Chapter 6

Search Space Analysis

Stochastic local search algorithms are among the most powerful methods for solving hard combinatorial problems like SAT or CSP. Thus, there is a considerable interest in improving the algorithms as well as in the understanding of the factors which influence their performance. The behaviour of local search algorithms crucially depends on structural aspects of the underlying search space. Recently, a growing number of researchers have been investigating the nature of this dependency for the propositional satisfiability (SAT) and constraint satisfaction problems (CSP) [CFG+96, Yok97, FCS97]. The main motivation for this kind of work lies in the assumption that knowledge about structural aspects of search spaces and their correlation to the performance of SLS algorithms can ultimately be exploited to improve the algorithms as well as SAT or CSP encodings of problems from other domains. Furthermore, this kind of knowledge can be used for evaluating SLS algorithms and encoding strategies.

In this chapter, we explore several approaches for search space analysis. After giving some motivation and background on the topic, we first analyse the dependence of local search performance on the number of solutions, an issue which has been investigated before [CFG+96]. Our study, however, avoids some weaknesses of this earlier work while extending both the scope and the depth of the analysis significantly, *e.g.*, by studying the observed phenomena for more powerful SLS algorithms and in dependence of problem size. Then, we introduce some novel approaches for search space analysis, based on analysing the trajectories of SLS algorithms. Our results indicate that the most dominant factor for local search performance is given by the number of solutions. However, other factors, like solution clustering, ruggedness of the search space, and local minima branching have a measurable influence on the behaviour of SLS algorithms.

test-set	instances	clause-len	vars	clauses
uf10-49	1,000	4	10	49
uf20-91	1,000	3	20	91
uf50-163	1,000	3	50	163
uf50-273	1,000	3	50	237

Table 6.1: Additional Uniform Random-3-SAT test-sets which are used for search space structure analysis.

6.1 Motivation and Background

Compared to the development of SLS algorithms for SAT, search space analysis is a rather young research subject. The methods we use for analysing search space structure and its impact on SLS behaviour are to some extent based on the approaches taken in previous work [CFG+96, Yok97, FCS97]. However, we find a number of weaknesses occurring in some or all of these studies:

- GSAT, or simplified versions of this algorithm are used as a reference local search procedure. While GSAT has the advantage of conceptual simplicity, its performance is considerably inferior to modern SLS algorithms like GSAT with random walk or WalkSAT, which in particular have the ability to escape from local minima without using a restart mechanism.
- The empirical methodology used to characterise SLS behaviour is often problematic. For example, [CFG+96] use only one maxSteps setting for all experiments, which is claimed to be optimal for the middle of the phase transition. While there is some doubt about the validity of this optimal value itself [HS98a], it is at least not clear, whether this value is also optimal for other clauses/variable ratios. On the other hand it is known that GSAT's performance critically depends on good maxSteps settings [GW95].
- Most of the studies are mainly based on the empirical analysis of Random-3-SAT instances, while other problem classes, in particular SAT-encoded problems from other domains, are somewhat neglected. Furthermore, the dependence of the results on the problem size is usually not investigated. This seems to be particularly problematic, since the importance of taking scaling properties into account is well-known in the field [Fuk97].

In our investigation presented here, we try to avoid these weaknesses. Instead of GSAT, we use GWSAT with approximately optimal noise settings for our experiments. While still being conceptionally very simple, the influence of maxSteps on the performance of this algorithm is minimal (*cf.* Chapter 5, [HS96, HS98a]). For our experiments we used instances and test-sets from the benchmark suite introduced in Chapter 4. Additionally, we generated two test-sets of very small Random-3-SAT instances (from the phase transition region) as well as two test-sets from the under- and over-constrained regions of Uniform Random-3-SAT (*cf.* Table 6.1). For these latter test-sets we chose the number of clauses such that the clauses/variable ratios are 25% below and above the phase transition point. Furthermore, for comparative analyses involving problems from other domains, we used instances from various other Uniform Random-3-SAT test-sets; this will be described in more detail in Section 6.4. All additional test-sets were generated as described in Chapter 4, *i.e.*, by filtering randomly generated Random-3-SAT formulae with a complete SAT algorithm.

If not indicated otherwise, the local search cost reported for individual problem instances was determined by running GWSAT (with approx. optimal noise setting) for at least 100 tries and estimating the expected local search cost per solution as defined in Chapter 2, taking into account the success-rate as well as the length of successful tries. Generally, maxSteps was chosen high enough ($\geq 10^6$) to ensure success probabilities close to one; this way, the error in extrapolating the local search cost is relatively small. As we have shown before, there is a strong correlation between GWSAT's performance and that of other modern SLS algorithms (*cf.* Chapter 4), therefore most of our results regarding local search cost will equally apply to other SLS algorithms for SAT, such as WalkSAT or Novelty.

When analysing search space structure and its impact on SLS performance, one of the most important techniques is the investigation and characterisation of correlations between structural features of the search space and local search cost. For our Random-3-SAT and Graph Colouring test-sets, we characterise these correlations establishing functional models using least-mean-squares fitting and regression analysis techniques known from literature [CFG+96].

6.2 The Number of Solutions

Intuitively, for a given problem instance, its number of solutions should have a considerable effect on local search performance: One would expect that instances with a high solution density, *i.e.*, a large number of solutions, are much easier to solve for SLS algorithms than instances with very few solutions. In previous work [CFG+96], it was shown that this intuition is correct regarding stochastic local search algorithms for SAT and CSP which are based on hill-climbing, such a GSAT. In this section, we first present the results of analysing the distribution of the number of solutions for different Random-3-SAT test-sets. Based on these results, we then discuss the correlation between local search cost and the number of solutions, refining and extending the methodology and results of [CFG+96].

Figure 6.1 (left) shows the empirical cumulative distributions of the number of solutions for the different test-sets at the phase transition for Random-3-SAT [CA96]. The coarse granularity which can be observed at the left end of the curves, especially for $n = 20$, is due to the fact that the number of solutions is a discrete measure and that for small problem sizes many instances have a very small number of solutions. The graphs show clearly

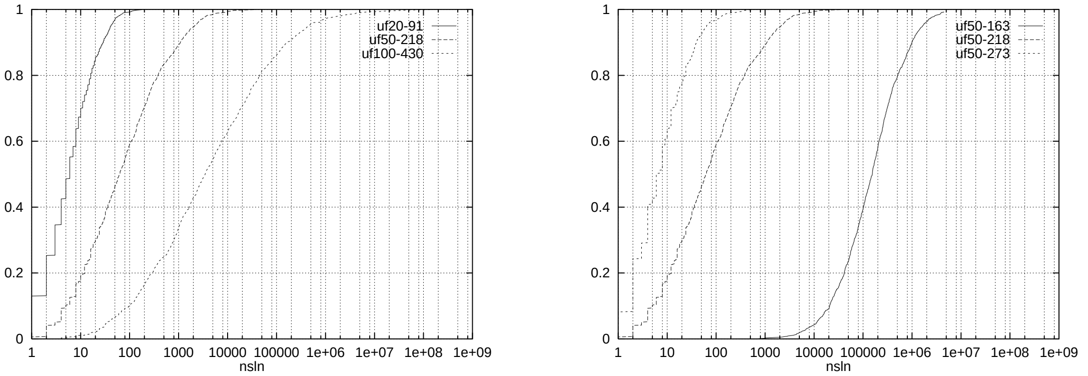

Figure 6.1: Distributions of the number of solutions for Random-3-SAT test-sets with n variables, k clauses (1,000 instances per test-set); *left:* different problem sizes at phase transition; *right:* for $n = 50$ across phase transition.

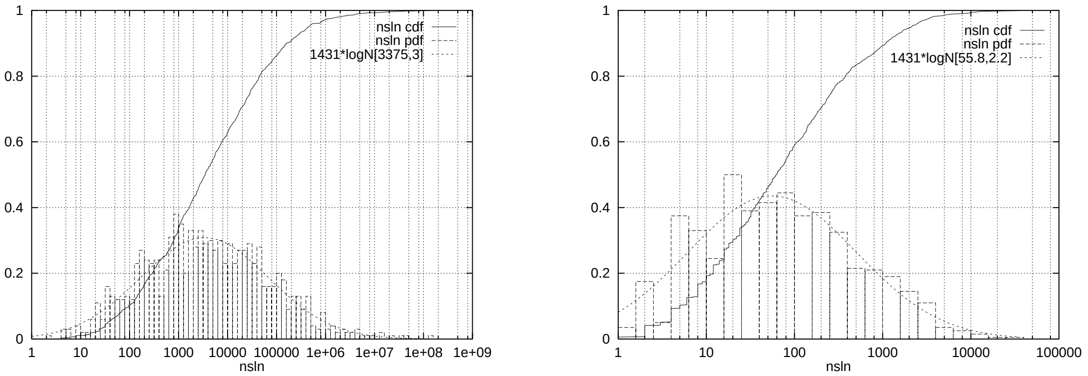

Figure 6.2: Number of solutions distributions for test-sets `uf100-430` *(left)* and `uf50-218` *(right)* at the phase transition (solid curves); the corresponding empirical probability density functions (bar diagrams) can be approximated by log-normal distributions (dashed curves).

that for increasing problem size, the distributions become progressively less steep, *i.e.*, the relative variance in the number of solutions increases significantly for larger problems. Note the difference between this scaling behaviour and the observations for the corresponding hardness distributions, where increasing the problem size affects mainly the tail.

At the first glance, the general shape of the distributions of local search cost and the number of solutions seems to be roughly similar (compare Figure 6.1 and Figure 4.13, page 95). But closer analysis of the empirical probability density functions of the number of solutions distributions indicates that (in contrast to the local search cost distributions) these can be reasonably well approximated using log-normal distributions (Figure 6.2). This is an effect of the random generation process, where each point in the search space (*i.e.*, each assignment) is uniformly affected by each independently generated clause. Note the absence of long and heavy tails, as observed for the corresponding hardness distributions, for which we could not find a functional approximation.

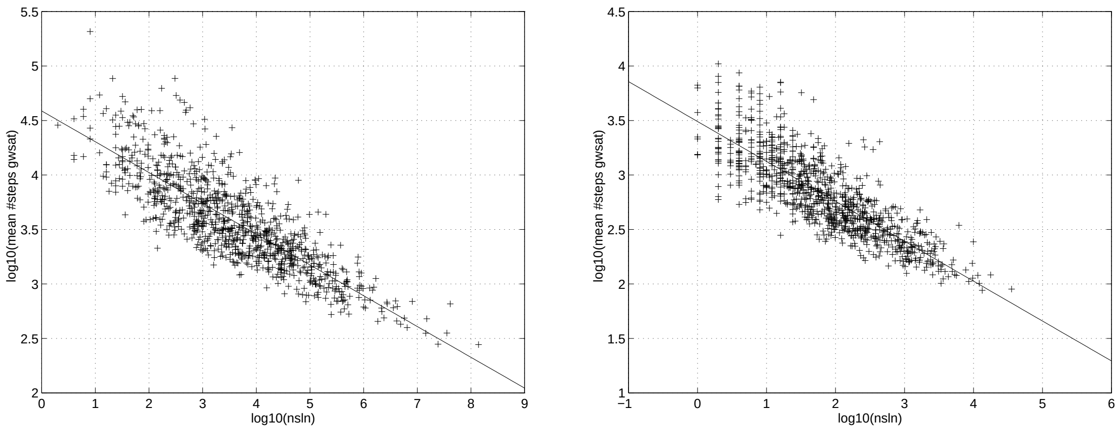

Figure 6.3: Correlation between average local search cost (vertical) and number of solutions (horizontal) for Random-3-SAT test-sets from the phase transition region with $n = 100$ *(left)* and $n = 50$ *(right)* and regression lines from *lms* regression fits.

test-set	a	b	r
uf20-91	-0.494	2.309	-0.82
uf50-218	-0.366	3.491	-0.83
uf100-430	-0.283	4.586	-0.83

test-set	a	b	r
uf50-163	-0.282	3.436	-0.90
uf50-218	-0.366	3.491	-0.83
uf50-273	-0.222	2.827	-0.51

Table 6.2: Data from regression analysis of the correlation between number of solutions and local search cost for Random-3-SAT test-sets; a, b are the parameters of the *lms* linear fit $ax+b$, r is the correlation coefficient; *left:* for different problem sizes at the phase transition, *right:* across the phase transition.

Combining the data from hardness and number of solutions analysis, we can now analyse the correlation between the logarithm of both values. Figures 6.3 depicts this correlation for various problem sizes at the phase transitions. The graphs show scatter plots of the data and the corresponding *lms* regression fits. For all problem sizes, there is a strong negative correlation between the number of solutions and the average local search cost per solution. Also note that while for a high number of solutions, the variability in local search cost is very small, it increases considerably for lower numbers of solutions. Again, the coarse granularity which can be observed at the left side of the graphs for the smaller problem size is caused by the large number of problem instances with few solutions.

As can be seen from Table 6.2 (left), we measure a stronger correlation between the number of solutions and average local search cost than reported in [CFG$^+$96] ($r = -0.83$ *vs* -0.77). At the same time, for $n = 100$ we find a significantly smaller gradient a for the regression fit (-0.283 *vs* -0.44 in [CFG$^+$96]). Remarkably, a decreases strictly monotonically with growing problem size n, meaning that for larger problems the functional dependency of the local search cost from the number of solutions gets weaker. The main reason for this stems from the fact that with growing problem size, the relative variance of the number

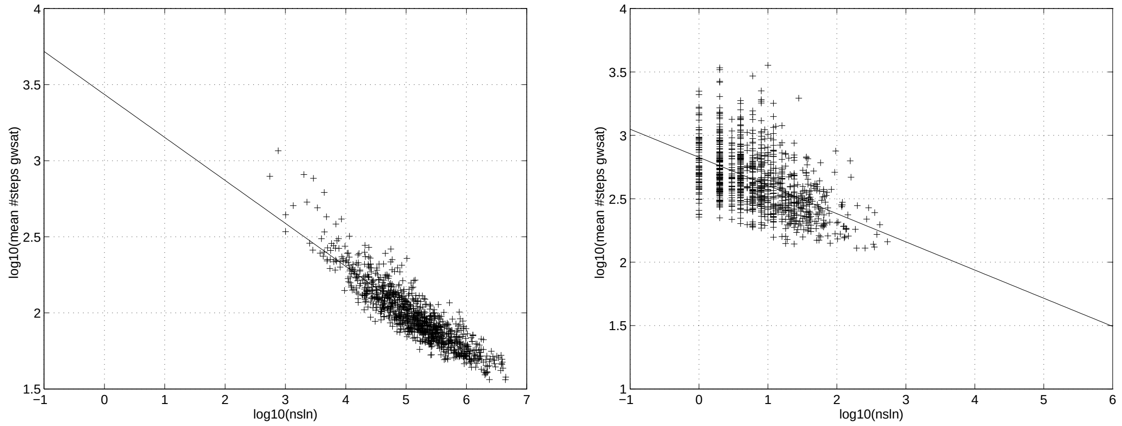

Figure 6.4: Correlation between local search cost (horizonal) and number of solutions (vertical) for Random-3-SAT test-sets with $n = 50$ from the underconstrained region *(left)* and from the overconstrained region *(right)*.

of solution distribution increases significantly faster (and more uniformly across the whole distribution) than that of the corresponding hardness distribution. The dependency between $\log a$ and $\log n$ appears to be roughly linear. More specifically, a very good *lms* fit can be obtained using the parametric function $a = \exp(u \cdot n^v)$, where the optimal parameters are determined as $u = 1.53$ and $v = -0.37$. Not surprisingly, the average local search cost for problem instances with exactly one solution (as estimated by the b parameter of the linear *lms* fit) also increases strictly monotonically with n. Interestingly, using the same fitting approach as above, the dependency between b and n can be approximated by $b = \exp(0.61 n^{0.44})$. This confirms earlier observations that the search cost of GSAT-like algorithms might grow slower than a simple exponential [GW93b, GMPW97]. Our results indicate furthermore that the same holds for the extrapolated local search cost for problem instances with exactly one solution.

Applying the same correlation analysis across the phase transition, *i.e.*, for different clauses per variable ratios, reveals some interesting facts. First, the results in Table 6.2 (right) show that the correlation coefficient r is clearly dependent on the clauses per variable ratio (*cvr*): for increasing *cvr*, r decreases monotonically. This can be also seen in Fig. 6.4, which shows considerably more noise in the scatter plot for the overconstrained than for the underconstrained test-set. One explanation for this phenomenon could be based on the simple fact that the number of solutions tends to decrease with a growing constrainedness, assuming that for a smaller number of solutions there is more variability in the average local search cost. This hypothesis is not only consistent with the results for $n = 50$ across the phase transition, but also with the fact that within all our test-sets, we found the variability in the number of solutions to be decreasing with growing number of solutions.

Another interesting observation can be made from Table 6.2 (right): The regression gradient a is maximal at the phase transition and significantly lower above and below the critical

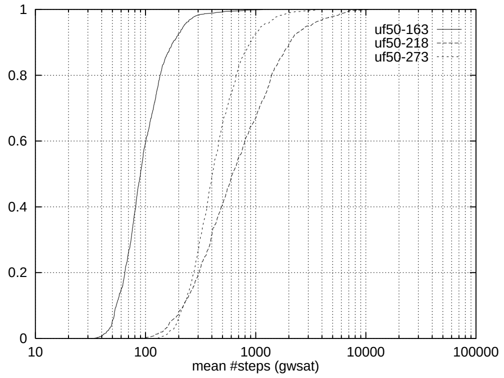

Figure 6.5: Hardness distributions (mean number of local search steps/solution using GWSAT, approx. optimal noise) for Random-3-SAT test-sets across the phase transition.

cvr. Consequently, the number of solutions has the most significant impact on local search cost at the phase transition. This result is a direct contradiction to the result for GSAT (without random walk) reported in [CFG+96], but consistent with their results for various CSP algorithms on Random-CSP problems. The most probable reason for this disagreement is given by the fact that they are using basic GSAT in combination with a rather crude methodology for evaluating the algorithm's behaviour. Note, that the extreme behaviour at the phase transition is also reflected in the hardness and number of solutions distributions (*cf.* Figures 6.5 and 6.1, right). We will come back to this issue in Section 6.3.

One last observation on the data presented in Table 6.2 (right) is the fact that the b parameters of the *lms* linear fits are roughly similar for test-sets `uf50-163` and `uf50-218`, while for `uf50-273` it is considerably lower. Since the b parameter gives an estimate for the average search cost on problem instances with exactly one solution, this observation implies that single solution instances tend to be harder when taken from the underconstrained region than when being overconstrained. One possible explanation for this phenomenon is given by the observation that underconstrained problems tend to have larger plateau regions [Yok97] which are probably more difficult to search for SLS algorithms like GWSAT or WalkSAT [FCS97].

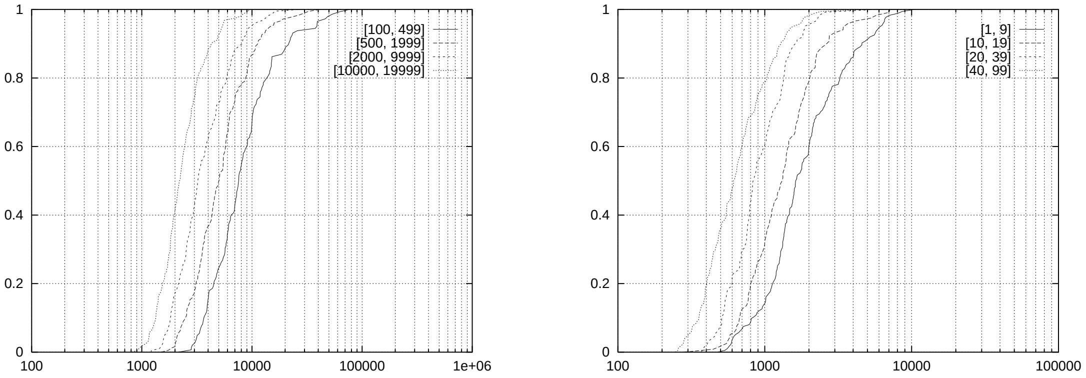

Figure 6.6: Hardness distributions for different strata $[l_i, u_i]$ of Random-3-SAT test-sets with $n = 100$ *(left)* and $n = 50$ *(right)* at the phase transition.

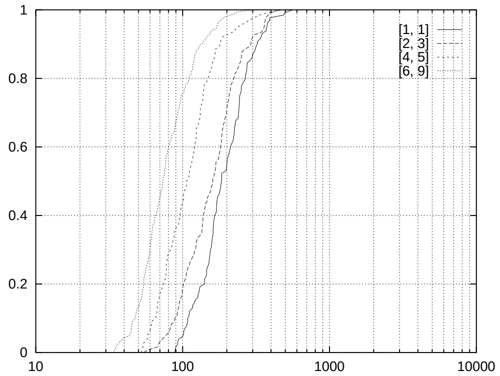

Figure 6.7: Same as Figure 6.6, but for $n = 20$.

6.3 Stratified Random-3-SAT

Although our results show a stronger correlation between the number of solutions and local search cost than reported in [CFG+96], we still observe a considerable variability in the local search cost for instances from the same test-set with a similar or equal number of solutions. To abstract from the predominant influence of the number of solutions on problem hardness, we study this phenomenon in more detail by stratifying our test-sets (defined by purely syntactic properties of the formulae) according to the number of solutions (a semantic property). More precisely, we measure hardness distributions (using the same method and parameters as in Chapter 4, Section 4.5), for the strata s_i obtained from the test-sets by restricting the number of solutions to an interval $[l_i, u_i]$.

For various problem sizes at phase transition, the results of this analysis are shown in Fig. 6.6 and 6.7. Surprisingly, in a semi-log plot, the shapes of these distributions are very similar for the different strata. For larger numbers of solutions, the corresponding curves are shifted to the left and they become slightly steeper, indicating that problems with many

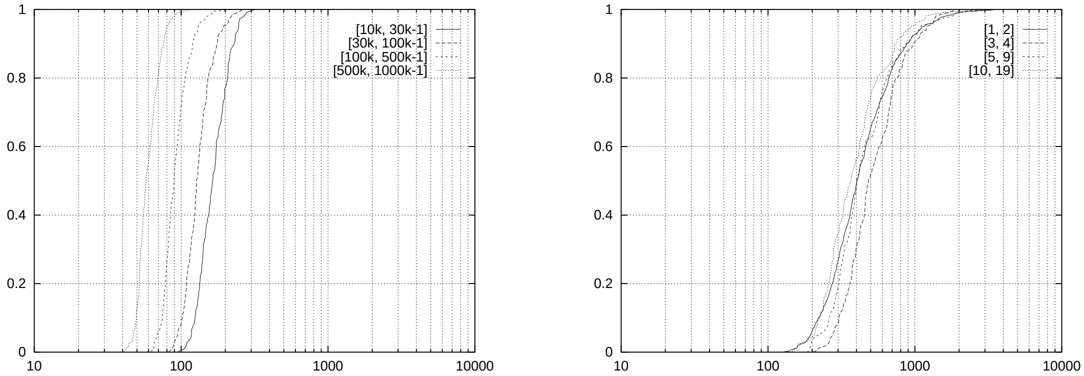

Figure 6.8: Hardness distributions for different strata of Random-3-SAT test-sets with $n = 50$ from the underconstrained region *(left)* and from the overconstrained region *(right)*.

solutions are uniformly easier and exhibit a smaller relative variance of instance hardness. Note also that the overall shape of the distributions for the strata is very similar to that of the overall hardness distributions (*cf.* Figure 6.5). However, the variance of the overall hardness distribution is significantly higher than for the strata; closer examination also reveals slightly heavier tails for the former.

Applying the same analysis across the phase transition, we observe an interesting difference (*cf.* Figure 6.8). While for the overall hardness distribution, minimal steepness (*i.e.*, maximal relative variance) can be observed at the phase transition (*cf.* Figure 6.5), the steepness of the corresponding distributions for the strata appears to be monotonically decreasing with growing constrainedness (*cf.* Figures 6.6–6.8). This means that across the phase transition, the shape of the overall distribution depends not only on the shape of the distributions for the strata, but also on the distribution of the number of solutions. In particular, the fact that the regression gradient for the correlation between local search cost and number of solutions is minimal at the phase transition (*cf.* Section 6.2) appears to be caused by the shape of the number of solutions distributions (*cf.* Figure 6.1, right).

Generally, the results reported in this section show that the noise observed in the correlation between local search cost and the number of solutions is mainly caused by some factor which is not related to the number of solutions. In particular, clustering of the solutions is most likely not the feature responsible for the variation in hardness within the strata: If clustering, *i.e.*, the distribution of the solutions within the search space, would have a major impact on local search cost, we should observe an increased solution cost variability for problems with higher numbers of solutions (where more pronounced differences in clustering can occur). But there is no significant difference between the hardness distributions on problems with only one solution (where clustering is irrelevant) and a higher number of solutions.

6.4 Sampling the Objective Function

Up to this point, our analysis was mainly concerned with the number of solutions and their distribution in the search space. But of course, other aspects of the search space topology play an important role in the context of SLS behaviour. One such factor is the ruggedness or smoothness of the objective function across the search space. It is intuitively clear that a mainly flat objective function which is only locally punctuated by the solutions should be much harder to deal with for any local search algorithm than one in which the solutions are surrounded by sloping basins. Unfortunately, the topological property corresponding to this intuitive notion of ruggedness is very difficult to measure directly.

We therefore use a simple indirect method for analysing the influence of objective function topology on SLS performance. This method essentially correlates the variance of the objective function, which is an indication of the ruggedness of the search space, to the average solution cost. For small problem instances ($n \leq 30$), we can use exhaustive search to determine the exact variance. For larger n, however, this becomes impractical; consequently, we use random sampling to estimate the objective function's variance. The intuition underlying this method is that flat, feature-less search spaces characterised by a low variance of the objective function should pose greater difficulties for SLS algorithms than rugged search spaces with a high objective function variance.

Random-3-SAT We performed this analysis for the Random-3-SAT test-sets for variable problem size and constrainedness. Because of computation time limitations, we reduced each test-set to 50 instances by selecting each 20th instance from the original test-set after ordering it according to the average local search cost for GWSAT (approx. optimal noise). We then measured the standard deviation of the objective function (number of unsatisfied clauses) for each remaining instance; for $n = 20$, the measurement is based on a systematic sample across the full search space (obtained by exhaustive search), for $n = 50$ and 100, it is based on a sample of 10^5 randomly picked assignments.

Preliminary results on single instances of the `uf50-218` test-set suggested a negative correlation between the standard deviation of the number of unsatisfied clauses ($sdnclu$) and the average local search cost. The result of a correlation analysis across the whole test-set confirms this hypothesis. Figure 6.9 shows the correlation between $sdnclu$ and the average local search cost (for GWSAT). The negative correlation, although quite noisy, is clearly observable (*cf.* Table 6.3 for correlation coefficient and parameters of the linear *lms* fit).

This result is consistent with our initial intuition that a smaller standard deviation indicates a more uniform search space structure, possibly providing less guidance for local search. Another explanation would suggest that for hard Random-3-SAT instances, a smaller variance of the objective function coincides with a lower number of solutions. This is consistent with our observation that the values of the objective function are approximately normally distributed:[1] in this situation, a smaller variance naturally implies smaller tail probabilities

[1] This is not very surprising, since for a given Random-3-SAT instance, clauses are generated randomly

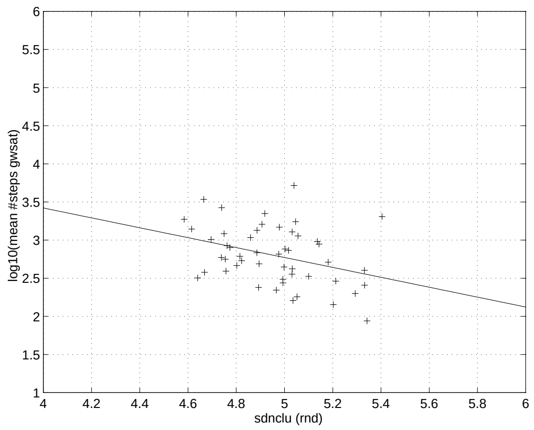

Figure 6.9: Correlation between *sdnclu* values (horizontal) and mean local search cost (vertical) for reduced test-set `uf50-218 (50)`.

test-set	r	a	b
uf20-91 (50)	-0.31	-0.42	3.28
uf50-218 (50)	-0.35	-0.65	6.02
uf100-430 (50)	-0.44	-0.97	10.31

test-set	r	a	b
uf50-163 (50)	-0.52	-0.56	4.37
uf50-218 (50)	-0.35	-0.65	6.02
uf50-273 (50)	-0.02	-0.02	2.75

Table 6.3: Correlation between *sdnclu* and average local search cost; a, b are the parameters of the linear fit $ax + b$, r is the correlation coefficient; *left:* for different problem sizes at phase transition, *right:* across phase transition

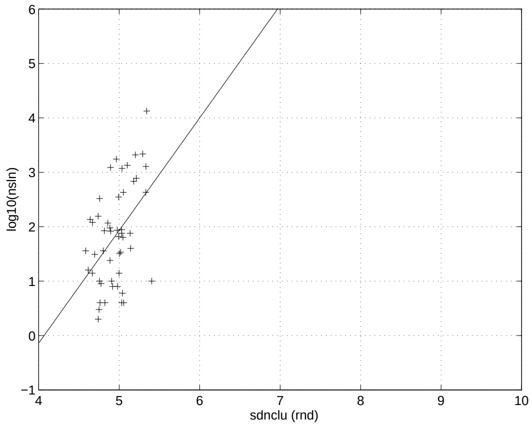

Figure 6.10: Correlation between *sdnclu* values (vertical) and number of solutions (horizontal) for reduced test-set `uf50-218 (50)`.

of this distribution and, consequently, a smaller number of solutions.

While both explanations could be correct, it is a priori not clear, whether one or the other plays a more important role in the given context. To further investigate this issue, we analysed the correlation between the *sdnclu* value and the number of solutions across the same test-set. The results of this analysis is shown in Figure 6.10: a positive correlation can be clearly observed (correlation coefficient $r = 0.46$) and a *lms* linear fit of the correlation data gives a gradient of 2.07. Note that this correlation is slightly stronger than the one between *sdnclu* and the average solution cost reported before. This suggests the following interpretation: For hard Random-3-SAT instances, a uniform search space structure, besides other factors, tends to come along with a small number of solutions, which itself has a major impact on the local search cost for finding a solution. Compared to this effect, the more uniform search space structure seems to play a minor role. This was verified by analysing the correlation between *sdnclu* and average solution cost for strata of the unreduced test-set `uf50-218` with equal or similar numbers of solutions. Here, a negative correlation between *sdnclu* and the average solution cost could not be observed. Thus, the *sdnclu* / solution cost correlation can apparently not be used to refine the analysis from Section 6.2.

Next, we studied the observed correlation for various problem sizes at the phase transition as well as across the phase transition. The results are shown in Figures 6.11–6.12 and in Table 6.3. While for the underconstrained test-set, the correlation between *sdnclu* and average solution cost is quite strong, it is getting weaker as the number of clauses increases. For the overconstrained test-set, no significant correlation could be observed. Note that this

---

and the influence of each clause on a given assignment is mostly independent of other clauses. Note, however, that by restricting the test-sets to satisfiable formulae, strict probablistic independence between the impact of clauses on the search space is lost.

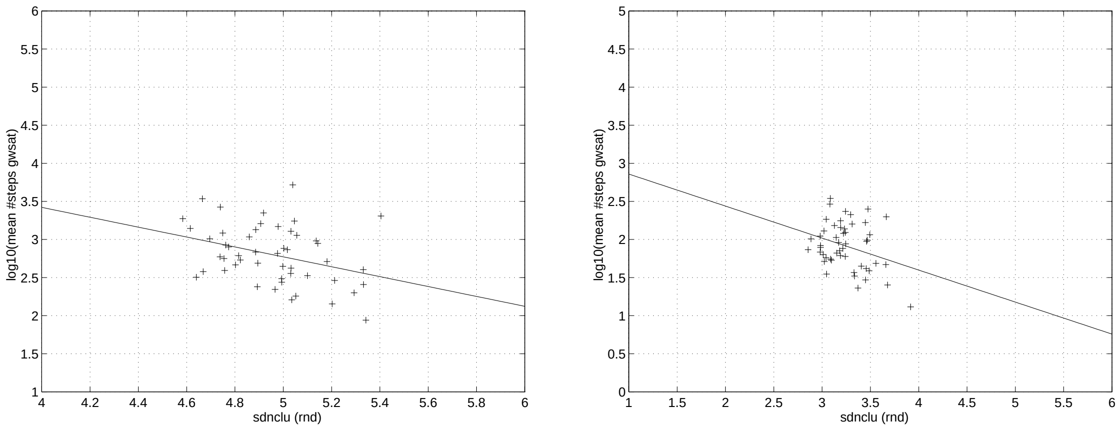

Figure 6.11: Correlation between *sdnclu* values (horizontal) and mean local search cost (vertical) for reduced test-sets `uf100-430 (50)` (*left*) and `uf20-91 (50)` (*right*).

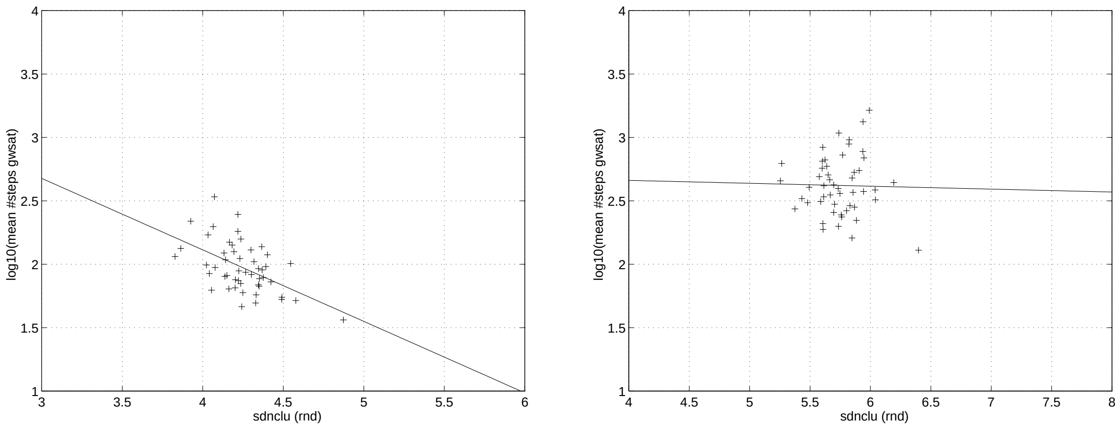

Figure 6.12: Correlation between *sdnclu* values (horizontal) and mean local search cost (vertical) for reduced test-sets `uf50-163 (50)` (*left*) and `uf50-273 (50)` (*right*).

instance	variables	clauses	solutions	avg *lsc*	*sdnclu*
flat30-60/n19	90	300	432	1,775.8	205.84
uf90-388/med	90	388	404	2,492.1	45.00
flat50-115/n4	150	545	5,892	4,539.2	421.20
uf150-645/med	150	645	5,884	10,231.9	70.05

Table 6.4: Comparison of the average local search cost and *sdnclu* values for Graph Colouring and Random-3-SAT instances.

result is analogous to our observations concerning the correlation between the number of solutions and average solution cost (see Section 6.2). Therefore, this result is consistent with the explanation for the *sdnclu* / solution cost correlation proposed above. The same holds with respect to varying problem size; here, we observe that the correlation gets stronger with increasing problem size. This is particularly interesting as, despite the huge differences in search space size between $n = 20$ and $n = 100$, our observations were obtained with a fixed sample size per instance. This is an indication that the observed correlations can be established for quite small relative sample sizes. At the same time, increasing the sample size for small problems does not significantly improve the observed correlation.

Graph Colouring Next, we applied the same analysis to problem instances from the Graph Colouring domain. Because of computational limitations we had to restrict the investigation to individual instances for different problem sizes. Our analysis is based on comparing problem instances with identical search space size (number of variables) and, to abstract from the dominant influence of this feature, a closely matching number of solutions. To achieve this, we newly generated two hard Random-3-SAT test-sets with 90 variables / 338 clauses and 150 variables / 645 clauses, consisting of 100 satisfiable instances (unforced, filtered generation) each. From these test-sets, we picked the median instances w.r.t. average local search cost for GWSAT (approx. optimal noise) and compared them with the instances from the `flat30-60` and `flat50-115` Graph Colouring test-sets which had the closest matching number of solutions, resp.

For these problem pairs, we compared the average local search cost for GWSAT (approx. optimal noise) and the *sdnclu* value. The results of this analysis are reported in Table 6.4; avg *lsc* refers to the average number of steps per solution when using GWSAT with approximately optimal noise settings. First, we observe that the more structured Graph Colouring instances tend to be easier to solve than corresponding Random-3-SAT instances with identical search space size and similar number of solutions. This effect apparently becomes more prominent with increasing problem size. Furthermore, the *sdnclu* values for the Graph Colouring instances are significantly higher than those for the Random-3-SAT instances. This confirms our initial intuition that the standard deviation of the objective function values should be negatively correlated with the hardness of an instance. Apparently, this effect, which we could not clearly detect within hard Random-3-SAT test-sets, can be observed when comparing structured and random instances.

instance	variables	clauses	solutions	avg *lsc*	*sdnclu*
bwp/anomaly	48	261	1	944.2	166.57
uf48-210/n49	48	210	1	1,509.5	24.84
bwp/medium	116	953	2	1,787.9	1,157.83
uf116-499/n92	116	499	16	12,484.5	57.31

Table 6.5: Comparison of the average local search cost and *sdnclu* values for Blocks World Planning and Random-3-SAT instances.

instance	variables	clauses	solutions	avg *lsc*	*sdnclu*
ais6	61	581	24	2,766.42	84.12
uf61-263/n75	61	263	24	1,695.37	28.98
ais8	113	1,520	40	64,167.11	3,711.34
uf113-486/n17:	113	486	40	58,267.11	51.74

Table 6.6: Comparison of the average local search cost and *sdnclu* values for All-Interval-Series and Random-3-SAT instances.

Blocks World Planning Applying the same methodology to small instances from the Blocks World Planning domain essentially confirms the observations from the previous section. The results are reported in Table 6.5; just as for the graph colouring domain, the Blocks World Planning instances are significantly more difficult for SLS algorithms than Random-3-SAT instances with identical search space size and a similar number of solutions. Again, with growing problem size, the difference in local search cost becomes more pronounced. At the same time, when compared with the corresponding Random-3-SAT instances, the Blocks World Planning instances show significantly higher average *sdnclu* values which also appear to grow faster with increasing problem size.

Note that because the Blocks World Planning instances have only one resp. two solutions, when taking into account the correlation between the number of solutions and average local search cost for Random-3-SAT, the corresponding Random-3-SAT instances are amongst the hardest of the respective test-sets. Therefore, we could not extend our analysis to the larger Blocks World Planning instances, because the corresponding Random-3-SAT problems are beyond the reach of even the most powerful SLS algorithms, given our computational resources and the type of analysis performed.

All-Interval-Series Performing the same analysis for the two smaller All-Interval-Series instances shows that for those, also much higher *sdnclu* values than for corresponding Random-3-SAT instances can be observed (*cf.* Table 6.6). However, the All-Interval-Series instances are more difficult to solve. This is not consistent with our observations for the instances from the Graph Colouring and Blocks World Planning domains and indicates that there are other factors which render the All-Interval-Series instances so hard for stochastic local search.

In summary, in this section we found some evidence that the standard deviation of the objective function (*sdnclu*), which intuitively corresponds to the ruggedness of the search space, is much lower for Random-3-SAT than for any of the more structured problem instances. The *sdnclu* value might also be correlated to the average local search cost for GWSAT. However, this correlation is somewhat noisy and does not account for the differences in hardness between Random-3-SAT instances with an identical or similar number of solutions or the relative hardness of All-Interval-Series instances when compared to instances from other structured problem domains. On the other hand, for Graph Colouring and Blocks World Planning instances, which are significantly easier to solve than Random-3-SAT instances with identical search space size and an almost identical number of solutions, we consistently observed significantly higher *sdnclu* values.

6.5 State Types and their Distribution

To further refine the analysis, in this section we classify search space states (or positions) according to the topology of their local neighbourhood and analyse the distributions of these state types. A similar approach has been followed in [Yok97]; it is based on the obvious fact that SLS algorithms are mainly guided by the local neighbourhood of the search space states along their trajectories. We classify the *state types* in the following way:

Definition 6.1 (Search Space State Types)

Let S be a search space, $N \subseteq S \times S$ a neighbourhood relation, and $f : S \mapsto \mathbb{N}$ an objective function on S. For a search space state $s \in S$, we define the following functions which determine the number of upwards, sidewards, and downward steps from s:

$$\begin{aligned} \mathit{upw}(s) &:= \#\{s' \in N(s) \mid f(s') > f(s)\} \\ \mathit{sidew}(s) &:= \#\{s' \in N(s) \mid f(s') = f(s)\} \\ \mathit{downw}(s) &:= \#\{s' \in N(s) \mid f(s') < f(s)\} \end{aligned}$$

Based on these functions, we define the following state-types:

$$\begin{aligned} \mathit{SLMIN}(s) &:\Leftrightarrow \mathit{downw}(s) = \mathit{sidew}(s) = 0 \\ \mathit{LMIN}(s) &:\Leftrightarrow \mathit{downw}(s) = 0 \wedge \mathit{sidew}(s) > 0 \wedge \mathit{upw}(s) > 0 \\ \mathit{IPLAT}(s) &:\Leftrightarrow \mathit{downw}(s) = \mathit{upw}(s) = 0 \\ \mathit{BPLAT}(s) &:\Leftrightarrow \mathit{downw}(s) > 0 \wedge \mathit{sidew}(s) > 0 \wedge \mathit{upw}(s) > 0 \\ \mathit{SLOPE}(s) &:\Leftrightarrow \mathit{downw}(s) > 0 \wedge \mathit{sidew}(s) = 0 \wedge \mathit{upw}(s) > 0 \\ \mathit{LMAX}(s) &:\Leftrightarrow \mathit{downw}(s) > 0 \wedge \mathit{sidew}(s) > 0 \wedge \mathit{upw}(s) = 0 \\ \mathit{SLMAX}(s) &:\Leftrightarrow \mathit{sidew}(s) = \mathit{upw}(s) = 0 \end{aligned}$$

□

The intuition behind these state types is that SLMIN and SLMAX states are strict local minima and strict local maxima, resp., LMIN and LMAX are (non-strict) local minima and local maxima which may occur on plateau regions; IPLAT states occur in the interior of plateaus, BPLAT characterises plateau border states, and SLOPE states correspond to sloping regions of the search space, where any direct step either increases or decreases the value of the objective function.

Obviously, for any given search space S, neighbourhood relation N, and objective function f, the classes of search space states induced by these predicates form a complete partition of S, *i.e.*, every search space state falls into exactly one of these types. Note also that these types can be weakly ordered according to the restrictiveness of their defining predicates when assuming that defining equalities are more restrictive than inequalities; in this respect, SLMIN, SLMAX, and IPLAT are most restricted, followed by LMIN, LMAX, and SLOPE, while BPLAT is least restricted. This ordering can be further refined by additionally assuming that conditions on the number of sideward steps are more restrictive than those on the number upwards or downward steps; the underlying intuition is that in the latter case are many options (i.e., objective function values), while for sideward steps, the objective function value is fixed. Thus, type SLOPE is more constrained than LMIN and LMAX, and IPLAT is more restrictive than SLMIN and SLMAX. For random search space structures, we would therefore expect a distribution of the state types according to this ordering, *i.e.*, the more constrained a state type is, the more seldomly it should occur.

However, it is not clear whether and to what extent the search spaces induced by the SAT instances from our benchmark suite, assuming the standard GSAT neighbourhood relation and objective function, exhibit random structure. Even for the Random-3-SAT instance distributions considered here, due to the restricted number of clauses (compared to the search space size, the number of clauses is logarithmic), the search space structure cannot expected to be truly random.[2] Therefore, to get a clearer picture of how local features of the search space topology affect SLS behaviour, we take an empirical approach to investigating state type distributions.

Before we report the results of our empirical analysis of state type distributions, we define the important notion of a *plateau region.*

Definition 6.2 (Plateau Region)

Let S be a search space, $N : S \mapsto S$ a neighbourhood relation, and $f : S \mapsto \mathbb{N}$ an objective function on S. A set $S' \subseteq S$ of search space states is a region, if it is connected by N, *i.e.*, for each pair of states $s', s'' \in S'$, a connecting path $s' = s_0, s_1, \ldots, s_k = s''$ exists, for which all s_i are in S' and all s_i, s_{i+1} are direct neighbours w.r.t. N. The *border* of a region S' is defined as the set of states within S' which have at least one neighbour state which does not belong to S'.

[2] One potential way for generating truly random search space structure is by means of random canonic normal form formulae; this approach would, however, require a number of clauses which is exponential larger than the number of variables.

instance	avg *lsc*	SLMIN	LMIN	BPLAT	IPLAT	SLOPE	LMAX	SLMAX
uf10-49/easy	5.48	0.10%	1.46%	87.21%	0%	10.35%	0.78%	0.10%
uf10-49/medium	24.59	0.10%	3.42%	83.50%	0%	11.82%	0.90%	0.20%
uf10-49/hard	244.15	0.20%	3.22%	86.52%	0%	7.51%	2.44%	0.10%
uf20-91/easy	13.05	0%	0.11%	99.27%	0%	0.59%	0.04%	$< 0.01\%$
uf20-91/medium	83.25	$< 0.01\%$	0.13%	99.40%	0%	0.31%	0.06%	$< 0.01\%$
uf20-91/hard	563.94	$< 0.01\%$	0.16%	99.23%	0%	0.56%	0.05%	$< 0.01\%$

Table 6.7: Distribution of search space state types for instances from Random-3-SAT test-sets `uf10-49` and `uf20-91`, based on complete exhaustive sampling.

Now, a plateau region is defined as a region S' for which

(i) all states $s' \in S'$ have the same objective function value, *i.e.*, $\exists l \in \mathbb{N} : \forall s' \in S' : f(s') = l$, and

(ii) no state s' from the border of S' has a neighbour state with the same level as s', *i.e.*, $\neg \exists s' \in S', s'' \in S - S' : N(s', s'') \wedge f(s') = f(s'')$. □

Obviously, the combined size of all plateau regions in S' is given by the sum of the number of IPLAT, BPLAT, LMIN, and LMAX states. In the following, we will use the term local minima region to refer to plateau regions containing LMIN states. Note that this definition covers the notions of both, local minima and benches, as defined in [FCS97]. As we will argue later (*cf.* Section 6.6), this approach makes sense in the light of our findings regarding the topology of local minima regions, considering that we are mainly concerned with SLS algorithms which can effectively escape local minima, such as GWSAT.

Random-3-SAT In a first experiment, we determined the complete state type distribution for the easy, medium, and hard instance of test-set `uf20-91` (the search space of these instances is small enough for exhaustive sampling). Table 6.7 shows the results of this analysis; *avg lsc* refers to the mean local search cost for GWSAT (measured in steps per solution, using approximately optimal noise, 100 tries per instance); the percentages for the state types specify the fraction of the sample matching this state type. Entries reading $< 0.01\%$ and $> 99.99\%$ indicate that the corresponding values are in the open intervals $(0\%, 0.01\%)$ and $(99.99\%, 100\%)$, respectively. The results are consistent with the ordering based on the restrictiveness of the state types discussed above: BPLAT states are predominant, followed by SLOPE, LMIN, and LMAX states; SLMIN and SLMAX states occur very rarely, and no IPLAT states were found for any of the instances analysed here or later. First of all, this suggests that the search spaces of Random-3-SAT instances show indeed structural properties (regarding state type distributions) similar to enirely random search spaces. But while random search spaces can be expected to contain equal numbers of LMIN and LMAX *resp.* SLMIN and SLMAX states, this is apparently not the case for Random-3-SAT search spaces. Here, LMIN states occur more more frequently than LMAX

instance	avg lsc	SLMIN	LMIN	BPLAT	IPLAT	SLOPE	LMAX	SLMAX
uf50-218/medium	615.25	0%	47.29%	52.71%	0%	$< 0.01\%$	0%	0%
uf100-430/medium	3,410.45	0%	43.89%	56.11%	0%	0%	0%	0%
uf150-645/medium	10,231.89	0%	41.95%	58.05%	0%	0%	0%	0%

Table 6.8: Distribution of search space state types for medium hardness problem instance from various Random-3-SAT test-sets, based on sampling along GWSAT trajectories (wp = 0.5).

states. This is most probably an effect of the CNF generation mechanism for Random-3-SAT instances; since each added three-literal CNF clause "lifts" the objective function for one eighth of the search space states by one, while the remaining states remain unaffected, local maxima are more likely to be eliminated when more and more clauses are added.

Our results also suggest that for Random-3-SAT at the phase transition, instance hardness might be correlated to the number of LMIN states, which conforms to the intuition that local minima states impede local search (*cf.* [Yok97]). But more interestingly, since we do not observe any IPLAT states, for algorithms using random walk (like GWSAT), there is always a direct escape route from any local minimum state. Thus, the common analogy relating local minima regions to basins is apparently not correct for Random-3-SAT instances from the phase transition, as each local minimum state has at least one neighbour which is not part of its plateau region.

We do not include analogous results for larger Random-3-SAT instances, since for these, exhaustive sampling is impractical (due to their huge search spaces) and random sampling finds almost exclusively BPLAT (and, very seldomly, SLOPE) states. Instead, in the following, we turn to sampling the state distributions from the trajectory of GWSAT using a noise setting of 0.5.[3] For obtaining these samples, we used 100 tries of the algorithm with a maximum of 1,000 steps each. If within one of these short tries a solution was found, this try was aborted as usual, to prevent solution states from being over-represented in the sample.[4] Therefore, the actual sample sizes vary; however, we made sure that each sample contained at least 50,000 search space states.

The results of this analysis, reported in Table 6.8, show clearly that GWSAT is strongly attracted to local minima states (as is to be expected, given its close relation to gradient descent). They also suggest that neither SLMIN, nor IPLAT or SLOPE states play a significant role in SLS behaviour on Random-3-SAT instances. This is consistent with our earlier observation for small instances, that these state types are extremely rare or do not occur at all.

[3] We chose this noise setting, because it is close to optimal for almost all the domains from our benchmark suite, and we prefer to use identical parameter settings for all problem domains to enhance comparability of our results across problem domains.

[4] GWSAT will, once it has found a solution, return to that solution over and over again with a very high probability, unless the search is restarted.

instance	avg lsc	SLMIN	LMIN	BPLAT	SLOPE
flat30/n19	1,775.8	0.44%	72.86%	26.58%	0.12%
uf90-388/medium	2,492.1	0%	49.41%	50.59%	0%
flat50/n4	4,539.2	1.30%	61.13%	37.10%	0.46%
uf150-645/medium	10,231.9	0%	41.95%	58.05%	0%
bwp/anomaly	944.2	18.68%	16.01%	54.94%	10.37%
uf48-210/n49	1,509.5	0%	46.01%	53.97%	0.02%
bwp/medium	1,787.9	6.56%	33.40%	56.31%	3.73%
uf116-499/n92	12,484.5	0%	44.77%	55.23%	0%
ais6	2,766.42	8.35%	40.78%	44.33%	6.55%
uf61-263/n75	1,695.37	0.03%	50.51%	49.46%	0%
ais8	64,147.11	4.73%	41.31%	48.35%	5.61%
uf113-486/n17	58,267.11	0%	42.14%	57.86%	0%

Table 6.9: Distribution of search space state types for instances from the Random-3-SAT (`uf-*`), Graph Colouring (`flat-*`), Blocks World Planning (`bwp-*`), and All-Interval-Series (`ais-*`) domains; based on sampling along GWSAT trajectories ($\texttt{wp} = 0.5$).

Graph Colouring, Blocks World Planning, and All-Interval-Series Applying an analogous analysis to the other problem domains from our benchmark suite, we find essentially the same situation (*cf.* Table 6.9). To abstract from the predominant influence of the number of solutions on local search behaviour, we used the same pairs of instances as in Section 6.4. IPLAT, MAX, and SLMAX states were not observed for any of the instances and are therefore not listed in the table. The results show several differences between the problem domains. First, while for Random-3-SAT, SLMIN states are almost never encountered, they regularly occur for the other problem domains. The same holds for SLOPE states. For the Blocks World Planning and All-Interval-Series instances, both observations can be intuitively explained by the fact that these have a significantly higher clauses per variable ratio than Random-3-SAT instances from the phase transition (*cf.* Chapter 4). Adding clauses tends to break up plateau regions [Yok97], which eventually converts some LMIN states into SLMIN states. Using a similar argument, it is easy to see that at the same time, BPLAT states are converted into SLOPE states. However, this explanation does not account for the corresponding observation regarding Graph Colouring instances, as these have a slightly lower clauses per variable ratio than Random-3-SAT at phase transition. As the All-Interval-Series instances are harder than corresponding Random-3-SAT instances, while Graph Colouring and Blocks World Planning instances are significantly easier, our results give no indication that the more frequent occurrence of SLMIN or SLOPE states for the SAT-encoded instances has any disadvantages, nor advantages w.r.t. SLS performance.

Second, the ratio between BPLAT and LMIN states is different for most of the domains: While for the Graph Colouring instances, more LMIN states are encountered, Blocks World Planning instances show more BPLAT states. For Random-3-SAT as well as All-Interval-Series, the ratio is more even, with a slight dominance on the side of the BPLAT states. However, since both, the Graph Colouring and the Blocks World Planning instances are significantly easier for GWSAT than the corresponding Random-3-SAT instances, this does not suggest any correlation between the hardness of a problem instance and the BPLAT to

LMIN ratio along GWSAT trajectories. Intuitively, this means that neither LMIN states nor BPLAT states induce significantly more difficulties for SLS algorithms.

Summarising these results, we found that IPLAT states do not occur for any of the instances analysed here, regardless of whether the search space is randomly sampled, or whether the samples are based on GWSAT trajectories. Furthermore, SLOPE as well as SLMIN states occur very seldomly for Random-3-SAT instances and only occasionally for SAT-encoded instances from any of the other domains when sampling GWSAT trajectories. Instead, the state type distributions thus obtained are dominated by BPLAT states, which mainly represent the gradient descent and plateau escape behaviour of GWSAT, and LMIN states, which characterise the plateau search phases of GWSAT's search. Somewhat surprisingly, our results regarding the ratio of BPLAT and LMIN states along GWSAT's trajectories when applied to instances from various problem domains, do not give clear evidence for the plateau search phase having a significantly larger or smaller impact on SLS performance than the gradient descent and plateau escape phases.

Thus, analysing state type distributions did not provide a simple feature that could account for the observed differences in SLS performance between various problem instances and domains. Consequently, we have to consider other, less localised features in our attempt to gain a better understanding of SLS behaviour. To this end, the fact that BLMIN states are never observed, provides an interesting starting point.

6.6 Local Minima Branching

Obviously, local minima regions play an important role for local search and it is a well-known fact that the number of local minima states and regions has a major impact on SLS performance [Yok97, FCS97]. But of course, not only the number of local minima states in the search space matters, but also their distribution. In earlier work, the metaphor of plateaus consisting of local minima was often used in the context of characterising SLS behaviour or search space structure. The number and size of these plateaus, as well as the number of their exits have been studied for Random-3-SAT [FCS97], while their geometry has not been addressed. This is somewhat surprising, since it is intuitively clear that, under the standard assumption that SLS algorithms spend most of their search effort in plateau regions, notions like the diameter, or the length of escape routes from a given plateau, should be extremely relevant. The topology of local minima regions is particularly interesting given the proven effectiveness of random walk techniques in the context of modern SLS algorithms, as, assuming a plateau-like, compact structure, random walk steps are only effective for escaping from a plateau if they are executed at its border. From Section 6.5 we know that interior plateau states are non-existent, or extremely rare, both for Random-3-SAT and SAT-encoded problems. Therefore, *every* state in a plateau is a border state — which partly explains the effectiveness of simple escape strategies.

In the following, we study the geometry of local minima regions for classes of random and

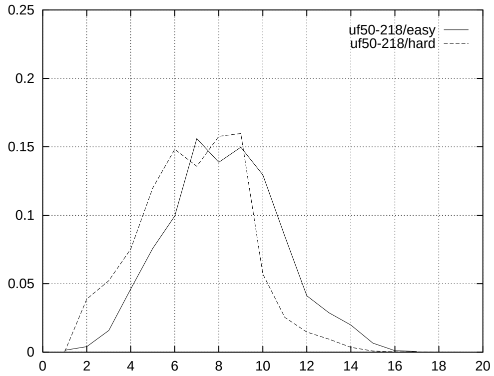

Figure 6.13: Empirical *blmin* probability distributions for easy and hard Random-3-SAT instances from `uf50-218` test-set.

structured SAT problems in more detail. Our analysis is based on measuring the branching of local minima states, where the *branching factor of a local minimum state* (*blmin*) is defined as the number of its direct neighbours with identical objective function value.[5] Thus, *blmin* is a positive integer between 0 and n (the number of variables for the given instance). It is quite obvious to assume that the branching, *i.e.*, the number of direct escapes from a given plateau state, should have an important impact on the effectiveness of escape mechanisms like random walk. But measuring *blmin* is not as simple as one might think; given the low relative proportion of local minima states when compared to the total size of the search space (*cf.* Section 6.5), random sampling is not an option. Therefore, we sample along trajectories of GWSAT (for approx. optimal noise); this has the additional effect of concentrating the analysis to the regions of the search space which are more relevant for SLS behaviour. If not indicated otherwise, our analyses are based on a sample obtained from 100 GWSAT tries with a maximal number of 1,000 steps per try. Thus, our samples usually contain 100,000 search space states; since in some of the tries, GWSAT finds a solution, the actual size of the samples might be slightly lower in some instances.[6]

Random-3-SAT To study the dependence of SLS performance on the branching of local minima, we investigate the correlation between the average solution cost for GWSAT and the average *blmin* values obtained by sampling GWSAT trajectories (as described above). First, we study some *blmin* distributions; later on, as in previous sections, we do a correlation analysis, computing correlation coefficients and performing linear regression. To keep the computation times within acceptable limits, we use the same reduced Random-3-SAT test-sets as in Section 6.4.

[5] Note that, for the SLS algorithms considered here, the number of direct neighbours is equal to the number of variables.

[6] To limit this effect, we had to keep the tries rather short.

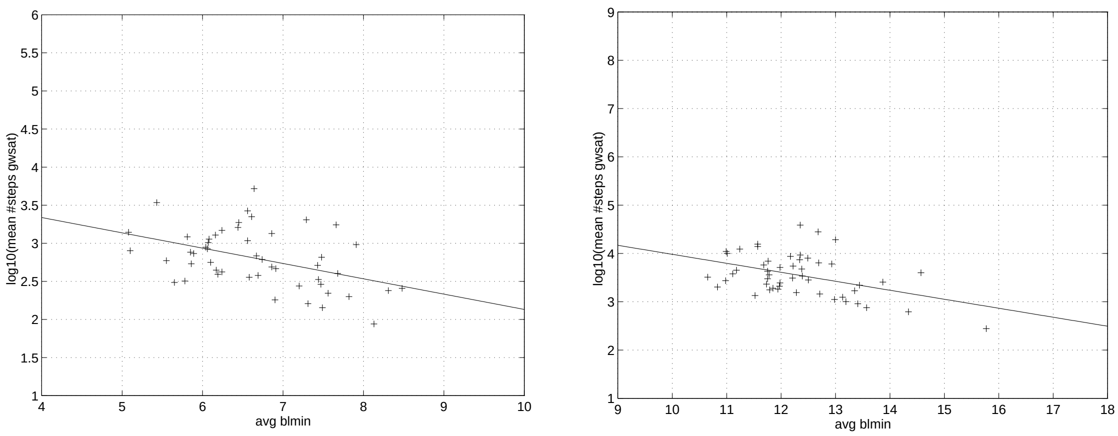

Figure 6.14: Correlation between average *blmin* and average local search cost for GWSAT (approx. optimal noise) for test-sets `uf50-218` *(left)* and `uf100-430` *(right)* (Random-3-SAT at phase transition).

Figure 6.13 shows two typical *blmin* distributions, here for the easy and hard instance from the `uf50-218` test-set. As can be seen from the plots, the average branching of LMIN states is quite low; this indicates that the overall structure of plateaus is very brittle, and for most problems, the LMIN states encountered have a large number of direct escapes when using random walk. Consequently, when encountering a local minimum region, an SLS algorithm's chance for escaping within a small number of random walk steps is rather high. Also, the branching seems to be even lower for the hard instance. This observation suggests a possible correlation between the average local minima branching and the hardness of a problem.

The results of an analysis of this correlation for the reduced Random-3-SAT test-sets with different problem sizes are shown in Figure 6.14. The scatter plots and *lms* fits indicate a negative correlation which gets stronger with increasing problem size. Table 6.10 shows the corresponding correlation coefficients and regression parameters; for the larger instances, also a greater inclination of the regression line is observed.

Generally, the observed correlation indicates that problems having a less branched plateau structure tend to be harder for GWSAT. This might seem a bit counter-intuitive, because using random walk, less branched plateau structures should be both easier to escape from and less difficult to search. But somehow this effect is either not present or dominated by another phenomenon. Since we know from Section 6.2 that the number of solutions is highly correlated with local search cost, one could hypothesise that, like the standard deviation of the objective function *sdnclu*, the average branching is somehow coupled with the number of solutions, which in turn dominates other influences on the mean local search cost.

Figure 6.15 (left) shows the correlation between the average local minima branching and the number of solutions for test-set `uf50-218`. The correlation coefficient of 0.68 indicates

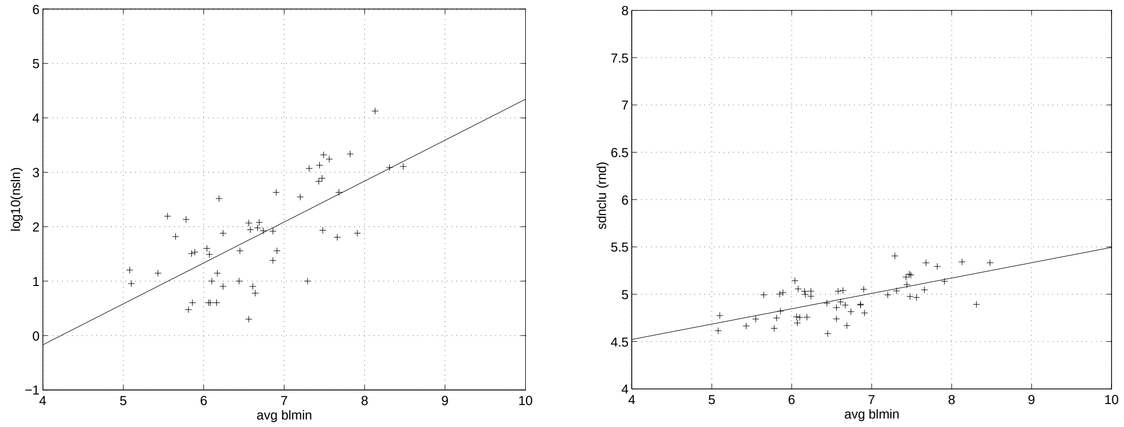

Figure 6.15: *Left:* Correlation between average *blmin* and number of solutions; *right:* same for average *blmin vs sdnclu*; all data for test-set `uf50-218` (Random-3-SAT at phase transition).

test-set	a	b	r
uf20-91/50	-0.06	2.13	-0.13
uf50-218/50	-0.20	4.14	-0.44
uf100-430/50	-0.19	5.84	-0.44

test-set	a	b	r
uf50-163/50	-0.14	3.48	-0.68
uf50-218/50	-0.20	4.14	-0.44
uf50-273/50	-0.14	3.30	-0.35

Table 6.10: Correlation between average *blmin* and the average local search cost, *left:* for different problem sizes at phase transition, *right:* across phase transition; a, b are the parameters of a *lms* linear fit using the function $ax + b$, r is the correlation coefficient.

that there is a slightly stronger correlation than the one observed between *sdnclu* and the number of solutions for the same test-set. As shown in Figure 6.15 (right), we also observe a rather strong correlation (correlation coefficient = 0.66) between the average local minima branching and *sdnclu* for the same test-set. This suggests the following explanation: For Random-3-SAT, a larger number of solutions tends to come along with clustering phenomena which, in turn, are characterised by both a larger branching of local minima as well as a higher standard deviation of the objective function values, *i.e.*, a more "rugged" search space topology. As a consequence, the average local minima branching can be used to estimate the number of solutions and therefore also the hardness of Random-3-SAT instances.

Determining the correlation between the average local minima branching and the mean local search cost for the 50 variable test-sets across the phase transition, we get similar results as the ones for *sdnclu* reported in Section 6.4. As can be seen from Figure 6.16 and the correlation data in Table 6.10, the correlation gets weaker with an increasing clauses per variable ratio. At the same time, for an increasing number of clauses, the average local minima branching decreases monotonically. This suggests that, for a fixed number of variables, the search space topology gets more and more rugged with an increasing number of clauses — an assumption which is confirmed by the results of [Yok97]. Furthermore,

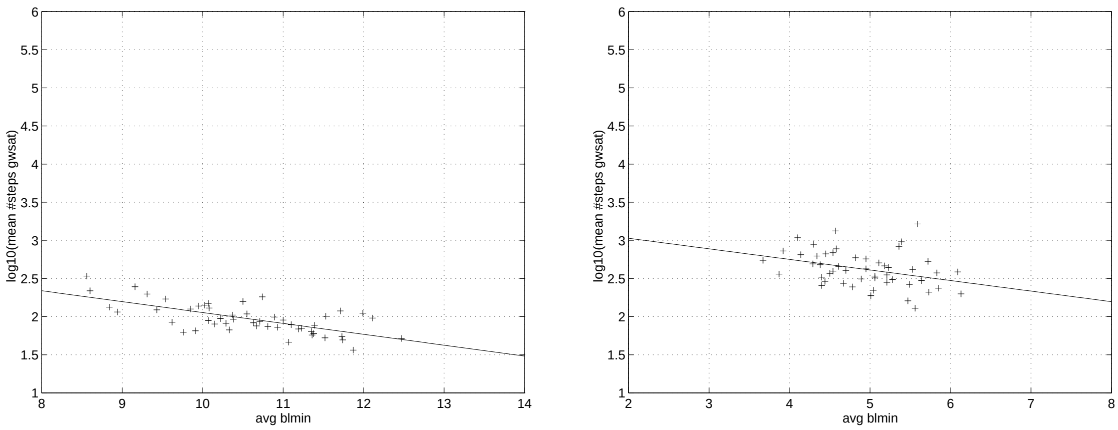

Figure 6.16: Correlation between average *blmin* and average local search cost (GWSAT, approx. optimal noise) for under- and overconstrained Random-3-SAT instances, test-sets `uf50-163` *(left)* and `uf50-273` *(right)*.

instance	variables	clauses	solutions	avg *lsc*	avg *blmin*
flat30-60/n19	90	300	432	1775.8	3.40
uf90-388/med	90	388	404	2492.1	10.35
flat50-115/n4	150	545	5892	4539.2	3.73
uf150-645/med	150	645	5884	10231.9	16.87

Table 6.11: Comparison of average local search cost and average local minima branching factors for Graph Colouring and Random-3-SAT instances.

at the phase transition the inclination of the regression line is maximal which indicates a stronger dependence of the average solution cost on the average local minima branching. Again, this observation is consistent with analogous observations for the dependence of the local search cost on the number of solutions and the explanation given above. Note that for high clauses per variable ratios, the average local minima branching seems to be a better estimator for the local search cost than *sdnclu* (*cf.* Tables 6.3 and 6.10).

Graph Colouring Next, we analysed the average local minima branching factor for instances from the Graph Colouring domain as compared to Random-3-SAT instances with the same search space size. The results are reported in Table 6.11; they indicate that Graph Colouring instances, which are significantly easier to solve for SLS algorithms, have a much lower average local minima branching factor than the hard Random-3-SAT instances. Also, with increasing problem size, the branching factor seems to grow faster for the Random-3-SAT instances than for the more structured Graph Colouring instances.

instance	variables	clauses	solutions	avg lsc	avg $blmin$
bwp/anomaly	48	261	1	944.2	2.69
uf48-210/n49	48	210	1	1,509.5	6.34
bwp/medium	116	953	2	1,787.9	3.44
uf116-499/n92	116	499	16	12,484.5	13.44

Table 6.12: Comparison of average local search cost and average local minima branching factors for Blocks World Planning and Random-3-SAT instances.

instance	variables	clauses	solutions	avg lsc	avg $blmin$
ais6	61	581	24	2,766.42	2.33
uf61-263/n75	61	263	24	1,695.37	7.19
ais8	113	1,520	40	64,147.11	2.83
uf113-486/n17	113	486	40	58,267.11	12.01

Table 6.13: Comparison of average local search cost and average local minima branching factors for All-Interval-Series and Random-3-SAT instances.

Blocks World Planning Performing a similar analysis for the two smallest Blocks World Planning instances, we obtained very similar results to those for the Graph Colouring domain (*cf.* Table 6.12). Clearly, the Blocks World Planning instances, which are significantly easier to solve for GWSAT, have a much lower average local minima branching factor than the hard Random-3-SAT instances. As for observed for the Graph Couloring instances, with increasing problem size, the branching factor seems to grow faster for Random-3-SAT than for the more structured Blocks World Planning domain.

All-Interval-Series Applying analogous methodology to the two smaller All-Interval-Series instances surprisingly gives a different result (*cf.* Table 6.13). Although compared to corresponding Random-3-SAT instances the local search cost (for WalkSAT, approx. optimal noise setting) is significantly higher, we observe much lower local minimum branching factors. This is not consistent with our observations for the other problem domains and strongly suggests that there are other factors which account for the relative hardness of instances from the All-Interval-Series domain.

Summing up the results of this section, by analysing the local minima branching factor $blmin$ along GWSAT trajectories we found evidence that the local minima structure is generally rather brittle than compact, *i.e.*, for the LMIN state visited by GWSAT, there is usually a relatively large number of direct escape routes when using random walk. In contrast, the often used plateau metaphor would suggest that the average $blmin$ value should be rather high, *i.e.*, close to n (the number of variables of the given problem instance), making it much more difficult for algorithms like GWSAT to escape from such plateaus. We also observed a positive correlation between the average $blmin$ value and the average local search cost for Graph Colouring and Blocks World Planning instances. This is consistent with our intuition that highly branched structures, which tend to have a higher potential for non-deterministic looping behaviour, are more difficult for SLS algorithms to search for solutions. However, for the All-Interval-Series domain, we observed very low branching factors despite the hardness of the instances. For Random-3-SAT, a negative correlation between the number of solutions and the average *blmin* value dominates our observed results, suggesting that the average local minima branching factor can be used for estimating the number of solutions for Random-3-SAT instances.

6.7 Related Work

The analysis of search space structure and its influence on SLS performance is a rather new area of research. There is, however, a number of studies which are related to the approaches followed here.

Section 6.2 is mainly based on methodology used by Clark *et al.* [CFG+96] for studying the correlation between the number of solutions and local search cost for CSP and SAT. However, reviewing the results reported in [CFG+96], we find that their results for SAT are considerably less conclusive than those for CSP: Mainly, the reported correlation is considerably weaker, and while for CSP, a minimal regression gradient can be found at the phase transition, an analogous phenomenon could not be observed for SAT. Our results reported in Section 6.2 eliminate these inconsistencies by overcoming several methodological weaknesses of their analysis. Nevertheless, their main result, the existence of a strong negative correlation between the number of solutions and average local search cost, is confirmed by our refined analysis.

Parkes [Par97] follows a different approach by analysing the clustering of solutions for Random-3-SAT instances. His results show that solutions usually occur in large clusters; however, from his study it remains unclear whether this is beneficial or detrimental for SLS performance. Our analysis of stratified Random-3-SAT test-sets suggests that, while clustering effects might have an influence on SLS performance, there have to be other, more dominant factors. These probably include clustering phenomena regarding non-solution plateaus, which could be related to Parkes' results concerning "failed clusters".

Our approach of analysing state type distributions and local minima structure is partly related to Yokoo's analysis of how constrainedness affects a simple hill-climbing algorithm's performance for Random-3-SAT and (un-encoded) Graph Colouring test-sets across the phase transition [Yok97]. Our approach of measuring search space state type distributions is related to his counting of local minima states. However, his analysis is solely based on the exhaustive analyis of search spaces for very small instances. He also observes that strict local minima are very rare, but does not investigate the frequency of occurrence for other state types. His main result, stating that with increasing constrainedness, the number of local minima states and the size of the local minima basins decrease, are complemented by our analysis presented here.

Probably the most detailed study of search space topology so far can be found in [FCS97, Fra97b]. Unlike our approach, this work focusses on more global features of search space structure, like the size and number of different types of plateau regions. Their analysis is mainly restricted to a number of random problem instance distributions, including hard Uniform Random-3-SAT, the search spaces of which are sampled using GSAT. While many aspects of their approach towards search space analysis appear to be promising, their results are very limited in their scope. They show that local minima regions as well as "benches", *i.e.*, plateau regions with neighbouring states that have a lower objective function value, are the characteristic structures for a large part of the search space and are highly variable in size. Their analysis addresses neither the correlation between these features and SLS performance, nor the topology of local minima regions, which is highly relevant in the context of SLS behaviour. It is also not clear how their results apply to SAT-encoded instances from other problem domains.

The importance of plateaus and local minima for SLS behaviour has been also stressed by Gent and Walsh [GW93b, GW93a], who established that GSAT spends most of its time searching plateaus. Many SLS methods are based on the notion that the ability of escaping from local minima leads to considerably improved performance. Among these are GSAT variants like GWSAT [SKC94], HSAT [GW93b], and WalkSAT; but also different approaches like Tabu Search [MSG97, Glo89], Simulated Annealing [KJV83, SKC93, Spe93], or the Breakout Method [Mor93]. Recently it could be shown that information on search space structure can be successfully used for automatically tuning search behaviour: Boyan's and Moore's STAGE approach [BM98] improves local search behaviour based on features of the underlying search spaces which are measured during the search.

6.8 Conclusions

In this chapter, we investigated various aspects of search space structure and studied their correlation to the search cost for SLS algorithms. Analysing the number of solutions per instance and its correlation to average local search cost for GWSAT across various Random-3-SAT test-sets, we confirmed earlier results for basic GSAT by observing a strong negative correlation. The strength of this correlation, however, decreases monotonically for growing constrainedness, indicating that for over-constrained problem instances, there is considerably more variance in local search cost than for critically or under-constrained instances with a similar or identical number of solutions. At the same time, we could show that the influence of the number of solutions is maximal at the phase transition region.

To abstract from the strong impact of the number of solutions on local search cost, in our further analysis, we used mainly sets or pairs of instances with a similar or identical number of solutions. Analysing hardness distributions for such subsets (strata) of Random-3-SAT, we found that the local search cost is still extremely variable within these strata. From an analysis of a single-solution stratum we found evidence that the clustering of solutions, as studied in [Par97], seems to have no major effect on local search cost for Random-3-SAT.

Consequently, there have to be other features of search space structure which are responsible for the large hardness variation within the strata.

As possible candidates for such features, we have studied the standard deviation of the objective function (*sdnclu*) over the search space and the average local minima branching factor (avg *blmin*) along GWSAT trajectories. To our best knowledge, these two measures have not been studied before in the context of search space structure. Intuitively, large *sdnlcu* values should indicate a rugged search space structure for which stochastic local search approaches based on hill-climbing are more effective than for featureless, flat search spaces with many plateaus. This intuition could be empirically confirmed for the Graph Colouring and Blocks World Planning domains, while for Random-3-SAT test-sets and All-Interval-Series instances we did not observe this correlation.

Since we know from previous work, that the number of local minima states has an impact on local search cost [Yok97], we extended these results by studying the structure of local minima regions. Surprisingly, we found that local minima regions have a very brittle structure, such that for each local minimum state only a small number of its neighbours belong to the same plateau region. Thus, the plateau regions resemble multi-dimensional systems of narrow canyons rather than compact basins. Consequently, using techniques like random walk, it should be extremely easy to directly escape from plateau regions with only a small number of escape steps.

For Graph Colouring and Blocks World Planning instances, we observed a positive correlation between the average *blmin* value and the average local search cost. This conforms to the intuition that highly branched structures are more difficult for SLS algorithms, because they have a higher potential for non-deterministic looping behaviour and a smaller number of escape routes per local minima state. For the Random-3-SAT and All-Interval-Series domains, our observations did not confirm this intuition. This indicates the existence of other search space features, which have a significant influence on SLS performance.

In summary, we identified and analysed a number of search space features influencing the hardness of instances for local search. Some of these features, like the average local minima branching, appear to be considerably different between Random-3-SAT and SAT-encoded instances from other domains. However, our results cannot explain all the observed differences in instance hardness within and across problem domains. Consequently, there have to be other, yet unknown factors which affect local search performance. In this sense, our analysis of search space structure is far from being complete, but extends the previous state of knowledge with new methodological approaches and novel insights.