# Chapter 5

# Characterising and Improving SLS Behaviour

Building on Chapter 4, here we extend and refine the analysis of SLS behaviour. We perform this analysis for the SLS algorithms for SAT introduced and discussed before; for empirical investigations we use the benchmark suite described in Section 4.4. However, the methodology used here is not restricted to SAT but can be rather easily applied to other problem domains and different SLS algorithms. In our refined analysis of SLS behaviour we first focus on qualitative norms of behaviour like approximate completeness. For the first time we give theoretical results concerning the approximate completeness of some of the most powerful currently known SLS algorithms for SAT. Next, we extend our empirical analysis of SLS behaviour by functionally characterising the algorithm's RTDs for optimal and sub-optimal noise parameter settings. Our analysis reveals a suprisingly regular behaviour of some of the best SLS algorithms for SAT when applied to hard instances from different benchmark domains. From these empirical results we derive a number of theoretically and practically interesting consequences, concerning parameterisation and parallelisation of these algorithms as well as an interpretation of their run-time behaviour. Finally, we show how, based on our characterisation results, some of the best currently known SLS algorithms for SAT can be further improved.

## 5.1 Characterising SLS Behaviour

In the last chapter we compared different SLS algorithms for SAT on our suite of benchmark problems; now, we analyse the behaviour of these algorithms in more detail. For this analysis we always use the algorithms without a restart mechanism, as the effects of the restart mechanism depend crucially on the run-time behaviour of the pure strategies and can be easily determined when its run-time distribution is known. In this section, we use the functional characterisation of run-time distributions as a central method, *i.e.*, we develop a mathematical model which describes the empirically observed behaviour correctly and accurately.

### 5.1.1 Asymptotic Run-time Behaviour

Local search algorithms as the ones we study here belong to the class of Las Vegas algorithms. As discussed in an abstract context in Chapter 2, these can be classified into three qualitative norms of behaviour with respect to their asymptotic behaviour for arbitrarily long run-times (see Section 2.1.1).

While all GSAT and WalkSAT variants are *incomplete*, when evaluating these algorithms we already observed that some of them appear to be *essentially incomplete*, while others seem to be *approximately complete*, *i.e.*, with increasing run-time the probability of finding a solution for a soluble instance approaches one. We will use the term *PAC behaviour*[1] to characterise a situation, where a Las Vegas algorithm, applied to a particular problem instance, shows approximately complete behaviour. Algorithms which show PAC behaviour for all soluble instances will be called *PAC algorithms.* Note, that empirical observations can neither prove PAC behaviour for a given algorithm applied to a particular instance (because based on a finite number of runs, it cannot be ruled out that for some very unlikely search trajectories, the algorithms gets stuck and will never find a solution), nor the PAC property of an algorithm. Essential incompleteness cannot be proven empirically, either: although it suffices to find one instance for which the RLD does not converge towards one, all experimental analyses have to be based on finite run-times – consequently, it can never be ruled out empirically that for even longer run-times convergence could be observed. Nevertheless, empirical results can indicate PAC behaviour and essential incompleteness; moreover, the significance of such evidence can be improved by increasing the number and run-time of experiments.

At the time of this writing we are not aware of any theoretical result concerning the approximate completeness or essential incompleteness of any of the SLS algorithms for SAT studied here. Our empirical evaluations (*cf.* Chapter 4) suggest the following hypotheses:

- plain GSAT is essentially incomplete;
- GWSAT is approximately complete for $\mathsf{wp} > 0$;
- GSAT+tabu is essentially incomplete for very small and very large tabu list lengths;
- WalkSAT is approximately complete for noise parameter settings $\mathsf{wp} > 0$;
- WalkSAT+tabu is essentially incomplete for very small and very large tabu list lengths;
- Novelty is essentially incomplete at least for some noise parameter settings;

[1]PAC is the abbreviation for Probabilistically Aproximately Complete; the term is inspired by *PAC learning*, a related concept from machine learning [KM90, Mit97].

- R-Novelty is essentially incomplete at least for some noise parameter settings.

While some of these hypotheses can be proven or refuted theoretically, for others it is not clear how to find a proof. To indicate how theoretical results concerning approximate completeness can be derived, we give proof sketches for some cases here. First we prove the approximate completeness for GSAT with random walk (GWSAT).

**Theorem 5.1**

*GWSAT is approximately complete for all* $\mathsf{wp} > 0$.

---

**Proof** For a given instance with $n$ variables and $k$ clauses, we show that there exists a $p' > 0$ such that from each non-solution assignment GWSAT can reach a solution with a probability $p \geq p'$: Let $a$ be the current (non-solution) assignment, and $s$ the solution with minimal hamming distance $h$ from $a$.

**Lemma 1**: For arbitrary $a$ and $s$ there is always one valid GWSAT step which decreases $h$ by one. In particular, there is always at least one such random walk step.

To see why this holds, assume that no such random walk step exists. Then none of the variables whose values are different in $a$ and $s$ can appear in an unsatisfied clause. But since $a$ is not a solution, there has to be at least one clause $c$ which is violated by $a$; now the variables appearing in $c$ have the same value in $a$ and $s$, therefore $s$ also violates $c$ and cannot be a solution. Since this contradicts the assumption, Lemma 1 has to be true.

Using Lemma 1 inductively, one can construct a sequence of $h$ random walk steps from $a$ to $s$. Next, we derive a lower bound for the probability with which GWSAT will execute this sequence.

Note first that for any assignment $a'$, the number of unsatisfied clauses is always less or equal to $k$, and the number of literals occurring in unsatisfied clauses is less or equal to $k \cdot l$, where $l$ is the length of the longest clause in the given instance. Therefore, if GWSAT decided to do a random walk step, the probability to select a variable which decreases $h$ is at least $1/(k \cdot l)$. Thus, the overall probability of executing a random walk step which decreases the hamming distance to the nearest solution is at least $\mathsf{wp}/(k \cdot l)$. Since GWSAT's steps depend only on the current assignment (Markov property), a lower bound for the probability of executing the correct $h$ step sequence to reach the solution can be estimated as $[\mathsf{wp}/(k \cdot l)]^h$.

Finally, note that $h$ is always less or equal to $n$, and therefore the probability of reaching a solution from any given assignment $a$ is at least $p' = [\mathsf{wp}/(k \cdot l)]^n$. Since this probability is independent from the number of steps which have been performed, the probability of finding a solution within $t$ steps is at least

$$\sum_{i=1}^{t} (1 - p')^{i-1} \cdot p' = p' \cdot \frac{(1 - p')^t - 1}{(1 - p') - 1}$$

For $t \to \infty$, this geometric series converges to $p'/p' = 1$, which proves GWSAT's approximate completeness. □

Note that this proof relies critically on the fact that the algorithm can decrease the hamming distance to the nearest solution in each step. Our proof shows that this is guaranteed if arbitrarily long sequences of random walk steps can be performed with a probability $p \geq p' > 0$ which is independent of the number of steps which have been performed in the past. Since our estimates for the probability of finding a solution did not depend on the actual variable selection strategy, the proof generalises easily to each algorithm which satisfies this condition.

The algorithms of the WalkSAT family, however, generally do not allow arbitrarily long sequences of random walk steps. In particular, for WalkSAT the variable selection strategy does not allow a random walk step if the selected clause contains a variable which can be flipped without breaking any currently satisfied clauses. Therefore, approximate completeness cannot be proven using the scheme given above. Actually, although our empirical results suggest that WalkSAT could be approximately complete, a proof seems to be difficult to find. For the other members of the WalkSAT family, we can prove their essential incompleteness using a very simple example instance. The proofs for WalkSAT+tabu, Novelty, and R-Novelty are very similar; therefore we give only the proof for Novelty here, and then discuss the corresponding results for the other algorithms.

**Theorem 5.2**

*Novelty is essentially incomplete for arbitrary noise parameter settings.*

---

**Proof** Let $F = \bigwedge_{i=1}^{6} c_i$ be the formula consisting of the clauses:

$$\begin{aligned}
c_1 &\equiv \neg x_1 \vee x_2 \\
c_2 &\equiv \neg x_2 \vee x_1 \\
c_3 &\equiv \neg x_1 \vee \neg x_2 \vee \neg y \\
c_4 &\equiv x_1 \vee x_2 \\
c_5 &\equiv \neg z_1 \vee y \\
c_6 &\equiv \neg z_2 \vee y
\end{aligned}$$

$F$ is satisfiable and has exactly one model $(x_1 = x_2 = \top, y = z_1 = z_2 = \bot)$. Now assume that the algorithm's current assignment is $A_1 \equiv (x_1 = x_2 = y = z_1 = z_2 = \top)$. In this situation, all clauses except $c_3$ are satisfied. Applying Novelty, $c_3$ will be selected and the variables receive the following scores: $x_1 : 0, x_2 : 0, y : -1$. Since regardless of the noise parameter, Novelty always flips the best or second-best variable, either $x_1$ or $x_2$ will be flipped.

Because both cases are symmetric, we assume without loss of generality that $x_1$ is flipped. This leads to the assignment $A_2 \equiv (x_1 = \bot, x_2 = y = z_1 = z_2 = \top)$ which satisfies all clauses except $c_2$. The scores for $x_1$ and $x_2$ are both 0, and since $x_1$ is the most recently flipped variable, $x_2$ will be picked now. The current assignment at this point is $A_3 \equiv (x_1 = x_2 = \bot, y = z_1 = z_2 = \top)$ which satisfies all clauses except $c_4$. Again, $x_1$ and $x_2$ have the same score of 0, but now $x_2$ is the most recently flipped variable, so $x_1$ will be flipped. Now, the current assignment is $A_4 \equiv (x_1 = \top, x_2 = \bot, y = z_1 = z_2 = \top)$, which leaves only $c_1$ unsatisfied. As before, $x_1$ and $x_2$ both receive a score of 0, and $x_2$ will be flipped. But this leads back to the assignment $A_1 \equiv (x_1 = x_2 = y = z_1 = z_2 = \top)$; therefore, Novelty got stuck in a loop.

Therefore, we found a soluble problem instance $F$ and an initial assignment $A_1$ for which Novelty will never reach a solution, regardless of how long it runs. Consequently, Novelty cannot be approximately complete but has to be essentially incomplete. $\Box$

Note that this result does not mean that Novelty will show essentially incomplete behaviour on each given problem instance. However, while for most of our benchmark problems, we observed approximately complete behaviour, for some instances Novelty got stuck in loops (*cf.* Chapter 4). Of course, in our example as well as in other cases, looping behaviour will usually only occur with a certain probability, since many trajectories of the algorithm avoid assignments leading into loops. Using the same example, one can easily prove:

**Corollary 5.3 WalkSAT**

*+tabu and R-Novelty are essentially incomplete for arbitrary tabu-list lengths and noise parameter settings, resp.*

Note that for R-Novelty, even the built-in deterministic loop breaking strategy (randomly picking a variable from the selected clause every 100th step) does not prevent the algorithm from getting stuck in loops, since these can be timed such that the loop breaker will never be activated when $c_3$ is violated — which would be the only way of flipping $y$ and reaching a solution. In the case of WalkSAT+tabu, the same loop will be observed for any tabu list length $\mathsf{tl} > 0$. For $\mathsf{tl} \geq 2$, the reason for this is the fact that when all variables in a clause are tabu, WalkSAT+tabu will not flip any variable at all (*cf.* Section 4.3); for $\mathsf{tl} = 1$, as for Novelty, $y$ can never be flipped when $c_3$ is selected.

So generally, although algorithms like WalkSAT+tabu, Novelty, and R-Novelty show mostly superior performance compared to GWSAT and WalkSAT, they suffer from essential incompleteness. However, as we will discuss in Section 5.2, based on the proofs outlined here we can overcome this weakness and thus substantially improve the most powerful SLS algorithms for SAT.

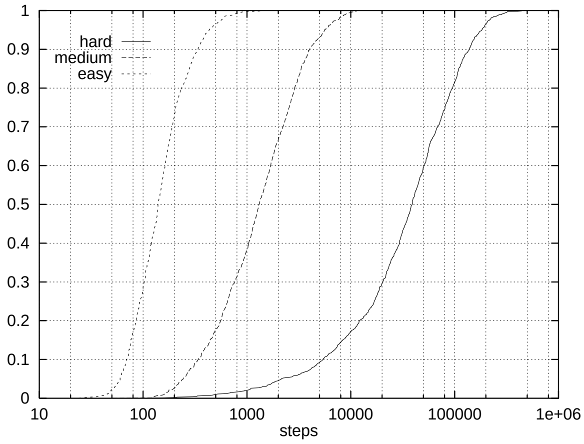


Figure 5.1: RLDs for WalkSAT (approx. optimal noise, `wp` $= 0.55$) applied to easy, medium, and hard problem instance from Random-3-SAT test-set `uf100-430`.

### 5.1.2 Functional Approximations for Optimal Noise

In the last section, we characterised the asymptotic behaviour of various modern SLS algorithms for SAT; now, we refine our analysis by functionally characterising their run-time behaviour. For this analysis, we apply the empirical methodology developed in Chapter 2, *i.e.*, for a given algorithm, we approximate the empirical RLDs for individual problem instances with continuous probability distributions; the goodness-of-fit of these approximations is tested using a standard $\chi^2$ test. Our empirical results show that for optimal noise parameter settings, the run-time distributions of the SLS algorithms studied here when applied to hard problem instances can be well aproximated by exponential distributions. We use the term *EPAC behaviour*[2] to characterise a situation, where the RLD for a given Las Vegas algorithm, when applied to a particular problem instance, can be well approximated by an exponential distribution. Informally, we refer to algorithms which show EPAC behaviour for a wide range of hard problem instances as *EPAC algorithms*. In the following sections, we will present the empirical results regarding the EPAC property of several modern SLS algorithms for the different problem domains from our benchmark suite.

**Random-3-SAT** Figure 5.1 shows the RLDs for WalkSAT (using an approximately optimal noise setting of 0.55, 1,000 tries and cutoff settings high enough to guarantee 100% success rate) when applied to the easy, medium, and hard problem instance from the `uf100-430` test-set. For the hard instance, the RLD can be approximated[3] using the cumulative form of an exponential distribution $ed[m](x) = 1 - 2^{-x/m}$ where $m$ is the median of the distri-

[2] EPAC is the abbreviation for Exponentially Probabilistically Aproximately Complete.

[3] All approximations were done using C. Gramme's "Gnufit" implementation of the Marquart-Levenberg algorithm.

| instance | median #steps $m$ | $\chi^2$ for $ed[m]$ | passed ($\alpha$) |
|---|---|---|---|
| uf100-430/easy | 132 | 501.68 | no |
| uf100-430/medium | 1,433 | 69.41 | no |
| uf100-430/hard | 61,082 | 27.51 | yes (0.05) |

Table 5.1: RLD approximations using exponential distributions $ed[m]$ for WalkSAT (approx. optimal noise) applied to the easy, medium, and hard instance from `uf100-430` test-set.

| test-set | acceptance level $\alpha$ | number passed |
|---|---|---|
| uf50-218 | 0.01 | 11.4% |
| uf50-218 | 0.05 | 7.1% |
| uf100-430 | 0.01 | 16.6% |
| uf100-430 | 0.05 | 10.3% |
| uf200-860 | 0.01 | 30% |
| uf200-860 | 0.05 | 26% |

Table 5.2: Fraction of instances passing the $\chi^2$ test for different Random-3-SAT test-sets.

bution and $x$ the number of steps required to find a solution.[4] For testing the goodness of this approximation we use a standard $\chi^2$-test [Roh76]. Basically, for a given empirical RLD this is done by estimating the parameter $m$ and comparing the deviations to the predicted distribution $ed[m]$. The result of this comparison is the $\chi^2$ value, where low $\chi^2$ values indicate a close correspondence between empirical and predicted distribution. Table 5.1 shows the estimated parameters $m$ and the $\chi^2$ values for the easy, medium, and hard instance mentioned before. It can be clearly seen that with increasing median search cost $m$, the $\chi^2$ values decrease, indicating that the harder the problem, the closer WalkSAT's RLD on this problem approximates an exponential distribution. For the medium and easy problem, this approximation is still reasonably good for the tail of the distribution (*i.e.*, for long runs), while for shorter runs the actual distribution is steeper than an exponential distribution. We conjecture that the deviations are caused by the initial hill-climb phase of the local search procedure (*cf.* [GW93a]).

This observation leads to the following hypothesis: *Applied to hard Random-3-SAT instances from the phase transition region, WalkSAT with optimal noise setting shows EPAC behaviour, i.e., its behaviour can be characterised using exponential distributions.* To test this hypothesis, we apply the methodology outlined above to the entire 50, 100, and 200 variable test-sets. The resulting correlation between median search cost and $\chi^2$ values can

[4] In the statistical literature, the exponential distribution $Exp(\lambda)$ is usually defined by $P(X \leq x) = 1 - e^{-\lambda x}$, which is equivalent to our representation $ed[m]$ using $m = \ln 2/\lambda$. A similar argument applies to the Weibull distribution mentioned later.

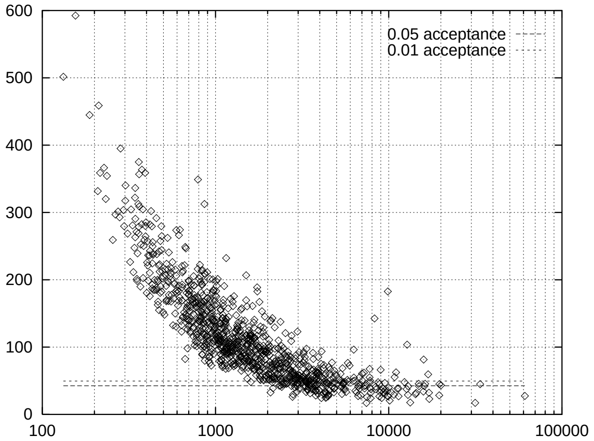


Figure 5.2: Correlation between hardness of problems (horizontal) and $\chi^2$ values (vertical) from testing the RLDs of individual instances versus a best-fit exponential distribution for test-set `uf100-430`. The horizontal lines indicate the acceptance thresholds for the 0.01 and 0.05 acceptance level (one of these lines is hard to see, since it coincides with the $\chi^2 = 50$ line).

be seen from the scatter plots given in Figures 5.2 and 5.3. Obviously, there is a strong negative correlation, indicating that, indeed, for harder problem instances, WalkSAT's behaviour can be more and more accurately characterised by exponential distributions. The figures also indicate two standard acceptance levels[5] for the $\chi^2$ test ($\alpha = 0.01$ and $\alpha = 0.05$). As can be seen from the plots, for high median search cost, almost all instances pass the $\chi^2$ test. Table 5.2 shows the overall percentage of the instances which passed the test for the different acceptance levels. The data suggests that for increasing problem hardness, a relatively higher number of instances pass the test, *i.e.*, the deviations of the RLDs from ideal exponential distributions apparently become less prominent for larger problems.

Table 5.3 reports the results of an analogous analysis for different algorithms when applied to the test-set `uf100-430`. This time, the $\chi^2$ test was restricted to the hardest 25% of the test-set. For WalkSAT+tabu and Novelty, stagnation behaviour was observed for some instances; these were removed from the test-set before approximating and testing the RLDs. As can be seen from Table 5.3, for ca. 50% of the tested instances the RLDs could be successfully approximated using exponential distributions. In the majority of the cases where the approximations failed the $\chi^2$ test, like pointed out earlier for WalkSAT, the reason lies in the deviations caused by the initial search phase. In summary, this shows clearly that the result established for WalkSAT generalises to the other, more powerful WalkSAT variants: Applied to hard Random-3-SAT instances, these show approximately

[5] The acceptance level $\alpha$ specifies the probability with which a statistical test incorrectely rejects the given hypothesis. Therefore, for lower acceptance levels, the test has to be more cautious with rejecting the hypothesis than for higher acceptance levels.

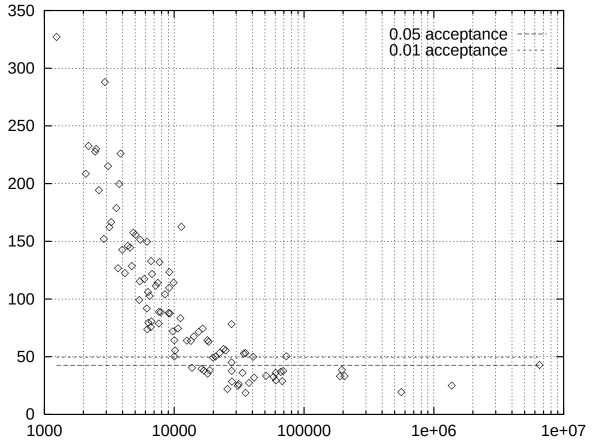


Figure 5.3: Correlation between hardness of problems (horizontal) and $\chi^2$ values (vertical) from testing the RLDs of individual instances versus a best-fit exponential distribution for test-set `uf200-860`. The horizontal lines indicate the acceptance thresholds for the 0.01 and 0.05 acceptance level.

exponential RLDs when using optimal noise settings. However, since these algorithms are essentially incomplete, there are instances for which they show stagnation behaviour and to which, consequently, this characterisation does not apply.

**Graph Colouring** In further experiments, we applied the same methodology to the `flat100-239` test-set from the graph colouring domain. As for the Random-3-SAT test-sets, we used approximally optimal noise parameter settings. For each algorithm we performed 1,000 tries, using an extremely large maxSteps value ($10^7$) to ensure maximal success rates for all instances. The results of the $\chi^2$ *vs* median search cost correlation analysis are summarised in Table 5.4. As for Random-3-SAT, we observe a strong negative correlation between the hardness of the problem instances and the $\chi^2$ values, indicating that with increasing median local search cost, the RLDs are more and more accurately approximated by exponential distributions. This holds for WalkSAT as well as for its more powerful, but essentially incomplete variants; for the latter the instances for which essentially incomplete behaviour is observed are not considered in the correlation analysis. Comparing the results for the `flat100-239` test-set with those for `flat50-115` (not reported here) confirm the observation we made for Random-3-SAT: with increasing problem size, the fraction of instances from the test-set for which the approximation using exponential distributions pass the $\chi^2$-test increases considerably. Note also that compared to the `uf100-430` test-set, the acceptance rates are generally higher for the `flat100-239` test-set. This is most probably caused by the fact that the `flat100-239` instances are larger and harder than the `uf100-430` instances.

| algorithm | acceptance level $\alpha$ | fraction passed | number removed |
|---|---|---|---|
| wsat(0.55) | 0.01 | 57.6% | 0 |
| | 0.05 | 34% | 0 |
| wsat+tabu(3) | 0.01 | 49.6% | 1 |
| | 0.05 | 36% | 1 |
| novelty(0.7) | 0.01 | 57.2% | 35 |
| | 0.05 | 40.2% | 35 |
| r-novelty(0.7) | 0.01 | 49.6% | 0 |
| | 0.05 | 34.8% | 0 |

Table 5.3: Fraction of instances from the hardest 25% of the `uf100-430` Random-3-SAT test-set passing the $\chi^2$ test for different algorithms with approx. optimal noise settings; the last column indicates the number of instances for which essentially incomplete behaviour was observed, these were removed from the test-set.

| algorithm | acceptance level $\alpha$ | fraction passed | number removed |
|---|---|---|---|
| wsat(0.5) | 0.01 | 52% | 0 |
| | 0.05 | 40% | 0 |
| wsat+tabu(3) | 0.01 | 64% | 1 |
| | 0.05 | 48% | 1 |
| novelty(0.6) | 0.01 | 84% | 1 |
| | 0.05 | 76% | 1 |
| r-novelty(0.6) | 0.01 | 100% | 1 |
| | 0.05 | 100% | 1 |

Table 5.4: Fraction of instances from the hardest 25% of the `flat100-239` Graph Colouring test-set passing the $\chi^2$ test for different algorithms with approx. optimal noise settings; the last column indicates the number of instances for which essentially incomplete behaviour was observed, these were removed from the test-set.

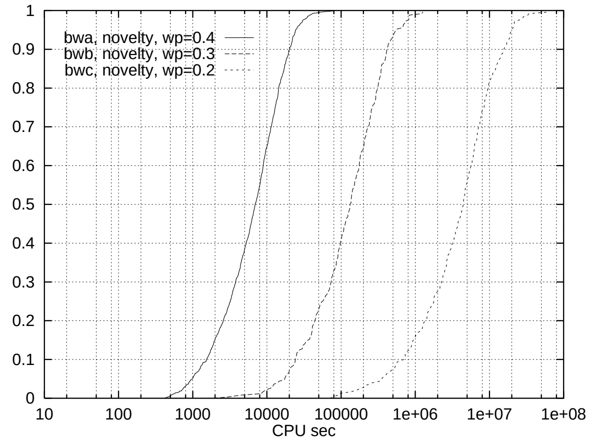


Figure 5.4: RLDs for Novelty with approx. optimal noise setting on SAT-encoded blocksworld planning problems.

| instance | median #steps $m$ | $\chi^2$ for $ed[m]$ | $\nu$ | passed ($\alpha$) |
|---|---|---|---|---|
| medium | 847 | 162.07 | 29 | no |
| huge | $14,905$ | 73.93 | 29 | no |
| bw_large.a | $6,839$ | 60.37 | 29 | no |
| bw_large.b | $119,680$ | 11.40 | 16 | yes (0.05) |
| bw_large.c | $4.27 \cdot 10^6$ | 8.71 | 16 | yes (0.05) |
| bw_large.d | $8.51 \cdot 10^6$ | 10.14 | 10 | yes (0.05) |

Table 5.5: RLD approximations using exponential distributions $ed[m]$ for Novelty (approx. optimal noise) applied to Blocks World Planning instances; last column indicates whether the approximation passed the $\chi^2$ test.

**Blocks World Planning** Fig. 5.4 shows the RLDs for Novelty when applied to SAT-encoded problem instances from the blocks-world planning domain. We used the optimal noise parameters from Section 4.5 and `maxSteps` settings high enough ($10^7$ and $10^8$, resp.) to ensure 100% success rate. For the smaller instances (up to `bw_large.a`) we used 1,000 runs, for `bw_large.b` and `bw_large.c` 250, and for `bw_large.d` 100 runs to approximate the actual RLDs as described before. The estimates for the parameter $m$ and the corresponding $\chi^2$ values for the approximation by exponential distributions are shown in Table 5.5. The critical $\chi^2$ values for a standard $\alpha = 0.05$ acceptance level are 42.6 for 1,000 tries ($\nu = 29$) and 26.3 for 250 tries ($\nu = 16$). This means, that only for the smaller instances the approximation does not pass the test. As for the easy Random-3-SAT instances, this can be explained by the initial hill-climb phase of local search.

Table 5.6 reports the results of approximating the RLDs of various SLS algorithms applied

| algorithm | median #steps $m$ | $\chi^2$ for $ed[m]$ | passed ($\alpha$) |
|---|---|---|---|
| wsat(0.35) | 119,680 | 11.40 | yes (0.05) |
| wsat+tabu(2) | 110,416 | 27.50 | yes (0.01) |
| novelty(0.3) | 133,453 | 14.47 | yes (0.05) |
| rnovelty(0.55) | 170,734 | 12.68 | yes (0.05) |

Table 5.6: RLD approximations using exponential distributions $ed[m]$ for various algorithms applied to Blocks World Planning instance `bw_large.b` (using approx. optimal noise); the last column indicates whether the approximation passed the $\chi^2$ test ($\nu = 16$).

| instance | algorithm | median #steps $m$ | $\chi^2$ for $ed[m]$ | $\nu$ | passed ($\alpha$) |
|---|---|---|---|---|---|
| ais6 | gwsat(0.5) | 1,898.1 | 33.34 | 29 | yes (0.05) |
| ais6 | wsat(0.5) | 857.78 | 28.31 | 29 | yes (0.05) |
| ais8 | gwsat(0.4) | 44,187.5 | 35.72 | 29 | yes (0.05) |
| ais8 | wsat(0.4) | 19,439.9 | 23.01 | 16 | yes (0.05) |
| ais10 | gwsat(0.4) | 343,096 | 25.51 | 16 | yes (0.05) |
| ais10 | wsat(0.2) | 118,237 | 19.29 | 16 | yes (0.05) |

Table 5.7: RLD approximations using exponential distributions $ed[m]$ for GWSAT and WalkSAT applied to All-Interval-Series instances (using approx. optimal noise); the last column indicates whether the approximation passed the $\chi^2$ test.

to instance `bw_large.b` with exponential distributions. The RLD data is based on 250 runs of each algorithm; the critical $\chi^2$ value for a standard $\alpha = 0.05$ acceptance level is 26.3. The approximations for all algorithms pass the test; only WalkSAT+tabu does not pass for the $\alpha = 0.05$ acceptance level, but for the $\alpha = 0.01$ acceptance level. The corresponding results of the larger Blocks World Planning instances are very similar. This confirms that our result concerning approximately exponential RLDs generalises to WalkSAT+tabu, Novelty, and R-Novelty also when applied to hard instances of the Blocks World Planning domain.

**All-Interval-Series** For the All-Interval-Series domain, we already know from Section 4.5 that only GWSAT and WalkSAT show approximately complete behaviour; all other algorithms suffer from stagnation behaviour. Therefore, we study only RLD characterisations for GWSAT and WalkSAT here. As can be seen from the approximation and test data reported in the Table 5.7, the exponential RLD approximations for both algorithms and all three instances passed the $\chi^2$ test for a standard acceptance level of $\alpha = 0.05$. Interestingly, for this problem domain, the effect of the initial search phase seems to be much less prominent than for the other problem classes studied here.

In summary, the results presented in this section show that some of the most powerful SLS

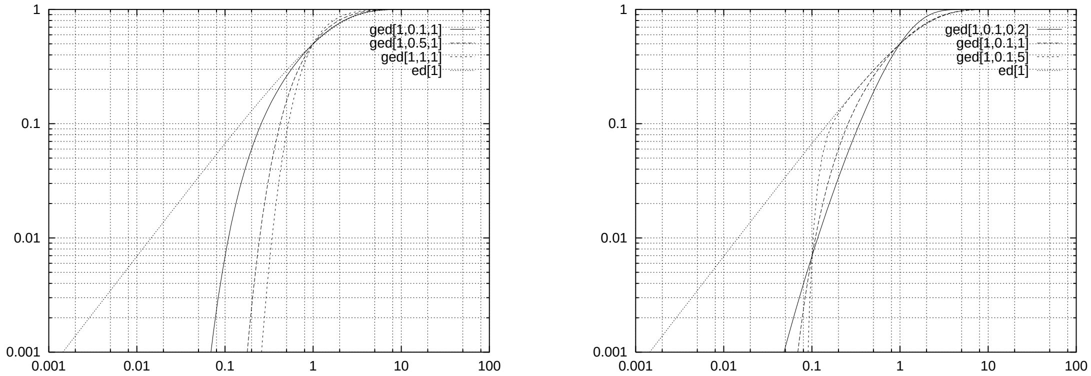


Figure 5.5: The new, generalised distribution class $ged[m, \gamma, \delta]$ for different $\gamma$ and $\delta$ values; note the difference to the standard exponential distribution.

algorithms for SAT show a very regular behaviour when applied to hard instances from a broad range of both problem domains: Using optimal noise parameter settings, the RLDs for these algorithms are approximately exponentially distributed. Furthermore, the quality of the corresponding functional approximations monotonically increases with the hardness of problem instances.

### 5.1.3 Modelling the Initial Search Phase

As pointed out before, the RLD approximations using exponential distributions show a systematic deviation at the left end of the RLD graphs which corresponds to the behaviour during the initial search phase. To study this phenomenon in more detail, we developed a new probability distribution which facilitates modelling the run-time behaviour during the initial search phase in great detail. This new distribution is based on the Weibull distribution[6] $wd[m, \beta](x) = 1 - 2^{-(x/m)^{\beta}}$, a well-known generalisation of the exponential distribution; the underlying intuition is that during the initial search phase, the performance of an SLS algorithm (*i.e.*, its probability of finding a solution within a fixed number of steps) is lower than later in the search process. This can be modeled by a Weibull distribution with a dynamically changing $\beta$ parameter (which controls the steepness of the cumulative distribution curve in a semi-log plot).

Our new model is defined by the following cumulative distribution function:[7]

$$ged[m, \gamma, \delta](x) = wd[m, 1 + (\gamma/x)^{\delta}](x) = 1 - 2^{-(x/m)^{1+(\gamma/x)^{\delta}}}$$

Although the full defining term looks quite complicated, the definition reflects exactly the idea of a Weibull distribution with dynamically changing $\beta$ parameter as given above.

[6] Weibull distributions are used in reliability theory to model failure rates in systems which are subject to aging phenomena [KGC77].

[7] We denote it *ged*, because it is a generalised form of the exponential distribution, as discussed later.

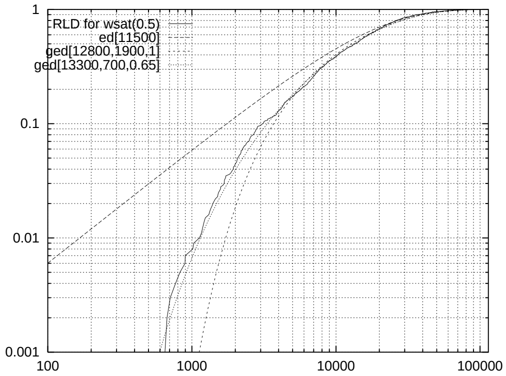


Figure 5.6: RLD for WalkSAT (approx. optimal noise) applied to Blocks World instance `bw_large.a` and three functional approximations.

| distribution | $\chi^2$ | $\nu$ | passed ($\alpha$) |
|---|---|---|---|
| $ed[11800]$ | 93.56 | 29 | no |
| $ged[12800, 1400, 1]$ | 66.70 | 28 | no |
| $ged[12800, 600, 0.6]$ | 37.53 | 27 | yes (0.05) |

Table 5.8: RLD approximations for WalkSAT (approx. optimal noise) applied to `bw_large.a`, using different distribution classes; parameters have been manually determined, the last column indicates whether the approximation passed the $\chi^2$ test. The $\nu$ values specify the degrees of freedom for the used distribution class; they are reported here for technical reasons only.

More precisely, the new distribution is obtained from a Weibull distribution by introducing a hyperbolically decaying $\beta$ parameter. Like for the exponential and Weibull distribution, $m$ in the formula above is the median of the distribution. The two remaining parameters intuitively correspond to the length of the initial search phase ($\gamma$) and to its impact on the overall behaviour ($\delta$). High $\gamma$ values indicate a long initial search phase, while high $\delta$ values are used to model a strong influence of the initial search phase. Figure 5.5 shows the new generalised distribution for various parameter values.

Note that the exponential distribution is a special case of this new class of distributions, since $ged[m, 0, \delta] = ed[m]$, while generally, Weibull distributions cannot be represented within this family. As can be easily seen from the definition, $ged[m, \gamma, \delta](x)$ asymptotically approaches an exponential distribution $ed[m](x)$ for large $x$, regardless of the values for $\gamma$ and $\delta$. For small and decreasing $x$, however, the relative difference between $ged[m, \gamma, \delta](x)$ and $ed[m](x)$ increases monotonically, which is qualitatively exactly what we observed for the empirical RLDs of SLS algorithms.

| problem instance | $\chi^2$ for $ed[m]$ | $\chi^2$ for $ged[m,\gamma,1]$ | $\chi^2$ for $ged[m,\gamma,\delta]$ | $\nu$ |
|---|---|---|---|---|
| medium | (198.47) | 40.68 | 33.57 | 29–27 |
| huge | (69.58) | 31.96 | 20.74 | 29–27 |
| bw_large.a | (74.21) | 35.57 | 25.31 | 29–27 |
| bw_large.b | 27.50 | 16.51 | 16.46 | 16–14 |
| bw_large.c | 21.43 | 21.43 | 21.43 | 16–14 |

Table 5.9: Blocks World Planning instances, functional approximation of RLDs for WalkSAT+tabu (approx. optimal noise) with different distribution classes; $\chi^2$ values in parentheses indicate that the approximation did not pass the test for $\alpha = 0.01$.

Because of computation time limitations and for brevity's sake we exemplify the RLD analysis based on the new distribution only for a selection of algorithms and benchmark problems. Figure 5.6 shows the RLD of WalkSAT (approx. optimal noise = 0.5) when applied to blocks world instance `bw_large.a` and three successively more precise functional approximations (these approximations have been manually determined). While the approximation using an exponential distribution fits very well for the upper part of the empirical RLD, it does not adequately model the algorithm's behaviour for shorter runs and consequently does not pass the $\chi^2$ test (*cf.* Table 5.8). The new, generalised distribution gives a significantly better approximation even when fixing the $\delta$ parameter to one. While this approximation is often sufficient for characterising the initial search phase, in this example, it still does not pass the $\chi^2$ test. However, using the $\delta$ parameter as an additional degree of freedom, we can model the given RLD so precisely that the corresponding approximation passes the $\chi^2$ test.

As shown in Table 5.9, using *ged* approximations we can characterise the behaviour of WalkSAT+tabu (approx. optimal noise) when applied to the Blocks World instances such that all approximations pass the $\chi^2$ test. Interestingly, we can achieve this even when fixing $\delta$ to one. Analogous results can be shown for the other SLS algorithms. The advantages of the *ged* approximation over *ed* approximations are reciprocal to the size of the instances. This is consistent with our earlier observation that for smaller and easier instances the initial search phase is more prominent. On the other hand, for large and/or very difficult problem instances, like `bw_large.c`, by using *ged* approximations no improvement of the approximation can be achieved. In particular, this can be observed for the All-Interval-Series instances; here, *e.g.*, when characterising WalkSAT's behaviour using using exponential approximations, these generally pass the $\chi^2$ test for all, even the smallest, instances. While using *ged* approximations, minor improvements can be achieved for the smallest instance, for the other instances these do not realise any advantage over standard exponential approximations.

Next, we used *ged* approximations of SLS behaviour for Random-3-SAT and Graph Colouring test-sets. Table 5.10 shows the results of approximating RLDs of R-Novelty applied to the `uf100-430` test-sets with different distribution classes. While using *ed* approximations, 12.9% of the instances passed the $\chi^2$ test for the $\alpha = 0.01$ acceptance level; by using *ged*'s

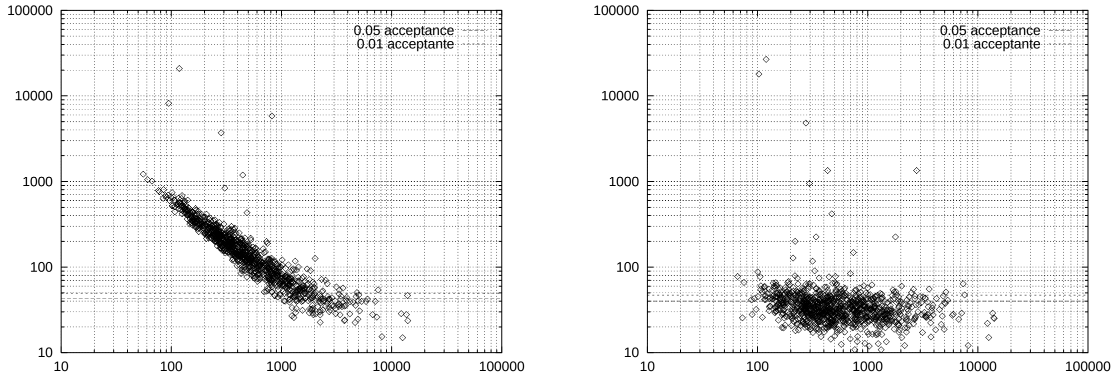


Figure 5.7: Correlation between hardness of problems (horizontal) and $\chi^2$ values (vertical) from testing the RLDs of individual instances versus a best-fit *ed (left)* and *ged* approximation *(right)* for R-Novelty (approx. optimal noise = 0.7) applied to Random-3-SAT test-set `uf100-430`. The dashed horizontal lines indicate the acceptance thresholds for the 0.01 and 0.05 acceptance level.

| distribution class | acceptance level $\alpha$ | fraction passed |
|---|---|---|
| $ed[m]$ | 0.01 | 12.9% |
| $ed[m]$ | 0.05 | 8.7% |
| $ged[m,\gamma,1]$ | 0.01 | 66.2% |
| $ged[m,\gamma,1]$ | 0.05 | 52.9% |
| $ged[m,\gamma,\delta]$ | 0.01 | 85.0% |
| $ged[m,\gamma,\delta]$ | 0.05 | 70.0% |

Table 5.10: Random-3-SAT, test-set `uf100-430`, functional approximation of RLDs for R-Novelty (approx. optimal noise = 0.7) with different distribution classes.

| distribution class | acceptance level $\alpha$ | fraction passed |
|---|---|---|
| $ed[m]$ | 0.01 | 2.5% |
| $ed[m]$ | 0.05 | 1.5% |
| $ged[m, \gamma, 1]$ | 0.01 | 63.4% |
| $ged[m, \gamma, 1]$ | 0.05 | 50.5% |
| $ged[m, \gamma, \delta]$ | 0.01 | 78.0% |
| $ged[m, \gamma, \delta]$ | 0.05 | 67.6% |

Table 5.11: Graph Colouring, test-set `flat50-115`, functional approximation of RLDs for Novelty (approx. optimal noise = 0.6) with different distribution classes; essentially incomplete behaviour was observed for 9 of 1,000 instances, these were removed from the test-set.

with $\delta = 1$, the acceptance rate could be increased to 66.2%; approximating the RLDs with unrestricted *ged*'s, 85.0% of the instances passed the test. At the same time, as can be seen in Figure 5.7, by using the new, generalised distribution, the negative correlation between the hardness of instances and the goodness of functional RLD approximations which is very prominent for exponential approximations is no longer observed. This indicates that, indeed, *ged* approximations are an adequate model for SLS behaviour over the whole range of instance hardness.

Applying the same correlation analysis to Novelty and Graph Colouring test-set `flat50-115` yields analogous results (*cf.* Table 5.11). Note that since these problem instances are very easy for Novelty, the effect of the initial search phase on the RLDs is very strong; consequently, the acceptance rates of the $\chi^2$ test, when applied to *ed* approximations, are very low. Here, by using *ged* approximations which adequately model the initial search phase an even more drastic improvement can be achieved: while for *ed* approximations, only 2.5% of the instances passed the $\chi^2$ test for the $\alpha = 0.01$ acceptance level, this acceptance rate is increased to 78.0% when using *ged*'s.

Nevertheless, one might ask why for a small fraction of problem instances even *ged* approximations are not good enough to pass the $\chi^2$ test. Figure 5.8 shows the RLD corresponding to the most extreme outlier from Fig. 5.7 with a $\chi^2$ value of 23,753.20 for the best *ged* approximation. This RLD is multi-modal and shows a premature stagnation behaviour of the algorithm between 200 and 1,000 steps; then, however, the perfomance improves again and the RLD converges to one. This example seems to be typical for the instances in which the given RLD cannot be adequately characterised by *ged* approximations. The intuitive interpretation of this behaviour is that initially the algorithm behaves in a regular way (which can be characterised by a *ged* approximation), but then gets stuck in some sort of "trap". It will, however, eventually escape from this trap and then continue its search in a regular way. These traps probably correspond to attractive non-solution areas of the search space (*e.g.*, complex local minima areas) which are hard to escape for the algorithm. As also shown in Fig. 5.8, a combination of two *ged* functions reflecting this interpretation

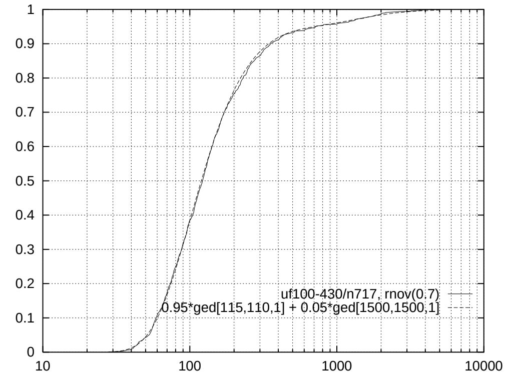


Figure 5.8: RLD for R-Novelty (approx. optimal noise = 0.7) applied to instance `uf100-430/n717`; note, how using a combination of two overlayed *ged* approximations, the observed behaviour can be almost perfectly modeled.

| algorithm | median of RLD | $m$ | $\chi^2$ for $ed[m]$ | passed ($\alpha$) |
|---|---|---|---|---|
| wsat(0.5) | 13,505 | 13,094.9 | 98.41 | no |
| wsat(0.6) | 15,339 | 16,605.5 | 76.17 | no |
| wsat(0.7) | 27,488 | 27,485.9 | 42.13 | yes (0.5) |
| wsat(0.8) | 72,571 | 74,289.8 | 20.78 | yes (0.5) |

Table 5.12: $ed[m]$ approximations of RLDs for WalkSAT applied to Blocks World Planning instance `bw_large.a`, using optimal and larger-than-optimal noise parameter settings.

gives an almost perfect approximation of the observed RLD. This strongly suggests that simple combinations of *ged* approximations can be used to adequately model and explain the irregular behaviour observed for a small fraction of problem instances from our test-sets.

Results analogous to the ones presented here could be obtained for other WalkSAT algorithms when applied to various test-sets and instances from our benchmark suite. In summary, our analysis shows that using the new probability distribution type *ged* developed here, we can model the run-time behaviour of state-of-the-art SLS algorithm for SAT in great detail. In particular, this extended model captures the SLS behaviour during the initial search phase very accurately.

### 5.1.4 Functional Approximations for Non-optimal Noise

While up to this point, we always concentrated on approximately optimal noise parameter settings, in the following, we present some results regarding SLS behaviour for non-optimal noise. For larger-than-optimal noise settings, we observe essentially the same EPAC be-

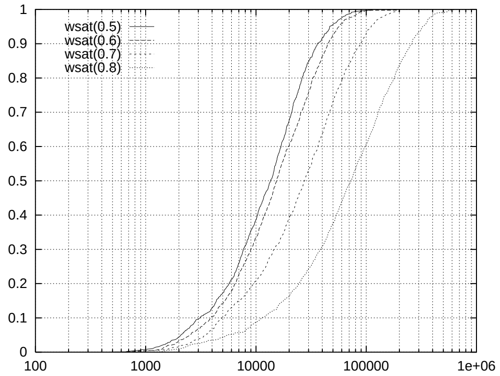


Figure 5.9: RLDs for WalkSAT applied to Blocks World Planning instance `bw_large.a`, using different noise parameter settings (0.5 is approx. optimal).

haviour as for optimal noise, but run-times compared to optimal noise are uniformly higher for all success probabilities. This can be easily seen in the semi-log plots of the corresponding RLDs, where for larger-than-optimal noise, the RLD curves have the same shape while being shifted to the right. At the same time, when increasing the noise, the effects of the initial search phase become less prominent — consequently, the approximations using exponential distributions are usually better for higher noise. This can be seen in Figure 5.9, showing the RLDs for WalkSAT with different noise settings when applied to Blocks World Planning instance `bw_large.a` (RLD data is based on 1,000 runs). The results for a $\chi^2$ test, using *ed* approximations, are reported in Table 5.12. For optimal noise, the *ed* approximation does not pass the test, because the deviation caused by the initial search phase is too strong. While this effect decreases for a slightly increased noise setting (as can be seen from the lower $\chi^2$ value), the *ed* approximation still does not pass the test. When further increasing the noise setting, the initial search phase becomes less prominent and the *ed* approximation passes the $\chi^2$ test. It should be noted, however, that when using *ged* approximations (*cf.* Section 5.1.3), the initial search phase can be correctly modeled in all three cases in such a way that the corresponding RLD approximations pass the $\chi^2$ test. Note also the increasing median run-times, indicating the decrease of performance for larger-than-optimal noise settings.

As a second example, we report the results of analysing the effects of increasing the noise parameter for R-Novelty, applied to the Graph Colouring test-set `flat50-115`. Table 5.13 shows the fraction of instances from this test-set, for which a best-fit *ed* approximation passed the $\chi^2$ test. As can be seen from this data, the fraction of the hardest 25% of the test-set which passed the test increases for larger-than-optimal noise, as the effects of the initial search phase become less prominent. At the same time the median hardness (mean number of steps / solution) increases considerably. Analogous results could be obtained for different SLS algorithms and other problem domains from our benchmark suite.

While for larger-than-optimal noise, the SLS performance decreases but the algorithms still

| algorithm | median hardness | acceptance level $\alpha$ | fraction passed |
|---|---|---|---|
| r-novelty(0.6) | 739.61 | 0.01 | 19.0% |
| | | 0.05 | 13.3% |
| r-novelty(0.7) | 832.92 | 0.01 | 23.8% |
| | | 0.05 | 14.9% |
| r-novelty(0.8) | 1,043.29 | 0.01 | 40.7% |
| | | 0.05 | 24.2% |

Table 5.13: Fraction of instances from the hardest 25% of the `flat50-115` Graph Colouring test-set passing the $\chi^2$ test for Novelty with optimal and larger-than-optimal noise settings; instances, for which essentially incomplete behaviour was observed, were removed from the test-set.

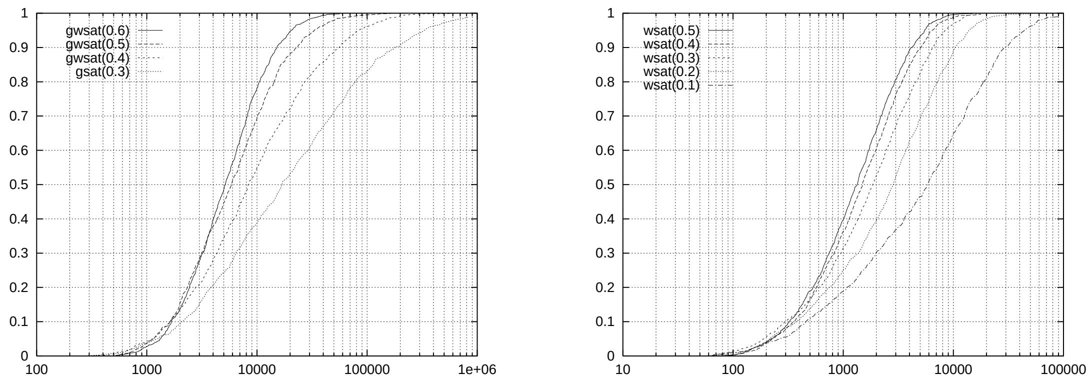


Figure 5.10: *Left:* RLDs for GWSAT applied to easy instance from Graph Colouring test-set `flat100-239`, using different noise parameter settings (0.6 is approx. optimal); *right:* same for WalkSAT applied to medium instance from Random-3-SAT test-set `uf100-430` (approx. optimal noise = 0.5).

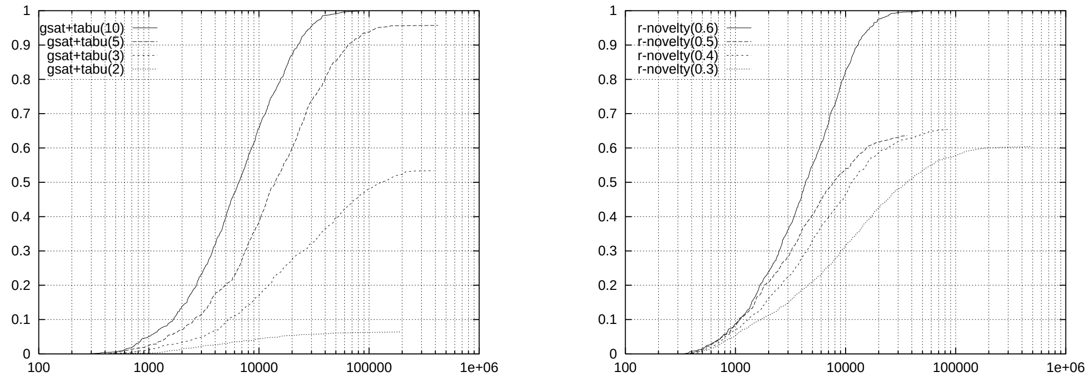


Figure 5.11: *Left:* RLDs for GSAT+tabu applied to medium instance from Graph Colouring test-set `flat100-239`, using different noise parameter settings (10 is approx. optimal); *right:* same for R-Novelty applied to Blocks World Planning instance `bw_large.a` (approx. optimal noise = 0.6).

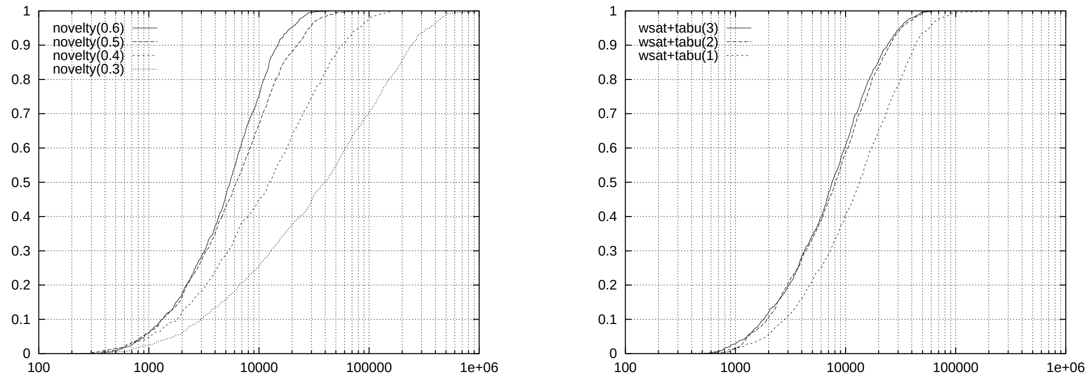


Figure 5.12: *Left:* RLDs for Novelty applied to medium instance from Graph Colouring test-set `flat100-239`, using different noise parameter settings (0.6 is approx. optimal); *right:* same for WalkSAT+tabu applied to Blocks World Planning instance `bw_large.a` (approx. optimal noise = 3).

| algorithm | mean | stddev | $\frac{\text{stddev}}{\text{mean}}$ | median | $Q_{25}$ | $Q_{75}$ | $Q_{10}$ | $Q_{90}$ |
|---|---|---|---|---|---|---|---|---|
| gwsat(0.2) | 595,388.77 | 580,061.58 | 0.97 | 413,064 | 168,375 | 797,305 | 66,640 | 1,431,559 |
| gwsat(0.3) | 69,778.76 | 73,354.57 | 1.05 | 47,511 | 20,531 | 96,839 | 7,134 | 158,442 |
| gwsat(0.4) | 64,167.11 | 64,661.00 | 1.01 | 43,128 | 19,079 | 85,901 | 7,150 | 143,875 |
| gwsat(0.5) | 79,258.66 | 82,593.77 | 1.04 | 50,887 | 22,172 | 109,799 | 8,387 | 181,464 |
| gwsat(0.6) | 169,139.57 | 170,347.50 | 1.01 | 115,024 | 49,558 | 229,213 | 17,247 | 389,196 |

Table 5.14: $ed[m]$ approximations of RLDs for GWSAT applied to All-Interval-Series instance `ais8`, using approx. optimal (=0.4) and non-optimal noise parameter settings.

show EPAC behaviour, for lower-than-optimal noise parameter settings we find a completely different situation. While for all algorithms, the performance decreases non-uniformly for different solution probabilities, some of them remain approximately complete, while others show essentially incomplete behaviour. We demonstrate this here by giving a small number of examples, which are nevertheless typical for our overall observations. As shown in Figure 5.10, GSAT and WalkSAT show a similar behaviour for smaller-than-optimal noise: While for short runs, the performance even improves, for longer runs and the corresponding higher success probabilities, the performance deteriorates considerably. Consequently, the RLDs are less steep than exponential distributions and suffer from an increasingly long and heavy tail. Both GSAT+tabu and R-Novelty show a similar, but additionally essentially incomplete behaviour; as can be seen from Figure 5.11, the maximal success probabilities for both algorithms are decreasing with the noise parameter. For Novelty, no increasingly essential incomplete behaviour is observed, but the performance also decays non-uniformly as for GSAT and WalkSAT (*cf.* Figure 5.12). In comparison, WalkSAT+tabu seems to be relatively mildly affected; the main reason for this might be that typically, both the clause-length and the optimal tabu-list-length are rather small.

Summarising these results, typically the detrimental effects of non-optimal noise parameter settings are much worse for lower-than-optimal noise settings than for larger-than-optimal noise. This is particularly the case for the higher percentiles of the corresponding RLDs (*cf.* Table 5.14); additionally, for essentially incomplete SLS algorithms, stagnation behaviour occurs more frequently with decreasing noise settings. However, the behaviour for very short runs is usually not affected by lower-than-optimal noise and sometimes, even performance improvements can be observed for the early phases of local search (this is consistent with the intuition that greedier local search behaviour should pay off during the initial search phase, where gradient descent dominates the search.)

### 5.1.5 Consequences and Interpretations

The empirical results presented in the previous sections have a number of theoretically and practically interesting consequences.

**Random restart** For all of the SLS algorithms studied here, the standard implementations include a restart mechanism which allows to restart the local search process after a fixed maximal number maxSteps of steps, if at that point no solution has been found. Restarting the local search is the most simple method for escaping from local minima, and for essentially incomplete algorithms like plain GSAT, using restart leads to significantly improved performance. For EPAC algorithms, however, restart is not effective, since in the case of exponential run-time distributions the probability of finding a solution within a fixed time interval is independent of the run-time spent before. These SLS algorithms are essentially memoryless, as for a given total time $t$, restarting at time $t' < t$ does not significantly influence the probability of finding a solution in time $t$.[8] Based on the results reported before, many modern SLS algorithms for SAT show EPAC behaviour for optimal and larger-than optimal noise settings when applied to our benchmark problems. Consequently, for these algorithms random restart is mostly ineffective. However, due to the effects of the initial search phase, this does not hold for relatively easy problem instances and short run-times. However, in practice the ineffectivity of random restart for hard instances can be easily verified over a broad range of maxSteps settings (this will be exemplified in Section 5.2.2 in a slightly different context). As we have seen in Section 5.1.4, the EPAC property is usually lost when using lower-than-optimal noise parameter; in this case, random restart is advantageous for appropriately chosen cutoff times (*cf.* Chapter 2).

**Search space reduction** One surprising consequence of our characterisation results is based on the well-known fact that blind guessing (*i.e.*, repeatedly testing a randomly selected variable assignment), one of the most naïve search methods, is also characterised by exponential RLDs. Therefore, our results suggest that the behaviour of modern, powerful SLS algorithms for optimal and larger-than-optimal noise settings can be interpreted as blind guessing in a significantly reduced search space. More precisely, these SLS algorithms, when applied to a problem instance with search space $S$ and search space size $|S|$, behave exactly like blind guessing in a "virtual search space" $S'$. Assuming that $S'$ has the same (known) number of solutions as $S$, the size $|S'|$ of the virtual search space can be easily calculated from the empirical RLD of the given SLS algorithm $A$ and the original search space size $|S|$. For a given problem instance, $|S'|$ gives an indication for the effectiveness of algorithm $A$: the smaller $|S'|$, the more effective is $A$. This can be easily generalised to test-sets of problems or problem distributions by combining the $|S'|$ values for the individual instances. Note that this interpretation holds for all situations where an SLS algorithm applied to a given problem instance or problem class shows EPAC behaviour, *i.e.*, it exhibits exponential RLDs.

This result immediately raises the following question: How is the virtual search space related to the actual search space? For now, we have no conclusive answer to this question. But ideally, we would like to be able to identify structural features of the original search space which can be shown to be tightly correlated with the virtual search space size. One would

[8] Because the exponential distribution is memoryless, it is often used in reliability theory to describe components that are not subject to aging phenomena like transistors [BP81].

assume that size and number of plateaus (including those consisting of solutions) in the original search space should be important factors. However, preliminary studies have shown that these features alone are not sufficient for explaining the observed search space reduction phenomenon.

## 5.2 Improving Existing SLS Algorithms

After characterising the behaviour of several state-of-the-art SLS algorithms for SAT, in this section we show how based on these results the algorithms can be further improved. In the following, we discuss three ways of improving SLS algorithms based on the empirical analyses given before: using hybrid GLSM extensions, multiple independent tries, and algorithm portfolios. While the first approach is used to improve the behaviour of specific SLS algorithms, the other two approaches are very general and can be exploited for parallel as well as sequential time-sliced processing.

### 5.2.1 Hybrid GLSM Extensions

As discussed in Section 5.1.1, some of the most powerful SLS algorithms for SAT, like WalkSAT+tabu, Novelty, and R-Novelty suffer from essential incompleteness. In practice, this can be easily overcome by introducing random restarts. Obviously, if for a given cutoff time $t$ the probability of finding a solution for a given problem instance is $p > 0$, the probability of finding a solution within $n$ independent runs is $p' = 1-(1-p)^n$ which converges to 1 for $n \rightarrow \infty$. Therefore, by introducing random restart after a fixed cutoff time, any Las Vegas algorithm can be made approximately complete. Moreover, the modified algorithm will have an approximately exponential run-time distribution for large numbers of restarts.

In the context of the GLSM model, random restart is realised by connecting at least one local search state with the initialisation state by a deterministic transition with condition mcount(maxSteps). This is realised in the implementations of all the SLS algorithms for SAT discussed in the context of this work (*cf.* Sections 4.2 and 4.3). In this sense, all the actual implementations correspond to hybrid GLSM extensions obtained from the pure local search strategy by adding restart.

Unfortunately, in general it is extremely difficult to find good cutoff times *a priori* (*a posteriori* they can be easily derived from the RLD data, as detailed before). Today, to our best knowledge, there exist no theoretical results on how to determine good cutoffs. The only empirical results we are aware of, are for GSAT when applied to hard Random-3-SAT problem distributions [GW93a]; but since these results are based on potentially problematic methodology, their significance is unclear (*cf.* Section 2.3). While using inappropriately chosen cutoffs generally still eliminates essential incompleteness, in practice, it leads to extremely poor performance.

Another way of overcoming essential incompleteness of a SLS algorithm is to extend it with a random walk state. This extension can again be interpreted in the GLSM framework by adding a random walk state and connecting it to the local search state(s) via unconditional probabilistic transitions. As an example, consider GSAT and GWSAT; this also demonstrates that adding random walk induces approximate completeness, as proven for GWSAT in Section 5.1.1. As mentioned there, the proof for GWSAT's approximate completeness easily generalises for all algorithms which guarantee that arbitrarily long sequences of random walk steps can be performed with a probability $p > 0$ which is independent of the number of steps that have been performed in the past. Thus, any GLSM $G$ satisfying the following conditions will provably correspond to a PAC algorithm:

1. $G$ has a random walk state $RW$;
2. $RW$ has a probabilistic, unconditional self-transition;
3. for any point in time, there is a positive, bounded probability for $G$ being in state $RW$, *i.e.*, $\forall t : \exists p' > 0 : P(G \text{ is in state } RW \text{ at time } t) \geq p'$.

As a convention, we assume that the random walk state has a bounded, positive probability of decreasing the hamming distance to the nearest solution in each step, as is the case for the random walk state used by GWSAT (*cf.* Chapter 4, Section 4.2 for formal definition). While these conditions are sufficient for guaranteeing approximate completeness, they can be further weakened. But because of the simplicity of the algorithms we are dealing with here, the conditions as given above are fully adequate for our purposes. More specifically, since all GSAT and WalkSAT variants discussed here are 1-state+restart GLSMs, we extend them with random walk using the generic scheme shown in Figure 5.13, where $LS$ is any local search state and $RW$ the random walk state as defined for GWSAT. It can be easily verified that this scheme satisfies the conditions for approximate completeness as given above; therefore, any instantiation will show approximately complete behaviour.

Random walk apparently has an advantage over random restart, since at least in GWSAT, it is more robust w.r.t. the additionally introduced parameter wp than random restart is w.r.t. maxSteps (the cutoff time). One reason for this empirically observed phenomenon is related to the inherent randomness of the random walk sequences; random restarts occur after a fixed cutoff time, whereas random walk sequences are probabilistically variable in their length and frequency of occurrence. Furthermore, when using random restart, the search process is re-initialised; consequently, a new try cannot benefit from the search effort spent in previous tries (unless information is carried from one run to the next, which is not the case for the algorithms considered here). The amount of perturbation introduced by a random walk sequence, however, probabilistically depends on the length of the sequence such that small pertubations are much more likely to occur.

We use the random walk extension here to improve the behaviour of Novelty and R-Novelty. For R-Novelty, we replace the deterministic loop breaking strategy, which is essentially equivalent to performing one random walk step each 100 steps, by a random walk extension as described above; for Novelty, we newly introduce an equivalent extension. We call

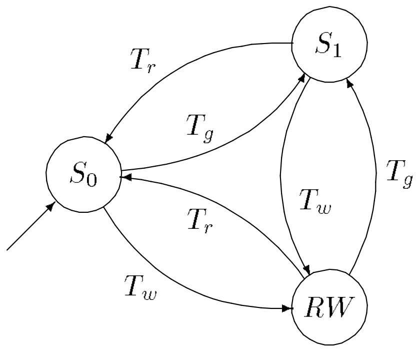


Figure 5.13: Random walk extension of a 1-state+restart GLSM; transition types: $T_r \equiv CPROB(C, 1)$ with condition $C \equiv \mathsf{mcount}(\mathsf{maxSteps}); T_w \equiv CPROB(\neg C, \mathsf{wp}); T_g \equiv CPROB(\neg C, 1 - \mathsf{wp})$.

these modified algorithms R-Novelty$^+$ and Novelty$^+$; their approximate completeness (PAC property) follows immediately using the argument given above.

Although these modified algorithms have theoretically considerable advantages, *a priori*, it is not clear how their performance compares with the original versions in practice. Table 5.15 shows the basic descriptive statistics for the mean local search cost (hardness distribution) across two Random-3-SAT and Graph Colouring test-sets (the data is based on 100 runs / instance, using $\mathsf{maxSteps} = 10^6$.). As can be seen when comparing the statistics for the original and modified versions of Novelty and R-Novelty, the latter show dramatically reduced means and standard deviations. The percentiles, however, are not too different, although for the Random-3-SAT test-set, the higher percentiles are also significantly lower. This indicates that the essentially incomplete behaviour which causes the high means and standard deviations of these hardness distributions is efficiently eliminated by the random walk extension, while for instances which do not suffer from essentially incomplete behaviour, the performance remains mainly unaffected. This interpretation is confirmed by the RLD statistics for Novelty and R-Novelty and their respective modified versions when applied to Blocks World Planning and All-Interval-Series instances (*cf.* Tables 5.16 and 5.17, all RLD data is based on 250 or more runs of each algorithm). Note that, as discussed earlier, even when using random restart, essentially incomplete behaviour makes an algorithm's performance extremely dependent on appropriately chosen cutoff parameters. This effect is especially drastic for extremely low asymptotic success probabilities, as observed for Novelty and R-Novelty when applied to large and hard Blocks World Planning and All-Interval-Series instances.

In summary, these results indicate that R-Novelty$^+$ and Novelty$^+$ show significantly improved performance over the original algorithms where these suffered from essentially incomplete behaviour, while in the remaining cases the performance is approximately identical. Thus, extending Novelty and R-Novelty with random walk generally improves their

| test-set | algorithm | mean | stddev | $\frac{\text{stddev}}{\text{mean}}$ | median | $Q_{25}$ | $Q_{75}$ | $Q_{10}$ | $Q_{90}$ |
|---|---|---|---|---|---|---|---|---|---|
| uf100-430 | novelty(0.6) | 28,257.51 | 191,668.71 | 6.78 | 851.87 | 479.14 | 1,845.30 | 302.88 | 4,390.39 |
| | novelty+(0.6) | 1,570.39 | 2,802.01 | 1.78 | 801.75 | 467.04 | 1,663.20 | 288.72 | 3,049.70 |
| flat50-115 | r-novelty(0.6) | 7,109.64 | 97,543.64 | 13.72 | 739.61 | 490.34 | 1,282.53 | 356.66 | 2,142.78 |
| | r-novelty+(0.6) | 1,084.01 | 1,053.71 | 0.97 | 747.37 | 486.65 | 1,245.42 | 358.74 | 2,167.39 |

Table 5.15: Performance comparison for Novelty and R-Novelty (approx. optimal noise) *vs* Novelty$^+$ and R-Novelty$^+$ for Random-3-SAT and Graph Colouring test-sets. The reported basic descriptive statistics refer to the hardness distributions (mean number of steps / solution) over the test-sets.

| instance | algorithm | $\widehat{p_s}^*$ | maxSteps | $\widehat{E_s}$ | mean$_c$ | stddev$_c$ | median$_c$ |
|---|---|---|---|---|---|---|---|
| bw_large.c | r-novelty(0.3) | 0.028 | $10^8$ | $3.47 \cdot 10^9$ | 169,810.86 | 157,752.57 | 32,334 |
| | r-novelty+(0.3) | 1.0 | $10^8$ | $8.09 \cdot 10^6$ | 8,086,468.24 | 8,414,928.96 | 5,292,830 |
| ais6 | novelty(0.7) | 0.101 | $10^6$ | $8.90 \cdot 10^6$ | 62.65 | 44.60 | 46 |
| | novelty+(0.7) | 1.0 | $10^6$ | $9.94 \cdot 10^3$ | 9,944.67 | 10,828.49 | 6,909 |
| ais8 | r-rnovelty(0.8) | 0.916 | $10^7$ | $1.06 \cdot 10^6$ | 139,779.01 | 135,746.86 | 99,880 |
| | r-novelty+(0.8) | 1.0 | $10^7$ | $1.51 \cdot 10^5$ | 151,051.11 | 152,669.01 | 112,413 |
| ais10 | r-rnovelty(0.7) | 0.012 | $10^8$ | $8.23 \cdot 10^9$ | 7,603.33 | 1,667.32 | 7,313 |
| | r-novelty+(0.7) | 1.0 | $10^8$ | $2.41 \cdot 10^6$ | 2,410,863 | 2,240,534 | 1,697,646 |

Table 5.16: Performance comparison for Novelty and R-Novelty (approx. optimal noise) *vs* Novelty$^+$ and R-Novelty$^+$ for Blocks World Planning and All-Interval-Series instances; $\widehat{p_s}^*$ indicates the asymptotic maximal success probability, $\widehat{E_s}$ denotes the expected number of steps for finding a solution (using random restart); the reported basic descriptive statistics refer to the corresponding *conditional* RLDs.

| instance | algorithm | mean | stddev | median | $Q_{25}$ | $Q_{75}$ | $Q_{10}$ | $Q_{90}$ |
|---|---|---|---|---|---|---|---|---|
| bw_large.a | novelty(0.4) | 9,315.07 | 8,587.17 | 6,932 | 3,202 | 12,877 | 1,534 | 20,202 |
| | novelty+(0.4) | 9,598.46 | 8,852.51 | 6,730 | 3,253 | 13,198 | 1,611 | 20,522 |
| bw_large.b | r-novelty(0.55) | 234,305.18 | 215,100.55 | 166,922 | 73,673 | 341,827 | 21,524 | 538,266 |
| | r-novelty+(0.55) | 223,493.79 | 251,273.29 | 141,353 | 53,223 | 315,074 | 22,499 | 496,085 |

Table 5.17: Performance comparison for Novelty and R-Novelty (approx. optimal noise) *vs* Novelty$^+$ and R-Novelty$^+$ for Blocks World Planning instances when no essentially incomplete behaviour is observed.

performance. However, it should be noted that these algorithms, as well as WalkSAT+tabu, can be also improved in a different, more specific way. Re-examining the proof for Novelty's essential incompleteness (*cf.* Section 5.1.1), one might have noted that the looping behaviour for the given example is caused by the fact that Novelty always picks between the best and second-best variable. What is needed to break the loop in the example given in the proof, is a mechanism which allows $y$ to be flipped when $c_3$ is selected. But in this situation, $y$'s score will always be lower than $x_1$'s and $x_2$'s. Therefore, to generally prevent this situation, a picking mechanism is needed which allows variables with arbitrarily bad scores to be picked once in a while. Similar considerations hold for R-Novelty. Thus, by modifying the picking mechanism according to this requirement, the essential incompleteness of both Novelty and R-Novelty can be removed. Note that adding a random walk state is equivalent to one specific modification of this kind. Other possibilities include weighting the probabilistic choice of the variable to be flipped by the score. However, it is not clear whether such variants show a best-case performance comparable to the original algorithms'.

For WalkSAT+tabu, the main reason for essential incompleteness when using long tabu-lists is the fact that, if all variables in a clause are tabu, none of them is flipped. This problem can be easily overcome by allowing a uniform or score-weighted random variable selection in this situation. Note, however, that — as opposed to the random walk extension discussed above — in general this will be insufficient to guarantee approximate completeness.

### 5.2.2 Parallelisation Based on Multiple Independent Tries

Randomised algorithms lend themselves to a straightforward parallelisation approach by performing independent runs of the same algorithm in parallel. In the given context, this is equivalent to using a homogenous cooperative GLSM model without communication as discussed in Chapter 3.1, Section 3.8. If the base algorithm is EPAC, this approach is particularly effective: Based on a well-known result from the statistical literature [Roh76], if for a given algorithm the probability of finding a solution in $t$ time units is exponentially distributed with parameter $m$, *i.e.*, the RLD can be characterised by $ed[m]$, then the probability of finding a solution in $p$ independent runs of time $t$ is distributed along $ed[m/p]$. Consequently, if we run such an algorithm once for time $t$, we get exactly the same success probability as when running the algorithm $p$ times for time $t/p$. This means that using multiple independent runs of an EPAC algorithm, an optimal speedup can be achieved for arbitrary numbers of processors.

In practice, however, SLS algorithms are not perfectly EPAC due to the systematic deviations during the initial search phase (*cf.* Section 5.1.3). Furthermore, usually a setup time (for initialising data structures *etc.*) has to be taken into account. Therefore, for very short runs (*i.e.*, very high numbers of processors), the speedup will generally be less than optimal. More precisely, for a given lower bound $s'$ on the speedup, a minimal cutoff time $t'$ and, subsequently, a maximal number of processors $p'$ can be derived from the RLD data such that for $p < p'$, a speedup $s \geq s'$ will be realised. However, if the RLDs converge against exponential distributions and the initial search phase and setup time are rather short compared

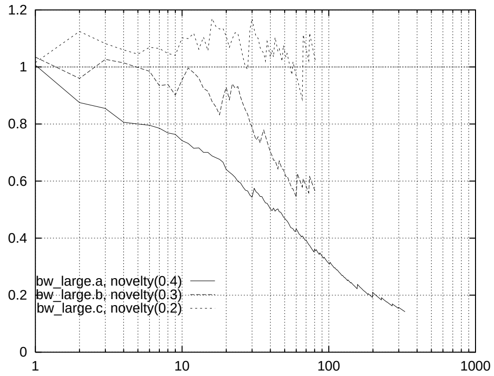


Figure 5.14: Efficiency of multiple independent try parallelisation (vertical) *vs* number of processors (horizontal) for Novelty (approx. optimal noise settings) applied to Blocks World Planning instances. An efficiency value of 1 indicates optimal speedup.

to the time required for solving typical problem instances, this effect becomes almost negligible. Based on the characterisations from the previous sections, this situation is given for most modern SLS algorithm for SAT when applied to hard problem instances from various domains. Furthermore, it holds for optimal as well as larger-than-optimal noise parameter settings (see also [HS98a]). Thus, parallelisation using independent tries will generally yield almost optimal speedup.

To illustrate this for a concrete example, we analysed the parallelisation efficiency for Novelty applied to several Blocks World instances based on the given RLDs (*cf.* Figure 5.4, page 121). Figure 5.14 shows the predicted parallelisation efficiency $E$ as a function of the number of processors $p$; $E(p)$ is defined as the speedup $S = T_1/T_p$ divided by $p$, where $T_1$ is the sequential and $T_p$ the parallel computation time, using $p$ processors. For Las Vegas algorithms like Novelty, it is reasonable to define the computation times such that $T_1$ is the cutoff time required to guarantee a solution probability of at least $p_s$, or technically: $T_1 = min(\{t' \mid P(RT \leq t') \geq p_s\})$. Given $T_1$ and the empirical RTD or RLD data, we can easily calculate $T_p$:

$$T_p = min(\{t' \mid \widehat{P}(RT \leq t') \geq 1 - (1 - p_s)^{1/p}\})$$

In our example, we chose $p_2 = 0.95$, *i.e.*, we want to guarantee a success probability of 95%. The curves were cut off at the right end to guarantee that the reported results were based on not less than ten runs of the algorithms; the underlying RLDs were based on 1,000 tries for the smallest instance and on 250 tries for the larger instances. As can be seen from the efficiency graphs, we observe an efficiency of one (corresponding to optimal speedup) up to a certain maximal number $p_{max}$ of processors. This $p_{max}$ value depends directly on the relative length of the initial search phase (*cf.* Sections 5.1.2 and 5.1.3) and, consequently, increases monotonically with instance hardness. In our example, for the

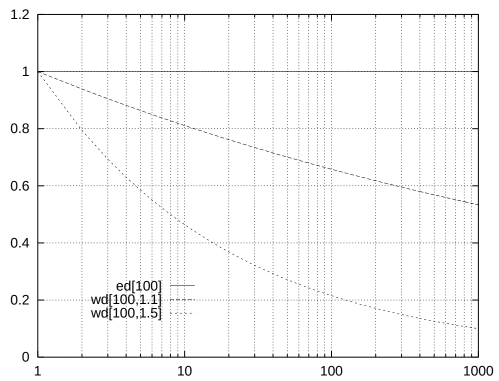


Figure 5.15: Efficiency of multiple independent try parallelisation (vertical) *vs* number of processors (horizontal) if the underlying (hypothetical) algorithm's run-time behaviour can be characterised by a Weibull distribution. An efficiency value of 1 indicates optimal speedup.

smallest instance, `bw_large.a`, the efficiency drops almost immediately when parallelising, while for the largest instance, `bw_large.c`, optimal speedup can still be achieved with ca. 100 processors. Furthermore, $p_{max}$ is positively correlated with the desired success probability $p_s$ such that for tighter success guarantees, the maximal number of processors such that optimal efficiency is still achieved, is higher.

However, in cases, where the RLDs are generally steeper than an exponential distribution, using independent tries will always lead to a suboptimal speedup (*cf.* Chapter 2); again, using RLD data, it is possible to predict the speedup as a function of the number of processors. This is shown for the example of Weibull distributions $wd[m, \beta]$ with $\beta \geq 1$ as RLDs in Figure 5.15; note that while for $\beta = 1$ (exponential distribution) optimal efficiency is observed, with increasing $\beta$ (*i.e.*, steepness of the distribution) the efficiency decays faster for large number of processors. However, leaving aside the effects of the initial search phase for EPAC algorithms (as discussed above), we have never observed this type of behaviour in our experiments with SLS algorithms for SAT.

For RLDs which are partially less steep than an exponential distribution, optimal cutoff times can be derived as detailed in Chapter 2, Section 2.2.2. This situation can be observed particularly for smaller-than-optimal noise parameter settings (*cf.* Section 5.1.4); here, the optimal cutoff implies an optimal number of processors for which a maximal speedup can be obtained when compared to the sequential case $p = 1$. This is exemplified in Figure 5.16, showing the RLD for GWSAT with a suboptimal noise setting when applied to a small Blocks World Planning instance. The "hump" in the multi-modal RLD indicates the existence of an optimal cutoff which corresponds to the optimal number of processors for multiple independent tries parallelisation represented by the maximum in the efficiency graph. Note that the efficiency compared to the sequential algorithm without restart is

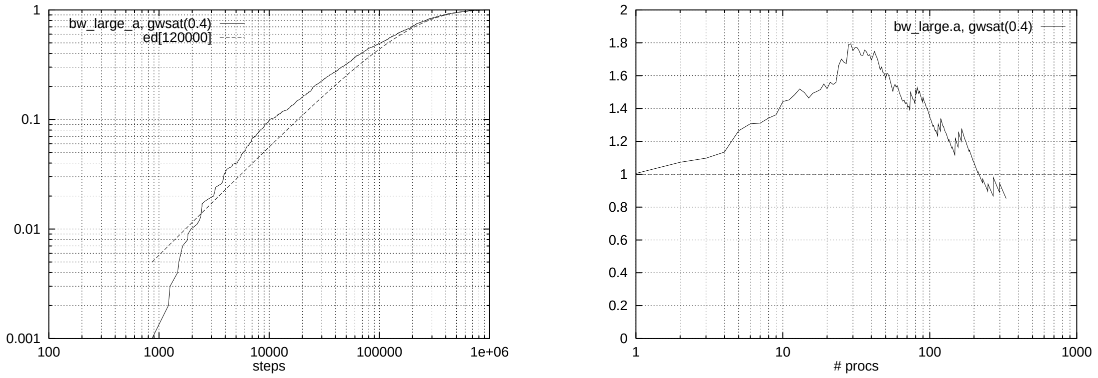


Figure 5.16: GWSAT with suboptimal noise setting (`wp` $= 0.4$) applied to Blocks World Planning instance `bw_large.a`. *Left:* RLD and corresponding asymptotic exponential distribution; *right:* parallelisation efficiency *vs* number of processors ($p_s = 0.95$).

greater than one, *i.e.* super-optimal, over a wide range of parallelisation.

Generally, using multiple independent tries is an extremely attractive model of parallel processing, since it involves basically no communication overhead and can be easily implemented and run on almost any parallel hardware platform, from networks of standard workstations to specialised MIMD machines with thousands of processors. Therefore, these results are not only relevant for the application of SLS algorithms to hard combinatorial problems like SAT to time-critical tasks (like robot control or online scheduling), but also for the distributed solving of very large and hard problem instances.

### 5.2.3 Parallelisation Based on Algorithm Portfolios

For SLS algorithms, algorithm portfolios are equivalent to the heterogeneous cooperative GLSM model without communication, introduced in Chapter 3. Intuitively, the idea is to execute a set of algorithms ("portfolio") in parallel, applying each individual algorithm to the same problem instance. Note that, like multiple independent tries parallelisation, this scheme can be easily realised on a single-processor system by using processor multiplexing to interleave the execution of the individual algorithms. Multiple independent tries parallelisation can be interpreted as a special case of an algorithm portfolio, where all individual algorithms are identical. Portfolios comprising different algorithms, or differently parameterised instances of an algorithm, only realise advantages over this simpler case if

1. no single individual algorithm dominates all the others, and
2. there are situations in which it is not possible to decide *a priori* which algorithm will be most effective.

If condition 1 is violated, *i.e.*, there is a completely dominant individual algorithm, obviously replacing all other elements of the portfolio with this algorithm will improve the overall

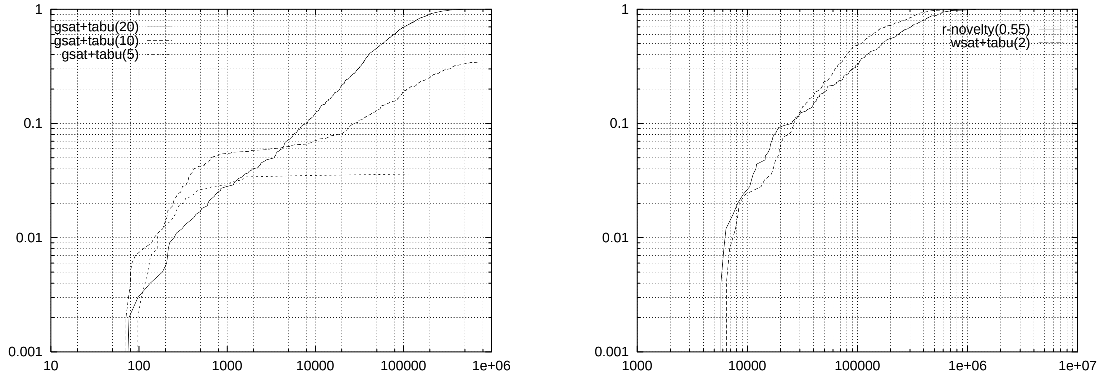


Figure 5.17: *Left:* Crossing RLDs for GSAT+tabu, applied to Random-3-SAT instance `uf100/hard`, using different tabu-list lengths; *right:* crossing RLDs for R-Novelty and WalkSAT+tabu (approx. optimal noise settings) when applied to Blocks World Planning instance `bw_large.b`.

performance. Condition 2 is considerably weaker. Of course, based on an RLD analysis, it is always possible to decide *a posteriori* which algorithm of a given portfolio is best for each problem instance and run-time interval. But only if this information is *a priori* available, it can be used to optimise the portfolio. Since generally, complete domination between algorithm seems to be rather the exception than the rule for hard combinatorial problems, the ability to successfully establish or approximate this *a priori* knowledge would be a great improvement for many problem classes. At least for SAT, the problem of finding methods for successfully predicting dominance relations between different algorithms, or even algorithm classes (such as complete search *vs* incomplete local search) is essentially unsolved. Therefore combining individual algorithms which show superior performance in certain situations or for certain domains into portfolios appears to be a promising method for solving hard SAT problems.

For the SLS algorithms considered here, we have observed different phenomena suggesting that the portfolio approach will be beneficial. First, while for a given problem instance, strict domination between different algorithms and parameterisations seems to be the rule, we occasionally observed crossing RLDs. Within our benchmark suite, such examples could be found for all algorithms presented here. Figure 5.17 shows two different cases for crossing RLDs: On the left-hand side, we see how GSAT+tabu applied to a hard Random-3-SAT instance exhibits crossing RLDs for different tabu-list lengths. Clearly, the shorter tabu-lists give superior performance for shorter run-times, but suffer from stagnation for longer runs. This phenomenon can also be observed for GSAT+tabu when applied to the Graph Colouring domain. The right-hand side of Figure 5.17 shows an example for crossing RLDs between different algorithms using optimal noise settings. Here, R-Novelty is superior to WalkSAT+tabu for short runs, while for long run-times, WalkSAT+tabu is significantly better. Again, a similar situation can be observed for other problem domains from our benchmark suite.

Apart from crossing RLDs, we also observed that for different problem domains, different SLS algorithms show the best performance. The results reported in Chapter 4 suggest that Novelty and R-Novelty are best-performing for random, unstructured problems, while WalkSAT+tabu and GSAT+tabu might be superior for relatively hard, large, structured problem instances, like the large Blocks World Planning instances, the hard Graph Colouring instances, and the All-Interval-Series instances. Thus, to optimise performance over a broad range of domains, or in a situation where the relative performance of the different algorithms is unknown, combining several of the best-performing algorithms mentioned above into a mixed portfolio is advantageous. Furthermore, such a portfolio can also include promising complete algorithms, like Satz [LA97] or Satz-Rand [GSK98], which outperform the best currently known SLS algorithms for certain highly structured problem domains, like Quasigroup Completion, or All-Interval-Series.

Given the current state of knowledge regarding the performance of SAT algorithms across different domains, we conjecture that for solving SAT instances from a broad range of problem domains, or instances the structure of which is largely unknown, heterogeneous portfolios comprising powerful algorithms, like Novelty$^+$, R-Novelty$^+$, WalkSAT+tabu, and GSAT+tabu, as well as state-of-the-art complete algorithms, like Satz-Rand, will prove to be the best and most robust solution methods. Such portfolios can probably be further enhanced by including different parameterisations (*e.g.*, noise settings) for the different algorithms; and in a last step, performance might be even further improved by automatically optimising the portfolios within single runs or over a number of runs, in the spirit of the evolutionary cooperative GLSM models discussed in Chapter 3.

## 5.3 Related Work

Although SLS algorithms for SAT, in particular GSAT and WalkSAT variants, have been intensively studied in the past, there is only a small number of studies which attempt to characterise SLS behaviour. One of these investigates the behaviour of GSAT and differentiates its behaviour into an initial hill-climbing and subsequent plateau phases [GW93a]. These phases are mathematically modeled; however, the characterisation is based on averaged local search trajectories on hard Random-3-SAT instances rather than on RLDs. A more recent study identifies invariants in local search behaviour [MSK97] for various algorithms of the WalkSAT family. Again, the characterisation is based on local search trajectories and no mathematical model is specified. However, it is empirically observed and conjectured that optimal behaviour of WalkSAT algorithms seems to occur for close-to-minimal mean to variance ratio of the objective function as sampled along the search trajectory. This is shown for single problem instances from various domains, including hard Random-3-SAT and Blocks World Planning.

Our approach of functionally approximating the run-time behaviour of SLS algorithms is partly related to the methodology used in [FRV97]. This concerns primarily the use of continuous probability distributions (which can also be found in [Hoo96b, HS96]) for approximating the behaviour of discrete algorithms and the $\chi^2$-test for testing the goodness-of-fit of these approximations. But while they study the search cost distribution of finding a satisfying assignment on an ensemble of instances for *complete* procedures for SAT, we characterise the behaviour of SLS algorithms applied to single problem instances based on run-time distributions. Gomes and Selman give a functional characterisation of the run-time behaviour of randomised complete search procedures for SAT applied to single problem instances from the *Quasigroup Completion Problem* [GSC97]. While they observe heavy-tailed behaviour which can be exploited using rapid random restarts, we did not observe heavy-tailed RLDs for SLS algorithms (using reasonably good parameter settings) when applied to any of our benchmark instances.

To our best knowledge, the approximative completeness of SLS algorithms for SAT has not been theoretically studied before. Thus, the results presented in Section 5.1.1 are novel and original. The same holds for the functional charactisations of SLS behaviour developed in Sections 5.1.2 and 5.1.3. Our main characterisation result, the EPAC behaviour of modern SLS algorithms (using optimal or larger-than optimal noise settings) when applied to hard problem instances from various domains, explains earlier observations regarding the effectiveness of random restart for GWSAT and WalkSAT [GW95, PW96] (*cf.* Section 5.1.5).

The improvements of SLS algorithms for SAT we discuss in Section 5.2 are inspired by previous work: Our study of multiple independent try parallelisation is conceptually related to Hogg's and Williams' work on the potential of parallel processing for *complete* Graph Colouring algorithms [HW94a]. Our discussion of algorithm portfolios is mainly based on ideas presented by Carla Gomes and Bart Selman in the context of complete algorithms for Quasigroup Completion Problems [GS97a]. Finally, the random walk extension of Novelty and R-Novelty is inspired by earlier work on GSAT variants [SKC93] and builds on McAllester's, Selman's, and Kautz's work on the WalkSAT family [MSK97]. Interestingly, combining our essential incompleteness result for WalkSAT+tabu with the ideas of random walk extension, sheds a new light on earlier work which claimed that adding random walk to tabu search algorithms for SAT does not improve SLS performance [SSS97]. Our results for SAT-encoded Graph Colouring instances show that essential incompleteness occurs in practice when using optimal tabu-list length (*cf.* Section 4.5); but the fact that essential incompleteness can generally be overcome by adding random walk (*cf.* Section 5.2) indicates that extending WalkSAT+tabu with random walk will improve the overall behaviour of the algorithm. The fact that in [SSS97] evidence for a contradicting conclusion could be observed is most probably a consequence of the problematic methodology their result is based on (*cf.* Chapter 2, Section 2.3).

## 5.4 Conclusions

In this chapter we presented three types of results regarding the run-time behaviour of modern SLS algorithms for SAT: theoretical results characterising their asymptotic behaviour, functional characterisations of their actual behaviour, and a number of strategies to improve existing algorithms.

Regarding the asymptotic behaviour of SLS algorithms, we proved a series of novel results, establishing the approximative completeness (PAC property) of GWSAT (for $\mathsf{wp} > 0$) as well as the essential incompleteness of WalkSAT+tabu (for all $\mathsf{tl} > 0$), Novelty, and R-Novelty (for arbitrary $\mathsf{wp}$). Although for the latter algorithms, which are among the most powerful SAT algorithms known today, essential incompleteness does not imply that stagnation of the local search will occur for each problem instance and each run of the algorithm, stagnation behaviour can be observed in practice (*cf.* Chapter 4) and severely affects the otherwise superior performance of these algorithms.

However, essential incompleteness can be easily overcome by extending the given algorithms with random walk, as shown in Section 5.2.1. We introduced this type of extension using the GLSM framework where it is easy to see under which conditions the resulting hybrid GLSM algorithm will be approximately complete. Of course, approximate completeness is also achieved by using a random restart mechanism; however, since in practice good cutoff values for the local search process are not known, and poorly chosen cutoffs lead to extremely poor performance, using random restart alone is substantially inferior to a random walk extension. To demonstrate the practical effectiveness of the random walk extension, we modified R-Novelty and Novelty accordingly and showed empirically that the extended variants achieve superior performance. To our best knowledge, these R-Novelty and Novelty variants are the most powerful currently known SLS algorithms for SAT. However, for certain problems, such as the All-Interval-Series instances, these algorithms, as all other SLS algorithms we are aware of, are substantially outperformed by state-of-the-art systematic SAT algorithms like Satz [LA97] or Satz-Rand [GSK98].

Regarding the functional characterisation of SLS behaviour, we empirically studied the run-time behaviour of WalkSAT, WalkSAT+tabu, Novelty, and R-Novelty. We could show that, using optimal noise parameter settings, the RLDs of these algorithms when applied to hard problem instances from various domains, can be approximated by exponential distributions (EPAC property). The same phenomenon is observed for larger-than-optimal noise settings, while for smaller-than-optimal noise, qualitatively different behaviour occurs. We further introduced a refined mathematical model based on a new distribution type which is asymptotically exponential, but allows to model the effects of the initial search phase. As we have shown, this extended model allows a precise characterisation of SLS behaviour for a vast majority of the problem instances from our benchmark suite. Thus, for the first time, we can model the behaviour of some of the most powerful and prominent SLS algorithms for SAT in great detail.

Our characterisation result has a number of interesting implications. As a direct consequence, random restart is ineffective for the algorithms studied here, when using optimal noise settings. Even worse, when the restart occurs during the initial search phase, *i.e.*, the cutoff occurs too early, the performance will be negatively effected. A second consequence of exponential RTDs is a new interpretation of the local search process as random picking in a drastically reduced, “virtual” search space. The size of this reduced search space can be calculated from the RTD data and, when compared to the actual search space size, gives an indication of the effectivity of the SLS algorithm. This novel interpretation of SLS behaviour raises the question, whether it is possible to identify structural features of the actual search spaces which directly correspond to the reduced search space. The analysis of search space structure is addressed in Chapter 6; however, so far we could not identify any features which are strongly correlated to the virtual search space size.

Finally, regarding the improvement of existing SLS algorithms for SAT, besides the hybrid GLSM extensions mentioned above, we discussed two forms of parallelisation. Multiple independent tries parallelisation, which is equivalent to using homogeneous cooperative GLSM models without communication (*cf.* Chapter 3), is particularly attractive for EPAC algorithms, since in this case optimal speedup and efficiency are obtained. Moreover, since no communication is involved, this model is simple to implement and extremely scalable. The second approach uses portfolios of different SLS algorithms; it can be easily modeled using heterogeneous cooperative GLSMs without communication. This model shows improved robustness and performance over the homogeneous model, when there is no clear domination relation between a set of individual algorithms, *i.e.*, when different algorithms show superior performance in different situations. As we have seen in Chapter 4, this is the case for modern SAT algorithms based on stochastic local search. Therefore, using the portfolio approach appears to be a promising way for improving robustness and performance of complex SAT algorithms.

Both parallelisation approaches are of particular interest in the context of time-critical application scenarios; however, using processor multiplexing, they can generally be also realised on single-processor systems. As a direct consequence of the EPAC behaviour observed for modern SLS algorithms, multiple independent tries will not improve performance on a single-processor system. For algorithm portfolios, the situation is different: based on the observations on our benchmark suite, improvements can be expected even for the single-processor case.

Summing up our results, we have shown how, based on a careful empirical analysis, SLS behaviour can be mathematically modeled in great detail by functionally approximating RTDs. Doing this, we find that the behaviour of some of the most powerful and popular SLS algorithms for SAT exhibit a surprising regular behaviour over a broad range of problem domains. Based on our characterisation of SLS behaviour, we developed and discussed a number of strategies for improving these algorithms. We expect (and have some preliminary evidence as of this writing) that our results on comparing, characterising, and improving SLS behaviour will generalise to SLS algorithms for problem domains like CSP or other hard combinatorial problems.