ISSN 1342-2812

Research Reports on Mathematical and Computing Sciences

A Message Passing Algorithm for MAX2SAT

Osamu Watanabe and Masaki Yamamoto

Dec. 2005, C–216

Department of
Mathematical and
Computing Sciences
Tokyo Institute of Technology

SERIES C: Computer Science

A Message Passing Algorithm for MAX2SAT

Osamu Watanabe*and Masaki Yamamoto
Dept. of Math. and Comp. Sci., Tokyo Inst. of Technology, Japan
(watanabe@is.titech.ac.jp)

Research Report C-216

Abstract

We propose a simple and deterministic algorithm for solving MAX2SAT, which runs $O(n(n+m))$ time where n and m are respectively the number of variables and clauses. For discussing its average case performance, we propose one natural "planted solution model"; a way to generate a MAX2SAT instance under a certain distribution defined by parameters p and r. We first show that if $p = \Omega(\ln^2 n/n)$ and $p \geq 9r$, then with high probability the planted solutions (there are four planted solutions) are only the optimal solution, unsatisfying $3rn^2$ clauses out of (on average) $2pn^2 + 8rn^2$ clauses. Then under this planted solution model we show that, for some constant $\epsilon > 0$, our algorithm yields one of the planted solutions with high probability if $p - r \geq n^{-1/2+\epsilon}$.

1 Introduction

Motivated by the recent work [OW05] on deriving a simple message passing algorithm for graph partitioning problems, we consider in this paper the MAX2SAT problem, one of the well-know NP-hard optimization problems, and propose a simple deterministic algorithm that runs in $O(n(n+m))$ time for a given 2-CNF formula with n variables and m clauses. For analyzing its average case performance, we propose one probabilistic model, a variation of the planted solution model [JS98] that has been used for the same purpose for the Graph Bisection problem. Then we prove that the algorithm produces one of the optimal solutions with high probability if instances are generated by our planted solution model with probability parameters satisfying a certain condition.

We introduce some notations and state our result more precisely. We will use standard notions and notations on propositional Boolean formulas and graphs without explanation. For Boolean formulas, the size parameter n determines the number of variables. Throughout this paper, for simplicity, we assume that a formula has even number of variables, and let $2n$ denote the number of variables of a given formula. On the other hand, we use m to denote the number of clauses of a given formula. We use $x_1, ..., x_{2n}$ for denoting Boolean variables. Since we consider only *2CNF formulas*, formulas defined as a conjunction of clauses of two literals, each clause is specified as $(x_i \vee x_j)$, $(x_i \vee \overline{x}_j)$, $(\overline{x}_i \vee x_j)$, or $(\overline{x}_i \vee \overline{x}_j)$, for $1 \leq i \leq j \leq 2n$, where x_i and $\overline{x}_i$ are called *positive* and *negative* literals respectively. A formula may contain

*Supported in part by a Grant-in-Aid for Scientific Research on Priority Areas "New Horizons in Computing" 2004-2006.

the same clause more than once. We assume that formulas are encoded appropriately. Now what follows is the description of the MAX2SAT problem.

MAX2SAT problem

Input: A 2CNF formula over Boolean variables $x_1, ..., x_{2n}$.
Task Find an assignment to $x_1, ..., x_{2n}$ maximizing the number of satisfied clauses.

In this paper, we reduce the MAX2SAT problem to some variable of graph partitioning problems. Consider any 2-CNF formula F with $2n$ variables. Corresponding to this F, consider the following directed graph $H_F = (U, A)$: A vertex set U is defined by $U \stackrel{\mathrm{def}}{=} U_1 \cup \widetilde{U}_1 \cup U_2 \cup \widetilde{U}_2$, where

$$\begin{array}{llll} U_1 \stackrel{\mathrm{def}}{=} \{1, \ldots, n\}, & \widetilde{U}_1 \stackrel{\mathrm{def}}{=} \{-1, \ldots, -n\}, \\ U_2 \stackrel{\mathrm{def}}{=} \{n+1, \ldots, 2n\}, & \widetilde{U}_2 \stackrel{\mathrm{def}}{=} \{-(n+1), \cdots, -2n\}. \end{array}$$

That is, $U = \{-2n, ..., -1, 1, ..., 2n\}$. An edge set A consists of directed edges (i, j) corresponding clauses $(\ell_{|i|} \to \ell_{|j|})$ in F. For example, suppose that F has a clause $(x_2 \vee \overline{x}_5)$, which is equivalent to both $(\overline{x}_2 \to \overline{x}_5)$ and $(x_5 \to x_2)$. Then corresponding to this clause, two edges $(-2, -5)$ and $(5, 2)$ are put into A.

Now consider any assignment a for F, which is considered as a mapping $\{1, ..., 2n\}$ to $\{-1, +1\}$; that is, $a(i) = +1$ and $a(i) = a(i) - 1$ means to assign respectively true and false to the variable x_i. This assignment defines the assignment t to vertices of H_F in the following natural way: for any $i \in U$, $t(i) = a(i)$ if $i > 0$ and $t(i) = -a(i)$ if $i < 0$. Consider the partition of U by the assignment t. Then it is easy to see that cut edges from a true vertex to a false vertex corresponds to clauses unsatisfied by the assignment a. (Precisely speaking, two cut edges corresponds to one unsatisfied clause.) Consider the converse. We say that an assignment t to U is consistent if $t(i) = -t(-i)$ for all $i \in U$. Then it is clear that any consistent assignment t defines some assignment a to F. Therefore, finding a consistent assignment to vertices (or a partition) that minimizes the number of cut edges from true to false vertices is to find the optimal assignment for F.

For this problem, we consider a simple message passing type algorithm[1] which computes $b(i)$, "belief" that a vertex i is assigned true. The computation is based on the following simple heuristic idea: if a graph H_F has an directed edge (j, i) and $b(i)$ is negative, i.e., we believe (for some reason) that this vertex i is assigned false, then (in order to minimize unsatisfied cut edges), we had better send a message to the vertex j, suggesting the false assignment to j. At each iteration, such messages are sent *in parallel* from vertices with negative beliefs to their connected vertices. Then at each vertex, its belief is updated based on the received messages. Note that we need to compute a consistent assignment; thus, beliefs should be consistent between i and $-i$, which is achieved by simply forcing $b(i) = -b(-i)$ at each step. After several iterations, if all beliefs get stabled, then we can determine the assignment to each vertex i based on the sign of $b(i)$. It is not so hard to see that the algorithm can be implemented so that each iteration needs $O(n + m)$ time by the standard unit cost RAM model.

Although simple, we think that this algorithm works quite well on average. For justifying our intuition, we introduce one scenario for discussing the average case performance of

[1]From the term "belief", one may expect some relation to the Perl's belief propagation algorithm [Pea88]. Our algorithm, though motivated by the one [OW05] that is indeed derived from the Perl's belief propagation algorithm, has nothing to do with it.

algorithms for the MAX2SAT problem, and prove that our algorithm indeed yields a correct answer with high probability. For the average case scenario, we propose some planted solution model, which has been proposed [Yam05] as a method for generating test instances for MAX2SAT algorithms. Also it is regarded as a variation of the the planted solution model [JS98] that has been used for the same purpose for the Graph Bisection problem. In general, a model for an average case scenario is a way to define a distribution of problem instances, and a planted solution model defines is by providing a way to generate problem instances. Intuitively, under a planted solution model, a target solution — which is called a *planted solution* — is first determined (or generated randomly), and a problem instance is generated randomly consistent with this solution. In our situation, we first fix one assignment, and then generate clauses independently following a certain distribution; roughly, clauses satisfied with the assignment are generated with probability p, and clauses unsatisfied with the assignment are generated with probability r. More precise description of the generation procedure is stated again in terms of graphs; see the next section for the details. Intuitively, if $p >> r$, then one can expect that the planted solution is the optimal assignment, satisfying $O(pn^2)$ clauses and unsatisfying $O(rn^2)$ clauses. In fact, we show the following theorem.

Theorem 1. For any probability parameters p and r satisfying $p = \Omega(\ln^2 n/n)$ and $p \geq 9r$, consider a randomly generated formula F under our planted solution model for the MAX2SAT problem. Then with high probability (i.e., with probability $1 - o(1)$ w.r.t. n), four planted solutions are optimal solutions for F; futhermore, there are no other optimal solutions.

Under this planted solution model (with probability parameters satisfying the above) the success probability of the algorithm is computed as the probability that it yields one of the planted solutions for randomly generated formulas. For some technical reason, we modify the algorithm so that it terminates after two iterations; see Section 3 for some other detail modifications. Even with such a strong time bound, we can show that the algorithm yields a correct answer (i.e., one of the planted solutions) with high probability if $p-r$ is large enough, which is stated more formally as follows.

Theorem 2. For any probability parameters p and r satisfying $p - r \geq n^{-1/2+\epsilon_p}$ for some constant $\epsilon_p > 0$, consider the execution of the algorithm (with MAXSTEP = 2) on a randomly generated formula F under our planted solution model for the MAX2SAT problem. Then with high probability (i.e., with probability $1 - o(1)$ w.r.t. n), it yields one of the planted solutions for F.

2 A Planted Solution Model for MAX2SAT

We explain our average case scenario or probability model, more specifically, a way of generating 2-CNF formulas for MAX2SAT instances. This model is regarded as a "planted solution model" for the MAX2SAT problem.

For a given $n \geq 1$, we discuss the way of generating a 2-CNF formula over $2n$ variables $X = \{x_1, \ldots, x_{2n}\}$. The outline of our generation is as follows. First generate a directed graph, and then transform the graph into a 2-CNF formula. We first explain the graph generation[2].

[2] The explanation here is for the simplified version, which determine U_1, $\widetilde{U}_1$, U_2, and $\widetilde{U}_2$ uniquely from n. In more general, we first generate an equal size partition T_1 and T_2 of $\{1, ..., 2n\}$ randomly, and for each $i \in T_1$ (resp., $i \in T_2$), with randomly chosen $s \in \{-1, +1\}$, assign $s \cdot i$ into U_1 and $-s \cdot i$ into $\widetilde{U}_1$ (resp., $s \cdot i$ into U_2 and $-s \cdot i$ into $\widetilde{U}_2$).

A generated graph $H = (U, D)$ is a directed graph of $4n$ vertices. The set U of vertices is determined from n by $U \stackrel{\text{def}}{=} U_1 \cup \widetilde{U}_1 \cup U_2 \cup \widetilde{U}_2$, where

$$\begin{array}{rclrcl} U_1 & \stackrel{\text{def}}{=} & \{1, \dots, n\}, & \widetilde{U}_1 & \stackrel{\text{def}}{=} & \{-1, \dots, -n\}, \\ U_2 & \stackrel{\text{def}}{=} & \{n+1, \dots, 2n\}, & \widetilde{U}_2 & \stackrel{\text{def}}{=} & \{-(n+1), \cdots, -2n\}. \end{array}$$

On the other hand, edges are generated randomly. There are two types of edges, and the set D of edges is defined by $D = I \cup C$ where I and C are generated as follows.

> Internal edges:
> $I \leftarrow \emptyset$;
> **for each** $V \in \{U_1, U_2\}$
> and **for each** $i, j \in V$ s.t. $i \neq j$ **do** {
> **repeat** $\lfloor pn \rfloor$ **times do** $I \leftarrow I \cup \{(i, j)\}$ with probability $1/n$;
> }
> Crossing edges:
> $C \leftarrow \emptyset$;
> **for each** V and V' from the following **do** {
> 1: (U_1, U_2), $(U_1, \widetilde{U}_2)$, $(\widetilde{U}_1, U_2)$, $(\widetilde{U}_1, \widetilde{U}_2)$,
> 2: $(U_1, \widetilde{U}_1)$, $(\widetilde{U}_1, U_1)$, and
> 3: $(U_2, \widetilde{U}_2)$, $(\widetilde{U}_2, U_2)$,
> **repeat** $\lfloor rn \rfloor$ **times do** {
> $f \leftarrow$ a random permutation mapping from V to V'
> **for each** $i \in V$ **do** $C \leftarrow C \cup \{(i, f(i))\}$;
> } }
> // Below we simply write, e.g., pn for $\lfloor pn \rfloor$.

Note that the graph may have multiple edges. There are rn^2 edges from, e.g, U_1 to U_2, and C has $8rn^2$ edges. On the other hand, the number of edges in I is from 0 to $2n^2$; but its expectation is $2pn^2$. We denote the distribution of graphs generated as above by $\mathcal{H}_{4n,p,r}$.

The transformation of a graph $H = (U, D)$ to a 2-CNF formula F is natural. For each edge $(i, j) \in D$ such that $i, j > 0$, a clause $(\overline{x_i} \vee x_j)$ is added to F. Similarly, for each edge $(i, -j)$ (resp., $(-i, j)$, $(-i, -j)$) such that $i, j > 0$, a clause $(\overline{x_i} \vee \overline{x_j})$ (resp., $(x_i \vee x_j)$, $(x_i \vee \overline{x_j})$) is added to F. This is our random generation of 2-CNF formulas, i.e., instances of the MAX2SAT problem. Note that F has $|D|$ edges, where $|D|$ is $2pn^2 + 8rn^2$ on average.

Consider any assignment to a to $2n$ Boolean variables of the generated formula F. That is, $a(i) \in \{-1, +1\}$ and $a(i) = +1 \iff x_i = 1$. We also regard it an assignment t to the vertices of H. For any $i \in U_1 \cup U_2$, we defined $t(i) = a(i)$, and for any $-i \in \widetilde{U}_1 \cup \widetilde{U}_2$, we define $t(-i) = -a(i)$. Vertices assigned true (i.e., $+1$) are called *true* and vertices assigned false (i.e., -1) are called *false*. Directed edges of H from a false vertex to a true vertex are called *unsatisfied edges*. Clearly each unsatisfied edge corresponds to a clause of F unsatisfied by the assignment a. On the other hand, any assignment t to vertices of H can be interpreted as an assignment to F's variables if $t(i) = -t(-i)$ for any $i \in U$. Such an assignment is called *consistent*. In particular, consistent assignments assigning the same values to all vertices in U_1 and U_2 respectively are important. There are four such assignments, and we call them *planted solutions*. The corresponding assignments to F are also called *planted solutions*. There are four planted solution. For example, assigning true to all vertices in U_1 and false to all in U_2

(hence, false to all in $\widetilde{U}_1$ and true to all in $\widetilde{U}_2$) is one of the four planted solutions. It is easy to see that any planted solution has rn^2 unsatisfied edges; thus, rn^2 clauses are unsatisfied by the corresponding assignment to F.

Now we claim that if p is large enough (compared with r), then planted solutions are optimal solutions (and no others) with high probability when MAX2SAT instances are generated as above.

Theorem 2.1. For any probability parameters p and r satisfying $p = \Omega(\ln^2 n/n)$ and $p \geq 9r$, consider a randomly generated formula F from a random graph of $\mathcal{H}_{4n,p,r}$. Then with high probability (i.e., with probability $1-o(1)$ w.r.t. n), four planted solutions are optimal solutions for F; futhermore, there are no other optimal solutions.

Proof. We show this by the well-known fact that a random graph is almost surely an expander. A directed graph $G = (V, E)$ is said to be a δ*-expander* if for every $S \subset V$ with $|S| \leq |V|/2$, the following holds:

$$\left|E\left(S, \overline{S}\right)\right| \geq \delta|S| \quad \text{and} \quad \left|E\left(\overline{S}, S\right)\right| \geq \delta|S|$$

Here by, e.g., $E(S, \overline{S})$ we mean the set of edges from vertices in S to vertices in $\overline{S}$.

We denote by $\mathcal{G}_{n,q,l}$ a distribution of graphs $G = (V, E)$ over n vertices that are generated as follows: for every pair (i, j) of vertices such that $i \neq j$, generate directed edges (i, j) l times independently with probability q and add them to E. Recall that when generating a graph of $\mathcal{H}_{4n,p,r}$, inner edges in U_1 and U_2 are generated in this way. We show here the following expansion property.

Claim 1. Let $l(n)$ be any function such that $l(n) = \Omega(\ln^2 n)$ and $l(n) \leq n$, and let $\delta(n)$ be any function such that $l(n)/2 - \delta(n) \geq c_0 l(n)$ for some constant $c_0 > 0$. Consider a a directed graph $G = (V, E)$ from $\mathcal{G}_{n,1/n,l(n)}$. Then with high probability (i.e., with probability $1 - o(1)$ w.r.t. n) G is a $\delta(n)$-expander.

Proof of the claim. Consider sufficiently large n, and let $l = l(n)$ and $\delta = \delta(n)$. Let S be a subset of V with size at most $n/2$. Let $\mathrm{Bad}(S)$ be an event that S does not meet the condition of a δ-expander, i.e., $|E(S, \overline{S})| < \delta|S|$ or $|E(\overline{S}, S)| < \delta|S|$. We estimate the upper bound of $\Pr\{|E(S, \overline{S})| < \delta|S|\}$. Note that the value of $\Pr\{\mathrm{Bad}(S)\}$ is at most two times of this value. This is done by using the standard Chernoff bound in the following way: For all pairs of $u \in S$ and $v \in \overline{S}$ and $1 \leq k \leq q$, we introduce independent random variables $Y_{u,v}^{(k)}$ such that $\Pr\{Y_{u,v} = 1\} = 1/n$ and $\Pr\{Y_{u,v} = 0\} = 1 - 1/n$. Let $Y \stackrel{\text{def}}{=} \sum_{u \in S, v \in \overline{S}} Y_{u,v}^{(k)}$. Then we have that $Y = |E(S, \overline{S})|$ and that $\mathrm{E}[Y] = l(1/n)(n - |S|)|S|$. Since we have

$$\begin{aligned}
\mathrm{E}[Y] - \delta|S| &= l(1/n)(n - |S|)|S| - \delta|S| \\
&\geq ((l/n)(n/2) - \delta) \cdot |S| \\
&\geq (l/2 - \delta) \cdot |S| \ > \ 0 \quad (\because l/2 > \delta),
\end{aligned}$$

we derive the following from the Chernoff bound:

$$\begin{aligned}
\Pr\left\{\left|E\left(S, \overline{S}\right)\right| < \delta|S|\right\} &= \Pr\{\mathrm{E}[Y] - Y > \mathrm{E}[Y] - \delta|S|\} \\
&< \exp\left(-\frac{(\mathrm{E}[Y] - \delta|S|)^2}{2\mathrm{E}[Y]}\right)
\end{aligned}$$

$$
\begin{aligned}
&\le \exp\left(-\frac{((l/2-\delta)|S|)^2}{2l(1/n)(n-|S|)|S|}\right) \\
&\le \exp\left(-\frac{(l/2-\delta)^2}{2l}\cdot|S|\right) \;\le\; \exp\left(-\frac{c_0^2 l}{2}\cdot|S|\right).
\end{aligned}
$$

Since the probability that G is not δ-expander is the probability of the union of $\mathrm{Bad}(S)$ over all $S \subset V$ with $|S| \le n/2$, it follows that

$$
\begin{aligned}
\Pr\left\{\bigcup_{S\subset V:|S|\le n/2} \mathrm{Bad}(S)\right\} &\le \sum_{s=1}^{n/2}\binom{n}{s}\Pr\{\mathrm{Bad}(S) : |S| = s\} \\
&\le \sum_{s=1}^{n/2}\left(\frac{en}{s}\right)^s \cdot 2\exp\left(-\frac{c_0^2 l(n)}{2}\cdot s\right) \\
&\le \sum_{s=1}^{n/2} 2\cdot\left(\frac{en}{s}\cdot\exp\left(-\frac{c_0^2 c\ln^2 n}{2}\right)\right)^s,
\end{aligned}
$$

where we used the assumption that $l(n) \ge c\ln^2 n$ for some constant c. Then we can conclude that the probability above is $o(1)$. □

Consider a random graph $H = (U, D)$ from $\mathcal{H}_{4n,p,r}$, and let F be its corresponding 2-CNF formula. Note that the induced graph $H[U_1]$ (resp., $H[U_2]$) of H on U_1 can be regarded as a random graph from $\mathcal{G}_{n,1/n,pn}$. On the other hand, $pn = \Omega(\ln^2 n/n)\cdot n = \Omega(\ln^2 n)$ by our assumption; hence from the above claim, we may assume that $H[U_1]$ and $H[U_2]$ are $pn/3$-expanders. That is, for each $V \in \{U_1, U_2\}$ and for every $S \subset V$, we have

$$
\left|D(S,\overline{S})\cap I\right| > \frac{pn}{3}|S| \quad\text{and}\quad \left|D(\overline{S},S)\cap I\right| > \frac{pn}{3}|S|.
$$

Now consider any consistent assignment t to U, which are different from the four planted solutions. By $\mathrm{unsat}_t(H)$ we denote the set of unsatisfied edges of H under t. From now on, we estimate a lower bound of $|\mathrm{unsat}_t(H)|$. Recall that $|\mathrm{unsat}_t(H)|$ is the same as the number of F's unsatisfied clauses. Thus, for the theorem, it suffices to show that $|\mathrm{unsat}_t(H)| > 3rn^2$.

Let a and b be the numbers of vertices of U_1 assigned true and false respectively under t. Similarly, let c and d be the numbers of vertices of U_2 assigned under true and false respectively under t. From the above expansion property, it is easy to see that the number of unsatisfied edges in U_1 (which are all internal edges of $H[U_1]$) is at least $(pn/3)\min\{a, b\}$. Similarly, the number of unsatisfied edges is U_2 is at least $(pn/3)\min\{c, d\}$.

For crossing edges, consider edges, for example, from U_1 to U_2 generated by one permutation mapping. Since any unsatisfied edge is from a true vertex to a false vertex, the number of unsatisfied edges is at least $a - d$ if $a \ge d$, and 0 otherwise. We here introduce a symbol '$\ominus$' and define $x \ominus y = x - y$ if $x \ge y$ and $x \ominus y = 0$ otherwise. Using this notation, the total number of unsatisfied edges from U_1 to U_2 is at least $rn(a \ominus d)$. It is similar for the other unsatisfied crossing edges. Summing up above, we have

$$
\begin{aligned}
|\mathrm{unsat}_t(H)| \;=\; & \frac{pn}{3}\min\{a,b\} + \frac{pn}{3}\min\{c,d\} \\
& + rn\left(|a-b| + |c-d| + (a\ominus c) + (a\ominus d) + (b\ominus c) + (b\ominus d)\right)
\end{aligned}
$$

Then it is not so hard to see that $|\text{unsat}_t(H)|$ is greater than rn^2 if $p \geq 9r$. Though easy, the derivation of this bound is tedious, we show below the analysis for one case that $a \leq b$ and $c \leq d$ and put the analyses of the other cases in the appendix. The condition of $a \leq b$ and $c \leq d$ implies $c \leq b$ and $a \leq d$ because of $a + b = c + d = n$. We further divide this case into two sub-cases: (i) $a \leq c$ and (ii) $c \leq a$. For the case of (i), since $a \leq c$ implies $d \leq b$, we have

$$\begin{aligned}
|\text{unsat}_t(H)| &> \frac{pn}{3}(a+c) + rn\left((b-a)+(d-c)+(b-c)+(b-d)\right) \\
&= \frac{pn}{3}(a+c) + 3rnb - rna - 2rnc \\
&= 3\left(\frac{pn}{9}a + rnb\right) + \left(\frac{pn}{9}c - rna\right) + 2\left(\frac{pn}{9}c - rnc\right) \\
&\geq 3rn(a+b) + \left(\frac{pn}{9} - rn\right)a + 2\left(\frac{pn}{9} - rn\right)c \\
&\geq 3rn^2.
\end{aligned}$$

On the other hand, for the case of (ii), since $c \leq a$ implies $b \leq d$, we have

$$\begin{aligned}
\text{unsat}_t(H) &\geq \frac{pn}{3}(a+c) + rn\left((b-a)+(d-c)+(a-c)+(b-c)\right) \\
&= \frac{pn}{3}(a+c) + 2rnb + rnd - 3rnc \\
&= 3\left(\frac{pn}{9}a + rnb\right) + rn(d-b) + 3\left(\frac{pn}{9}c - rnc\right) \\
&\geq 3rn(a+b) + rn(d-b) + 3\left(\frac{pn}{9} - rn\right)c \\
&\geq 3rn^2.
\end{aligned}$$

Note that the inequality is strict because either $c > 0$ or $d > b$. With the analyses for the other cases given in the appendix, we have $\text{unsat}_t(H) > 3rn^2$ for any assignment t. ■

3 Algorithm and Its Average Performance

We state our algorithm `algoMP_MAX2SAT` and prove our main theorem. That is, if a formula F is generated under our planted solution model with p and r satisfying $p - r \geq n^{-1/2+\epsilon}$, then with high probability, the algorithm yields one of the planted solutions, which is (again with high probability) the optimal solution for F.

The description of `algoMP_MAX2SAT` is shown in Figure 1. As explained in Introduction, the algorithm is based on the following simple heuristic idea: if a 2-CNF formula F has a clause $(l \to l')$ and we believe (for some reason) literal l' to be assigned false, then (in order to satisfy as many clauses as possible), we suppose that we had better assign false to literal l. This idea is implemented as statement (1); for any vertex corresponding literal l' believed to be false, a message that supports assignment false is sent from this vertex to all vertices corresponding literals l such that $(l \to l') \in F$ while no message is sent from vertices believed to be true. Note that beliefs should be consistent between i and $-i$, i.e., vertices for the same variable; this consistency is forced by statement (2).

For understanding the algorithm, some more detail explanations are needed.

1. A graph $H_F = (U, A)$ constructed from F is different from the one $H = (U, D)$ for generating F. We use the same vertex set U; on the other hand, for each clause $(x_i \vee x_j)$, for example, H_F has two edges (i, j) and $(-j, -i)$. Thus, each edge in D corresponds to two edges in A.

2. For any $C_k = (l \vee l')$, let $e(C_k)$ denote a directed edge corresponding to $(\bar{l} \to l')$, and $\overline{e}(C_k)$ is a directed edge corresponding to $(\overline{l'} \to l)$. By $N^{-1}(u)$ we mean the set of vertices j having a direct edge to i. The function $\mathrm{sign}(z)$ returns $+1$ if $z > 0$ and -1 otherwise.

3. For our theoretical analysis, we make the following modifications: (i) set MAXSTEP $= 2$, (ii) statement (2) is not executed for the first round, and (iii) statement (3) is inserted.

4. Due to our simplified version for generating instances (see the footnote of the previous section), we could assume that $x_1 = -1$ and $x_{n+1} = -1$. For the general instances, while we may still fix x_1, we would have to run the algorithm by fixing $x_1 = \pm 1$ and $x_j = \pm 1$ for all $j \in \{1, ..., 2n\} \backslash \{i\}$.

```
procedure algoMP_MAX2SAT(F);
// An input F = C_1 ∧ ··· ∧ C_m is a 2-CNF formula over variables x_1, ··· , x_2n.
// The algorithm assumes that x_1 = −1 (i.e., false) and x_{n+1} = −1 (i.e., false).
begin
  Construct a directed graph H_F = (U, A),
    where U = U_1 ∪ Ũ_1 ∪ U_2 ∪ Ũ_2, and A = {e(C_k), ē(C_k) : 1 ≤ k ≤ m};
  Set b(i) to 0 for all i ∈ U;
  Set b(1) = b(n + 1) = −1;
  repeat MAXSTEP times do {
    for each i ∈ {1, ··· , 2n}\{1, n + 1} do {
      // The following update is made in parallel.
      b(i)  ← Σ_{j∈N^{-1}(i)} min(0, b(j));    }
      b(−i) ← Σ_{j∈N^{-1}(−i)} min(0, b(j));   }  — (1)
      b(i)  ← b(i) − b(−i);  }
      b(−i) ← −b(i);         }  — (2)
    }
    if all sign(i) are stabilized then break;
    // Set b(1), b(−1), b(n + 1), b(−(n + 1)) to 0;   — (3)
  }
  output( −1, sign(b(2)), ··· , sign(b(n)), −1, sign(b(n + 2)), ··· , sign(b(2n)) );
end-procedure
```

Figure 1: Message passing algorithm for the MAX2SAT problem

It is easy to see that the running time of the algorithm (for the unit cost RAM model) is $O(n + m)$; thus, the total running time for the general instances is $O((n + m)n)$.

Now we analyze the performance of the algorithm and prove the main theorem. From now on, we consider sufficiently large n and a random formula F generated by our planted solution model with parameters p and r. We assume that p and r satisfies the condition of the theorem and $p \geq 9r$. That is, $p - r > n^{-1/2+\epsilon_p}$ for some $\epsilon_p > 0$; hence, clearly $p > n^{-1/2+\epsilon_p}$. Let $H = (U, A)$ be a graph constructed in the algorithm from F. (For simpicity we omit the subscript F; this H is different from the one used for generating F.) We denote by I a set of edges within $V \in \{U_1, \widetilde{U}_1, U_2, \widetilde{U}_2\}$ and define $C = A \backslash I$.

Consider any $i \in \{1, \cdots, 2n\} \backslash \{1, n+1\}$. Let b_i be a random variable denoting the value obtained as $b(i)$ after the execution. Its expectation can be calculated as follows.

Lemma 3.1. For any $i \in \{1, \cdots, 2n\} \backslash \{1, n+1\}$, we have $\mathrm{E}[b_i] = -((p-r)^2 n - 2p(p-r))$.

Proof. Let A_{ij} for any (ordered) pair of $i, j \in V \in \{U_1, \widetilde{U}_1, U_2, \widetilde{U}_2\}$ be a random variable related to an internal edge such that $A_{ij} = k$ if the number of edges of I from i to j is k. Similarly, let B_{ij} for each pair of $i \in V$ and $j \in V'$ where (V, V') from the followings

$$\begin{array}{rl} 1: & (U_1, U_2),\ (U_1, \widetilde{U}_2),\ (\widetilde{U}_1, U_2),\ (\widetilde{U}_1, \widetilde{U}_2), \\ 1': & (\widetilde{U}_2, \widetilde{U}_1),\ (U_2, \widetilde{U}_1),\ (\widetilde{U}_2, U_1),\ (U_2, U_1), \\ 2: & (U_1, \widetilde{U}_1),\ (\widetilde{U}_1, U_1), \\ 3: & (U_2, \widetilde{U}_2),\ (\widetilde{U}_2, U_2), \end{array}$$

be a random variable related to a crossing edge such that $B_{ij} = k$ if the number of edges of C from i to j is k. Then, for any $i \in U_1 \backslash \{1\}$, we have $b_i = b_i^+ + (-b_i^-)$, where

$$\begin{aligned} -b_i^+ \;=\; & \sum_{j \in U_1} A_{ij} A_{j1} + \sum_{j \in \widetilde{U}_1} B_{ij} B_{j1} + \sum_{j \in U_2} B_{ij} B_{j1} + \sum_{j \in \widetilde{U}_2} B_{ij} B_{j1} \\ & + \sum_{j \in U_2} B_{ij} A_{jn+1} + \sum_{j \in \widetilde{U}_2} B_{ij} B_{jn+1} + \sum_{j \in U_1} A_{ij} B_{jn+1} + \sum_{j \in \widetilde{U}_1} B_{ij} B_{jn+1}, \end{aligned} \tag{1}$$

and

$$\begin{aligned} -b_i^- \;=\; & \sum_{j \in U_1} B_{-ij} A_{j1} + \sum_{j \in \widetilde{U}_1} A_{-ij} B_{j1} + \sum_{j \in U_2} B_{-ij} B_{j1} + \sum_{j \in \widetilde{U}_2} B_{-ij} B_{j1} \\ & + \sum_{j \in U_2} B_{-ij} A_{jn+1} + \sum_{j \in \widetilde{U}_2} B_{-ij} B_{jn+1} + \sum_{j \in U_1} B_{-ij} B_{jn+1} + \sum_{j \in \widetilde{U}_1} A_{-ij} B_{jn+1}. \end{aligned} \tag{2}$$

Recall our generation of a random graph from which F is obtained. Since for any internal edge from i to j we independently generate pn times a direct edge (i, j) with probability $1/n$, we have $\mathrm{E}[A_{ij}] = (pn) * (1/n) = p$. Moreover, A_{ij} is independent of any $A_{i'j'}$ s.t. $(i', j') \notin \{(i, j), (-j, -i)\}$ and any $B_{i''j''}$. On the other hand, since for crossing edges we independently generate rn random permutations, we have $\mathrm{E}[B_{ij}] = (rn) * (1/n) = r$ for any crossing edge from i to j. Moreover, any B_{ij} is independent of any B_{jk}. Thus, we have

$$\begin{aligned} \mathrm{E}[-b_i^+] &= p^2(n-2) + 2pr(n-1) + 5r^2 n, \\ \mathrm{E}[-b_i^-] &= 4pr(n-1) + 4r^2 n, \end{aligned}$$

from the fact that $\mathrm{E}[YY'] = \mathrm{E}[Y]\mathrm{E}[Y']$ if Y is independent of Y'. Therefore, we conclude $\mathrm{E}[b_i] = -(p^2(n-2) + 2pr(n-1) + 5r^2 n) + 4pr(n-1) + 4r^2 n = -((p-r)^2 n - 2p(p-r))$.

It is similarly proven for $i \in U_2 \backslash \{n+1\}$. We put a formula of b_i for $i \in U_2 \backslash \{n+1\}$ in the appendix. We'll see there that $\mathrm{E}[b_i]$ for $i \in U_2 \backslash \{n+1\}$ has the same value. ■

From our choice of p and r, we have $\mathrm{E}[b_i] = -((p-r)^2 n - 2p(p-r)) < -1$ because the value of b_i is integer. Thus, the algorithm yields *on average* all false assignment, which is one of the planted solutions. Now for showing that the algorithm surely yields this planted solution, we will discuss below concentration of b_i around its average. More specifically, for $i \in U_1 \backslash \{1\}$, we estimate the following probability: the value of $\Pr\{b_i > (1-\alpha)\mathrm{E}[b_i]\}$ for any $\alpha > 0$. (The analysis is similar for $i \in U_2 \backslash \{n+1\}$.)

Since the expectation of $-b_i$ is positive, we deal with $-b_i$ not b_i for our convenience. Consider the following cases: for an arbitrary constant $\epsilon > 0$,

$$(*) \ \cdots \ \left\{ \begin{array}{lcl} |(\sum_{j \in U_1 \backslash \{1,i\}} A_{j1}) - p(n-2)| & < & \epsilon p(n-2), \\ |(\sum_{j \in U_2 \backslash \{n+1\}} A_{jn+1}) - p(n-1)| & < & \epsilon p(n-1), \end{array} \right.$$

and

$$(**) \ \cdots \ \left\{ \begin{array}{lclllcl} \max\{A_{j1} : j \in U_1\} & \le & \ln n, & & \max\{A_{jn+1} : j \in U_2\} & \le & \ln n, \\ & & & & \max\{B_{jn+1} : j \in U_1\} & \le & \ln n, \\ \max\{B_{j1} : j \in \widetilde{U}_1\} & \le & \ln n, & & \max\{B_{jn+1} : j \in \widetilde{U}_1\} & \le & \ln n, \\ \max\{B_{j1} : j \in U_2\} & \le & \ln n, & & & & \\ \max\{B_{j1} : j \in \widetilde{U}_2\} & \le & \ln n, & & \max\{B_{jn+1} : j \in \widetilde{U}_2\} & \le & \ln n. \end{array} \right.$$

We denote by $\mathrm{Good}(H)$ the event that all the events of $(*)$ and $(**)$ simaltaniously occur. As is shown in Lemma 3.2, the probability of $\overline{\mathrm{Good}(H)}$ (for any $\epsilon > 0$) is less than $1/n^2$. (We can actually prove much smaller probability. But the value of $1/n^2$ is sufficiently small for our purpose.) Thus, we have

$$\begin{array}{rcl} \Pr\{-b_i < (1-\alpha)\mathrm{E}[-b_i]\} & = & \Pr\{\mathrm{Good}(H)\} \Pr\{-b_i < (1-\alpha)\mathrm{E}[-b_i] | \mathrm{Good}(H)\} \\ & & \quad + \Pr\left\{\overline{\mathrm{Good}(H)}\right\} \Pr\left\{-b_i < (1-\alpha)\mathrm{E}[-b_i] \middle| \overline{\mathrm{Good}(H)}\right\} \\ & < & 1 \cdot \Pr\{-b_i < (1-\alpha)\mathrm{E}[-b_i] | \mathrm{Good}(H)\} \\ & & \quad +(1/n^2) \cdot \Pr\left\{-b_i < (1-\alpha)\mathrm{E}[-b_i] \middle| \overline{\mathrm{Good}(H)}\right\} \\ & \le & \Pr\{-b_i < (1-\alpha)\mathrm{E}[-b_i] | \mathrm{Good}(H)\} + 1/n^2. \end{array}$$

Lemma 3.2. The probability that at least one of $(*)$ and $(**)$ is not satisfied is less than $1/n^2$, i.e., $\Pr\{\overline{\mathrm{Good}(H)}\} < 1/n^2$.

Proof. Consider the first case of $(*)$, i.e., $|\sum_{j \in U_1 \backslash \{1,i\}} A_{j1} - p(n-2)| < \epsilon p(n-2)$. According to our generation of internal edges, A_{j1} can be regarded as the sum of independent pn random variables $Y_j^1, \cdots, Y_j^{pn}$ such that $\Pr\{Y_j^k = 1\} = 1/n$ and $\Pr\{Y_j^k = 0\} = 1 - 1/n$ for $1 \le k \le pn$. Note that those $pn * (n-2)$ random variables $\bigcup_{j \in U_1 \backslash \{1,i\}} \{Y_j^1, \cdots, Y_j^{pn}\}$ are independent. Applying the standard Chernoff bound to it, we have

$$\Pr\left\{ \left| \sum_{j \in U_1} A_{j1} - p(n-2) \right| > \epsilon p(n-2) \right\} \ < \ 2\exp\left(-\frac{\epsilon^2 p(n-2)}{3} \right).$$

It is exactly the same for the other case of $(*)$.

Next, consider the first case of $(**)$, i.e., $\max\{A_{j1} : j \in U_1\}$. Since we have

$$\Pr\{\max\{A_{j1} : j \in U_1\} \le \ln n\} \;\le\; \sum_{j \in U_1} \Pr\{A_{j1} \le \ln n\}, \tag{3}$$

we estimate the value of $\Pr\{A_{j1} \le \ln n\}$ for any $j \in U_1$. Again, A_{j1} can be regarded as the sum of independent pn random variables $Y_j^1, \cdots, Y_j^{pn}$ such that $\Pr\{Y_j^k = 1\} = 1/n$ and $\Pr\{Y_j^k = 0\} = 1 - 1/n$ for $1 \le k \le pn$. Applying to it corollary 4.2 in the Appendix with $1 + \delta = \ln n/\mu$ (> 1) where $\mu = p$, we have

$$\begin{aligned} \Pr\{A_{j1} \le \ln n\} &< \left(\frac{e}{\ln n/p}\right)^{(\ln n/p)p} \\ &< \left(e \cdot \frac{p}{\ln n}\right)^{\ln n}. \end{aligned}$$

Since the inequality above is satisfied for any $j \le U_1$, we conclude from (3) that

$$\Pr\{\max\{A_{j1} : j \in U_1\} \le \ln n\} \;<\; n \cdot \left(e \cdot \frac{p}{\ln n}\right)^{\ln n}.$$

It is exactly the same for $\Pr\{\max\{A_{jn+1} : j \in U_2\} \le \ln n\}$. Consider the next case of $(**)$, i.e., $\max\{B_{jn+1} : j \in U_1\}$. Since we have

$$\Pr\{\max\{B_{jn+1} : j \in U_1\} \le \ln n\} \;\le\; \sum_{j \in U_1} \Pr\{B_{jn+1} \le \ln n\}, \tag{4}$$

we estimate the value of $\Pr\{B_{jn+1} \le \ln n\}$ for any $j \in U_1$. Similarly, B_{jn+1} can be regarded as the sum of independent rn random variables $Y_j^1, \cdots, Y_j^{rn}$ such that $\Pr\{Y_j^k = 1\} = 1/n$ and $\Pr\{Y_j^k = 0\} = 1 - 1/n$ for $1 \le k \le rn$. Applying to it corollary 4.2 in the Appendix with $1 + \delta = \ln n/\mu$ (> 1) where $\mu = r$, we have

$$\begin{aligned} \Pr\{B_{jn+1} \le \ln n\} &< \left(\frac{e}{\ln n/r}\right)^{(\ln n/r)r} \\ &< \left(e \cdot \frac{r}{\ln n}\right)^{\ln n}. \end{aligned}$$

Since the inequality above is satisfied for any $j \le U_1$, we conclude from (4) that

$$\Pr\{\max\{B_{j1} : j \in U_1\} \le \ln n\} \;<\; n \cdot \left(e \cdot \frac{r}{\ln n}\right)^{\ln n}.$$

It is exactly the same for the other cases. At last, summing up all the probability we have calculated above, we have that $\Pr\{\overline{\text{Good}(H)}\}$ is less than,

$$4 \exp\left(-\frac{\epsilon^2 p(n-2)}{2}\right) + 2n \cdot \left(e \cdot \frac{p}{\ln n}\right)^{\ln n} + 6n \cdot \left(e \cdot \frac{r}{\ln n}\right)^{\ln n} \;<\; \frac{1}{n^2}.$$

■

Therefore, we'll show that the conditional probability of $\Pr\{-b_i < (1-t)\mathrm{E}[-b_i]\}$ given $\text{Good}(H)$ is small. That is:

Lemma 3.3. With high probability, say, $1 - o(1)$, we have $-b_i > 1$ and $b_{-i} > 1$ for all $i \in \{1, \cdots, 2n\}$.

Proof. The idea of estimating the deviation is the following observation: If each term of b_i shown in (1) and (2) little deviate from each expectation, then b_i also little deviate from its expectation. That is:

Claim 2. Let $Y_1^+, \cdots, Y_{k^+}^+$ be (not necessarily independent) random variables taking positive values or zero, and $Y_1^-, \cdots, Y_{k^-}^-$ be (not necessarily independent) be random variables taking negative values or zero. Let $Y \stackrel{\text{def}}{=} \sum_{k'} Y_{k'}^+ + \sum_{k'} Y_{k'}^-$. If we have $\Pr\{Y_{k'}^+ < (1-\epsilon_{k'}^+)\mathrm{E}[Y_{k'}^+]\} < p_{k'}^+$ for $1 \le k' \le k^+$ and $\Pr\{Y_{k'}^- < (1+\epsilon_{k'}^-)\mathrm{E}[Y_{k'}^-]\} < p_{k'}^-$ for $1 \le k' \le k^-$, then

$$\Pr\left\{Y < \sum_{k'}(1-\epsilon_{k'}^+)\mathrm{E}[Y_{k'}^+] + \sum_{k'}(1+\epsilon_{k'}^-)\mathrm{E}[Y_{k'}^-]\right\} \;<\; \sum_{k'} p_{k'}^+ + \sum_{k'} p_{k'}^-.$$

Proof of the claim. It is obvious that if $Y_{k'}^+ \ge (1-\epsilon_{k'}^+)\mathrm{E}[Y_{k'}^+]$ for $1 \le k' \le k^+$ and $Y_{k'}^- \ge (1+\epsilon_{k'}^-)\mathrm{E}[Y_{k'}^-]$ for $1 \le k' \le k^-$, then $Y \ge \sum_{k'}(1-\epsilon_{k'}^+)\mathrm{E}[Y_{k'}^+] + \sum_{k'}(1+\epsilon_{k'}^-)\mathrm{E}[Y_{k'}^-]$. Taking the contraposition of this, we have that if $Y < \sum_{k'}(1-\epsilon_{k'}^+)\mathrm{E}[Y_{k'}^+] + \sum_{k'}(1+\epsilon_{k'}^-)\mathrm{E}[Y_{k'}^-]$, then $Y_{k'}^+ < (1-\epsilon_{k'}^+)\mathrm{E}[Y_{k'}^+]$ for some $1 \le k' \le k^+$ or $Y_{k'}^- < (1+\epsilon_{k'}^-)\mathrm{E}[Y_{k'}^-]$ for some $1 \le k' \le k^-$. That probability is less than $\sum_{k'=1}^{k^+} p_{k'}^+ + \sum_{k'=1}^{k^-} p_{k'}^-$. $\square$

Consider first a term $\sum_{j \in U_1} A_{ij}A_{j1}$ of b_i under Good(H). Since we are given Good(H), we suppose that each value of A_{j1} is arbitrarily given such that

$$\sum_{j \in U_1 \setminus \{1,i\}} A_{j1} = (1 \pm \epsilon)p(n-2) \quad \text{and} \quad \max\{A_{j1} : j \in U_1\} \le \ln n.$$

We denote this event above by Good$(A_{U_1 1})$. According to our generation of internal edges, for $j \in U_1 \setminus \{1, i\}$, $A_{ij}A_{j1}$ can be seen as the sum of independent pn random variables $Y_j^1, \cdots, Y_j^{pn}$ such that $\Pr\{Y_j^k = A_{j1}\} = 1/n$ and $\Pr\{Y_j^k = 0\} = 1 - 1/n$ for $1 \le k \le pn$. Thus, we have for any $y > 0$,

$$\Pr\left\{\sum_{j \in U_1} A_{ij}A_{j1} < y \,\middle|\, \mathrm{Good}(H)\right\} \;=\; \Pr\left\{\sum_{j \in U_1}\sum_{k=1}^{pn} Y_j^k < y \,\middle|\, \mathrm{Good}(A_{U_1 1})\right\}.$$

Note that those $pn * (n-2)$ random variables $\bigcup_{j \in U_1 \setminus \{1,i\}} \{Y_j^1, \cdots, Y_j^{pn}\}$ are independent. We are now in the position that we can apply corollary 4.5 to the RHS of the equation above. Since the target value of y we are about to prove is $(1-\alpha_1)p^2(n-2)$ for $\alpha_1 > 0$, and

$$\begin{aligned}
\mathrm{E}\left[\sum_{j \in U_1}\sum_{k=1}^{pn} Y_j^k\right] \;&=\; \sum_{j \in U_1}\sum_{k=1}^{pn} \mathrm{E}\left[Y_j^k\right] \;=\; \sum_{j \in U_1}\sum_{k=1}^{pn} A_{j1} \cdot \frac{1}{n} \\
&=\; \sum_{j \in U_1} A_{j1} \cdot \sum_{k=1}^{pn} \frac{1}{n} \;=\; p \cdot \sum_{j \in U_1} A_{j1} \;=\; p \cdot (1 \pm \epsilon)p(n-2) \;=\; (1 \pm \epsilon)p^2(n-2),
\end{aligned}$$

by applying the corollary with $\delta = \alpha_1$ s. t. $1 - \alpha_1 = (1-\delta)(1 \pm \epsilon)$, we have for some constant c_1,

$$\Pr\left\{\sum_{j \in U_1}\sum_{k=1}^{pn} Y_j^k < (1-\delta)(1 \pm \epsilon)p^2(n-2) \,\middle|\, \mathrm{Good}(A_{U_1 1})\right\}$$

$$< \exp\left(-\frac{\delta^2(1\pm\epsilon)p^2(n-2)}{2\ln n}\right) = \exp\left(-\frac{\left(1-\frac{1-\alpha_1}{1\pm\epsilon}\right)^2(1\pm\epsilon)p^2(n-2)}{2\ln n}\right)$$

$$= \exp\left(-\frac{(\alpha_1\pm\epsilon)^2p^2n}{(1\pm\epsilon)c_1\ln n}\right) = \exp\left(-\frac{(\alpha_1\pm\epsilon)^2n^{\epsilon_p}}{(1\pm\epsilon)c_1\ln n}\right).$$

Next, consider, for example, a term $\sum_{j\in U_2} B_{ij}A_{jn+1}$ of b_i under Good(H). (It is exactly the same for the other terms of type $\sum_{j\in V} A_{*j}B_{j*}$ and type $\sum_{j\in V} B_{*j}A_{j*}$ for $V\in\{U_1,\widetilde{U}_1,U_2,\widetilde{U}_2\}$.) Since we are given Good(H), we suppose that each value of A_{jn+1} is arbitrarily given such that

$$\sum_{j\in U_2} A_{jn+1} = (1\pm\epsilon)p(n-1) \quad \text{and} \quad \max\{A_{jn+1} : j\in U_2\} \le \ln n.$$

We denote this event above by Good(A_{U_2n+1}). According to our generation of crossing edges, $\sum_{j\in U_2} B_{ij}A_{jn+1}$ can be seen as the sum of independent rn random variables $Y^1,\cdots,Y^{rn}$ such that $\Pr\{Y^k = A_{jn+1}\} = 1/n$ for $1\le k\le rn$ and $j\in U_2\backslash\{n+1\}$. Thus, we have for any $y>0$,

$$\Pr\left\{\sum_{j\in U_2} B_{ij}A_{jn+1} < y \middle| \text{Good}(H)\right\} = \Pr\left\{\sum_{k=1}^{rn} Y^k < y \middle| \text{Good}(A_{U_2n+1})\right\}.$$

We are now in the position that we can apply corollary 4.5 to the RHS of the equation above. Since the target value of y we are about to prove is $(1-\alpha_2)pr(n-1)$ for $\alpha_2>0$, and

$$\mathrm{E}\left[\sum_{k=1}^{rn} Y^k\right] = \sum_{k=1}^{rn}\mathrm{E}\left[Y^k\right] = \sum_{k=1}^{rn}\sum_{j\in U_2} A_{jn+1}\cdot\frac{1}{n}$$

$$= \sum_{k=1}^{rn}\frac{1}{n}\cdot\sum_{j\in U_2} A_{jn+1} = \frac{1}{n}\cdot rn\cdot(1\pm\epsilon)p(n-1) = (1\pm\epsilon)pr(n-1),$$

by applying the corollary with $\delta=\alpha_2$ s. t. $(1-\alpha_2)=(1-\delta)(1\pm\epsilon)$, we have for some constant c_2,

$$\Pr\left\{\sum_{k=1}^{rn} Y_i^k < (1-\delta)(1\pm\epsilon)pr(n-1) \middle| \text{Good}(A_{U_2n+1})\right\}$$

$$< \exp\left(-\frac{\delta^2(1\pm\epsilon)pr(n-1)}{2\ln n}\right) = \exp\left(-\frac{\left(1-\frac{1-\alpha_2}{1\pm\epsilon}\right)^2(1\pm\epsilon)pr(n-1)}{2\ln n}\right)$$

$$= \exp\left(-\frac{(\alpha_2\pm\epsilon)^2prn}{(1\pm\epsilon)c_2\ln n}\right) = \exp\left(-\frac{(\alpha_2\pm\epsilon)^2rn^{1/2+\epsilon_p}}{(1\pm\epsilon)c_2\ln n}\right).$$

Finally, consider, for example, a term $\sum_{j\in\widetilde{U}_1} B_{ij}B_{j1}$ of b_i under Good(H). (It is exactly the same for the other terms of type $\sum_{j\in V} B_{*j}B_{j*}$ for $V\in\{U_1,\widetilde{U}_1,U_2,\widetilde{U}_2\}$.) From our generation of crossing edges and condition Good(H), we suppose that each value of B_{j1} is arbitrarily given such that

$$\sum_{j\in\widetilde{U}_1} B_{j1} = rn \quad \text{and} \quad \max\{B_{j1} : j\in\widetilde{U}_1\} \le \ln n.$$

We denote this event above by $\mathrm{Good}(B_{\widetilde{U}_1 1})$. According to our generation of crossing edges, $\sum_{j\in\widetilde{U}_1} B_{ij}B_{j1}$ can be seen as the sum of independent rn random variables $Y^1, \cdots, Y^{rn}$ such that $\Pr\{Y^k = B_{j1}\} = 1/n$ for $1 \le k \le rn$ and $j \in \widetilde{U}_1$. Thus, we have for any $y > 0$,

$$\Pr\left\{\sum_{j\in\widetilde{U}_1} B_{ij}B_{j1} < y \middle| \mathrm{Good}(H)\right\} \;=\; \Pr\left\{\sum_{k=1}^{rn} Y^k < y \middle| \mathrm{Good}(B_{\widetilde{U}_1 1})\right\}.$$

We are now in the position that we can apply corollary 4.5 to the RHS of the equation above. Since the target value of y we are about to prove is $(1-\alpha_3)r^2n$ for $\alpha_3 > 0$, and

$$\begin{aligned}\mathrm{E}\left[\sum_{k=1}^{rn} Y^k\right] &= \sum_{k=1}^{rn}\mathrm{E}\left[Y^k\right] = \sum_{k=1}^{rn}\sum_{j\in\widetilde{U}_1} B_{j1}\cdot\frac{1}{n}\\ &= \sum_{k=1}^{rn}\frac{1}{n}\cdot\sum_{j\in\widetilde{U}_1} B_{j1} = \frac{1}{n}\cdot rn\cdot rn = r^2n,\end{aligned}$$

by applying the corollary with $\delta = \alpha_3$, we have for some constant c_3,

$$\Pr\left\{\sum_{k=1}^{pn} Y^k < (1-\delta)r^2n \middle| \mathrm{Good}(B_{\widetilde{U}_1 1})\right\} \;<\; \exp\left(-\frac{\delta^2r^2n}{2\ln n}\right) \;=\; \exp\left(-\frac{\alpha_3^2r^2n}{c_3\ln n}\right).$$

At last, from the observation we have seen in Claim 2, summing up all the probability we have calculated above, the probability of the event

$$-b_i \;<\; \left\{\begin{array}{l}(1-\alpha_1)p^2(n-2)\\ \;+\;2(1-\alpha_2)pr(n-1)\\ \;+\;5(1-\alpha_3)r^2n\end{array}\right\} \;-\; \left\{\begin{array}{l}4(1+\alpha_4)pr(n-1)\\ \;+\;4(1+\alpha_5)r^2n\end{array}\right\},$$

which is equivalent to

$$-b_i \;<\; \mathrm{E}[-b_i] \;+\; \left\{\begin{array}{l}-\;\alpha_1 n^{\epsilon_p}\\ -\;2\alpha_2 rn^{1/2+\epsilon_p}\\ -\;5\alpha_3 r^2n\\ -\;4\alpha_4 rn^{1/2+\epsilon_p}\\ -\;4\alpha_5 r^2n\end{array}\right\} \;+\; \left\{\begin{array}{l}-\;(1-\alpha_1)2p^2\\ -\;2(1-\alpha_2)pr\\ +\;4(1+\alpha_4)pr\end{array}\right\}$$

is less than

$$\left\{\begin{array}{l}\exp\left(-\frac{(\alpha_1\pm\epsilon)^2n^{\epsilon_p}}{(1\pm\epsilon)c_1\ln n}\right)\\ \;+\;2\exp\left(-\frac{(\alpha_2\pm\epsilon)^2rn^{1/2+\epsilon_p}}{(1\pm\epsilon)c_2\ln n}\right)\\ \;+\;5\exp\left(-\frac{\alpha_3^2r^2n}{c_3\ln n}\right)\end{array}\right\} \;+\; \left\{\begin{array}{l}4\exp\left(-\frac{(\alpha_4\pm\epsilon)^2rn^{1/2+\epsilon_p}}{(1\pm\epsilon)c_2\ln n}\right)\\ \;+\;4\exp\left(-\frac{\alpha_5^2r^2n}{c_3\ln n}\right)\end{array}\right\}.$$

We carefully choose parameters $\alpha_1, \cdots, \alpha_5$ so that the deviation probability above is sufficiently small, say, $1/n^2$. We set α_1 to some constant, and the others are set as

$$\begin{array}{lllll}\alpha_2 &=& \alpha_4 &=& r^{-1/2}n^{-1/4+\epsilon_p/4}\mp\epsilon \quad (>0)\\ \alpha_3 &=& \alpha_5 &=& r^{-1}n^{-1/2+\epsilon_p/2} \quad\quad\;\; (>0)\end{array}$$

so that the deviation probability is less than

$$\left\{ \begin{array}{l} \exp\left(-\frac{(\alpha_1 \pm \epsilon)^2 n^{\epsilon_p}}{(1\pm\epsilon)c_1 \ln n}\right) \\ \;+\; 2\exp\left(-\frac{n^{\epsilon_p}}{(1\pm\epsilon)c_2 \ln n}\right) \\ \;+\; 5\exp\left(-\frac{n^{\epsilon_p}}{c_3 \ln n}\right) \end{array} \right\} + \left\{ \begin{array}{l} 4\exp\left(-\frac{n^{\epsilon_p}}{(1\pm\epsilon)c_2 \ln n}\right) \\ \;+\; 4\exp\left(-\frac{n^{\epsilon_p}}{c_3 \ln n}\right) \end{array} \right\} < \frac{1}{n^2}.$$

Since we have the above for all $i \in \{1, \cdots, 2n\}\backslash\{1, n+1\}$, we have with probability $1 - 1/n$ that for all $i \in \{1, \cdots, 2n\}\backslash\{1, n+1\}$,

$$\begin{array}{rcl} -b_i & \geq & n^{2\epsilon_p} + \left\{ \begin{array}{l} -\ \alpha_1 n^{\epsilon_p} \\ -\ 6\alpha_2 r n^{1/2+\epsilon_p} \\ -\ 9\alpha_3 r^2 n \end{array} \right\} + \left\{ \begin{array}{l} -\ (1-\alpha_1)2p^2 \\ -\ 2(1-\alpha_2)pr \\ +\ 4(1+\alpha_4)pr \end{array} \right\} \\ & \geq & n^{2\epsilon_p} + \left\{ \begin{array}{l} -\ \alpha_1 n^{\epsilon_p} \\ -\ 6r^{1/2} n^{1/4+5\epsilon_p/4} \pm 6\epsilon r n^{1/2+\epsilon_p} \\ -\ 9rn^{1/2+\epsilon_p/2} \end{array} \right\} - 1 \\ & \geq & n^{2\epsilon_p} + \left\{ \begin{array}{l} -\ \alpha_1 n^{\epsilon_p} \\ -\ 6n^{7\epsilon_p/4} - 6\epsilon r n^{1/2+\epsilon_p} \\ -\ 9n^{1/2+\epsilon_p/2} \end{array} \right\} - 1 \\ & \geq & n^{2\epsilon_p} - \alpha_1 n^{\epsilon_p} - 6n^{7\epsilon_p/4} - 6\epsilon n^{2\epsilon_p} - 9n^{3\epsilon_p/2} - 1 \\ & > & 1. \end{array}$$

■

Now we summarize what we have obtained is enough for proving the main theorem. First from Lemma 3.1 and by our choice of p and r, if each b_i is close to its expectation, then the assignment that the algorithm yields is one of the planted solution, i.e., all false assignment. Secondly, from Lemma 3.3, if H (i.e., H_F constructed from F in the algorithm) satisfies some condition, then the deviation of b_i from its expectation is small enough. Finally, Lemma 3.2 guarantees that such a good situation occurs with high probability. Therefore we have our theorem.

References

[HM98] M. Habib et al. Ed., "Probabilistic Methods for Algorithmic Discrete Mathematics", Springer, 1998.

[JS98] M. Jerrum and G. Sorkin, The Metropolis algorithm for graph bisection, *Discrete Appl. Math* 82(1-3), 155–175, 1998.

[MR95] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge Univ. Press, 1995.

[Pea88] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers Inc., 1988.

[OW05] M. Onsjö and O. Watanabe, Simple algorithms for graph partition problems, Research Report C-212, Dept. of Math. and Comput. Sci., Tokyo Inst. of Tech, 2005.

[Yam05] M. Yamamoto, Generating instances for MAX2SAT with optimal solutions, *Theory of Comput. Syst.*, to appear.

4 Appendix

Complete Proof of Theorem 2.1

We present a proof of the lower bound of $|\mathrm{unsat}_t(H)|$ for the rest case. This completes the proof of Theorem 2.1. Here, we again describe the target equation:

$$\begin{aligned}|\mathrm{unsat}_t(H)| &> \frac{pn}{3}\min\{a,b\} + \frac{pn}{3}\min\{c,d\} \\ &\quad + rn\left(|a-b| + |c-d| + (a \ominus c) + (a \ominus d) + (b \ominus c) + (b \ominus d)\right).\end{aligned}$$

We complete the proof of the claim that $|\mathrm{unsat}_t(H)|$ is greater than rn^2 if $p \geq 9r$. We have proven it for the case that $a \leq b$ and $c \leq d$. In this appendix, we prove it for the other cases, that is, (1): $a \leq b$ and $d \leq c$, (2): $b \leq a$ and $c \leq d$, and (3): $b \leq a$ and $d \leq c$. For the case of (1), the condition of $a \leq b$ and $d \leq c$ implies $d \leq b$ and $a \leq c$ because of $a+b=c+d=n$. We further divide this case into two sub-cases: (1-i) $a \leq d$ and (1-ii) $d \leq a$. For the case of (1-i), since $a \leq d$ implies $c \leq b$, we have

$$\begin{aligned}|\mathrm{unsat}_t(H)| &> \frac{pn}{3}(a+d) + rn\left((b-a)+(c-d)+(b-c)+(b-d)\right) \\ &= \frac{pn}{3}(a+d) + 3rnb - rna - 2rnd \\ &= 3\left(\frac{pn}{9}a + rnb\right) + \left(\frac{pn}{9}d - rna\right) + 2\left(\frac{pn}{9}d - rnd\right) \\ &\geq 3rn(a+b) + \left(\frac{pn}{9} - rn\right)a + 2\left(\frac{pn}{9} - rn\right)d \\ &\geq 3rn^2.\end{aligned}$$

On the other hand, for the case of (1-ii), since $d \leq a$ implies $b \leq c$, we have

$$\begin{aligned}\mathrm{unsat}_t(H) &> \frac{pn}{3}(a+d) + rn\left((b-a)+(c-d)+(a-d)+(b-d)\right) \\ &= \frac{pn}{3}(a+d) + 2rnb + rnc - 3rnd \\ &= 3\left(\frac{pn}{9}a + rnb\right) + rn(c-b) + 3\left(\frac{pn}{9}d - rnd\right) \\ &\geq 3rn(a+b) + rn(c-b) + 3\left(\frac{pn}{9} - rn\right)d \\ &\geq 3rn^2.\end{aligned}$$

For the case of (2), the condition of $b \leq a$ and $c \leq d$ implies $c \leq a$ and $b \leq d$ because of $a+b=c+d=n$. We further divide this case into two sub-cases: (2-i) $b \leq c$ and (2-ii) $c \leq b$. For the case of (2-i), since $b \leq c$ implies $d \leq a$, we have

$$\begin{aligned}|\mathrm{unsat}_t(H)| &> \frac{pn}{3}(b+c) + rn\left((a-b)+(d-c)+(a-c)+(a-d)\right) \\ &= \frac{pn}{3}(b+c) + 3rna - rnb - 2rnc\end{aligned}$$

$$\begin{aligned}
&= 3\left(\frac{pn}{9}b + rna\right) + \left(\frac{pn}{9}c - rnb\right) + 2\left(\frac{pn}{9}c - rnc\right) \\
&\geq 3rn(a+b) + \left(\frac{pn}{9} - rn\right)b + 2\left(\frac{pn}{9} - rn\right)c \\
&\geq 3rn^2.
\end{aligned}$$

On the other hand, for the case of (2-ii), since $c \leq b$ implies $a \leq d$, we have

$$\begin{aligned}
|\mathrm{unsat}_t(H)| &> \frac{pn}{3}(b+c) + rn\left((a-b) + (d-c) + (a-c) + (b-c)\right) \\
&= \frac{pn}{3}(b+c) + 2rna + rnd - 3rnc \\
&= 3\left(\frac{pn}{9}b + rna\right) + rn(d-a) + 3\left(\frac{pn}{9}c - rnc\right) \\
&\geq 3rn(a+b) + rn(d-a) + 3\left(\frac{pn}{9} - rn\right)c \\
&\geq 3rn^2.
\end{aligned}$$

For the case of (3), the condition of $b \leq a$ and $d \leq c$ implies $d \leq a$ and $b \leq c$ because of $a + b = c + d = n$. We further divide this case into two sub-cases: (3-i) $b \leq d$ and (3-ii) $d \leq b$. For the case of (3-i), since $b \leq d$ implies $c \leq a$, we have

$$\begin{aligned}
|\mathrm{unsat}_t(H)| &> \frac{pn}{3}(b+d) + rn\left((a-b) + (c-d) + (a-c) + (a-d)\right) \\
&= \frac{pn}{3}(b+d) + 3rna - rnb - 2rnd \\
&= 3\left(\frac{pn}{9}b + rna\right) + \left(\frac{pn}{9}d - rnb\right) + 2\left(\frac{pn}{9}d - rnd\right) \\
&\geq 3rn(a+b) + \left(\frac{pn}{9} - rn\right)b + 2\left(\frac{pn}{9} - rn\right)d \\
&\geq 3rn^2.
\end{aligned}$$

On the other hand, for the case of (3-ii), since $d \leq b$ implies $a \leq c$, we have

$$\begin{aligned}
|\mathrm{unsat}_t(H)| &= \frac{pn}{3}(b+d) + rn\left((a-b) + (c-d) + (a-d) + (b-d)\right) \\
&= \frac{pn}{3}(b+d) + 2rna + rnc - 3rnd \\
&= 3\left(\frac{pn}{9}b + rna\right) + rn(c-a) + 3\left(\frac{pn}{9}d - rnd\right) \\
&> 3rn(a+b) + rn(c-a) + 3\left(\frac{pn}{9} - rn\right)d \\
&\geq 3rn^2.
\end{aligned}$$

Complete Proof of Lemma 3.1

We present the value of $\mathrm{E}[b_i]$ for $i \in U_2 \backslash \{n+1\}$. This completes the proof of Lemma 3.1. For any $i \in U_2 \backslash \{n+1\}$, we have $b_i = b_i^- + b_i^+$, where

$$-b_i^- = \sum_{j \in U_1} B_{ij}A_{j1} + \sum_{j \in \widetilde{U}_1} B_{ij}B_{j1} + \sum_{j \in U_2} A_{ij}B_{j1} + \sum_{j \in \widetilde{U}_2} B_{ij}B_{j1}$$

$$+\sum_{j\in U_2} A_{ij}A_{jn+1}+\sum_{j\in\widetilde{U}_2} B_{ij}B_{jn+1}+\sum_{j\in U_1} B_{ij}B_{jn+1}+\sum_{j\in\widetilde{U}_1} B_{ij}B_{jn+1}, \quad (5)$$

and

$$\begin{aligned} b_i^+ &= \sum_{j\in U_1} B_{-ij}A_{j1}+\sum_{j\in\widetilde{U}_1} B_{-ij}B_{j1}+\sum_{j\in U_2} B_{-ij}B_{j1}+\sum_{j\in\widetilde{U}_2} A_{-ij}B_{j1} \\ &\quad +\sum_{j\in U_2} B_{-ij}A_{jn+1}+\sum_{j\in\widetilde{U}_2} A_{-ij}B_{jn+1}+\sum_{j\in U_1} B_{-ij}B_{jn+1}+\sum_{j\in\widetilde{U}_1} B_{-ij}B_{jn+1}. \quad (6)\end{aligned}$$

By the same argument as the case for $i \in U_1\backslash\{1\}$, we have $\mathrm{E}[b_i] = -((p-r)^2 n - 2p(p-r))$.

Some variants of the Chernoff bound

We present some variants of the Chernoff bound we use in the main section. We follow the argument in [MR95] and in the part by C. McDiarmid of [HM98].

Proposition 4.1. Let $Y_1, \cdots, Y_k$ be independent random variables such that $0 \le Y_i \le 1$ and $\mathrm{E}[Y_i] = \mu_i$ for $1 \le i \le k$. Then, for $Y \stackrel{\text{def}}{=} \sum_i Y_i$, $\mu \stackrel{\text{def}}{=} \mathrm{E}[Y]$, and for any $\delta > 0$,

$$\Pr\{Y > (1+\delta)\mu\} < \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^\mu .$$

Proof. We make use of the following relationship: $Y > (1+\delta)\mu \Leftrightarrow \exp(tY) > \exp(t(1+\delta)\mu)$ for any $t > 0$. By Markov inequality and the independence of $\{Y_1, \cdots, Y_k\}$, we have

$$\begin{aligned}\Pr\{Y > (1+\delta)\mu\} &= \Pr\{\exp(tY) > \exp(t(1+\delta)\mu)\} \\ &< \frac{\mathrm{E}[\prod_i \exp(tY_i)]}{\exp(t(1+\delta)\mu)} \\ &= \frac{\prod_i \mathrm{E}[\exp(tY_i)]}{\exp(t(1+\delta)\mu)}.\end{aligned}$$

We study the numerator of RHS of the above. Suppose that Y_i takes $y_1, y_2 \cdots \in [0,1]$ with probability $p_1, p_2 \cdots$, respectively. Then, we have

$$\begin{aligned}\mathrm{E}[\exp(tY_i)] &= \exp(ty_1)p_1 + \exp(ty_2)p_2 + \cdots \\ &< (1 - y_1 + y_1 e^t)p_1 + (1 - y_2 + y_2 e^t)p_2 + \cdots .\end{aligned}$$

The last inequality comes from $e^{ty} \le 1 - y + ye^t$ for any $0 \le y \le 1$. (This fact comes from the convexity of e^{ty}.) It follows that

$$\begin{aligned}\mathrm{E}[\exp(tY_i)] &< (p_1 + p_2 + \cdots) - (y_1p_1 + y_2p_2 + \cdots) + e^t(y_1p_1 + y_2p_2 + \cdots) \\ &= 1 + (e^t - 1)\mu_i \\ &\le \exp((e^t - 1)\mu_i).\end{aligned}$$

The last inequality comes from $1 + x \le e^x$ with $x = \mu_i(e^t - 1)$. Thus,

$$\Pr\{Y > (1+\delta)\mu\} < \frac{\prod_i \exp((e^t-1)\mu_i)}{\exp(t(1+\delta)\mu)}$$

$$
\begin{aligned}
&= \frac{\exp((e^t-1)\mu)}{\exp(t(1+\delta)\mu)} \\
&= \left(\frac{\exp(e^t-1)}{\exp(t(1+\delta))}\right)^{\mu}.
\end{aligned}
$$

Differentiating the last formula, we see that it is minimized by setting $t = \ln(1+\delta)$. We notice that $t > 0$ iff $\delta > 0$. Substituting this value for t, we obtain the proposition. ■

Corollary 4.2. If we have $\Pr\{Y_i = 1\} = p_i$ and $\Pr\{Y_i = 0\} = 1 - p_i$ for $1 \le i \le k$, then we have the same bound as the proposition above. In particular, for any $\delta > 0$,

$$
\Pr\{Y > (1+\delta)\mu\} \;<\; \left(\frac{e}{1+\delta}\right)^{(1+\delta)\mu}.
$$

Proposition 4.3. Let $Y_1, \cdots, Y_k$ be independent random variables such that $0 \le Y_i \le 1$ and $\mathrm{E}[Y_i] = \mu_i$ for $1 \le i \le k$. Then, for $Y \stackrel{\mathrm{def}}{=} \sum_i Y_i$, $\mu \stackrel{\mathrm{def}}{=} \mathrm{E}[Y]$, and for any $0 < \delta \le 1$,

$$
\Pr\{Y < (1-\delta)\mu\} \;<\; \exp\left(-\frac{\delta^2\mu}{2}\right)
$$

Proof. We almost follow the proof of the previous proposition. In this case, we make use of the following relationship: $Y < (1-\delta)\mu \Leftrightarrow \exp(-tY) > \exp(-t(1-\delta)\mu)$ for any $t > 0$. By Markov inequality and the independence of $\{Y_1, \cdots, Y_k\}$, we have

$$
\begin{aligned}
\Pr\{Y < (1-\delta)\mu\} &= \Pr\{\exp(-tY) > \exp(-t(1-\delta)\mu)\} \\
&< \frac{\prod_i \mathrm{E}[\exp(-tY_i)]}{\exp(-t(1-\delta)\mu)}.
\end{aligned}
$$

From the proof of the previous proposition, $\mathrm{E}[\exp(-tY_i)] < \exp((e^{-t}-1)\mu_i)$. It follows that

$$
\begin{aligned}
\Pr\{Y < (1-\delta)\mu\} &< \frac{\exp((e^{-t}-1)\mu)}{\exp(-t(1-\delta)\mu)} \\
&= \left(\frac{\exp(e^{-t}-1)}{\exp(-t(1-\delta))}\right)^{\mu}.
\end{aligned}
$$

Differentiating the last formula, we see that it is minimized by setting $t = \ln(1/(1-\delta))$. We notice that $t > 0$ iff $\delta > 0$. Substituting this value for t, we obtain the following:

$$
\Pr\{Y < (1-\delta)\mu\} \;<\; \left(\frac{\exp(-\delta)}{(1-\delta)^{(1-\delta)}}\right)^{\mu}.
$$

Simplifying the formula above by the fact that $(1-\delta)^{(1-\delta)} > \exp(-\delta + \delta^2/2)$ for $0 < \delta \le 1$ (this comes from $\ln(1-\delta) = -(\delta + \delta^2/2 + \delta^3/3 + \cdots)$), we obtain the proposition. ■

Corollary 4.4. If we have $\Pr\{Y_i = 1\} = p_i$ and $\Pr\{Y_i = 0\} = 1 - p_i$ for $1 \le i \le k$, then we have the same bound as the proposition above.

Corollary 4.5. Let $Y_1, \cdots, Y_k$ be independent random variables such that $c \stackrel{\text{def}}{=} \max_i\{Y_i\}$ and $\mathrm{E}[Y_i] = \mu_i$ for $1 \le i \le k$. Then, for $Y \stackrel{\text{def}}{=} \sum_i Y_i$, $\mu \stackrel{\text{def}}{=} \mathrm{E}[Y]$, and for any $0 < \delta \le 1$,

$$\Pr\{Y < (1-\delta)\mu\} \; < \; \exp\left(-\frac{\delta^2\mu}{2c}\right)$$

Proof. We set $Y_i' = Y_i/c$ for all $1 \le i \le k$, and let $Y' \stackrel{\text{def}}{=} \sum_i Y_i'$ and $\mu' = \mathrm{E}[Y']$ so that we can apply the previous proposition to Y'. Note that $\mu' = \mu/c$. Then, we have

$$\begin{aligned}\Pr\{Y < (1-\delta)\mu\} &= \Pr\{Y' < (1-\delta)\mu'\} \\ &< \exp\left(-\frac{\delta^2\mu'}{2}\right) \\ &= \exp\left(-\frac{\delta^2\mu}{2c}\right).\end{aligned}$$

■