

According to (3), the number of nodes in T_r is bounded by $n|T_r| = \mathcal{O}^*\left(|T_r|\right)$ and hence the total running time is thus bounded by

$$\mathcal{O}^*\left(|A||T_r|\right) = \mathcal{O}^*\left(\left[2(\alpha\varrho)^{\varrho}\left(1-\varrho\right)^{1-\varrho}\right]^n\right).$$

This expression is minimal for $\varrho \approx 0.26$, yielding the bound of $\mathcal{O}^*\left(1.481^n\right)$ in Dantsin et. al. [1].

Similarly, replacing α by β from Theorem 1, we obtain for $\varrho \approx 0.264$ an exact algorithm that runs in $\mathcal{O}^*\left(1.473^n\right)$.

3 Simple partial assignments

We will prove Theorem 1 by induction on $r \geq 0$. The basic idea is as follows. We first try to find a "simple truth assignment" by fixing as few as possible of the variables to $x_i = 1$ (exactly one per false clause). In case we do not succeed, we will exhibit a "good" clause to branch on.

We start by analyzing the structure of $\mathcal{C}$ and introduce some notation. Let $\mathcal{F} \subseteq \mathcal{C}$ denote the set of false clauses (at $x = 0$). We may assume w.l.o.g. that each $F \in \mathcal{F}$ is a 3-clause $F = (x_i \vee x_{i'} \vee x_{i''})$, because otherwise, as we observed already in section 2, branching on a false clause of length at most 2 yields the recursion $|T_r| \leq 2|T_{r-1}|$ and Theorem 1 follows by induction.

Secondly, we may assume that the clauses $F \subseteq \mathcal{F}$ are pairwise disjoint. Indeed, if $F = (x_i \vee x_{i'} \vee x_{i''})$ and $F' = (x_j \vee x_{j'} \vee x_{j''})$ intersect, say $x_i = x_j$, then branching on F at φ and on F' at $\varphi_2 = \varphi\left[x_{i'} = 1, x_i = 0\right]$ and $\varphi_3 = \varphi\left[x_{i''} = 1, x_i = 0, x_{i'} = 0\right]$ yields a search tree as indicated in figure 3.

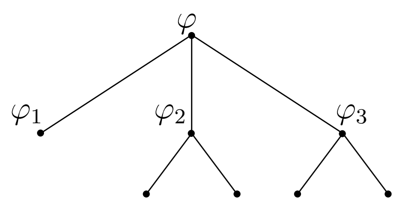

Fig. 3. Branching on intersecting clauses

The corresponding recursion is $|T_r| \leq |T_{r-1}| + 4|T_{r-2}|$ and, again, Theorem 1 follows inductively.

Thus in what follows, we may (and will) assume that φ is *regular* in the sense that $\mathcal{F}$ consists of pairwise disjoint 3-clauses. We often identify such a clause $F = (x_i \vee x_{i'} \vee x_{i''}) \in \mathcal{F}$ with its corresponding set of variables $F = \{x_i, x_{i'}, x_{i''}\}$ or with the corresponding set of elements (indices) $F = \{i, i', i''\}$. The elements i, i', i'' covered by a false clause $F \in \mathcal{F}$ are *neighbors* of each other. The elements $i \in \{1, \ldots, n\}$ covered by false clauses are called *internal* elements. We denote by $I = I_\varphi \subseteq \{1, \ldots, n\}$ the set of internal elements. The elements in $\{1, \ldots, n\} \setminus I$ are called *external*.

Recall that, as mentioned above, we first try to construct a truth assignment for φ by fixing some variable to $x_i = 1$ (one per false clause in $\mathcal{F}$). In general, fixing some variables, say $x_{i_1} = 1, \ldots, x_{i_t} = 1$, results in a new formula $\varphi' = \varphi\,[x_{i_1} = 1, \ldots, x_{i_t} = 1]$ whose clauses are obtained from the clauses in $\mathcal{C}$ by fixing $x_{i_1} = 1, \ldots, x_{i_t} = 1$ in each clause. This way each clause $C \in \mathcal{C}$ *reduces* to a corresponding clause $C' = C\,[x_{i_1} = 1, \ldots, x_{i_t} = 1] \in \mathcal{C}' = \mathcal{C}_{\varphi'}$. We say that C reduces to $C' = 1$ (a fixed *true* clause) if C contains some $x_i, i \in \{i_1, \ldots, i_t\}$. Similarly, C reduces to $C' = 0$, the empty (fixed *false*) clause if C contains only negated literals $\overline{x}_i, i \in \{i_1, \ldots, i_t\}$. Note that $C \in \mathcal{C}$ reduces to $C' \in \mathcal{F}_{\varphi'}$ if and only if all negated variables $\overline{x}_i$ in C are indexed by $i \in \{i_1, \ldots, i_t\}$.

Definition 2 (Simple partial assignment) *A* simple partial assignment *(SPA) of φ is a formula*

$$\varphi' = \varphi\,[x_{i_1} = 1, \ldots, x_{i_t} = 1]$$

that fixes at most one variable per false clause to $x_i = 1$, without creating any new false clauses, i.e., such that the following hold:

$$(S1)\ \{i_1, \ldots, i_t\} \subseteq I$$

$$(S2)\ |F \cap \{i_1, \ldots, i_t\}| \leq 1 \quad \textit{for each } F \in \mathcal{F}_\varphi$$

$$(S3)\ \mathcal{F}_{\varphi'} \subseteq \mathcal{F}_\varphi.$$

There are certain clauses in $\mathcal{C} \setminus \mathcal{F}$ that are "irrelevant" in the sense that they never reduce to a false clause by fixing $x_{i_1} = 1, \ldots, x_{i_t} = 1$ as long as (S1) and (S2) hold: A clause $C \in \mathcal{C} \setminus \mathcal{F}$ is called *externally true* if $C = (\overline{x}_l \vee \ldots)$ with $l \in \{1, \ldots, n\} \setminus I$ being external. A clause $C \in \mathcal{C} \setminus \mathcal{F}$ is *internally true* if $C = (\overline{x}_i \vee \overline{x}_j \vee \ldots)$ with $i, j \in I$ being neighbors. Clearly, an externally and/or internally true $C \in \mathcal{C}$ reduces to a true clause $C' \in \mathcal{C}_{\varphi'}$ whenever $\varphi' = \varphi\,[x_{i_1} = 1, \ldots, x_{i_t} = 1]$ satisfies (S1) and (S2). We let $\mathcal{E} \subseteq \mathcal{C} \setminus \mathcal{F}$ denote the set of externally and/or internally true clauses.

The remaining set $\mathcal{R} = \mathcal{C} \setminus (\mathcal{F} \cup \mathcal{E})$ is called the set of *relevant* clauses. We will use these clauses to guide our search process, i.e., we will construct T_r by "branching along relevant clauses" as indicated already in section 2. We first treat the so-called "pure case", where each relevant clause contains only negated variables. This is the case where the bound (2) is tight in the approach of Dantsin et. al. [1].

4 The pure case

A regular φ is called *pure* if every $R \in \mathcal{R} = \mathcal{R}_\varphi$ contains only negated variables. Throughout this section, we assume that φ is (regular and) pure and hence so is any *SPA* φ' of φ.

We say that $R \in \mathcal{R}$ *intersects* $F = (x_i \vee x_{i'} \vee x_{i''}) \in \mathcal{F}$ if R contains one of $\overline{x}_i, \overline{x}_{i'}, \overline{x}_{i''}$. Recall that R cannot contain two of these since it would then be internally true. To motivate the following, consider an *SPA* $\varphi' = \varphi[x_i = 1]$ of φ. Any $R \in \mathcal{R}$ reduces to a true clause in φ' due to (S3). If R intersects the unique false clause $F = (x_i \vee x_{i'} \vee x_{i''})$ covering i, then either R becomes an externally true clause in φ' (namely when R contains either $\overline{x}_{i'}$ or $\overline{x}_{i''}$) or R reduces to an "even more" relevant clause $R' \in \mathcal{R}_{\varphi'}$. For example, $R = (\overline{x}_i \vee \overline{x}_j \vee \overline{x}_k)$ reduces to $R' = (\overline{x}_j \vee \overline{x}_k) \in \mathcal{R}_{\varphi'}$.

Let $\varphi' = \varphi[x_{i_1} = 1, \ldots, x_{i_t} = 1]$ be an *SPA* of φ and let $F_{i_1}, \ldots, F_{i_t} \in \mathcal{F}$ be the unique clauses covering $i_1, \ldots, i_t$, resp. We say that φ' is *proper* if every $R \in \mathcal{R}$ that intersects some $F \in \{F_{i_1}, \ldots, F_{i_t}\}$ reduces to an externally true clause $R' \in \mathcal{R}_{\varphi'}$ (so R must contain some $\overline{x}_i$ with $i \in I$ being a neighbor of an element in $\{i_1, \ldots, i_t\}$).

Lemma 3 *For any two proper* SPA*'s φ' and φ'' of φ there exist a proper* SPA *$\overline{\varphi}$ with $\mathcal{F}_{\overline{\varphi}} = \mathcal{F}_{\varphi'} \cap \mathcal{F}_{\varphi''}$.*

PROOF. Let $\mathcal{F}_\varphi = \{F_1, \ldots, F_f\}$ with $F_i = (x_i \vee x_{i'} \vee x_{i''}), i = 1, \ldots, f$, and assume that, say,

$$\begin{aligned}\varphi' &= \varphi[x_1 = 1, \ldots, x_s = 1],\\ \varphi'' &= \varphi[x_{s+1} = 1, \ldots, x_t = 1, x_{j_1} = 1, \ldots, x_{j_l} = 1],\end{aligned}$$

with $j_1, \ldots, j_l$ being covered by $F_1, \ldots, F_s$. We define $\overline{\varphi}$ as

$$\overline{\varphi} = \varphi[x_1 = 1, \ldots, x_t = 1].$$

Clearly, $\overline{\varphi}$ satisfies (S1) and (S2). We verify (S3) by showing that any $R \in \mathcal{R}_\varphi$ reduces to a true clause $\overline{R} \in \mathcal{R}_{\overline{\varphi}}$. Indeed, we will show that any $R \in \mathcal{R}_\varphi$ intersecting $F_1 \cup \ldots \cup F_t$ reduces (even) to an externally true clause in $\overline{\varphi}$, thus showing at the same time that $\overline{\varphi}$ is proper.

Let $R \in \mathcal{R}_\varphi$ intersect $F_i \in \{F_1, \ldots, F_t\}$. If $i \leq s$, then R reduces to an externally true clause in φ' (since φ' is proper) and hence to an externally true clause in $\overline{\varphi}$. On the other hand, if R does not intersect $F_1 \cup \ldots \cup F_s$ (but $F_{s+1} \cup \ldots \cup F_t$), then R reduces to the same clause in $\overline{\varphi}$ as in φ''. So again, the claim follows, as φ'' is proper. $\square$

Lemma 3 is useful in constructing proper *SPA*'s $\overline{\varphi}$ with smaller and smaller sets $\mathcal{F}_{\overline{\varphi}}$. Ideally, we would like to arrive at $\mathcal{F}_{\overline{\varphi}} = \emptyset$, in which case $\overline{\varphi}$ defines a truth assignment for φ. To describe our search process for proper *SPA*'s of φ, we introduce the notion of "b-blocking".

Definition 4 (b-blocking) *Consider a clause $R \in \mathcal{R}_\varphi$.*

(1) If $R = (\overline{x}_i \vee \ldots)$ then R 0-blocks *$i \in I$.*

(2) If $R = (\overline{x}_i \vee \ldots)$ has length at most two, then R b-blocks i for all $b \geq 0$.

(3) If $R = (\overline{x}_i \vee \ldots)$ has length three, i.e. $R = (\overline{x}_i \vee \overline{x}_j \vee \overline{x}_k)$ for some $j, k \in I$ with neighbors j', j'' and k', k'', resp., then R b-blocks *i, if each of j', j'', k' and k'' is $(b-1)$-blocked by some clause in $\mathcal{R}_{\varphi[x_i=1]}$.*

We call $i \in I$ b-blocked by *$\mathcal{R}_\varphi$ if there exists some $R \in \mathcal{R}_\varphi$ (of arbitrary length) that b-blocks i.*

Example. Assume $\mathcal{F} = \mathcal{F}_\varphi$ consists of three clauses $(x_i \vee x_{i'} \vee x_{i''})$, $(x_j \vee x_{j'} \vee x_{j''})$ and $(x_k \vee x_{k'} \vee x_{k''})$. Furthermore, assume that $\mathcal{R} = \mathcal{R}_\varphi$ consists of three clauses $R = (\overline{x}_i \vee \overline{x}_j \vee \overline{x}_k)$, $R' = (\overline{x}_{i'} \vee \overline{x}_{j'} \vee \overline{x}_{k'})$ and $R'' = (\overline{x}_{i''} \vee \overline{x}_{j''} \vee \overline{x}_{k''})$. Then each element in $I = I_\varphi$ is 0-blocked, but none is 1-blocked. Indeed, consider, e.g. $\varphi' = \varphi[x_i = 1]$. Then R' and R'' reduce to externally true clauses in φ'. So $\mathcal{R}_{\varphi'} = \{(\overline{x}_j \vee \overline{x}_k)\}$ and, for example, j' is not 0-blocked by $\mathcal{R}_{\varphi'}$. For this reason (see the general construction described below), it is easy to find a truth assignment for φ (e.g. by setting $x_i = 1, x_{j'} = 1, x_{k'} = 1$).

For $b \geq 0$, we let $U_b \subseteq I$ denote the set of elements $i \in I$ that are not b-blocked by $\mathcal{R}_\varphi$. We call these elements *b-unblocked* (by $\mathcal{R}_\varphi$). Let $\mathcal{U}_b \subseteq \mathcal{F}$ denote the set of false clauses $F \in \mathcal{F}$ that cover some b-unblocked $i \in I$. We also call these false clauses *b-unblocked*. By definition, we have $U_0 \subseteq U_1 \subseteq \ldots$ and also $\mathcal{U}_0 \subseteq \mathcal{U}_1 \subseteq \ldots$

Note that we can compute the set $U_b \subseteq I$ for $b \geq 0$ along with a b-blocking clause $R \in \mathcal{R}_\varphi$ for every $i \in I \setminus U_b$ in time $\mathcal{O}\left(n^{b+3}\right)$. Indeed, for $b = 0$, it suffices to scan the $\mathcal{O}(n^3)$ clauses in $\mathcal{R} = \mathcal{R}_\varphi$.

We proceed by induction on $b \geq 0$. Thus assume $b \geq 1$ and let $i \in I$ and $\varphi' = \varphi[x_i = 1]$. By induction, the set $U'_{b-1} \subseteq I_{\varphi'}$ of elements that are $(b-1)$-unblocked by $\mathcal{R}_{\varphi'}$ can be computed in time $\mathcal{O}\left(n^{b+2}\right)$. We then check for each of the $\mathcal{O}(n^2)$ 3-clauses $R = (\overline{x}_i \vee \overline{x}_j \vee \overline{x}_k)$ whether some element from $\{j', j'', k', k''\}$ is in U'_{b-1} or not. This takes (at most) $\mathcal{O}(n^2)\,\mathcal{O}(n) = \mathcal{O}(n^3)$ in total. Hence the total time needed to check $i \in I$ is $\mathcal{O}\left(n^{b+2}\right) + \mathcal{O}(n^3) = \mathcal{O}\left(n^{b+2}\right)$ and the claim follows.

The next result is crucial:

Theorem 5 *For each $b \geq 0$ there exists a proper* SPA *φ' of φ with $\mathcal{F}_{\varphi'} \subseteq \mathcal{F}_\varphi \setminus \mathcal{U}_b$.*

PROOF. By induction on $b \geq 0$. Assume first that $b = 0$. Let $F \in \mathcal{U}_0$, say $F = (x_i \vee x_{i'} \vee x_{i''})$ with $i \in U_0$. Then $\varphi' = \varphi[x_i = 1]$ is, by definition of U_0, a proper *SPA* and $\mathcal{F}_{\varphi'} = \mathcal{F} \setminus \{F\}$. The claim now follows from Lemma 3 and induction.

Next assume $b \geq 1$. Let $F = (x_i \vee x_{i'} \vee x_{i''}) \in \mathcal{U}_b$ with $i \in U_b$. As before, due to Lemma 3, it suffices to show that there is a proper *SPA* φ' of φ with $\mathcal{F}_{\varphi'} \subseteq \mathcal{F} \setminus \{F\}$. Let $\varphi_1 := \varphi[x_i = 1]$. Clearly, φ_1 is an *SPA* of φ. (Otherwise there were a clause $(\overline{x}_i) \in \mathcal{R}$. But such a clause would b-block i contradicting $i \in U_b$.) Let $U^1_{b-1} \subseteq I_{\varphi_1}$ and $\mathcal{U}^1_{b-1} \subseteq \mathcal{F}_{\varphi_1}$ denote the set of elements in I_{φ_1} resp. clauses in $\mathcal{F}_{\varphi_1}$ that are $(b-1)$-unblocked by $\mathcal{R}_{\varphi_1}$. By induction on b, there is a proper *SPA* φ'_1 of φ_1 with $\mathcal{F}_{\varphi'_1} \subseteq \mathcal{F}_{\varphi_1} \setminus \mathcal{U}^1_{b-1}$. We claim that actually φ'_1 is a proper *SPA* of φ. Clearly, φ'_1 is an *SPA* of φ (as any *SPA* of an *SPA* is an *SPA*).

To show that φ'_1 is proper, assume that

$$\varphi'_1 = \varphi_1[x_{i_1} = 1, \ldots, x_{i_t} = 1] = \varphi[x_i = 1, x_{i_1} = 1, \ldots, x_{i_t} = 1]$$

and let $F_i, F_{i_1}, \ldots, F_{i_t} \in \mathcal{F}$ denote the unique clauses in $\mathcal{F}$ covering $i, i_1, \ldots, i_t$, resp. Let $R \in \mathcal{R}_\varphi$ intersect $F_i \cup F_{i_1} \cup \ldots \cup F_{i_t}$. We are to show that R reduces to an externally true clause R'_1 in φ'_1.

Assume first that R intersects $F_i = (x_i \vee x_{i'} \vee x_{i''})$. If R contains either $\overline{x}_{i'}$ or $\overline{x}_{i''}$, the claim is obviously true. Thus assume $R = (\overline{x}_i \vee \ldots) \in \mathcal{R}$. Since $i \in U_b$, R must be a 3-clause $R = (\overline{x}_i \vee \overline{x}_j \vee \overline{x}_k)$. So R reduces to $R_1 = (\overline{x}_j \vee \overline{x}_k)$ in φ_1. As $i \in U_b$, at least one neighbor of either j or k is in $\mathcal{U}^1_{b-1}$, i.e., either $F_j = (x_j \vee x_{j'} \vee x_{j''})$ or $F_k = (x_k \vee x_{k'} \vee x_{k''})$ is in $\mathcal{U}^1_{b-1} \subseteq \mathcal{F}_{\varphi_1}$. So $\mathcal{F}_{\varphi'_1} \subseteq \mathcal{F}_{\varphi_1} \setminus \mathcal{U}^1_{b-1}$ implies that φ'_1 fixes at least one variable from either F_j or F_k to 1, i.e., either F_j or F_k occurs in $\{F_{i_1}, \ldots, F_{i_t}\}$. Thus $R_1 = (\overline{x}_j \vee \overline{x}_k)$ reduces to an externally true clause R'_1 in φ'_1 (as φ'_1 is a proper *SPA* of φ) and hence so does R.

Next assume that R does not intersect F_i. Then $R \in \mathcal{R}_\varphi$ and the claim follows immediately from the fact that φ_1' is a proper *SPA* of φ_1. $\square$

Corollary 6 *If* $\mathcal{U}_b = \mathcal{F}$ *for some* $b \geq 0$*, then* φ *has a truth assignment that can be computed in time* $\mathcal{O}\left(n^{b+3}\right)$*.* $\square$

We are now ready to prove Theorem 1 in the pure case. Let $b \geq 0$ (to be specified later on) and assume there exists some $F = (x_i \vee x_{i'} \vee x_{i''}) \in \mathcal{F} \setminus \mathcal{U}_b$. (Otherwise a truth assignment exists and there is no need to construct a search tree.) We then branch on F at the root node φ of T_r, branching to $\varphi_1 = \varphi\,[x_i = 1]$, $\varphi_2 = \varphi\,[x_{i'} = 1]$ and $\varphi_3 = \varphi\,[x_{i''} = 1]$.

Since $F \notin \mathcal{U}_b$, the elements i, i' and i'' are b-blocked by $\mathcal{R}_\varphi$. Let $R \in \mathcal{R}$ b-block i. If R is a 1-clause, i.e. $R = (\overline{x}_i)$, then the subtree rooted at φ_1 is empty. If R is a 2-clause, i.e. $R = (\overline{x}_i \vee \overline{x}_j)$, then branching on $F_1 = (x_j \vee x_{j'} \vee x_{j''})$ at φ_1 yields a search tree as indicated in figure 4. Thus we obtain a recursion $|T_r| \leq 2|T_{r-1}| + 2|T_{r-2}|$ and Theorem 1 follows inductively.

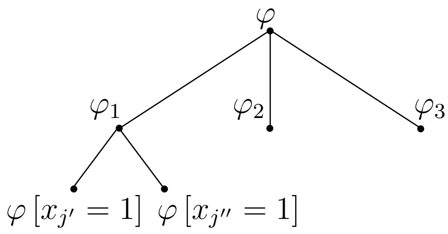

Fig. 4. When i is blocked by a 2-clause.

Hence assume that $R = (\overline{x}_i \vee \overline{x}_j \vee \overline{x}_k)$ b-blocks i. In this case we obtain a search tree as in figure 2 by branching on F_1 at φ_1 and on $F_1' = (x_k \vee x_{k'} \vee x_{k''})$ at $\varphi_1' = \varphi_1\,[x_j = 1]$.

Let us denote the size of the subtree rooted at φ_1 by $|T_{r-1}^{(b)}|$ to indicate that $\varphi_1 = \varphi\,[x_i = 1]$ is obtained by fixing x_i with i being b-blocked by $\mathcal{R}_\varphi$. We thus get the recursion

$$|T_{r-1}^{(b)}| \leq 2|T_{r-2}^{(b-1)}| + 2|T_{r-3}|, \tag{4}$$

as both j' and j'' are $(b-1)$-blocked by $\mathcal{R}_{\varphi_1}$. Furthermore, of course $|T_r| \leq 3|T_{r-1}^{(b)}|$ holds, since also i' and i'' are b-blocked by $\mathcal{R}_\varphi$.

Iterating (4), we obtain for $r \geq b + 2$

$$
\begin{aligned}
|T_{r-1}^{(b)}| &\leq 2\left[2|T_{r-3}^{(b-2)}| + 2|T_{r-2}|\right] + 2|T_{r-3}| \\
&\vdots \\
&\leq 2^b|T_{r-b-2}| + \ldots + 2|T_{r-3}| + 2^b|T_{r-b-1}^{(0)}| \\
&\leq 2^b|T_{r-b-2}| + \ldots + 2|T_{r-3}| + 2^b|T_{r-b-1}|,
\end{aligned}
$$

where the last inequality follows from $|T_k^{(0)}| \leq |T_k|$.

Assuming inductively that $|T_k| \leq c\beta^k$ for $k < r$, we get

$$
\begin{aligned}
|T_r| &\leq 3|T_r^{(b-1)}| \\
&\leq 3c\beta^r \left[\frac{2^b}{\beta^{b+1}} + \sum_{k=1}^{b} \frac{2^k}{\beta^{k+2}}\right] \\
&= 3c\beta^r \left[\frac{2^b}{\beta^{b+1}} + \frac{2 - 2^{b+1}\beta^{-b}}{\beta^3 - 2\beta^2}\right].
\end{aligned}
$$

For β as in Theorem 1 and $b \geq 4$ we have for the term in the brackets

$$
\frac{2^b}{\beta^{b+1}} + \frac{2 - 2^{b+1}\beta^{-b}}{\beta^3 - 2\beta^2} < \frac{1}{3}.
$$

So $|T_r| \leq c\beta^r$ follows inductively.

5 The general case

In the general case, when φ is regular, but not necessarily pure, we proceed as follows. As in section 4 we say that $i \in I$ is *blocked* by $R \in \mathcal{R}$ if $R = (\overline{x}_i \vee \ldots)$. Let $U \subseteq I$ denote the elements that are *unblocked*, i.e. not blocked by any $R \in \mathcal{R}$ and let $\mathcal{U} \subseteq \mathcal{F}$ denote the set of clauses $F \in \mathcal{F}$ that contain some $i \in U$.

If $\mathcal{F} = \mathcal{U}$, a truth assignment is easily obtained by fixing exactly one unblocked i per clause $F \in \mathcal{F}$ to $x_i = 1$. Hence assume $\mathcal{F}^* = \mathcal{F} \setminus \mathcal{U} \neq \emptyset$ in what follows and let $I^* \subseteq I$ denote the elements covered by clauses in $\mathcal{F}^*$. We distinguish two cases:

Case 1. There exists an element $i \in I^*$ that is blocked by some $R \in \mathcal{R}$ which is *not* of the form $R = (\overline{x}_i \vee \overline{x}_j \vee \overline{x}_k)$ with $j, k \in I$.

In this case we branch on the unique clause $F \in \mathcal{F}^*$ covering i. Branching along blocking clauses as in section 4 then proves Theorem 1 inductively.

Indeed, assume that i is blocked by a clause of type $R = (\overline{x}_i \vee \overline{x}_j \vee x_k)$ with $j, k \in I$. Note that j is then covered by a clause $F_1 \neq F$ since otherwise R were internally true. We then branch on $F_1 = (x_j \vee x_{j'} \vee x_{j''})$ at $\varphi_1 = \varphi\,[x_i = 1]$ and on the false 1-clause (x_k) at $\varphi_1' = \varphi_1\,[x_j = 1]$. The resulting search tree then differs from the one in figure 2 in that one of the two subtrees of φ_1' is eliminated, yielding a recursion

$$|T_r| \leq 6|T_{r-2}| + 5|T_{r-3}|,$$

assuming the "worst case scenario", where both i' and i'' are blocked by 3-clauses with three negated variables each. In this case, Theorem 1 follows inductively (by choice of β). It is straightforward to verify that this is indeed the worst case scenario for case 1).

Case 2. All blocking clauses for elements in I^* have three negated variables each.

In this case, let $\mathcal{R}^*$ denote the set of clauses $R = (\overline{x}_i \vee \overline{x}_j \vee \overline{x}_k) \in \mathcal{R}$ with $i, j, k \in I^*$. Let φ^* denote the formula defined by the clauses $\mathcal{C}^* = \mathcal{F}^* \cup \mathcal{R}^*$. In particular, φ^* is pure. Let $\mathcal{U}_b^* \subseteq \mathcal{F}$ denote the clauses in $\mathcal{F}^*$ that are b-unblocked by $\mathcal{R}_{\varphi^*}$.

Lemma 7 *If $\mathcal{U}_b^* = \mathcal{F}^*$, then φ has a truth assignment.*

PROOF. By Theorem 5, φ^* has a proper *SPA*

$$\varphi' = \varphi^*\,[x_{i_1} = 1, \ldots, x_{i_t} = 1]$$

defining a truth assignment for φ^* (see also Corollary 6).

To define a truth assignment for φ, pick elements $j_1, \ldots, j_s \in U$, one from each clause in $\mathcal{U}$, and let

$$\overline{\varphi} = \varphi\,[x_{i_1} = 1, \ldots, x_{i_t} = 1, x_{j_1} = 1, \ldots, x_{j_s} = 1]\,.$$

We claim that $\overline{\varphi}$ defines a truth assignment for φ, i.e. that $\mathcal{F}_{\overline{\varphi}} = \emptyset$. Assume to the contrary that $R \in \mathcal{R}$ reduces to a false clause in $\overline{\varphi}$. Clearly, $R \notin \mathcal{R}^*$ must hold, since any clause in $\mathcal{R}^*$ reduces to an (externally) true clause in φ' and hence to a true clause in $\overline{\varphi}$. However, if $R \in \mathcal{R} \setminus \mathcal{R}^*$, case 2) implies that $R = (\overline{x}_i \vee \ldots)$ with $i \in I \setminus I^*$. In particular, i is blocked by R and so $i \notin \{j_1, \ldots, j_s\}$. Thus, R reduces to a true clause in $\overline{\varphi}$. $\square$

Due to Lemma 7, we may assume w.l.o.g. that $\mathcal{U}_b^* \neq \mathcal{F}^*$. Thus we may choose $F \in \mathcal{F}^* \setminus \mathcal{U}_b^*$ for branching at the root node φ of T_r and continue branching on false clauses in $\mathcal{F}^*$ along clause $\mathcal{R}^*$ as if we were searching for a truth assignment for φ^*. Theorem 1 thus follows inductively also in the general case.

References

[1] E. Dantsin, A. Goerdt, E.A. Hirsch, R. Kannan, J. Kleinberg, C. Papadimitriou, O. Raghavan, U. Schoening [2002]: *A deterministic* $(2-2/(k+1))^n$ *algorithm for k-SAT based on local search.* In: Theoretical Computer Science 289 (2002), 69-83. Elsevier Science B.V.

[2] T. Hofmeister, U. Schoening, R. Schuler, O. Watanabe [2002]: *A Probabilistic 3-SAT Algorithm Further Improved.* In: H. Alt, A. Ferreira (Eds.): STACS 2002, LNCS 2285, 192-202. Springer-Verlag Berlin Heidelberg.

[3] U. Schoening [2002]: *A Probabilistic Algorithm for k-SAT Based on Limited Local Search and Restart.* In: Algorithmica 32 (2002), 615-623. Springer-Verlag New York Inc.