SIAM J. DISC. MATH.
Vol. 2, No. 4, pp. 508–523, November 1989

009

AVERAGE PERFORMANCE OF HEURISTICS FOR SATISFIABILITY*

RAJEEV KOHLI† AND RAMESH KRISHNAMURTI‡

Abstract. Distribution-free tight lower bounds on the average performance ratio for random search, for a greedy heuristic and for a probabilistic greedy heuristic are derived for an optimization version of satisfiability. On average, the random solution is never worse than $\frac{1}{2}$ of the optimal, regardless of the data-generating distribution. The lower bound on the average greedy solution is at least $\frac{1}{2}$ of the optimal, and this bound increases with the probability of the greedy heuristic selecting the optimal at each step. In the probabilistic greedy heuristic, probabilities are introduced into the search strategy so that a decrease in the probability of finding the optimal solution occurs only if the nonoptimal solution becomes closer to the optimal. Across problem instances, and regardless of the distribution giving rise to data, the minimum average value of the solutions identified by the probabilistic greedy heuristic is no less than $\frac{2}{3}$ of the optimal.

Key words. satisfiability, greedy heuristics, probabilistic heuristics, average performance

AMS(MOS) subject classification. 68Q25

1. Introduction.

This paper examines the average performance of random search, of a greedy heuristic and of a probabilistic version of a greedy heuristic for an optimization version of satisfiability. We derive tight lower bounds on the average performance of each heuristic. The analysis assumes no specific data-generating distributions and therefore is valid for all distributions.

A variety of analytic approaches have recently been pursued to analyze the average-case performance of heuristics. These include representing the execution of algorithms by Markov chains (Coffman, Leuker, and Rinnooy Kan [5]), obtaining the performance bound for a more tractable function that dominates the performance of the heuristic for each problem instance (Bruno and Downey [3], Boxma [2]), and obtaining the performance bound for a simpler, more easily analyzed heuristic which dominates the heuristic of interest for each problem instance (Csirik et al. [6]). Bounds that hold for most problem instances have also been employed to obtain asymptotic bounds for the average-case performance of various heuristics (Bentley et al. [1] and Coffman and Leighton [4]). A number of results from applied probability theory have been used for average-case analyses by Frenk and Rinnooy Kan [10], Karp, Luby, and Marchetti-Spaccamela [14], Shor [17], and Leighton and Shor [15]. The vast majority of these approaches begins by assuming independent, identically distributed data from a given density function. The subsequent analyses are often difficult, and one rarely finds an explicit formula for the quantity of interest. One reason for this is that conditional probabilities arise in the analyses, and after a sufficient number of steps, the conditioning can make the analyses formidable. Appropriate choice of distributional assumptions also is difficult, as are inferences regarding the robustness of results for a given distribution to other distributions.

A well-known algorithm for solving satisfiability is the Davis–Putnam Procedure (Davis, Logemann, and Loveland [7]). Goldberg, Purdom, and Brown [11], and Franco and Paull [9] have analyzed the average-case *complexity* of variants of this procedure for solving satisfiability. Johnson [13] considers an optimization version of satisfiability, called maximum satisfiability, proposes two heuristics for solving the maximum satisfiability problem, and proves tight worst-case bounds on the performances of these heu-

* Received by the editors December 15, 1988; accepted for publication February 28, 1989. This research was supported by the Natural Sciences and Engineering Research Council of Canada under grant OGP0036809.

† Graduate School of Business, University of Pittsburgh, Pittsburgh, Pennsylvania 15260.

‡ School of Computing Science, Simon Fraser University, Burnaby, British Columbia, V5A 1S6, Canada.

ristics. One of these heuristics is the greedy heuristic that we use in this paper. If each clause contains at least l variables, Johnson [13] shows a tight worst-case bound of $l/(l+1)$ for the greedy heuristic. Since we consider the most general optimization version of satisfiability, where unary clauses (clauses with just one variable) are allowed, this bound reduces to $\frac{1}{2}$. As one of our results, we derive this bound using a different approach. Lieberherr and Specker [16] provide the best possible polynomial-time algorithm for the maximum satisfiability problem where unary clauses are allowed, but the set of clauses must be 2-satisfiable, i.e., any two of the clauses are simultaneously satisfiable. The lower bound obtained for their algorithm is 0.618.

In the present analyses, we consider the lower bound of the average performance making no assumption regarding the data-generating distribution. For two of the three procedures (random search and the probabilistic greedy heuristic), we also make no assumption regarding the independence of data. For the third (the greedy heuristic), we assume independence, but only in a certain "aggregate" sense, which we discuss later. Each of the bounds we obtain is tight. Our central results are as follows. Random search, which has an arbitrarily bad performance in the worst case, provides solutions that, on average, are never worse than $\frac{1}{2}$ of the optimal. The greedy heuristic can potentially improve on this performance. Although the lower bound on its average performance ratio can be $\frac{1}{2}$ of the optimal, this lower bound increases with the probability of the heuristic selecting the optimal at each step. A probabilistic algorithm related to the greedy heuristic is then described. The probabilities are introduced into the search strategy so that a decrease in the probability of finding the optimal solution occurs only if the non-optimal solution becomes closer to the optimal. The search probabilities are not fixed a priori but exploit the structure of the data to force a trade-off for every problem instance. Across problem instances, and regardless of the distribution giving rise to the data, the average performance of the algorithm is never less than $\frac{2}{3}$ of the optimal.

Section 2 describes the maximum satisfiability problem, the random search procedure, and obtains a tight lower bound on its average performance. Section 3 introduces the greedy heuristic, derives its worst-case bound, and a tight lower bound on its average performance. Section 4 describes the probabilistic greedy heuristic and derives a tight lower bound on its average performance.

2. The Msat problem. Consider the following optimization version of satisfiability: given n clauses, each described by a disjunction of a subset of k variables or their negations, find a truth assignment for the variables that maximizes the number of clauses satisfied. The above problem, which is the most general version of maximum satisfiability, is NP-complete (Johnson [13]). We call this Msat.

We use the following tabular representation of Msat. For a problem involving n clauses and k variables, construct a table T_k with n rows and $2k$ columns. The ith row is associated with clause i, $i = 1, \cdots, n$. A pair of columns, u_j, $\bar{u}_j$, is associated with the jth variable, $j = 1, \cdots, k$. Let t_{ij} denote the entry in the cell identified by row i and column u_j, and let $\bar{t}_{ij}$ denote the entry in the cell identified by row i and column $\bar{u}_j$. For $i = 1, \cdots, n, j = 1, \cdots, k$, define

$$\begin{cases} t_{ij} = 1, \bar{t}_{ij} = 0, & \text{if clause } i \text{ contains variable } j, \\ t_{ij} = 0, \bar{t}_{ij} = 1, & \text{if clause } i \text{ contains the negation of variable } j, \\ t_{ij} = 0, \bar{t}_{ij} = 0, & \text{if clause } i \text{ contains neither variable } j \text{ nor its negation.} \end{cases}$$

A truth assignment for satisfiability results in the jth variable being assigned a T (*True*) or an F (*False*), $j = 1, \cdots, k$. This corresponds to selecting either column u_j

(if the jth variable is assigned a T) or $\bar{u}_j$ (if the jth variable is assigned an F), $j = 1, \cdots, k$, for Msat. Consequently, selecting u_j or $\bar{u}_j$ for each $j, j = 1, \cdots, k$, such that the maximum number of rows in these columns have at least one 1, corresponds to solving Msat for T_k; i.e., finding a truth assignment that maximizes the number of clauses satisfied.

Let $T(u_k)(T(\bar{u}_k))$ denote the table obtained by deleting from T_k all rows with a 1 in column $u_k(\bar{u}_k)$, and deleting *both* u_k and $\bar{u}_k$. Let the resulting table be denoted T_{k-1}. That is,

$$T_{k-1} = \begin{cases} T(u_k), & \text{if column } u_k \text{ is chosen from } T_k, \\ T(\bar{u}_k), & \text{if column } \bar{u}_k \text{ is chosen from } T_k. \end{cases}$$

In general, let $T(u_j)(T(\bar{u}_j))$ denote the table obtained by deleting from T_j all rows with a 1 in column $u_j(\bar{u}_j)$, and deleting both u_j and $\bar{u}_j, j = 1, \cdots, k$. Let the resulting table be denoted T_{j-1}. That is,

$$T_{j-1} = \begin{cases} T(u_j), & \text{if column } u_j \text{ is chosen from } T_j, \\ T(\bar{u}_j), & \text{if column } \bar{u}_j \text{ is chosen from } T_j. \end{cases}$$

Let x_j denote the number of 1's in u_j and let n_j denote the total number of 1's across columns u_j and $\bar{u}_j$ in table $T_j, j = 1, \cdots, k$. Without loss of generality, assume that the columns $u_j, j = 1, \cdots, k$, comprise the optimal solution for Msat described by T_k. Let m_k denote the optimal solution to Msat described by T_k. In general, let m_j denote the value of the optimal solution to Msat described by $T_j, j = 1, \cdots, k$. Also, let $a_j(\bar{a}_j)$ denote the value of the optimal solution to Msat described by $T(u_j)(T(\bar{u}_j)), j = 1, \cdots, k$. That is,

$$m_{j-1} = \begin{cases} a_j, & \text{if column } u_j \text{ is chosen from } T_j, \\ \bar{a}_j, & \text{if column } \bar{u}_j \text{ is chosen from } T_j. \end{cases}$$

Example 1. Consider a problem consisting of three variables x_1, x_2, and x_3 and seven clauses given by $x_1 + x_2 + x_3$, $\bar{x}_1 + x_2 + \bar{x}_3$, $x_1 + \bar{x}_2 + \bar{x}_3$, $\bar{x}_1 + x_3$, $x_2 + \bar{x}_3$, $x_1 + x_2$, and $\bar{x}_1$. Table T_3 for this problem is given in Fig. 1.

For the table T_3 above, table $T(u_3)$ is obtained by deleting rows 1 and 4 and the columns u_3 and $\bar{u}_3$. Table $T_2 = T(u_3)$ is given in Fig. 2. Similarly, we can obtain table $T(\bar{u}_3)$ by deleting rows 2, 3, and 5 and the columns u_3 and $\bar{u}_3$. Table $T_2 = T(\bar{u}_3)$ is given in Fig. 3.

Consider a random procedure that selects column u_j or $\bar{u}_j$ with probability $\frac{1}{2}, j = 1, \cdots, k$. The procedure can easily be seen to select an arbitrarily bad solution in the

Row	u_3	$\bar{u}_3$	u_2	$\bar{u}_2$	u_1	$\bar{u}_1$
1	1	0	1	0	1	0
2	0	1	1	0	0	1
3	0	1	0	1	1	0
4	1	0	0	0	0	1
5	0	1	1	0	0	0
6	0	0	1	0	1	0
7	0	0	0	0	0	1

FIG. 1. *Table T_3 for Example* 1.

Row	u_2	$\bar{u}_2$	u_1	$\bar{u}_1$
1	1	0	0	1
2	0	1	1	0
3	1	0	0	0
4	1	0	1	0
5	0	0	0	1

FIG. 2. *Table* $T_2 = T(u_3)$ *for Example* 1.

worst case. But how poorly does it do on the average? That is, if *any* data-generating mechanism is used to construct instances of Msat, and if for each problem instance a random solution is selected, what is the average value of the ratio of the random solution value to the optimal solution value? Theorem 1 shows that it is never less than $\frac{1}{2}$. That is, if f_k is the random solution to Msat described by T_k, then the ratio $r_k = f_k/m_k$ has an expected value $E[r_k] \geqq \frac{1}{2}$. We begin by proving the following lemma.

LEMMA 1. $a_j = m_j - x_j$ *and* $\bar{a}_j \geqq \max\{0, m_j - n_j\}, j = 1, \cdots, k$.

Proof. As u_j, the optimal column for Msat described by table T_j, has x_j 1's, the optimal solution value to Msat described by table $T(u_j)$ is, trivially, $a_j = m_j - x_j$. Also, $x_j \leqq n_j$ (by definition), so that $m_j - x_j \geqq m_j - n_j$. Of the $m_j - x_j$ rows with at least one 1 in $T(u_j)$, at most $n_j - x_j$ can also have 1's in column $\bar{u}_j$ of T_j. Hence the Msat problem described by $T(\bar{u}_j)$ has an optimal solution with value $\bar{a}_j$ no smaller than $m_j - x_j - (n_j - x_j) = m_j - n_j$. As n_j can exceed m_j, and as the value of the optimal solution to Msat described by T_j is nonnegative, $\bar{a}_j \geqq \max\{0, m_j - n_j\}, j = 1, \cdots, k$. □

THEOREM 1. $E[r_k] \geqq \frac{1}{2}$ *for all* k.

Proof. We prove the theorem by induction on the number of variables.

Base case. $E[r_1] \geqq \frac{1}{2}$.

As each column of T_1 is selected with probability $\frac{1}{2}$, the expected value of the random solution is

$$E[r_1] = \frac{1}{2}x_1 + \frac{1}{2}(n_1 - x_1) = \frac{n_1}{2}.$$

As the optimal solution value, corresponding to u_1, is $m_1 = x_1 \leqq n_1$, the expected performance ratio is

$$E[r_1] = \frac{(n_1/2)}{x_1} \geqq \frac{(n_1/2)}{n_1} = \frac{1}{2}.$$

Induction hypothesis. $E[r_l] \geqq \frac{1}{2}$ for all $l \leqq k - 1$.

Induction step. To prove $E[r_k] \geqq \frac{1}{2}$.

Row	u_2	$\bar{u}_2$	u_1	$\bar{u}_1$
1	1	0	1	0
2	0	0	0	1
3	1	0	1	0
4	0	0	0	1

FIG. 3. *Table* $T_2 = T(\bar{u}_3)$ *for Example* 1.

By the induction hypothesis, the random procedure applied to table $T(u_k)(T(\bar{u}_k))$ has an expected solution value no smaller than $\frac{1}{2}a_k(\frac{1}{2}\bar{a}_k)$. Hence the random solution, for table T_k, has an expected solution

$$E[f_k] \geqq \frac{1}{2}\left(x_k + \frac{a_k}{2}\right) + \frac{1}{2}\left(n_k - x_k + \frac{\bar{a}_k}{2}\right).$$

As $a_k = m_k - x_k$ by Lemma 1,

$$E[f_k] \geqq \frac{1}{2}\left(x_k + \frac{m_k - x_k}{2}\right) + \frac{1}{2}\left(n_k - x_k + \frac{\bar{a}_k}{2}\right).$$

Also, $\bar{a}_k \geqq \max\{0, m_k - n_k\}$ by Lemma 1. Consider $m_k > n_k$. Then $\bar{a}_k \geqq m_k - n_k > 0$, and the above inequality for $E[f_k]$ simplifies to

$$E[f_k] \geqq \frac{1}{2}\left(x_k + \frac{m_k - x_k}{2}\right) + \frac{1}{2}\left(n_k - x_k + \frac{m_k - n_k}{2}\right),$$

or

$$E[f_k] \geqq \frac{m_k}{2} + \frac{n_k}{4} - \frac{x_k}{4}.$$

The right-hand side attains the least value at $x_k = n_k$, at which value

$$E[f_k] \geqq \frac{m_k}{2},$$

and

$$E[r_k] = \frac{E[f_k]}{m_k} \geqq \frac{1}{2}.$$

Now consider $m_k \leqq n_k$. Then $\bar{a}_k \geqq 0$ $(\geqq m_k - n_k)$, and hence

$$E[f_k] \geqq \frac{1}{2}\left(x_k + \frac{m_k - x_k}{2}\right) + \frac{1}{2}(n_k - x_k).$$

Simplifying,

$$E[f_k] \geqq \frac{m_k}{4} + \frac{n_k}{2} - \frac{x_k}{4}.$$

Since $m_k \leqq n_k$, the right-hand side attains the least value at $x_k = n_k = m_k$, at which value

$$E[f_k] \geqq \frac{m_k}{2},$$

and $E[r_k] = E[f_k]/m_k \geqq \frac{1}{2}$. □

The lower bound on the average value $E[r_k]$ of r_k, is tight and is illustrated by the example in Fig. 4. Assume that the data pattern shown in the figure is generated each time; i.e., the data-generating mechanism presents the same pattern with x 1's in column u_2 and $(n_2 - x)$ 1's in column u_1, where x can range from 1 to n_2. The probability of a particular value of x for a problem instance is determined by the distribution of the random variable x. The average performance of the random solution is the average of the performance ratio across the four solutions that can be selected, each solution being selected with probability $\frac{1}{4}$. The average performance ratio can be verified to be $\frac{1}{2}$, which is the lower bound on the expected performance ratio for random search.

Row	u_2	$\bar{u}_2$	u_1	$\bar{u}_1$
1	1	0	0	0
·	·	·	·	·
·	·	·	·	·
x	1	0	0	0
$x+1$	0	0	1	0
·	·	·	·	·
·	·	·	·	·
n_2	0	0	1	0

FIG. 4. *Worst case example for random procedure. x is a discrete random variable ranging from 1 to n_2.*

3. The greedy heuristic. Random search, of course, appears to be a simplistic procedure for solving the problem. A greedy heuristic that selects columns based on the number of 1's they contain is presented next (Johnson [13]), and its worst-case performance and average performance are analyzed. We begin by describing the greedy heuristic.

Initialization. Order the columns of T_k so that n_k, the number of 1's across u_k and $\bar{u}_k$, is largest among all pairs of columns u_l, $\bar{u}_l$, $l = 1, \cdots, k$ (the ordering plays no role in the analysis and is used merely to detect the termination of the algorithm efficiently). If $x_k \geqq n_k - x_k$, select column u_k; otherwise, select column $\bar{u}_k$. Eliminate u_k and $\bar{u}_k$, and all rows with a 1 in the chosen column. Note that the resulting table is denoted by T_{k-1}. That is,

$$T_{k-1} = \begin{cases} T(u_k), & \text{if column } u_k \text{ is chosen from } T_k, \\ T(\bar{u}_k), & \text{if column } \bar{u}_k \text{ is chosen from } T_k. \end{cases}$$

Recursion. Order the columns of T_j and rename the variables u_1 through u_j so that n_j, the number of 1's across u_j and $\bar{u}_j$, is largest among all pairs of columns u_l, $\bar{u}_l$, $l = 1, \cdots, j$. If $x_j \geqq n_j - x_j$, select column u_j; otherwise, select column $\bar{u}_j$. Eliminate u_j and $\bar{u}_j$, and all rows with a 1 in the chosen column. Again, note that the resulting table is denoted by T_{j-1}. That is,

$$T_{j-1} = \begin{cases} T(u_j), & \text{if column } u_j \text{ is chosen from } T_j, \\ T(\bar{u}_j), & \text{if column } \bar{u}_j \text{ is chosen from } T_j. \end{cases}$$

Termination. Stop if T_j contains no 1's, or if $j = 0$.

Note that

$$m_{j-1} = \begin{cases} a_j, & \text{if } T_{j-1} = T(u_j), \\ \bar{a}_j, & \text{if } T_{j-1} = T(\bar{u}_j), \end{cases}$$

denotes the value of the optimal solution to Msat described by T_{j-1}, $j = 1, \cdots, k$. Let f_j denote the value of the greedy solution, and let $r_j = f_j/m_j$ denote the performance ratio of the greedy heuristic for Msat described by T_j. Theorem 2 shows that the worst-case bound for the greedy heuristic is $\frac{1}{2}$ of the optimal. We also show by an example that this lower bound on the performance of the greedy heuristic is tight. We begin by proving the following lemma.

LEMMA 2. $m_{j-1} \geqq m_j - n_j$, $j = 1, \cdots, k$.

Proof. By Lemma 1, $a_j = m_j - x_j$. As $x_j \leqq n_j$ (by definition), $a_j \geqq m_j - n_j$. Also, $\bar{a}_j \geqq \max\{0, m_j - n_j\} \geqq m_j - n_j$ by Lemma 1. As m_{j-1}, the value of the optimal solution to Msat described by T_{j-1}, is either a_j or $\bar{a}_j$, it follows that $m_{j-1} \geqq m_j - n_j$. □

THEOREM 2. *$r_k \geq \frac{1}{2}$ for all k.*

Proof. We prove the theorem by induction on the number of variables.

Base case. $r_1 \geq \frac{1}{2}$.

The single-variable problem is described by table T_1 with column u_1 containing x_1 1's, and column $\bar{u}_1$ containing $(n_1 - x_1)$ 1's. The greedy heuristic selects the column with more 1's, which also is the optimal column. Thus

$$f_1 = m_1 = \max\{x_1, n_1 - x_1\},$$

and hence

$$r_1 = \frac{f_1}{m_1} = 1 \geq \frac{1}{2}.$$

Induction hypothesis. $r_j \geq \frac{1}{2}$ for all $j \leq k - 1$.

Induction step. To prove $r_k \geq \frac{1}{2}$.

At the first step, the greedy heuristic chooses u_k or $\bar{u}_k$, whichever has the larger number of 1's. Hence

$$f_k = \max\{x_k, n_k - x_k\} + f_{k-1}.$$

By the induction hypothesis, $r_{k-1} \geq \frac{1}{2}$, so that

$$f_{k-1} = r_{k-1} m_{k-1} \geq \frac{1}{2} m_{k-1}.$$

Hence,

$$f_k \geq \max\{x_k, n_k - x_k\} + \frac{1}{2} m_{k-1}.$$

As the maximum of two numbers is no smaller than their mean, $\max\{x_k, n_k - x_k\} \geq \frac{1}{2} n_k$. Also, by Lemma 2, $m_{k-1} \geq m_k - n_k$. Thus,

$$f_k \geq \frac{1}{2} n_k + \frac{1}{2}(m_k - n_k) = \frac{m_k}{2}.$$

Thus $r_k = f_k / m_k \geq \frac{1}{2}$. □

The bound specified by Theorem 2 is tight, and is illustrated by the example in Fig. 5. The optimal solution is $m_k = 2^k$, corresponding to columns u_j, $j = 1, \cdots, k$. The greedy solution is $f_k = 2^{k-1} + 1$, corresponding to column $\bar{u}_k$. Hence $r_k = f_k / m_k = \frac{1}{2} + \varepsilon$, where $\varepsilon = 1/2^k$. Since ε can be made to approach 0 arbitrarily closely by increasing k, r_k can be made to approach $\frac{1}{2}$ from above arbitrarily closely, giving rise to an asymptotic upper bound of $\frac{1}{2}$ for the worst-case performance of the greedy heuristic. Observe that the worst-case bound for the greedy heuristic equals the average-case bound for the random solution.

We are now ready to prove the lower bound on the average performance of the greedy heuristic. Assume that a probabilistic data-generating mechanism is used to obtain instances of Msat. Specifically, assume that the mechanism generates a larger number of 1's in u_j with probability p, and generates a larger number of 1's in $\bar{u}_j$ with probability $(1 - p)$, $j = 1, \cdots, k$. Note that we assume that p does not vary with j. However, we make no distributional assumptions about the data-generating process. Theorem 3 characterizes the lower bound on the average performance ratio for the greedy heuristic.

THEOREM 3.

$$E[r_k] \geq \frac{1}{2 - p}.$$

Row	u_k	$\bar{u}_k$	u_{k-1}	$\bar{u}_{k-1}$	· ·	u_2	$\bar{u}_2$	u_1	$\bar{u}_1$
1	0	1	0	1	· ·	0	1	0	1
2	0	1	0	1	· ·	1	0	0	0
3	0	1	0	1	· ·	0	0	0	0
·	·	·	·	·	· ·	·	·	·	·
·	·	·	·	·	· ·	·	·	·	·
2^{k-2}	0	1	0	1	· ·	0	0	0	0
$2^{k-2}+1$	0	1	1	0	· ·	0	0	0	0
·	·	·	·	·	· ·	·	·	·	·
·	·	·	·	·	· ·	·	·	·	·
·	·	·	·	·	· ·	·	·	·	·
2^{k-1}	0	1	1	0	· ·	0	0	0	0
$2^{k-1}+1$	1	0	0	0	· ·	0	0	0	0
·	·	·	·	·	· ·	·	·	·	·
·	·	·	·	·	· ·	·	·	·	·
·	·	·	·	·	· ·	·	·	·	·
2^k	1	0	0	0	· ·	0	0	0	0
2^k+1	0	1	0	1	· ·	0	1	0	1

FIG. 5. *Worst case example for the greedy heuristic with* $n_k = 2^k + 1$ *clauses.*

Proof. We prove the theorem by induction on the number of variables.

Base case. $E[r_1] \geqq 1/(2-p)$.

For a single-variable problem, the optimal column u_1 has at least as many 1's as the nonoptimal column $\bar{u}_1$. Therefore the value of the greedy solution equals the number of 1's in the optimal column. Thus, the expected performance ratio of the greedy heuristic is

$$E[r_1] = \frac{E[f_1]}{m_1} = 1 \geqq \frac{1}{2-p} \quad \text{for any } p, \qquad 0 \leqq p \leqq 1.$$

Induction hypothesis. $E[r_l] \geqq 1/(2-p)$ for all $l \leqq k-1$.

Induction step. To prove $E[r_k] \geqq 1/(2-p)$.

If the greedy heuristic selects column u_k from T_k, it guarantees a solution value of at least x_k. In addition, table $T_{k-1} = T(u_k)$, generated at the first step, describes an Msat problem for which the expected value of the greedy solution is, by the induction hypothesis, no less than $[1/(2-p)]a_k$. Hence if column u_k is selected at step 1, the expected value of the greedy solution is no less than $x_k + [1/(2-p)]a_k$. By a similar argument, if the greedy heuristic selects $\bar{u}_k$ at step 1, the expected value of its solution is no less than $n_k - x_k + [1/(2-p)]\bar{a}_k$. Now u_k is selected with probability p, and $\bar{u}_k$ is selected with probability $(1-p)$. The expected value of the greedy solution is therefore

$$E[f_k] \geqq p\left(x_k + \frac{1}{2-p}a_k\right) + (1-p)\left(n_k - x_k + \frac{1}{2-p}\bar{a}_k\right).$$

Noting that $a_k = m_k - x_k$ by Lemma 1,

$$E[f_k] \geqq p\left(x_k + \frac{1}{2-p}(m_k - x_k)\right) + (1-p)\left(n_k - x_k + \frac{1}{2-p}\bar{a}_k\right).$$

Also, $\bar{a}_k \geqq \max\{0, m_k - n_k\}$ by Lemma 1. Consider $m_k > n_k$. Then $\bar{a}_k \geqq m_k - n_k > 0$,

and the above inequality for $E[f_k]$ becomes

$$E[f_k] \geqq p\left(x_k + \frac{1}{2-p}(m_k - x_k)\right) + (1-p)\left(n_k - x_k + \frac{1}{2-p}(m_k - n_k)\right).$$

Simplifying,

$$E[f_k] \geqq p\left(\frac{1-p}{2-p}x_k + \frac{1}{2-p}m_k\right) + (1-p)\left(n_k - x_k + \frac{1}{2-p}(m_k - n_k)\right).$$

Noting that $x_k \geqq (n_k)/2$ if column u_k is chosen and $n_k - x_k \geqq (n_k)/2$ if column $\bar{u}_k$ is chosen, we get

$$E[f_k] \geqq p\left(\frac{1-p}{2-p}\frac{n_k}{2} + \frac{1}{2-p}m_k\right) + (1-p)\left(\frac{n_k}{2} + \frac{1}{2-p}(m_k - n_k)\right)$$

which implies that

$$E[f_k] \geqq \frac{m_k}{2-p}.$$

Hence

$$E[r_k] = \frac{E[f_k]}{m_k} \geqq \frac{1}{2-p}.$$

Now consider $m_k \leqq n_k$. Then $\bar{a}_k \geqq 0$ ($\geqq m_k - n_k$), which implies

$$E[f_k] \geqq p\left(\frac{1-p}{2-p}\frac{n_k}{2} + \frac{1}{2-p}m_k\right) + (1-p)\left(\frac{n_k}{2}\right).$$

Simplifying,

$$E[f_k] \geqq \frac{pm_k + (1-p)n_k}{2-p}.$$

As $m_k \leqq n_k$, the right side of the above expression has a minimum at $n_k = m_k$. Hence

$$E[f_k] \geqq \frac{pm_k + (1-p)m_k}{2-p} = \frac{m_k}{2-p}.$$

Thus, $E[r_k] = (E[f_k])/m_k \geqq 1/(2-p)$. □

The lower bound obtained in Theorem 3 is tight. To illustrate, consider the following example involving k variables and $n_k = 2^s - 1$ rows, where $s > k$. Generate the data as follows. For $j = 1, \cdots, k$, set

$$\begin{cases} t_{i,k-j+1} = 0, \bar{t}_{i,k-j+1} = 1, & \text{if } i = 1, \cdots, 2^{s-j} - 1 \\ t_{i,k-j+1} = 1, \bar{t}_{i,k-j+1} = 0, & \text{if } i = 2^{s-j} + 1, \cdots, 2^{s-j+1} - 1 \\ t_{i,k-j+1} = 1, \bar{t}_{i,k-j+1} = 0 \text{ with probability } p, & \text{if } i = 2^{s-j} \\ t_{i,k-j+1} = 0, \bar{t}_{i,k-j+1} = 1 \text{ with probability } 1-p, & \text{if } i = 2^{s-j}. \end{cases}$$

Generate 0's in all remaining cells.

Figure 6 is an example of the data generated for $k = 2$, $n_2 = 2^3 - 1 = 7$. The data generated in this manner for arbitrary k and $n_k = 2^{k+1} - 1$ is shown in Fig. 7.

Since only one 1 is generated probabilistically in column u_j or $\bar{u}_j$, the probability of the greedy heuristic choosing u_j is p, and the probability of choosing $\bar{u}_j$ is $(1-p)$, $j = 1, \cdots, k$. Note that each row has a 1 in either column u_k or column $\bar{u}_k$. The value of

Row	u_2	$\bar{u}_2$	u_1	$\bar{u}_1$
1	0	1	0	1
2	0	1	x_1	y_1
3	0	1	1	0
4	x_2	y_2	0	0
5	1	0	0	0
6	1	0	0	0
7	1	0	0	0

FIG. 6. *Data generated for* $k = 2$, $n_2 = 2^3 - 1$. x_i y_i *equals* 1 0 *with probability* p, 0 1 *with probability* $1 - p$, *for* $i = 1, 2, \cdots, k$.

Row	u_k	$\bar{u}_k$	u_{k-1}	$\bar{u}_{k-1}$	u_{k-2}	$\bar{u}_{k-2}$	· ·	u_1	$\bar{u}_1$
1	0	1	0	1	0	1	· ·	0	1
2	0	1	0	1	0	1	· ·	x_1	y_1
3	0	1	0	1	0	1	· ·	1	0
4	0	1	0	1	0	1	· ·	0	0
·	·	·	·	·	·	·	· ·	·	·
·	·	·	·	·	·	·	· ·	·	·
$\frac{n_k+1}{8}-1$	0	1	0	1	0	1	· ·	0	0
$\frac{n_k+1}{8}$	0	1	0	1	x_{k-2}	y_{k-2}	· ·	0	0
$\frac{n_k+1}{8}+1$	0	1	0	1	1	0	· ·	0	0
·	·	·	·	·	·	·	· ·	·	·
·	·	·	·	·	·	·	· ·	·	·
$\frac{n_k+1}{4}-1$	0	1	0	1	1	0	· ·	0	0
$\frac{n_k+1}{4}$	0	1	x_{k-1}	y_{k-1}	0	0	· ·	0	0
$\frac{n_k+1}{4}+1$	0	1	1	0	0	0	· ·	0	0
·	·	·	·	·	·	·	· ·	·	·
·	·	·	·	·	·	·	· ·	·	·
·	·	·	·	·	·	·	· ·	·	·
$\frac{n_k+1}{2}-1$	0	1	1	0	0	0	· ·	0	0
$\frac{n_k+1}{2}$	x_k	y_k	0	0	0	0	· ·	0	0
$\frac{n_k+1}{2}+1$	1	0	0	0	0	0	· ·	0	0
·	·	·	·	·	·	·	· ·	·	·
·	·	·	·	·	·	·	· ·	·	·
·	·	·	·	·	·	·	· ·	·	·
n_k	1	0	0	0	0	0	· ·	0	0

FIG. 7. *Data generated for arbitrary* k *and* $n_k = 2^{k+1} - 1$. x_i y_i *equals* 1 0 *with probability* p, 0 1 *with probability* $1 - p$, *for* $i = 1, 2, \cdots, k$.

the optimal solution to Msat described by T_k is $m_k \geqq n_k - k$, because the collection of columns u_j, $j = 1, \cdots, k$, has at least $n_k - k$ nonoverlapping 1's. The expected performance of the greedy heuristic is

$$\begin{aligned} E[f_k] &= p\frac{n_k+1}{2} + (1-p)\frac{n_k+1}{2} + p\left(p\frac{n_k+1}{4} + (1-p)\frac{n_k+1}{4}\right) \\ &\quad + p^2\left(p\frac{n_k+1}{8} + (1-p)\frac{n_k+1}{8}\right) + \cdots \\ &\quad + p^{k-1}\left(p\frac{n_k+1}{2^k} + (1-p)\frac{n_k+1}{2^k}\right) \\ &= \frac{n_k+1}{2}\left(1 + \frac{p}{2} + \left(\frac{p}{2}\right)^2 + \cdots + \left(\frac{p}{2}\right)^{k-1}\right). \end{aligned}$$

Since

$$\left(1 + \frac{p}{2} + \left(\frac{p}{2}\right)^2 + \cdots + \left(\frac{p}{2}\right)^{k-1}\right) = \frac{2}{2-p}\left(1 - \left(\frac{p}{2}\right)^k\right),$$

and

$$E[r_k] = \frac{E[f_k]}{m_k},$$

we get

$$E[r_k] \leqq \frac{n_k+1}{2m_k}\frac{2}{2-p}\left(1 - \left(\frac{p}{2}\right)^k\right).$$

As $m_k \geqq n_k - k$,

$$E[r_k] \leqq \frac{n_k+1}{n_k-k}\frac{1}{2-p}\left(1 - \left(\frac{p}{2}\right)^k\right).$$

Let $\varepsilon_1 = (k+1)/(n_k - k)$ and $\varepsilon_2 = (p/2)^k$. Since $n_k > 2^k - 1$ and $p \geqq 0$, it follows that $\varepsilon_1 > 0$ and $\varepsilon_2 \geqq 0$ for all $k \geqq 1$. $E[r_k]$ may then be written as

$$E[r_k] \leqq (1+\varepsilon_1)(1-\varepsilon_2)\frac{1}{2-p}$$

implying that

$$E[r_k] \leqq (1+\varepsilon_1)\frac{1}{2-p}.$$

Since $n_k > 2^k - 1$, n_k grows exponentially faster than k. Consequently, $\varepsilon_1 = (k+1)/(n_k - k)$ approaches 0 for large k. Hence the upper bound on $E[r_k]$ can be made arbitrarily close to $1/(2-p)$. As it is also a lower bound on $E[r_k]$ by Theorem 3, $1/(2-p)$ is a tight lower bound on $E[r_k]$.

How small can p, and hence $1/(2-p)$, be? For the data-generating mechanism described above, p can be arbitrarily small. The lower bound on the average performance ratio for the greedy heuristic then approaches $\frac{1}{2}$, the same as the lower bound on the average performance ratio for the random procedure. However, the data-generating mechanism described above is "perverse." Other mechanisms can possibly guarantee higher minimum values for p, and hence a higher minimum performance ratio for the greedy heuristic. One mechanism, similar to that used in Goldberg, Purdom, and Brown

[11], is as follows: for all $j = 1, \cdots, k$, generate

$$\begin{cases} t_{ij} = 1, \bar{t}_{ij} = 0, & \text{with probability } q, \\ t_{ij} = 0, \bar{t}_{ij} = 1, & \text{with probability } q, \\ t_{ij} = 0, \bar{t}_{ij} = 0, & \text{with probability } 1 - 2q. \end{cases}$$

The choice of q is arbitrary, and as in many simulations may be based on random sampling from a parametric distribution. In this case, the probability that u_j has a larger number of 1's than $\bar{u}_j$ is $\frac{1}{2}$, $j = 1, \cdots, k$, and hence $E[r_k] \geqq \frac{2}{3}$.

Is there a way to improve the lower bound on $E[r_k]$ for the greedy heuristic? So long as p is governed by "nature" (i.e., by a data-generating mechanism which the algorithm cannot control), there appears to be no way. But there is no reason why the choice of p should not be made a part of the heuristic. For instance, we may introduce probabilistic choice at each step of the greedy heuristic so that, whatever p is, the heuristic selects a solution with a probability *it chooses*. The perversity of a data-generating mechanism may then be superseded by the heuristic. We pursue this approach below by describing a probabilistic version of the greedy heuristic, which we call the *probabilistic greedy heuristic*.

4. The probabilistic greedy algorithm. Like the greedy heuristic, the probabilistic greedy heuristic selects at step j column u_j or $\bar{u}_j$ from table T_j, $j = 1, \cdots, k$. However, each column is selected with probability proportional to the number of 1's it contains. That is, u_j is chosen with probability $p = x_j/n_j$, and $\bar{u}_j$ is chosen with probability $1 - p = (n_j - x_j)/n_j$. We describe the heuristic more formally below.

Initialization. Order the columns of T_k so that n_k, the number of 1's across u_k and $\bar{u}_k$, is largest among all pairs of columns u_l, $\bar{u}_l$, $l = 1, \cdots, k$. Select column u_k with probability $p = x_k/n_k$, and select column $\bar{u}_k$ with probability $1 - p = (n_k - x_k)/n_k$. Eliminate u_k and $\bar{u}_k$, and all rows with a 1 in the chosen column, to obtain table T_{k-1}, where, as before,

$$T_{k-1} = \begin{cases} T(u_k), & \text{if column } u_k \text{ is chosen from } T_k, \\ T(\bar{u}_k), & \text{if column } \bar{u}_k \text{ is chosen from } T_k. \end{cases}$$

Recursion. Order the columns of T_j so that n_j, the number of 1's across u_j and $\bar{u}_j$, is largest among all pairs of columns u_l, $\bar{u}_l$, $l = 1, \cdots, j$. Select column u_j with probability $p = x_j/n_j$, and select column $\bar{u}_j$ with probability $1 - p = (n_j - x_j)/n_j$. Eliminate u_j and $\bar{u}_j$, and all rows with a 1 in the chosen column, to obtain table T_{j-1}, where, as before,

$$T_{j-1} = \begin{cases} T(u_j), & \text{if column } u_j \text{ is chosen from } T_j, \\ T(\bar{u}_j), & \text{if column } \bar{u}_j \text{ is chosen from } T_j. \end{cases}$$

Termination. Stop if T_j contains no 1's, or if $j = 0$.

The probabilistic greedy heuristic forces a trade-off between the probability of selecting the optimal solution and the value of the nonoptimal solution it identifies. We illustrate the trade-off below for the Msat problem described by T_k. Assume that at each of the first $k - 1$ steps the probabilistic greedy heuristic chooses the optimal column. At step k, the probabilistic greedy heuristic chooses column u_1 with probability $p = x_1/n_1$, and column $\bar{u}_1$ with probability $1 - p = (n_1 - x_1)/n_1$. Hence the expected performance ratio is

$$E[r_k] = \frac{x_1}{n_1} \frac{x_1 + (m_1 - x_1)}{m_1} + \frac{n_1 - x_1}{n_1} \frac{(n_1 - x_1) + (m_1 - x_1)}{m_1}$$

where $m_1 - x_1$ is the number of clauses satisfied by the optimal columns selected by the greedy heuristic in steps 1 to $k-1$, and hence $x_1 + m_1 - x_1$ is the value of the greedy solution if column u_1 is selected and $n_1 - x_1 + m_1 - x_1$ is the value of the greedy solution if column $\bar{u}_1$ is selected. The trade-off can be seen in the expression for $E[r_k]$. The probability of selecting the optimal column u_1 decreases as x_1 decreases. However, as x_1 decreases, the value of the nonoptimal solution $n_1 - x_1 + m_1 - x_1 = n_1 + m_1 - 2x_1$ increases. The lower bound on $E[r_k]$ is obtained by choosing x_1 so that $E[r_k]$ has its smallest value. It can be verified that $E[r_k]$ is minimized by setting $x_1 = 3n_1/4$, at which value, $E[r_k] = 1 - n_1/8m_1$. Hence $m_1 \geqq x_1 = 3n_1/4$. Thus

$$\min E[r_k] = 1 - \frac{n_1}{8m_1} \geqq 1 - \frac{n_1}{8(3n_1/4)} = \frac{5}{6}.$$

That is, the lower bound on the expected performance ratio is $\frac{5}{6}$ when the first $k-1$ columns selected by the greedy heuristic are optimal. As described below, the trade-off between the probability of selecting the optimal solution and the value of the nonoptimal solution occurs in general for the probabilistic greedy heuristic.

THEOREM 4. *$E[r_k] \geqq \frac{2}{3}$ for all k.*

Proof. We prove the theorem by induction on the number of variables.

Base case. $E[r_1] \geqq \frac{2}{3}$.

For the single-variable problem, the optimal solution to Msat described by T_1 is $m_1 = x_1$, and corresponds to column u_1 of T_1 as per our assumption. As the probabilistic greedy heuristic selects u_1 with probability $p = x_1/n_1$, and selects $\bar{u}_1$ with probability $1 - p = (n_1 - x_1)/n_1$, the expected value of its solution is

$$E[f_1] = px_1 + (1-p)(n_1 - x_1),$$

and the expected performance ratio of the heuristic is

$$E[r_1] = \frac{E[f_1]}{m_1} = \frac{[(x_1/n_1)]x_1 + [(n_1 - x_1)/n_1](n_1 - x_1)}{x_1}.$$

Given n_1, the lower bound on $E[r_1]$ is obtained by minimizing the above expression with respect to x_1, which can be verified to occur at $x_1 = n_1/\sqrt{2}$. Substituting this value of x_1 in $E[r_1]$ and simplifying yields

$$E[r_1] \geqq 2\sqrt{2} - 2 \geqq \frac{2}{3}.$$

Induction hypothesis. $E[r_l] \geqq \frac{2}{3}$ for all $l \leqq k-1$.

Induction step. $E[r_k] \geqq \frac{2}{3}$ for all k.

If the probabilistic greedy heuristic selects column u_k from T_k, it guarantees a solution value of at least x_k. In addition, $T_{k-1} = T(u_k)$, generated at the first step, describes an Msat problem for which the expected value of the heuristic solution is, by the induction hypothesis, no less than $\frac{2}{3}a_k$. Hence if column u_k is selected at step 1, the expected value of the heuristic solution is no less than $x_k + \frac{2}{3}a_k$. By a similar argument, if the greedy heuristic selects $\bar{u}_k$ at step 1, the expected value of its solution is no less than $n_k - x_k + \frac{2}{3}\bar{a}_k$. Now u_k is selected with probability $p = x_k/n_k$, and $\bar{u}_k$ is selected with probability $1 - p = (n_k - x_k)/n_k$. The expected value of the heuristic solution is therefore

$$E[f_k] \geqq \frac{x_k}{n_k}\left(x_k + \frac{2}{3}a_k\right) + \frac{n_k - x_k}{n_k}\left(n_k - x_k + \frac{2}{3}\bar{a}_k\right).$$

As $a_k = m_k - x_k$ by Lemma 1,

$$E[f_k] \geqq \frac{x_k}{n_k}\left(x_k + \frac{2}{3}(m_k - x_k)\right) + \frac{n_k - x_k}{n_k}\left(n_k - x_k + \frac{2}{3}\bar{a}_k\right).$$

Also, $\bar{a}_k \geqq \max\{0, m_k - n_k\}$ by Lemma 1. Consider $m_k > n_k$. Then $\bar{a}_k \geqq m_k - n_k > 0$, and the above inequality for $E[f_k]$ becomes

$$E[f_k] \geqq \frac{x_k}{n_k}\left(\frac{2m_k}{3} + \frac{x_k}{3}\right) + \frac{(n_k - x_k)^2}{n_k} + \frac{2(n_k - x_k)}{3n_k}(m_k - n_k).$$

Simplifying,

$$E[f_k] \geqq \frac{(x_k)^2}{3n_k} + \frac{2m_k x_k}{3n_k} + \frac{(n_k - x_k)^2}{n_k} + \frac{2(n_k - x_k)}{3n_k}(m_k - n_k).$$

The right side of the above expression can be verified to obtain its minimum value when $x_k = n_k/2$, at which value of x_k,

$$E[f_k] \geqq \frac{2m_k}{3},$$

and hence

$$E[r_k] = \frac{E[f_k]}{m_k} \geqq \frac{2}{3}.$$

Now consider $m_k \leqq n_k$. Then $\bar{a}_k \geqq 0$ $(\geqq m_k - n_k)$, and hence

$$E[f_k] \geqq \frac{x_k}{n_k}\left(x_k + \frac{2}{3}(m_k - x_k)\right) + \frac{n_k - x_k}{n_k}(n_k - x_k).$$

Simplifying

$$E[f_k] \geqq \frac{(x_k)^2}{3n_k} + \frac{2m_k x_k}{3n_k} + \frac{(n_k - x_k)^2}{n_k}.$$

The right side of the above expression can be verified to obtain its minimum value when $x_k = (3n_k - m_k)/4$, at which value of x_k

$$E[f_k] \geqq \frac{n_k}{4} + \frac{m_k}{2} - \frac{m_k^2}{12n_k}.$$

The right side of the above expression takes its smallest value when $n_k = m_k$, for which

$$E[f_k] \geqq \frac{m_k}{4} + \frac{m_k}{2} - \frac{m_k}{12} = \frac{2}{3}m_k.$$

Therefore $E[r_k] = E[f_k]/m_k \geqq \frac{2}{3}$. □

It can be verified that for the data in Fig. 8, the expected performance of the probabilistic greedy heuristic is

$$E[f_k] = 2\left(\frac{1}{2}\right)\frac{n_k}{2} + 2\left(\frac{1}{2}\right)^2\frac{n_k}{2^2} + \cdots + 2\left(\frac{1}{2}\right)^k\frac{n_k}{2^k} + \left(\frac{1}{2}\right)^k\frac{n_k}{2^k}$$

$$= \frac{2n_k}{3} + \frac{n_k}{3}\left(\frac{1}{4^k}\right).$$

Noting that $m_k = n_k$, the expected performance ratio equals $E[r_k] = \frac{2}{3} + \varepsilon$, where $\varepsilon = \frac{1}{3}(1/4^k)$. Since ε can be made to approach 0 arbitrarily closely by increasing k, $E[r_k]$ can be made to approach $\frac{2}{3}$ from above arbitrarily closely. Since the asymptotic upper bound for the probabilistic greedy heuristic is $\frac{2}{3}$, the lower bound of $\frac{2}{3}$ specified by Theorem 4 is tight.

As the data-generating mechanism plays no role in determining the lower bound of the performance ratio for the probabilistic greedy heuristic, the bound obtained by

Row	u_k	$\bar{u}_k$	u_{k-1}	$\bar{u}_{k-1}$	· ·	u_2	$\bar{u}_2$	u_1	$\bar{u}_1$
1	0	1	0	1	· ·	0	1	0	1
2	0	1	0	1	· ·	1	0	0	0
3	0	1	0	1	· ·	0	0	0	0
·	·	·	·	·	· ·	·	·	·	·
·	·	·	·	·	· ·	·	·	·	·
2^{k-2}	0	1	0	1	· ·	0	0	0	0
$2^{k-2}+1$	0	1	1	0	· ·	0	0	0	0
·	·	·	·	·	· ·	·	·	·	·
·	·	·	·	·	· ·	·	·	·	·
·	·	·	·	·	· ·	·	·	·	·
2^{k-1}	0	1	1	0	· ·	0	0	0	0
$2^{k-1}+1$	1	0	0	0	· ·	0	0	0	0
·	·	·	·	·	· ·	·	·	·	·
·	·	·	·	·	· ·	·	·	·	·
·	·	·	·	·	· ·	·	·	·	·
2^k	1	0	0	0	· ·	0	0	0	0

FIG. 8. *Worst case example for the probabilistic greedy heuristic with* $n_k = 2^k$ *clauses.*

Theorem 4 also holds if the *same* problem is sampled repeatedly; i.e., if the probabilistic greedy heuristic is implemented multiple times to solve the same problem, the average heuristic solution will be no smaller than $\frac{2}{3}$ of the optimal. Of course, in this case the solution with the largest value is of greater interest than the average value of the solutions across trials. For a large number of trials, the distribution of the largest value of the probabilistic greedy solution corresponds to the extreme value distribution for the largest among a sample of n observations. Since the largest value of the probabilistic greedy solution is bounded from above by the value of the optimal, the distribution in this case is characterized by the limited-value distribution (Gumbel [12]), which corresponds to the type-three distribution in the Fisher and Tippett characterization of extreme-value distributions (Fisher and Tippett [8]). Thus, regardless of the data-generating distribution, the asymptotic cumulative distribution function of the largest value of the probabilistic greedy solution is

$$H[z] = \exp\left(-\left(\frac{m_k - z}{m_k - v}\right)^w\right),$$

with corresponding density

$$h(z) = \frac{w}{m_k - v}\left(\frac{m_k - z}{m_k - v}\right)^{w-1} H(z),$$

where z is the largest value of the probabilistic greedy solution across trials, $H(v) = 1/e = 0.36788$, and $w > 0$ is the shape parameter of the distribution (see, e.g., Gumbel [12, pp. 164–165; p. 275]).

5. Conclusion. Two aspects of the probabilistic greedy heuristic should perhaps be mentioned. First, it guarantees an average solution value of no less than $\frac{2}{3}$ of the optimal value regardless of the distribution of data. Second, the trade-off it forces between the probability of selecting the optimal solution and the value of the nonoptimal solution is a feature that is not evidently observed in other heuristics. Indeed, it is this feature of

the heuristic that ensures that its average performance is never too bad. In contrast, while the greedy heuristic can do well, its ability to do so depends on the value of p. For independent observations from parametric distributions with $p = \frac{1}{2}$, it does as well, on average, as the probabilistic greedy heuristic. But for perverse distributions, the greedy heuristic on average can do as poorly as random search. On the other hand, for an "unintelligent" procedure, the random search does quite well to ensure an average solution of no less than $\frac{1}{2}$ of the optimal, regardless of the data-generating distribution. It remains an open question whether relaxing the independence assumption, or assuming specific distributions, strengthens the bounds on the average performance for the greedy heuristic. It may also be possible to strengthen the average bound for the greedy heuristic with restricting assumptions on problem instances, such as when the set of clauses are 2-satisfiable (Lieberherr and Specker [16]), or when each clause contains at least l variables, $1 \leqq l \leqq k$ (Johnson [13]).

Acknowledgments. The authors thank Pavol Hell, Tiko Kameda, Art Liestman, Joe Peters, and Art Warburton for numerous helpful suggestions. Special thanks are due to Tiko Kameda and Joe Peters for carefully reading the paper and giving detailed comments that have helped improve the presentation.

REFERENCES

[1] J. BENTLEY, D. S. JOHNSON, F. T. LEIGHTON, AND L. A. MCGEOCH, *Some unexpected expected behavior results for bin packing*, Proc. 16th ACM Sympos. Theory of Computing, 1984, pp. 279–288.

[2] O. J. BOXMA, *A probabilistic analysis of multiprocessing list scheduling: the erlang case*, Stochastic Models, 1 (1985), pp. 209–220.

[3] J. L. BRUNO AND P. J. DOWNEY, *Probabilistic bounds for dual bin packing*, Acta Inform., 22 (1985), pp. 333–345.

[4] E. G. COFFMAN AND F. T. LEIGHTON, *A provably efficient algorithm for dynamic storage allocation*, Proc. 18th ACM Sympos. Theory of Computing, 1986, pp. 77–90.

[5] E. G. COFFMAN, G. S. LUEKER, AND A. H. G. RINNOOY KAN, *Asymptotic methods in the probabilistic analysis of sequencing and packing heuristics*, Management Sci., 3 (1988), pp. 266–290.

[6] J. CSIRIK, J. B. G. FRENK, A. FRIEZE, G. GALAMBOS, AND A. H. G. RINNOOY KAN, *A probabilistic analysis of the next fit decreasing bin packing heuristic*, Oper. Res. Lett., 5 (1986), pp. 233–236.

[7] M. DAVIS, G. LOGEMANN, AND D. LOVELAND, *A machine program for theorem proving*, Comm. ACM, 5 (1962), pp. 394–397.

[8] R. A. FISHER AND L. H. C. TIPPETT, *Limiting forms of the frequency distribution of the largest or smallest member of a sample*, Proc. Cambridge Phil. Soc., 24 (1928), pp. 180–190.

[9] J. FRANCO AND M. PAULL, *Probabilistic analysis of the Davis Putnam procedure for solving the satisfiability problem*, Discrete Applied Math., 5 (1983), pp. 77–87.

[10] J. B. G. FRENK AND A. H. G. RINNOOY KAN, *The asymptotic optimality of the LPT rule*, Math. Oper. Res., 12 (1987), pp. 241–254.

[11] A. GOLDBERG, P. PURDOM, AND C. BROWN, *Average time analysis of simplified Davis–Putnam procedures*, Inform. Process. Lett., 15 (1982), pp. 72–75.

[12] E. J. GUMBEL, *Statistics of Extremes*, Columbia University Press, New York, 1958.

[13] D. S. JOHNSON, *Approximation algorithms for combinatorial problems*, J. Comput. System Sci., 9 (1974), pp. 256–278.

[14] R. M. KARP, M. LUBY, AND A. MARCHETTI-SPACCAMELA, *A probabilistic analysis of multidimensional bin packing problems*, Proc. 16th ACM Sympos. Theory of Computing, 1984, pp. 289–298.

[15] F. T. LEIGHTON AND P. W. SHOR, *Tight bounds for minimax grid matching with applications to the average-case analysis of algorithms*, Proc. 18th ACM Sympos. Theory of Computing, 1986, pp. 91–103.

[16] K. J. LIEBERHERR AND E. SPECKER, *Complexity of partial satisfaction*, J. Assoc. Comput. Mach., 28 (1981), pp. 411–421.

[17] P. W. SHOR, *The average-case analysis of some on-line algorithms for bin packing*, Combinatorica, 6 (1986), pp. 179–200.