An Analysis of Backtracking with Search Rearrangement*

by

Paul Walton Purdom, Jr.

and

Cynthia A. Brown

Computer Science Department

Indiana University

Bloomington, Indiana 47405

TECHNICAL REPORT NO. 89

AN ANALYSIS OF BACKTRACKING WITH SEARCH REARRANGEMENT

Paul Walton Purdom, Jr. and Cynthia A. Brown

Revised: September 1982

---

*Research reported herein was supported in part by the National Science Foundation under grant number MCS 79-06110.

An Analysis of Backtracking with Search Rearrangement

by

Paul Walton Purdom, Jr.

and

Cynthia A. Brown

Abstract. The search rearrangement backtracking algorithm of Bitner and Reingold introduces at each level of the backtrack tree a variable with a minimal number of remaining values; search order may differ on different branches. For conjunctive normal form formulas with v variables, s literals per term ($s \geq 3$), and v^{α} terms ($\frac{s}{2} < \alpha < s$), the average number of nodes in a search rearrangement backtrack tree is $\exp\left[\Theta(v^{\frac{s-\alpha-1}{s-2}})\right]$. (I.e., for some positive constants a_1, a_2, and v_0, when $v \geq v_0$ the number of nodes is between $\exp(a_1 v^{\frac{s-\alpha-1}{s-2}})$ and $\exp(a_2 v^{\frac{s-\alpha-1}{s-2}})$.) For $1 < \alpha \leq \frac{s}{2}$ the average number of nodes is between $\exp\left[\Theta(v^{\frac{s-\alpha-1}{s-2}})\right]$ and $\exp\left[\Theta((\ln v)^{\frac{s-1}{s-2}} v^{\frac{s-\alpha-1}{s-2}})\right]$. This compares with $\exp\left[\Theta(v^{\frac{s-\alpha}{s-1}})\right]$ for ordinary backtracking. For $1 < \alpha < s$, simple search rearrangement has approximately the same effect on speeding up backtracking as does reducing the problem complexity by decreasing the number of literals per term by one. Thus simple search rearrangement backtracking leads to a dramatic improvement in the expected running time.

This research was supported in part by the National Science Foundation under grant number MCS 7906110.

1. Introduction

Many problems can be regarded as a search for all the solutions to an equation of the form $\underline{P}(x_1, \ldots, x_v) = \text{true}$, where $\underline{P}$ is a v-ary predicate over an ordered set of variables $S = \{x_i\}_{1 \leq i \leq v}$, and each x_i has finitely many possible values. A straightforward, but exponentially costly, way to solve such a problem is to generate and test each combination of values of the variables. For most problems, backtracking can reduce the amount of time required to find the solutions.

To perform backtracking it is necessary to have, in addition to the problem predicate $\underline{P}$, an intermediate predicate $\underline{P}_A$ for each subset A of S. The predicate $\underline{P}_A$ must have the value true for any assignment of values to the variables in A that can be extended to a solution to $\underline{P}$. An intermediate predicate is powerful if it is false for most other assignments of values. Powerful intermediate predicates that can be calculated efficiently eliminate false starts quickly, making backtracking feasible for large problems.

In backtracking each variable starting with the first is set to its initial value. As each variable is set, the appropriate intermediate predicate is tested. If it is true, the next variable is set. If it is false, the current variable is reset to its next value; if the current variable has no more values, it is removed from the set of variables with values, and the previous variable is set to its next value. Knuth [4] gives a good introduction to backtracking.

Bitner and Reingold [1] consider a modification of backtracking which we call (simple) search rearrangement backtracking. (For other modifications, see [6].) Instead of setting the variables in a fixed order, the unset variables are tested at each step to select one that produces the fewest true values for the corresponding intermediate predicate. Bitner and Reingold showed experimentally that this search rearrangement process improves the efficiency of backtracking. In this paper we study search rearrangement backtracking analytically, and compare its performance with that of ordinary backtracking, which we analyzed in [2].

2. Model and Notation

To compare backtracking methods, we analyze their average performance on the problem of finding all solutions to conjunctive normal form formulas having t terms and s literals per term over a set S of v variables. Duplication is permitted in the terms for a predicate and in the literals within a term, so there are $(2v)^{st}$ predicates in the set. For each predicate $\underline{P}$ and set of variables $A \subseteq S$, the intermediate predicate $\underline{P}_A$ is the conjunction of those terms of $\underline{P}$ for which all the variables are in the set A. (If there are no such terms, then $\underline{P}_A$ is by definition true.) These intermediate predicates are natural and efficiently calculable. The set of predicates is NP complete (for $s \geq 3$, $t > v$, and v increasing), so some problems in the set are hard. We find that the backtrack trees for most of the problems in this set are similar to those encountered in realistic problems, and this set of problems is suitable for analysis. We therefore believe these problems are a good model for our study of backtracking.

For purposes of illustration we use the predicate $E = T_1 \wedge T_2 \wedge T_3 \wedge$

... $\wedge T_{11}$, where $T_1 = (v_1 \vee v_1 \vee \neg v_2)$, $T_2 = (v_1 \vee v_1 \vee \neg v_5)$, $T_3 = (v_1 \vee \neg v_3 \vee v_5)$, $T_4 = (v_1 \vee v_2 \vee \neg v_6)$, $T_5 = (\neg v_1 \vee \neg v_1 \vee v_2)$, $T_6 = (\neg v_1 \vee \neg v_1 \vee v_2)$, $T_7 = (\neg v_1 \vee v_3 \vee v_4)$, $T_8 = (\neg v_1 \vee v_3 \vee v_5)$, $T_9 = (\neg v_1 \vee \neg v_4 \vee \neg v_5)$, $T_{10} = (\neg v_1 \vee \neg v_3 \vee v_5)$, and $T_{11} = (\neg v_1 \vee v_4 \vee v_6)$. Notice the duplication in the literals within a clause (as in $(v_1 \vee v_1 \vee \neg v_2)$) and in the clauses in the predicate ($T_5 = T_6$). Intermediate predicate $E_{\{1,2,5\}}$ is $T_1 \wedge T_2 \wedge T_5 \wedge T_6$; intermediate predicate $E_{\{1,3\}}$ is the empty predicate, which is identically true.

Simple search rearrangement backtracking is done as follows. Let S' be the set of variables without values (the unset variables) and S'' the set of variables with values. Let w range over the problem variables; denote the value of problem variable w by Value[w] for $1 \leq w \leq v$. The set S'' is maintained as a stack.

Simple Search Rearrangement

Step 1. (Initialize.) Set S'' to empty and S' to S.

Step 2. (Solution?) If S' is not empty, go to Step 3. Otherwise, the current values in Value constitute a solution. Go to Step 6.

Step 3. (Find best variable.) For each variable w in S' do the rest of this step. (The order in which the variables are tested is immaterial for our purposes.) For both Value[w] $\leftarrow$ false and Value[w] $\leftarrow$ true, compute $\underline{E}_{S'' \cup \{w\}}$. If the result is false in both cases, exit the loop for w and go to Step 6. If it is false in one and true in the other, remember the value of w that gives true and exit the loop for w and go to Step 5.

Step 4. (Binary node.) Let w be the smallest element of S'. Set $S'' \leftarrow S'' \cup \{w\}$ and $S' \leftarrow S' - \{w\}$. Set Value[w] $\leftarrow$ false, mark w as binary, and go to Step 2.

Step 5. (Unary node.) Set Value[w] to the unique value that makes $P_{S'' \cup \{w\}}$ true. (This value is remembered from Step 3). Set $S'' \leftarrow S'' \cup \{w\}$ and $S' \leftarrow S' - \{w\}$. Mark w as unary, and go to Step 2.

Step 6. (Next value.) If S'' is empty, stop. Otherwise, set w $\leftarrow$ top (S''). If w is marked as binary and Value[w] = false, set Value[w] $\leftarrow$ true and go to Step 2.

Step 7. (Backtrack.) Set $S'' \leftarrow S'' - \{w\}$, $S' \leftarrow S' \cup \{w\}$, and go to Step 6.

Figure 1 shows the backtrack tree obtained by applying the simple search rearrangement algorithm to E, with the convention that unset variables are tested in numerical order according to their subscripts. Initially, no variables are set. The first time Step 3 is executed it does not find a variable with zero or one values, so in Step 4 v_1 is selected and set to its first value (false). Returning to Step 3 by way of Step 2, testing $E_{\{1,2\}} = T_1 \wedge T_5 \wedge T_6$ shows that setting v_2 to true makes the predicate false. Therefore v_2 is selected and set to false in Step 5. In a similar way, v_5, v_3, and v_6 are forced to assume the value false. The only remaining variable, v_4, is not constrained and so is not selected in Step 3; it is selected as a binary variable in Step 4, leading to two solutions. The remainder of the backtrack tree is produced in a similar way.

We will concentrate on the number of binary nodes in the average search rearrangement backtrack tree. Let $N_{\underline{P}}$ be the number of binary nodes in the backtrack tree for predicate $\underline{P}$. The total number of nodes in the tree is between $N_{\underline{P}}$ and $(2v-1) N_{\underline{P}}$. Assuming that for each $A \in S$, $\underline{P}_A$ can be evaluated in constant time, the total running time of the algorithm is between $c_1 N_{\underline{P}}$ and $c_2 v^2 N_{\underline{P}}$ for some constants c_1 and c_2. We compute N, the average value of $N_{\underline{P}}$. The average values for both the total number of nodes and the total running time are N times some polynomially bounded function of v.

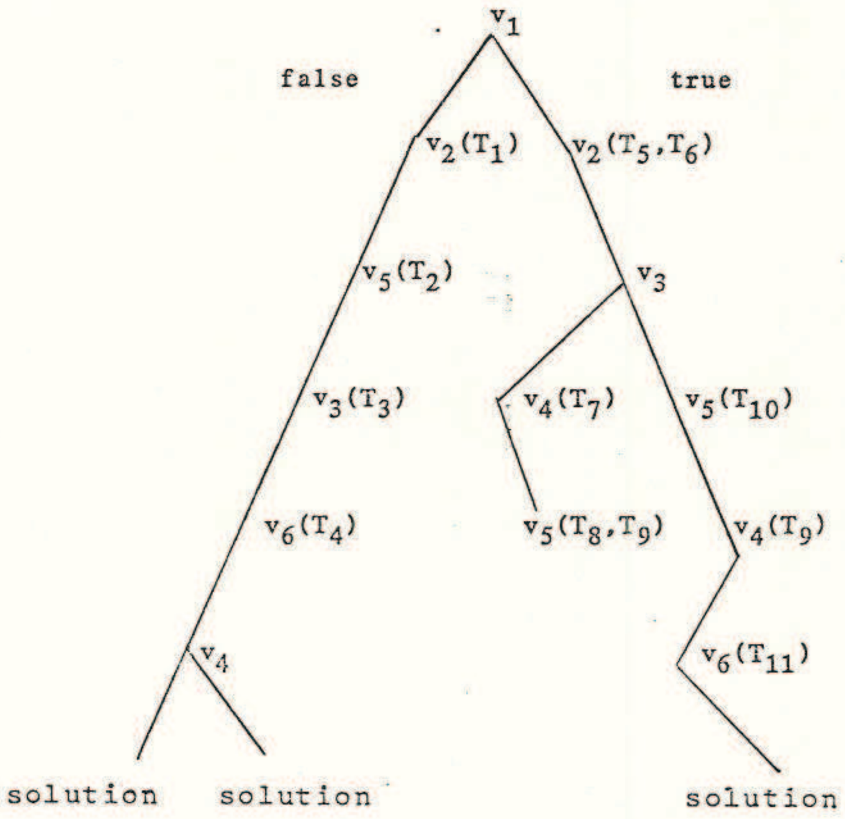
v1
false
true
v2(T1)
v2(T5,T6)
v5(T2)
v3
v3(T3)
v4(T7)
v5(T10)
v6(T4)
v5(T8,T9)
v4(T9)
v4
v6(T11)
solution
solution
solution

Figure 1. The backtrack tree obtained by applying a simple search rearrangement backtrack algorithm to predicate E. Each interior node is labelled with the name of the variable that is set as that node. At nodes where a variable is forced to take on a particular value, the clause or clauses that cause the forcing are shown in parentheses. At the leaf nodes where a variable with no values is discovered, the clauses that eliminate both values are shown.

In [3] we give the results of a statistical study of the ratio of the number of binary nodes to the number of unary nodes and to the number of predicate evaluations.

3. The Average Number of Binary Nodes

We now derive an exact formula for N, the average number of binary nodes. The exact formula has an exponential number of terms, so we also derive upper and lower limits with a polynomial number of terms.

Our method is to consider each possible binary node and to count the predicates that have that node in their backtrack trees. Summing over all binary nodes and dividing by the number of predicates ($(2v)^{st}$) gives the average number of binary nodes in a tree.

We begin by counting the predicates for which a particular variable w occurs as a binary node with all the other variables in S′ set to false. (Later we multiply by an appropriate factor to allow for other settings of the variables.) For this to occur, the algorithm must reach Step 4 with every variable less than w set to false, with the value of w unconstrained, and with none, some, or all of the variables greater than w set to false. Any variable $y > w$ which is set to false must have been set in Step 5: it must have been forced to assume the value false by a clause made up of literals with truth value false, the literal $\neg y$, and nothing else. The variables less than w may have been forced to false in Step 5 or may correspond to a binary node where the false branch is being explored; the step at which these variables received their values does not affect the analysis.

Let q_0 equal the number of variables that precede w (recall that they are all set to false). Let q_1 equal q_0 plus the number of variables whose values are forced to false when just the first q_0 variables are set. Let q_i be q_{i-1} plus the number of variables whose values are forced to false by the q_{i-1} variables. Let m $(0 \leq m < v)$ be the smallest value of i such that $q_{i+1} = q_i$. We call m the number of rounds of forcing.

To illustrate, consider the binary node for v_4 on the left branch of the backtrack tree in Fig. 1. Variables v_1, v_2, and v_3 precede v_4 in the ordered set of variables, so $q_0 = 3$. With these variables set to false, v_5 is forced to false by T_2 and v_6 is forced to false by T_4. Thus $q_1 = 5$. Setting v_5 and v_6 to false does not force any more variables, so $m=1$. (A second round of forcing would have occurred if the predicate had contained a term such as $(v_5 \vee v_6 \vee \neg v_7)$.) Notice that the rounds of forcing used in the analyses do not necessarily correspond to the order in which variables are set by the algorithm. They are a device to facilitate counting the predicates that have a particular binary node. The analysis starts with the assumption that a node such as v_4 is binary and counts the number of ways in which this can happen.

Assume that the variables forced during rounds 1 through m are consecutive variables. (We later multiply by a factor to account for the number of ways of choosing these variables.) For the rest of this section let the variables be renumbered so that the variable for the binary node is q_0, so that the forced variables start with q_0+1. Recall that each predicate in the class we are considering has t clauses. For purposes of counting we divide these clauses into two classes. Given m and $q_0, q_1, \ldots q_m$, the predicate must contain clauses that force the variables from $q_0 + 1$ to q_m to the value false. We call these required clauses. For $1 \leq i \leq m$ and $1 \leq x \leq q_i - q_{i-1}$, let $j_{ix} \geq 1$ be the number of clauses that force variable $q_{i-1}+x$ to false, given only that the first q_{i-1} variables are set to false. Notice that for $(i,x) \neq (i',x')$ the clauses counted by j_{ix} are disjoint from those counted by $j_{i'x'}$.

Remaining clauses in the predicate are called permitted clauses. Permitted clauses are any clauses other than required clauses that are compatible with the assumptions we make about the node we are considering. Whether a clause is permitted or required depends on the binary node being considered. The same clause might be a permitted clause for one node and a required clause for another. In predicate E, when v_1 is set to false and $w = v_4$ the variables v_2, v_3, v_5 and v_6 are forced. Since v_2 and v_3 precede v_4 we do not care whether they were forced or set, and so the clauses that force them are not required clauses. Variable v_5 is forced by T_2, and v_6 by T_4, so $j_{11} = 1$ and $j_{12} = 1$. The rest of the clauses are permitted clauses.

The inital counting arguments are summarized in Table 1. First consider the required clauses for round i of forcing. Variable $q_{i-1} + x$ $(1 \leq x \leq q_i - q_{i-1})$ is forced to false by clauses containing some of the first q_{i-1} variables, all unnegated, and the negated literal for the forced variable. Since clauses containing only the first q_{i-1} variables (and not the forced variable) are not suitable, there are $Q_{i-1} = (q_{i-1} + 1)^s - q_{i-1}{}^s$ such clauses. The Q_{i-2} of these that contain only the first q_{i-2} variables and the forced variable are also unsuitable, since they would force the variable on an earlier round than the i^{th}. Therefore there are $R_{i-1} = Q_{i-1} - Q_{i-2}$ clauses that force variable $q_{i-1} + x$ on round i. (Use 0 for Q_{-1}.) For each variable $q_{i-1}+x$ forced on round i, the predicate contains j_{ix} clauses chosen (with replacement) from a set of R_{i-1} clauses, the sets corresponding to distinct values of x are disjoint. The total number of required clauses in the predicate is $\sum_{i,x} j_{ix}$.

The remaining clauses in the predicate (the permitted clauses) can be selected from any of the $(2v)^s$ clauses except the following. There are q_m^s clauses made up entirely of false (unnegated) literals for the first q_m variables. A predicate containing such a clause would be false and would not reach Step 4 under our assumptions. For each of the $(v-q_m)$ unset variables, there are Q_m clauses that would force it to true on an (m+1)st round of forcing. This accounts for $(v-q_m)Q_m$ clauses. Immediately after round i $(0\le i\le m)$ there are $(v-q_i)$ unset variables and R_i clauses that would force them to false on round $i+1$. (We exclude the required clauses from the class of permitted clauses, along with unwanted forcing clauses. The required clauses are counted explicitly by the other factors of the formula.) This accounts for $\sum_{0\le i\le m} (v-q_i)R_i$ clauses.

Much of the complexity of our analysis results from dividing the forcing into rounds. This division is necessary in order to count the required and permitted clauses correctly. A variable can only be forced by being in a clause where all variables for the other literals have already received values. One of the approximate formulas we present later was obtained by dropping the requirement that variables be forced in a legal order.

The total number of predicates that satisfy our current assumptions is

$$p^{t-\sum_{i,x} j_{ix}} \prod_{1\le i\le m} R_{i-1}^{\sum_{x} j_{ix}} ,$$

where

$$P = (2v)^s - q_m^s - (v-q_m)Q_m - \sum_{0\le i\le m} (v-q_i)R_i$$

$$= (2v)^s - q_m^s - 2(v-q_m)Q_m - \sum_{0\le i<m} (q_{i+1}-q_i)Q_i \ .$$

The total number of binary nodes in all the $(2v)^{st}$ backtrack trees, which is $(2v)^{st}N$, can be obtained by multiplying this formula by the appropriate factors and summing. The result is

$$(2v)^{st}N = \sum_{0\le m<v} \ \sum_{0\le q_0<q_1<\ldots<q_m<v} \ \sum_{j_{11}\ne 0,\ldots,j_{m,q_m-q_{m-1}}\ne 0} 2^{q_m} \binom{t}{j_{11},\ldots,j_{m,q_m-q_{m-1}},\ t-\sum_{i,x} j_{ix}} \binom{v-q_0-1}{q_1-q_0,\ldots,q_m-q_{m-1},v-q_m-1} P^{t-\sum_{i,x} j_{ix}} \prod_{1\le i\le m} R_{i-1}^{\sum_x j_{ix}} ,$$

where the factor 2^{q_m} accounts for the number of ways to assign values to the q_m set variables, the first multinomial accounts for the number of ways to order the terms ($t_1 \wedge t_2$ is different from $t_2 \wedge t_1$, for $t_1 \ne t_2$), and the second multinomial accounts for the number of ways to select which variables are forced on each of the m rounds. The initial q_0 variables and the variable for the binary node at Step 4 are not available for selection. The sums over j require that there is at least one term to force each forced variable. The sums over the q_i require that at least one variable be forced on each round.

We adopt the convention that limits on a summation variable are not shown explicitly when we intend that the sum be taken over all values of the variable that result in a non-zero value of the summand. If, for example, the variable appears as the bottom of a binomial coefficient, the implied range is between the value of the top of the binomial and zero.

The sums over the j_{ix} can be done using the binomial theorem and subtracting the $j_{ix}=0$ term. The sum over all the j_{1x} for $1 \le x \le q_1 - q_0$ is

$$(2v)^{s}t_N = \sum_{0 \le m < v} \; \sum_{0 \le q_0 < q_1 < \ldots < q_m < v} \; \sum_{j_{21} \ne 0, \ldots, j_{m,q_m - q_{m-1}} \ne 0} \; \sum_{i_1} 2^{q_m} \binom{q_1 - q_0}{i_1} (-1)^{q_1 - q_0 + i_1} \binom{t}{j_{21}, \ldots, j_{m,q_m - q_{m-1}}, t - \sum_{i>1,x} j_{ix}}$$

$$\cdot \binom{v - q_0 - 1}{q_1 - q_0, \ldots, q_m - q_{m-1}, v - q_m - 1} (P + i_1 R_0)^{t - \sum_{i>1,x} j_{ix}} \prod_{2 \le i \le m} R_{i-1}^{\sum_x j_{ix}} .$$

Summing over all j_{ix} and combining the binomials with the second multinomial gives

$$\left.\begin{aligned} (2v)^{s}t_N = \sum_{0 \le m < v} \; \sum_{0 \le q_0 < q_1 < \ldots < q_m < v} \; \sum_{i_1, \ldots, i_m} 2^{q_m} (-1)^{q_m - q_0 + \sum_n i_n} \\ \binom{v - q_0 - 1}{q_1 - q_0 - i_1, \ldots, q_m - q_{m-1} - i_m, i_1, \ldots, i_m, v - q_m - 1} \Big(P + \sum_{1 \le n \le m} i_n R_{n-1}\Big)^t . \end{aligned}\right\} (1)$$

The sum $\sum_{1 \le n \le m} i_n R_{n-1}$ equals $\sum_{1 \le n \le m} (i_n - i_{n+1}) Q_{n-1}$, where $i_{m+1} \equiv 0$.

4. Lower and Upper Limits

Formula 1 is not very useful for $v \gg 10$ because the number of terms increases exponentially with increasing v. We obtain a lower limit on its value by noticing that, for each fixed value of m, the corresponding partial sum is positive. (It equals the number of binary nodes that are immediately preceded by m rounds of forcing.) The *k-sum lower limit* is obtained by replacing $\sum_{0 \le m < v}$ in Formula 1 by $\sum_{0 \le m < k}$.

In particular, the one-sum lower limit is

$$(2v)^s t_N \geq \sum_{0 \le q_0 < v} 2^{q_0} P_0^t \,, \qquad (2)$$

where $P_0 = (2v)^s - q_0^s - 2(v-q_0)Q_0$, and the two-sum lower limit is

$$(2v)^s t_N \geq \sum_{0 < q_0 \le q_1 < v} \sum_{i_1} 2^{q_1} (-1)^{q_1 - q_0 + i_1} \binom{v - q_0 - 1}{q_1 - q_0 - i_1, i_1, v - q_1 - 1} (P_1 + i_1 Q_0)^t \,, \qquad (3)$$

where $P_1 = (2v)^s - q_1^s - 2(v-q_1)Q_1 - (q_1 - q_0)Q_0$. The $m=0$ and $m=1$ sums have been combined.

Upper limits can be obtained by allowing "sloppy counting" of required and permitted clauses. If some permitted clauses are misclassified as required clauses, the total number of predicates that meet the criteria increases: all the original predicates are still counted (though some of their clauses are put in a different class) and some additional, spurious predicates are also included. If some clauses are classified as both permitted and required, the number of spurious predicates increases even more. Another simplification that increases the size of the sum is to disregard the necessity for legal rounds of forcing. We combine these approaches to obtain upper limits with polynomially many terms.

The one-sum upper limit is obtained by ignoring altogether the idea of required vs. permitted terms and the idea of rounds of forcing. All terms that are not false and that do not force unset variables are used, and all variables except the one for the binary node are available for forcing. Once the variable for the binary node is chosen (and choosing it accounts for the factor of v in Formula 4) there are v-1 variables available to be set or forced, or left unset (choosing these accounts for the factor of $\binom{v-1}{q_0}$). If q_0 variables are being set or forced, there are 2^{q_0} branches. Finally, a factor P_0 is needed to count the number of legal clauses under these assumptions. The derivation of P_0 is summarized in Table 2; $P_0 = (2v)^s - q_0^s - 2(v-q_0)Q_0$, and P_0^t is the number of predicates made up of such clauses. The final formula is

$$(2v)^{st}N \le \sum_{0\le q_0<v} v \; 2^{q_0} \binom{v-1}{q_0} P_0^t \; . \qquad (4)$$

Notice that the factor of v was needed here because the q_0 variables that are set or forced are chosen arbitrarily and do not necessarily precede the variable for the binary node, nor does the value of q_0 determine which variable is the one for the binary node. In all our other formulas variable q_0+1 (before renumbering) is the variable for the binary node and so the factor of v is not needed.

In the two-sum upper limit we have $q_1 - q_0$ sets of required clauses, each with $q_1^s - (q_1-1)^s$ elements (and zero elements when $q_1 = 0$). Here we ignore the different rounds of forcing: each set has all the clauses that can force a variable if it is the last one to be forced (that is, the clauses contain literals of the other forced variables). Thus we over-count the actual number of forcing clauses available. These required sets are not disjoint; they contain $q_1^s - q_0^s$ terms in all. Table 3 summarizes the initial analysis. The final formula is

$$(2v)^s t_N \le \sum_{0 \le q_0 \le q_1 < v} \sum_{i_1} 2^{q_1} (-1)^{q_1 - q_0 + i_1} \binom{v - q_0 - 1}{q_1 - q_0 - i_1, i_1, v - q_1 - 1} \qquad (5)$$

$$[P_u + i_1 [q_1^s - (q_1 - 1)^s]]^t ,$$

where $P_u = (2v)^s - 2(v - q_1)Q_1 - 2q_1^s + q_0^s$. In this formula q_1 plays a role similar to that of q_m in Formula 1; the role played by q_0 is similar in both formulas.

Another upper limit can be obtained from the analysis in Table 4. It leads to the following formula:

$$(2v)^s t_N \le \sum_{0 \le q_0 \le q_1 < v} \; \sum_{0 \le i_1 \le 1, \ldots, 0 \le i_{q_1 - q_0} \le 1} 2^{q_1} (-1)^{q_1 - q_0 + \sum_n i_n} \frac{(v - q_0 - 1)!}{(v - q_1 - 1)!}$$

$$[(2v)^s - 2(v - q_1)Q_1 - 2q_1^s + q_0^s + \sum_{1 \le n \le q_1 - q_0} i_n \times$$

$$[(q_0 + i_n)^s - (q_0 + i_n - 1)^s]]^t .$$

We do not, however, analyze the asymptotic behavior of this formula.

The two-sum upper limit can be improved in various ways. For example, one can treat the first few rounds of forcing exactly and the remaining rounds approximately. Another approach would have an approximate treatment for forcing the first half of the forced variables, followed by an approximate treatment for the second half. We have not investigated which of these methods gives the most precise answer for a fixed amount of computation.

5. Asymptotic Analysis

We now consider the asymptotic behavior of the formulas for the one- and two-sum lower and upper limits (four cases), holding s fixed, letting $t=v^{\alpha}$ for a fixed α, and allowing v to become large. We find only the leading term in the exponential dependence of the result.

In each of the four formulas (once the sum over i_1 is done, if necessary) the sums contain no more than v^2 terms, where each term is positive. Therefore, to the required accuracy (ignoring polynomial factors), each sum is equal to the largest term. Most details of the analysis are omitted; they are similar to those in [2]. We assume $s \geq 3$ and $1<\alpha<s-1$.

Briefly, the procedure is: (1) sum over i_1 (if necessary); (2) expand multinomials into factorials and use Stirling's approximation for the factorials; (3) take the logarithm of the summand; (4) take the derivatives of the logarithm (with respect to q_0 and to q_1) and set them to zero; (5) solve the equations asymptotically to find the values of q_0 and q_1 that maximize the summand; (6) substitute the values of q_0 and q_1 into the log of the summand to find the log of the maximum term; and (7) obtain the final value by increasing the error term to $\log v$ (if necessary) to allow for missing polynomial factors and exponentiate the result. Do 40 pages of calculations without error and obtain the results given below. (Appendix 2 outlines the necessary steps.)

The one-sum lower limit summand is maximized with

$$q_0 = \left(\frac{2^{s-1}\,\ell n\ 2}{s(s-1)}\right)^{\frac{1}{s-2}} v^{\frac{s-\alpha-1}{s-2}} + \Theta\left(v^{\frac{s-2\alpha}{s-2}}\right) + \Theta(1) .$$

This gives

$$N \geq \exp\left[\left(\frac{2\ \ell n\ 2}{s(s-1)}\right)^{\frac{1}{s-2}} \frac{2(s-2)\,\ell n\, 2}{s-1}\ v^{\frac{s-\alpha-1}{s-2}} + \Theta\left(v^{\frac{s-2\alpha}{s-2}}\right) + \Theta(1)\right] . \qquad (6)$$

The one-sum upper limit summand is maximized with

$$q_0 = 2\left(\frac{2\ \ell n\ F(v)}{s(s-1)(s-2)}\right)^{\frac{1}{s-2}} v^{\frac{s-\alpha-1}{s-2}} + \Theta\left(\left(\ell n\, F(v)\right)^{\frac{2}{s-2}} v^{\frac{s-2\alpha}{s-2}}\right) + \Theta(1) ,$$

where $F(v)$ is the solution to the equation

$$F(v) = \frac{a(v)}{\ln F(v)} \text{ with } a(v) = \frac{s(s-1)(s-2)}{2} v^{\alpha-1} .$$

For any $\varepsilon > 0$ we have for large v

$$\frac{a(v)}{\ln a(v)} < F(v) < \frac{a(v)}{\ln a(v)} \left(1 + (1+\varepsilon)\frac{\ln \ln a(v)}{\ln a(v)}\right) ,$$

so $\ln F(v) \sim (s-1)\ln v$, where $x(v) \sim y(v)$ means $\lim\limits_{v\to\infty} \frac{x(v)}{y(v)} = 1$.

This gives

$$\begin{aligned} N \leq \exp\Bigg[&\frac{2}{s-1}\left(\frac{2}{s(s-1)(s-2)}\right)^{\frac{1}{s-2}} \left(\ell n\ F(v)\right)^{\frac{s-1}{s-2}} v^{\frac{s-\alpha-1}{s-2}} \\ &+ 2\left(\frac{2}{s(s-1)(s-2)}\left(\ell n\ F(v)\right)\right)^{\frac{1}{s-2}} v^{\frac{s-\alpha-1}{s-2}} \\ &+ \Theta\left(\left(\ell n\ F(v)\right)^{\frac{s}{s-2}} v^{\frac{s-2\alpha}{s-2}}\right) + \Theta\left(\ell n\ v\right)\Bigg] . \end{aligned} \qquad (7)$$

The leading term in the exponent for the one-sum upper limit is larger than the corresponding term in the one-sum lower limit by the slowly increasing factor $(\ln F(v))^{\frac{s-1}{s-2}}$.

The two-sum cases require approximating a sum of the form

$$\sum_i \binom{a}{i} (-1)^i (1+ic)^t = (-1)^a \sum_j \binom{t}{j} \left\{ {j \atop a} \right\} a! c^j \ ,$$

where a, c, and t are function of v, q_0, and q_1 $(t=v^q)$, and where $\left\{ {j \atop a} \right\}$ is a Stirling number of the second kind [4]. Since each term on the right side is positive, a lower limit is the first non-zero term $(j=a)$. The first non-zero term is also a good approximation for the total sum if the quantity act approaches zero as v becomes large (in our application this happens when $\alpha > \frac{s}{2}$) . For the two-sum lower limit we use

$$(-1)^a \sum_i \binom{a}{i} (-1)^i (1+ic)^t > \binom{t}{a} a! c^a \ .$$

The two-sum lower limit summand is (approximately) maximized with

$$q_0 = \frac{2(s-1)}{y+s-1-\ln 2} \left[\frac{2y}{s(s-1)} \right]^{\frac{1}{s-2}} v^{\frac{s-\alpha-1}{s-2}} \quad \text{, and}$$

$$q_1 = 2 \left[\frac{2y}{s(s-1)} \right]^{\frac{1}{s-2}} v^{\frac{s-\alpha-1}{s-2}} \quad ,$$

where y is the solution of the equation

$$y + \ln\left(1 - \frac{\ln 2}{y}\right) + (s-2)\ln\left(1 + \frac{v-\ln 2}{s-1}\right) = 0 \ .$$

The value of y is between $\ln 2$ and $\ln 3$ for all $s \geq 3$; $y \simeq 1.0106135875$ for $s = 3$, and $y \simeq .9965516271$ for $s=4$. This gives

$y \rightarrow \ln 2$. This gives

$$N \geq \exp\left[2\left[\frac{2y}{s(s-1)}\right]^{\frac{1}{s-2}} \left[\ln 2 + \frac{y - \ln 2}{y+s-1-\ln 2}\,(2-y \ln 2) - \frac{y}{(s-1)}\right] v^{\frac{s-\alpha-1}{s-2}} + \Theta\left(v^{\frac{s-2\alpha}{s-2}}\right) + \Theta(\ln v)\right]. \tag{8}$$

It is difficult to obtain an asymptotic upper limit from limit (5) due to the problem of approximating the sum over i, for values of $q_1 \geq v^{\frac{s-\alpha}{s}}$. Combining the derivation of limits (4) and (5), however, gives

$$(2v)^{st} N \leq \sum_{0 \leq q_0 \leq q_1 < q_*} \sum_{i_1} 2^{q_1} (-1)^{q_1-q_0+i_1} \binom{v-q_0-1}{q_1-q_0-i_1,\, i_1,\, v-q_1-1} [P_u + i_1[q_1^s-(q_1-1)^s]]^t + v \sum_{q_* \leq q_0 < v} 2^{q_0} \binom{v-1}{q_0} P_0^t \;. \tag{9}$$

Using $i' = q_1-q_0-i_1$ and approximating the sum over i' gives

$$(2v)^{st} N \leq \sum_{0 \leq q_0 \leq q_1 < q_*} 2^{q_1} \frac{(v-q_0-1)!}{(v-q_1-1)!} \binom{t}{q_1-q_0} [q_1^s-(q_1-1)^s]^{q_1-q_0} [P_u + (q_1-q_0)[q_1^s-(q_1-1)^s]]^{t-q_1+q_0} \left(1 + O\left(\frac{q_1^s t}{v^s}\right)\right) + v \sum_{q_* \leq q_0 < v} 2^{q_0} \binom{v-1}{q_0} P_0^t \;; \tag{10}$$

provided $\frac{q_1^s t}{v^s}$ decreases as v increases. For $q_* = v^{\frac{s-\alpha}{s} - \varepsilon}$, $\varepsilon > 0$ and small, and $\alpha > \frac{s}{2}$, the approximation is valid. Also under these conditions the second summation will be small enough to be absorbed in the final error term. The first sum is maximized with

$$q_0 = 2\left(1 - \frac{\ln 2}{s-1}\right)\left(\frac{4 \ln 2}{s(s-1)}\right)^{\frac{1}{s-2}} v^{\frac{s-\alpha-1}{s-2}} + \Theta(1) \text{ , and}$$

$$q_1 = 2\left(\frac{4 \ln 2}{s(s-1)}\right)^{\frac{1}{s-2}} v^{\frac{s-\alpha-1}{s-2}} + \Theta(1) .$$

This gives

$$N \le \exp\left[\frac{2(s-2)\ln 2}{s-1}\left(\frac{4 \ln 2}{s(s-1)}\right)^{\frac{1}{s-2}} v^{\frac{s-\alpha-1}{s-2}} + \Theta(\ln v)\right] . \tag{11}$$

for $\frac{s}{2} < \alpha < s-1$. Unfortunately, we do not have a good approximation for $1 < \alpha \le \frac{s}{2}$.

6. Conclusion

For $\frac{s}{2} < \alpha < s-1$ and $s \ge 3$ we have shown that

$$N = \exp\left[\Theta\left(v^{\frac{s-\alpha-1}{s-2}}\right)\right] ,$$

while for $1 < \alpha \le \frac{s}{2}$ and $s \ge 3$ we have

$$\exp\left[\Theta\left(v^{\frac{s-\alpha-1}{s-2}}\right)\right] \le N \le \exp\left[\Theta\left\{\left(v^{\frac{s-\alpha-1}{s-2}}\right)(\ell n\ v)^{\frac{s-1}{s-2}}\right\}\right] .$$

The coefficient of the leading term of the two-sum upper limit is exactly $2^{\frac{1}{s-2}}$ times the coefficient of the leading term of the one-sum lower limit, so the upper and lower bounds on the coefficient are close for large s. Table 5 shows the value of the leading coefficients for Formulas 6, 8, and 11 for small values of s. The two-sum lower limit gives only a small improvement over the one-sum lower limit.

The time for simple backtracking on these problems [2] is

$$N = \exp\left[\Theta\left(v^{\frac{s-\alpha}{s-1}}\right)\right] ,$$

so the speed-up from using simple search rearrangement is comparable to that obtained by switching to simpler problems with s reduced by one. (For $\frac{s}{2} < \alpha < s$ the leading coefficient is also somewhat smaller for the search rearrangement algorithm.) Since this change is in the exponent the improvement in expected running time obtained by using simple search rearrangement is dramatic.

The values of q_0 and q_1 at the maximum in the two-sum approximations suggest that in search rearrangement backtracking there is typically a large number of unary nodes preceding each binary node. This conclusion is confirmed by our experimental measurements.

An interesting question for future research is whether there are backtracking algorithms (such as perhaps the ones reported in [6]) that use average time $\exp[\Theta(v^{(s-\alpha-k+1)/(s-k)})]$) for each fixed $k \le s-1$. Experimental results show that the algorithms in [6] perform better than simple search rearrangement on very large problems, but they have not been analyzed as yet.

Acknowledgement

We wish to thank our referee, Prof. James Fill, for his careful reading of this paper and his numerous helpful comments.

Required terms: $j_{ix} \ge 1$ for $1 \le i \le m$, $1 \le x \le q_i - q_{i-1}$

Size of set	Reason
$R_{i-1} = Q_{i-1} - Q_{i-2}$, where $Q_i = (q_i+1)^s - q_i^s$, $Q_{-1} = 0$.	Term must have false literals combined with literals for forced variables. The forced variable cannot be forced on earlier round.

Permitted terms: $t - \sum j_{ix}$

Size of set	Reason
$(2v)^s$	Total number of terms, minus :
$-q_m^s$	terms made of false literals ,
$-(v - q_m)Q_m$	terms that force variables to true, and
$-\sum_{0 \le i \le m} (v-q_i)R_i$	terms that force variables to false .

TABLE 1

A summary of the analysis for the exact formula

<table>
<tr><td colspan="2">Required terms: None</td></tr>
<tr><td colspan="2">Permitted terms: t</td></tr>
<tr><th>Size of set</th><th>Reason</th></tr>
<tr><td>$(2v)^s$</td><td>Total number of terms, minus:</td></tr>
<tr><td>$-q_0^s$</td><td>terms made of false literals, and</td></tr>
<tr><td>$-2(v - q_0)Q_0$</td><td>terms that force unforced variables.</td></tr>
</table>

TABLE 2

A summary of the analysis for the one-sum upper limit formula

Required terms: $j_{1x} \geq 1$ for $1 \leq x \leq q_1 - q_0$	
Size of set	Reason
$q_1^s - (q_1 - 1)^s$	Each forced variable must occur in a term with all other literals false.
Permitted terms: $t - \sum_x j_{1x}$	
Size of set	Reason
$(2v)^s$	Total number of terms, minus :
$-q_1^s$	terms made of false literals ,
$-2(v - q_1)Q_1$	terms that force unforced variables , and
$-(q_1^s - q_0^s)$	total number of required terms.

TABLE 3

A summary of the analysis for the two-sum upper limit formula

Required terms: $j_x \geq 1$ for $q_0 < x \leq q_1$	
Size of set	Reason
$x^s - (x-1)^s$	Each forced variable must occur in a term with all other literals false, where all other variables are already set.
Permitted terms: Same as in Table 3	

TABLE 4
A summary of an alternate two-sum upper limit

s	one-sum lower	two-sum lower	two-sum upper
3	0.1602	0.2465	0.3203
4	0.3141	0.3922	0.4442
5	0.4271	0.5226	0.5381
6	0.5142	0.5760	0.6115
7	0.5840	0.6398	0.6708
8	0.6415	0.6924	0.7200
9	0.6899	0.7367	0.7617
10	0.7314	0.7748	0.7976
100	1.2534	1.2595	1.2624
1000	1.3663	1.3670	1.3672
∞	$2\ln 2 \approx 1.3863$	$2\ln 2 \approx 1.3863$	$2\ln 2 \approx 1.3863$

TABLE 5

Coefficients c for $N \sim \exp cv^{\frac{s-\alpha-1}{s-2}}$

Appendix 1

A number of numerical checks were performed to verify the results in this paper.

Equation 1 was verified by a computer program that performed direct enumeration for $s = 1$, $1 \leq t \leq 4$, $1 \leq v \leq 8$; $s = 2$, $1 \leq t \leq 2$, $1 \leq v \leq 8$; $s = 2$, $t = 3$, $1 \leq v \leq 6$; $s = 3$, $t = 1$, $1 \leq v \leq 8$; and $s = 3$, $t = 2$, $1 \leq v \leq 4$. A statistical check with one percent accuracy was done for $s = 3$, $t = v^{3/2}$, $v = 4$ and 9.

Formula 6 was compared with the maximum term in the one-sum lower limit. At $v = (1337)^2$, $s = 3$, $t = (1337)^3$, the ratio of the logarithm of Formula 6 to the logarithm of the maximum term was 1.0013, and converging.

Formula 7 was compared with the one-sum upper limit (Formula 4). At $v = (34)^2$, $s = 3$, $t = (34)^3$ the ratio of the logarithms achieved a minimum of 0.988. For higher v it slowly increased.

Formula 8 was compared with the maximum term in the two-sum lower limit (Formula 3). At $v = 64$, $s = 3$, $t = 512$ the ratio of the logarithms was 1.5 and decreasing. Roundoff error prevented measurements at significantly larger v.

Formula 11 was compared with the two-sum upper limit (Formula 5). At $v = (11)^2$, $s = 3$, $t = (11)^3$ the ratio was 0.805 and erratically increasing.

Appendix 2

In this appendix we give more details of the derivation of the asymptotic results mentioned in Section 5. The one-sum lower limit is

$$\sum_{0 \le q_0 < v} 2^{q_0} \cdot P_0^t$$

$$= \sum_{0 \le q_0 < v} 2^{q_0} \left((2v)^s - q_0^s - 2(v - q_0)\left((q_0 + 1)^s - q_0^s\right)\right)^t .$$

The logarithm of a term is

$$q_0 \ln 2 + t \ln\left[(2v)^s - q_0^s - 2(v - q_0)\left((q_0 + 1)^s - q_0^s\right)\right] .$$

Set $t = v^\alpha$, take the derivative with respect to q_0, and set it equal to 0.

Assuming $q_0 \ll v$, the largest terms of the derivative give

$$2s(s-1)v^{\alpha+1} q_0^{s-2} = (2v)^s \ln 2 ,$$

or

$$q_0 = \left(\frac{2^{s-1} \ln 2}{s(s-1)}\right)^{\frac{1}{s-2}} v^{\frac{s-\alpha-1}{s-2}} .$$

Substitute this value back into the original derivative; keeping the largest terms gives

$$q_0 = \left(\frac{2^{s-1} \ln 2}{s(s-1)}\right)^{\frac{1}{s-2}} v^{\frac{s-\alpha-1}{s-2}} \left(1 + \Theta\left(v^{\frac{-(\alpha-1)}{s-2}}\right) + \Theta\left(v^{-\frac{(s-\alpha-1)}{s-2}}\right)\right) .$$

Substitute this value into the original formula to get the logarithm of the maximum term; the most significant terms give

$$v^{\alpha}\ell n\,(2v)^{s} + \left(\frac{2\ \ell n\ 2}{s(s-1)}\right)^{\frac{1}{s-2}} (2\ \ell n\ 2)\left(\frac{s-1}{s-2}\right) v^{\frac{s-\alpha-1}{s-2}} + \Theta\ v^{\frac{s-2\alpha}{s-2}} + \Theta\,(1)\ .$$

Since all the terms are positive, the value of one term is a lower limit on the value of the sum.

Formula (4) gives the one-sum upper limit. Using Stirling's approximation for the binomial, the log of a term is $\ell n v - \frac{1}{2}\ \ell n(2\pi) + (v - \frac{1}{2})\ell n(v-1)$ $-(q_0 + \frac{1}{2})\,\ell n\ q_0 - (v - q_0 - \frac{1}{2})\,\ell n(v - q_0 - 1) + q_0\ \ell n\ 2 + v^{\alpha}\ \ell n((2v)^{s} - q_0^{s}$ $-2(v - q_0)((q_0 + 1)^{s} - q_0^{s})) + O(\frac{1}{q_0}) + O(\frac{1}{v - q_0})$. Take the derivative with respect to q_0 and set it equal to zero. Assuming $q_0 \ll v$, and keeping the asymptotically most significant terms, shows that q_0 near the peak satisfies the relation.

$$q_0^{s-2} \sim \frac{2^{s-1}}{s(s-1)}\ v^{s-\alpha-1}\ \ell n\left(\frac{2v}{q_0}\right)\ .$$

To obtain an approximation of q_0 at the peak, let $a(v) = \frac{2}{s(s-1)(s-2)}\ v^{1-\alpha}$ and let $F(v)$ be the solution to the equation

$$\frac{1}{F(v)} = a(v)\ \ell n F(v)\ .$$

Then

$$q_0 \sim 2v\ \ a(v)^{\frac{1}{s-2}}\ (\ell n\ F(v))^{\frac{1}{s-2}}\ \ ;$$

attention to the O terms gives the value of q_0 given in the text preceding Formula (7). The local maximum at $q_0 = v-1$ is even less significant for this limit than it was for the one-sum lower limit. The value of the upper limit is no more than v times the value of the maximum term, and this factor is absorbed in the $\Theta(\ln v)$ term.

The two-sum lower limit is given in Formula (3). It can be rewritten in the following form by expanding the multinomial:

$$\sum_{0 \le q_0 \le q_1 < v} \sum_i 2^{q_1} \binom{v-q_0-1}{q_1-q_0} \binom{q_1-q_0}{i} (-1)^{q_1-q_0+i} (P_1 + i\,Q_0)^t \ .$$

Using the transformation $\sum_i \binom{a}{i}(-1)^i (1+ic)^t = \sum_j (-1)^a \binom{t}{j} \left\{ {j \atop a} \right\} a!\, c^j$,

with $a = q_1 - q_0$, $c = \frac{Q_0}{P_1}$, gives, for the sum over i ,

$$P_1^{\,t}(-1)^{q_1-q_0} \sum_i \binom{q_1-q_0}{i} (-1)^i \left(1 + i\,\frac{Q_0}{P_1}\right)^t = \sum_j \binom{t}{j} \left\{ {j \atop q_1-q_0} \right\} (q_1-q_0)!\,Q_0^{\,j} P_1^{\,t-j} \ .$$

A lower limit for this sum is $\binom{t}{q_1-q_0}(q_1-q_0)!\,Q_0^{\,q_1-q_0}\,P^{\,t-q_1+q_0}$. We thus obtain the lower limit

$$\sum_{0 \le q_0 \le q_1 < v} 2^{q_1} \binom{v-q_0-1}{q_1-q_0} \binom{t}{q_1-q_0} (q_1-q_0)!\; Q_0^{\,q_1-q_0}\; P_1^{\,t-q_1+q_0} \ .$$

Using Stirling's approximation, the log of a term is $q_1 \ln 2$

$$\begin{aligned}
&(v-q_0-\tfrac{1}{2}) \ln(v-q_0-1) + (v^\alpha + \tfrac{1}{2}) \ln v^\alpha - \tfrac{1}{2} \ln 2\pi - (q_1-q_0+\tfrac{1}{2}) \ln(q_1-q_0) - \\
&(v-q_1-\tfrac{1}{2}) \ln(v-q_1-1) - (v^\alpha-q_1+q_0+\tfrac{1}{2}) \ln(v^\alpha-q_1+q_0) + (q_1-q_0) \ln((q_0+1)^s-q_0^{\,s}) + \\
&(v^\alpha-q_1+q_0) \ln[(2v)^s-q_1^{\,s}-2(v-q_1)((q_1+1)^s-q_1^{\,s}) - (q_1-q_0)((q_0+1)^s-q_0^{\,s})] + \\
&\Theta\left(\frac{1}{v-q_0-1}\right) + \Theta\left(\frac{1}{v^\alpha}\right) + \Theta\left(\frac{1}{q_1-q_0}\right) + \Theta\left(\frac{1}{v-q_1-1}\right) + \Theta\left(\frac{1}{v^\alpha-q_1+q_0}\right) .
\end{aligned}$$

Take the derivative with respect to q_1 , set it equal to 0 , and solve, retaining asymptotically important terms. This gives

$$\ell n \left(\frac{s\, q_0^{s-1}}{2^{s-1}\, v^{s-\alpha-1} (q_1 - q_0)} \right) = \frac{s(s-1)\, q_1^{s-2}}{v^{s-\alpha-1}\, 2^{s-1}} \ .$$

The same procedure on the q_0 derivative gives

$$0 = \ell n \left(\frac{2^{s-1}\, v^{s-\alpha-1}\, (q_1 - q_0)}{s\, q_0^{s-1}} \right) + \ell n\, 2 + (s-1) \left(\frac{q_1}{q_0} - 1 \right) - \frac{v^{\alpha-s}\; s}{2^s}$$

$$\left((s-1) q_1\, q_0^{s-2} - q_0^{s-1} \right) .$$

Use the value of the ℓn term from the q_1 derivative to replace the term in the q_0 derivative. Identify the asymptotically important terms using $q_0 \sim v^{\frac{s-\alpha-1}{s-2}}$ and $q_1 \sim v^{\frac{s-\alpha-1}{s-2}}$. This gives

$$q_0 \sim \frac{q_1}{\dfrac{s\, q_1^{s-2}}{v^{s-\alpha-1}\, 2^{s-1}} - \dfrac{\ell n\, 2}{s-1} + 1} \ .$$

Substitute this value of q_0 into the formula for the q_1 derivative. Changing the variable to $x = \dfrac{s\, q_1^{s-2}}{2^{s-1}\, v^{s-\alpha-1}}$ gives the equation

$$(s-1)x = \ell n \frac{x}{\left(x + 1 - \frac{\ell n\, 2}{s-1} \right)^{s-2} \left(x - \frac{\ell n\, 2}{s-1} \right)} \ .$$

This equation can be solved for x in terms of s . Setting $y = (s-1)x$ gives the results in the text. We have not calculated the errors in the values for q_0 and q_1 that maximize the value of a term; this is not necessary since the value of any term in a sum of positive terms is a lower limit on the value of the sum.

The two-sum upper limit is given in Formula 5. Because of limitations of our asymptotic methods, we derive an upper bound by using terms from Formula 5 over a part of the range and from Formula 4 on the remainder, as indicated in Formula 9. This is possible because each individual term in Formula 4 for a fixed q_0 bounds the partial sum in Formula 1 with q_m fixed to that same value. Likewise, a partial sum in Formula 5 with q_1 fixed to some value bounds the partial sum in Formula 1 with q_m fixed to that value.

To remove the sum over i_1 in Formula 9, use the transformation $(-1)^a \sum_i \binom{a}{i}(-1)^i(1+ic)^t = \binom{t}{a} a!c^a(1+\Theta(act))$, with $a = q_1 - q_0$ and $c = \frac{q_1^s - (q_1-1)^s}{P_2}$. This transformation is valid for $\alpha > \frac{s}{2}$ and $q_1 < v^{\frac{s-\alpha}{s} - \varepsilon}$, where $\varepsilon > 0$.

To find the values of q_1 and q_0 that maximize the first term in the resulting formula (Formula 10), analyze it as follows. Using Stirling's approximation, the log of a term is $q_1 \ln 2 - \frac{1}{2} \ln 2\pi +$ $(v - q_0 + \frac{1}{2}) \ln(v - q_0 - 1) + (v^\alpha + \frac{1}{2}) \ln v^\alpha - (v - q_1 - \frac{1}{2}) \ln(v - q_1 - 1) -$ $(q_1 - q_0 - \frac{1}{2}) \ln(q_1 - q_0) - (v^\alpha - q_1 + q_0 - \frac{1}{2}) \ln(v^\alpha - q_1 + q_0) + (q_1 - q_0)$ $\ln(q_1^s - (q_1-1)^s) + (v^\alpha - q_1 + q_0) \ln[P_2 + (q_1 - q_0)(q_1^s - (q_1-1)^s)] + \Theta(\frac{1}{v - q_0 - 1}) +$ $\Theta(\frac{1}{v^\alpha}) + \Theta(\frac{1}{v - q_1 - 1}) + \Theta(\frac{1}{q_1 - q_0}) + \Theta\left(\frac{1}{v^\alpha - q_1 + q_0}\right) +$ $\Theta\left(\frac{(q_1-q_0)v^\alpha(q_1^s - (q_1-1)^s)}{P_2 + q_1 - q_0(q_1^s - (q_1-1)^s)}\right)$. Take the derivative with respect to q_1 and set it equal to zero; the important terms give

$$0 = \ell n\left(\frac{sq^{s-1}}{(q_1-q_0)2^{s-1}\ v^{s-\alpha-1}}\right) + \frac{(q_1-q_0)(s-1)}{q_1} - \frac{s(s-1)q_1^{s-2}}{2^{s-1}\ v^{s-\alpha-1}}$$

$$+\,O\left(v^{\frac{1-\alpha}{s-2}}\right) + O\left(v^{\frac{-(s-\alpha-1)}{s-2}}\right).$$

The important terms from the q_0 derivative give

$$0 = \ell n\left(\frac{(q_1-q_0)2^{s}\ v^{s-\alpha-1}}{s\ q_1^{s-1}}\right) + O\left(v^{\frac{1-\alpha}{s-2}}\right) + O\left(v^{\frac{-(s-\alpha-1)}{s-2}}\right) .$$

This implies the log term is of the same order as the O terms. Use this to rewrite the q_1 equation and solve it for q_0 in terms of q_1, giving

$$q_0 = (1 + \frac{\ell n\ 2}{s-1})q_1 - \frac{s\ q_1^{s-1}}{2^{s-1}\ v^{s-\alpha-1}} + O\left(v^{\frac{s-2\alpha}{s-2}}\right) + O(1) .$$

Since our approximation is only good for $s-2\alpha < 0$, the first O term can be dropped.

Substitute this value into the equation from the q_0 derivative, exponentiate each side of the resulting equation, and expand the right side in a power series. This gives

$$\frac{\left(-(\frac{\ell n\ 2}{s-1}) + \frac{s\ q_1^{s-2}}{2^{s-1}\ v^{s-\alpha-1}} + O\left(v^{\frac{-(s-\alpha-1)}{s-2}}\right)\right) 2^{s}\ v^{s-\alpha-1}}{s\ q_1^{s-2}}$$

$$= 1 + O\left(v^{\frac{1-\alpha}{s-1}}\right) + O\left(v^{\frac{-(s-\alpha-1)}{s-2}}\right) .$$

Letting $x = \frac{s\, q_1^{s-2}}{2^{s-1}\, v^{s-\alpha-1}}$, we obtain

$$x = \frac{2 \ln 2}{s-1} + O\left(v^{\frac{-(s-\alpha-1)}{s-2}}\right) ,$$

or

$$q_1 = \left(\frac{2^s \ln 2}{s(s-1)}\right)^{\frac{1}{s-2}} v^{\frac{s-\alpha-1}{s-2}} + O(1) ,$$

and

$$q_0 = (1 - \frac{\ln 2}{s-1})\, q_1 + O(1) .$$

By substituting these values into Formula 10 and ignoring the second sum we obtain Formula 11. That sum is small relative to the terms we retain. The maximum term in the sum is obtained by setting $q_0 = q_*$. The sum is no bigger than v times its maximum term. The log of the sum is $2s^{-1/s} \left(\frac{\alpha \ln v}{s} + 1 + \ln 2\right) v^{\frac{s-\alpha}{s}} + O\left(v^{\frac{\alpha-s}{s}}\right)$. For $\alpha > \frac{s}{2}$, this can be neglected in relation to the terms in Formula 11 (since, for example, $\exp(v^2 + O(1)) + \exp(v) = \exp(v^2 + O(1))$ asymptotically).

References

1. J.R. Bitner and E.M. Reingold, *Backtrack programming techniques*, CACM 18 (1975), pp. 651-655.
2. Cynthia A. Brown and Paul W. Purdom, Jr., *An average time analysis of backtracking*, SIAM J. Comput. 3 (1981) pp. 583-593.
3. Cynthia A. Brown and Paul W. Purdom, Jr., *An empirical comparison of backtracking algorithms*, IEEETPAMI 4 (1982), pp. 309-316.
4. D.E. Knuth, *Estimating the efficiency of backtracking programs*, Math. Comput. 29 (1975), pp. 121-136.
5. D.E. Knuth, *The Art of Computer Programming*, vol. 1 (1973), p. 65.
6. Paul W. Purdom, Jr., Cynthia A. Brown, and Edward L. Robertson, *Backtracking with multi-level search rearrangement*, Acta Informat. 15 (1981), pp. 99-113.