FITNESS LANDSCAPE ANALYSIS OF A CLASS OF NP-HARD PROBLEMS

by

KHULOOD ALYAHYA

A thesis submitted to
The University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
The University of Birmingham
June 2016

Abstract

A number of fitness landscape properties of randomly generated instances of a class of NP-hard combinatorial optimisation problems are empirically studied in this research. We believe that the studied properties give insight into the structure of the problem landscape and can be representative of the problem difficulty, in particular with respect to local search algorithms. The properties include: types of search position, number of local and global optima and plateaux, quality of optima and plateaux, basin size and its correlation with fitness, time to local optima, cost of finding the global solution, and the quality of optima obtained with a fixed budget search. Our work focuses on studying how these properties vary with different values of problem parameters. We also compare these properties across different landscapes that were induced by different neighbourhood operators or different penalty functions of the following problems: the number partitioning problem, the binary knapsack problem, and the quadratic binary knapsack problem. Unlike existing studies of these problems, we study instances generated at random from various distributions. We found a general trend where in all the three problems, some of their landscape features were found to vary between the different distributions. We captured this variation by a single, easy to calculate, parameter and we showed that it has a potentially useful application in guiding the choice of the neighbourhood operator of local search heuristics.

ACKNOWLEDGEMENTS

I would like to thank everyone who directly or indirectly helped me during my PhD studies. I thank my advisor professor Jonathan Rowe for providing me with guidance and support and my RSMG members: professor Xin Yao and Dr. John Bullinaria for their constructive comments and questions. I thank all of my friends and family who supported me during this period, with special gratitude to my father.

CONTENTS

NOMENCLATURE

Δ	Density of the profit matrix in Quadratic 0-1 Knapsack
$\lvert \cdot \rvert$	Cardinality of a set
β	Number of distinct optima seen
β_1	Number of optima seen once
$\hat{p}$	Sample proportion
$\hat{v}^{JK}$	Jackknife estimation of number of optima
$\hat{v}^{SRS}$	Simple random sampling estimation of number of optima
λ	Knapsack capacity constraint coefficient
$\langle p \rangle^{\mathrm{NPP}}$	Theoretical average strict local optima proportion of NPP with uniform weights
$\mathrm{CI}_{\mathrm{AC}}$	*Agresti-Coull* confidence interval
CI_{s}	Standard confidence interval
$\tilde{p}$	*Agresti-Coull* sample proportion
C	Knapsack capacity
CV	Coefficient of variation
e	The natural logarithm base
h	Hamming distance

k	Control parameter of the phase transition in Number Partitioning and Subset Sum ($k = log_2 M/n$)
M	Range of the weights
n	Problem size
$N(x)$	Neighbours set of configuration x
p	Population proportion
p_{ij}	Profit when both item i and j are selected
p_i	Item i profit
S	Population size
s	Sample size to estimate the number of optima
s_0	Infinite population sample size
s_1	Finite population sample size
v	Number of optima
w_i	Item i weight
X	Set of all configurations
x	Configuration in the search space
Y	Number of sampled optima using simple random sampling
$z_{\alpha/2}$	z-score for $(1-\alpha)100\%$ confidence level
H1+2	1+2-bit flip neighbourhood operator
H1	1-bit flip neighbourhood operator

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Meta-heuristics are general approximate optimisation techniques. They have been successfully applied to find good approximate solutions to many hard optimisation problems. There is a vast number of available meta-heuristics techniques with a vast number of different variations to each technique. This number is growing every year with new proposed techniques and variations. The most common way to evaluate the performance of a new proposed algorithm is through "up-the-wall" game [13], where an extensive number of experiments is performed on benchmark problems and the results are then compared against the performance of other existent algorithms. The goal of the game is to claim further up the wall than the others. Hooker [49] refers to this approach as "competitive testing" and he argues that this approach is harmful for research as it gives little or no insight into why or how the algorithm under test is better or worse than the others. This line of research is increasingly being criticised for not advancing our knowledge and understanding of the algorithm behaviour [108, 109, 1, 8], which in turns contributes to making the selection of the most suitable algorithm for a given optimisation problem even more challenging. Finding a single algorithm that has the best performance on every optimisation problem is ruled out by the no free lunch theorem [134], which states that an algorithm that has superior performance over the others for all the possible optimisation problems does not exist. Therefore, there is a need for a deep understanding of the problem features and how that relates to the algorithm behaviour. This kind of research is encouraged by an ongoing initiative to direct the research in the field of meta-heuristics towards this direction [108, 1, 109, 49].

Fitness landscape analysis provides a great analytical framework to address the need of understanding the relation between problem features and algorithm performance. An extensive amount of research has been carried out in the past two decades, where new approaches were proposed to characterise the problem fitness landscape and relate that to its difficulty with respect to certain meta-heuristics. The aim is that the gained insight can help in the selection process of the best suited algorithm/configurations, or the design of new more suitable techniques. Several predictive measures of problem difficulty have been proposed over the years. According to [55] these measures fall into two types, descriptive and analytical. Descriptive methods try to classify a problem according to some properties of the landscape, while analytical methods define a measurable output to classify problems (e.g. summary statistic). However, and despite the extensive work being done in this area, until now none of the proposed measures seem to be satisfactory [46]. Sometimes a measure cannot even accurately predict the difficulty of all instances of a single problem class as [55] points out. For most of the measures, computing the exact value of the measure is exponential with the size of the problems since it requires exhaustive search of the whole search space [46, 55]. On the other hand, approximate measures, which are computed on a sample of the search space, do not always give the correct indication of the problem difficulty. Even exact computations of many difficulty measures can be very misleading sometimes [55]. A major problem with the analytical measures is that they try to compress large and complex information into a single number and a lot of information gets lost in that process. But most importantly, a general measure that accurately predicts the difficulty of all the problems and that can be computed in polynomial-time cannot exist unless P = NP, as rigorously proven by He et al. [46]. Therefore, the emphasis should be on designing difficulty measures for a broad class of problems on which the predictability of the measure is consistent. Studying more than one measure or feature can help in getting a broader perspective and increase the chances of capturing various aspects of the problem difficulty.

In this thesis, we provide a fitness landscape analysis of three NP-hard problems that fall into a class of NP-hard binary packing problems related to the 0-1 knapsack problem, namely: the number partitioning problem (NPP), the binary knapsack problem (0-1KP), and the quadratic binary knapsack problem (0-1QKP). We study a set of landscape properties with respect to local search. We believe these properties to be representative of the problems difficulties and to give an insight into the structure of the problem landscape. The set includes: types of search position, number of local and global optima and plateaux, quality of optima and plateaux, basin size and its correlation with fitness, time to local optima, cost of finding the global solution, and quality of optima obtained with a fixed budget search. Our work focuses on studying how these properties vary with different values of problem parameters, where one of these parameter being phase transition control parameter. We also compare these properties across different landscapes that were induced by different neighbourhood operators or different penalty functions. Most of the existing studies of these problems only consider instances where the weights are drawn at random from a uniform distribution. We study instances generated by drawing the weights at random from various distributions. In all of the three problems, we found that some of the landscape features vary greatly between some of the distributions. We proposed and demonstrated that the use of a single parameter that is easy to calculate and does not require the knowledge of the underlying distribution of the weights, namely the coefficient of variation of the weights CV, captures most of this variability. We also show that the CV has a potentially useful application in guiding the choice of the neighbourhood operator of local search heuristics. The problem of estimating the number of local optima often arises during fitness landscape analysis. We evaluate the performance of estimating the number of local optima by estimating their proportion using simple random sampling and discuss the choice of different confidence intervals in this thesis.

CHAPTER 2

LITERATURE SURVEY

2.1 Fitness Landscape Analysis

Fitness landscape is a powerful metaphor which facilitates the visualisation of the relationship between configuration space and fitness values. The concept of fitness landscape was first introduced by biologist Sewall Wright in 1932 [135]. Visualisation of the fitness landscape offers an intuitive way of understanding a problem (though caution is required in high dimensional spaces as it can be misleading [101]). Formally, the fitness landscape is a triple (X, N, f), where X is the search space, N is the neighbourhood operator function, and f is the objective function $f : X \rightarrow R$ [107]. The neighbourhood operator represents how the solutions are connected in the landscape, and how one can move from one solution to another. Analysing the fitness landscape to identify the landscape features that are related to problem difficulty has been an active area of research over the last two decades. Stadler [104] notes that the fitness landscape can be studied in two ways. The first is from a static point of view, in which the focus is to analyse the geometric features of the landscape. The second is from a dynamic point of view, in which the focus is to study the dynamic features of the search method, for instance an evolving population searching the given landscape. The challenge lies in combining the two methods together, and determining how the geometric features of the landscape influence the dynamic behaviour of the search method. One of the first studied features of the fitness landscape is deceptiveness. Goldberg [39] was the first to introduce the concept of deceptive problems where the global structure of the problem leads the algorithm away from the global optima. Deceptiveness can make a problem difficult to optimise and could render the search algorithm less efficient than random search [127]. However, that does not apply to all deceptive problems. Wilson [132] proposed a deceptive function that is easy to optimise for a genetic algorithm. Also, Vose and Wright [123] have shown a non-deceptive function, yet it is difficult to optimise for a genetic algorithm.

The rest of this section gives an overview of some of the concepts, predictive measures, and sampling methods that have been proposed over the years to study the fitness landscape and characterise the problem difficulty. Note that some of the predictive measures study the fitness function only and majority of them study the fitness landscape. For a more detailed overview the reader is advised to look at [85, 124, 96].

2.1.1 Modality

When trying to assess the difficulty of a problem, a feature of the fitness landscape that might seem to be the most obvious, is the number of local optima. One would think that a unimodal problem with single optima would be easier to search than a multimodal problem, but this is not always true. Horn and Goldberg [53] have shown a unimodal problem that is hard to search and an extremely maximally multimodal problem, where half of the points in the search space are local optima, yet it is easy to search. Therefore, relying on the number of local optima alone as an indicator of problem hardness is neither sufficient nor necessary [58]. Nevertheless, local optima and their number could still provide valuable information about the landscape. Examples of the information that can be studied about the local optima, alongside their frequency, are: the distribution of the optima over the search space, examining if fitter ones cluster together, examining if they form a valley in the search space (in which case, search algorithms such as tabu search could be more suitable), and the difference in quality between the local and the global optimum. A method to examine the distribution of the optima over the landscape is proposed in [110], where they study the average distance between local optima and combined that with the optima entropy (as a measure of their diversity) to estimate how they are distributed over the landscape. The local optima of many combinatorial optimisation problems tend to cluster together and form “big valley” (or “central massif” if maximising). This has been observed in NK family of landscapes, travelling salesman problem, graph bi-partitioning and flow-shop scheduling [125].

2.1.2 Basins of attractions and barriers

Another important feature of an optimum is its basin of attraction, unlike modality, basins of attraction could provide more information about the landscape. For an optimum $x^* \in X$, its basin of attraction $B(x^*)$ is the set of points that leads to it after applying local search to them, $B(x^*) = \{x \in X \mid \text{localsearch}(x) = x^*\}$ [85, 78]. Pitzer et al. [86] note that a basin of attraction can be classified as strong or weak according to the set of points that belongs to it. A strong basin of attraction is a basin on which all the points in the basin converge exclusively to a certain optimum. A weak basin of attraction is a basin, which has some points that could converge to another optimum depending on the algorithm or operators used. The search space X can be seen as the union of all the weak basins $X = \cup_{i \epsilon X^*} B(i)$, where X^* is the set of all the optima. Basins of attraction have several interesting properties such as size, shape, stability and distribution, which can be studied to drive difficulty measures [37]. For example if the basin size of the global optimum is relatively small then the problem can be difficult. A reverse hill-climbing algorithm to determine recursively the basin of attraction of a given point in the landscape is developed in [56]. To examine the quality of the intermediate points between local optima, [69] proposed a directed walk between two optima and then compared the fitness trajectory of this walk to the fitness trajectory of a random walk starting from the same optimum. The underlying idea behind this directed walk, is that it could give an indication whether a crossover between the two optima is useful or not.

The concept of barriers is borrowed from physics where it describes the energy barriers between metastable states [94, 85]. In fitness landscapes and assuming minimization, a fitness barrier is defined as the minimum fitness value required to go from one optimum to another through any random path [94]. Fitness barriers between the the set of local minima form an ultra-metric distance measure [94, 85] and it can be used to build a barrier tree that represents a hierarchy of local optima. This ultra-metric distance measure can be used to indicate the difficulty of moving between optima. However, it has some drawback [85]. First, it does not take into account how far away the two optima are, which is a factor that could affect the difficulty of moving between optima. Second, it does not take into account the fitness values of the optima. If the difference between the fitness values of the optima is large then moving from one to another might be easier than the other way around [85]. A proposed extension to make the method of barrier trees more useful in studying heuristic optimization algorithms is presented in [44].

2.1.3 Ruggedness

Another feature of fitness landscapes that is related to local optima is ruggedness (rugged landscapes assumed to have many local optima). Ruggedness is one of the first proposed methods to measure problem hardness. The first measures of ruggedness was introduced by Weinberger [126], where he defines the autocorrelation function and the correlation length. Autocorrelation function measures the correlation of fitness values of neighbouring steps in a random walk [52]. An important assumption for this measure to accurately characterise the correlation structure of the entire landscape, is that the fitness landscape should be statistically isotropic. This means that the random walk is “representative” of the fitness landscape, regardless of the starting point [126]. The correlation length can be defined as the maximal distance between two points in the walk where the correlation between them is still statistically significant [52]. Smoother landscapes have larger correlation lengths. Autocorrelation and correlation length have been used successfully as a measure of problem hardness for some problem classes [71, 21]. For other problems they could not predict the problem difficulty [66]. In problems such as the needle in a haystack where the whole search space is flat except for a single point, correlation length for such problems, indicates that the problem is easy while it is in fact very difficult [56]. Also, auto-correlation fails to reflect the problem difficulty or detect the changes in the problem difficulty over phase transitions as shown in [116].

2.1.4 Neutrality

Neutrality is another feature of fitness landscapes, which refers to the amount of neutral areas or plateaux in the landscape [32]. The problem with neutrality is that it does not provide any guidance for search heuristics which could lead the search process to wander randomly in the neutral areas for a long time without much progress. Nevertheless, exploring neutral areas could be useful sometimes by allowing the search process to reach better quality solutions and escape a nearly local optima [101]. A neutral walk is introduced in [93], which is a variation of a random walk to explore neutral areas. A neutral walk starts at a random point and continuously moves to neighbouring points with equal fitness such that the distance to the starting point is maximised. The maximum distance obtained from this walk could then be used as a measure of neutrality. There are interesting features to study about neutral areas in a given landscape such as the size of neutral areas, types of neutral areas, the maximum and average distances between two points in the neutral area [85].

2.1.5 Position types and their distributions

We have seen how local optima and neutrality can be useful for characterising landscapes. Another view that could help in gaining more insight about the structure of the landscape is to look at the distribution of different search position types in the search space. For a given point in the landscape, according to the topology and fitness of its direct neighbourhood, it can belong to seven different types of search position [51]. Collecting information about these types could be helpful in trying to understand the behaviour of local search algorithms in particular, since they are guided mainly by their direct neighbourhood [50].

2.1.6 Fitness distance correlation

Fitness distance correlation (FDC) is perhaps one of the most popular measures of problem hardness. It was proposed by Jones and Forrest [57] to measure the correlation between fitness values and the distances to the global optimum. The main motivation behind this measure is that the relation between distances and fitness values can be an attribute of problem difficulty. Considering a maximisation problem, a large and positive correlation coefficient indicates a misleading problem, a correlation coefficient near to zero indicates a difficult problem, while a large and negative correlation coefficient indicates a straightforward problem [57]. An obvious drawback of this measure, is that it requires the knowledge of the global optimum. This could be alleviated by considering the best known solution since in many situations when applying meta-heuristics, the goal is to find a good enough solution. However, if the best known solution is far from the global optimum then this might result in an incorrect and misleading output [51]. FDC measure has been quite successful as a measure of difficulty in some problems and has given significant insights into the behaviour of some meta-heuristics [85, 51]. However, in some situations it fails to predict the problem difficulty [3, 76]. An example, is the function developed by [3] on which the fitness values and the distances to the global optimum are not correlated indicating that it is a difficult problem yet it is easy to solve. In addition, obtaining the correlation coefficient approximately might give inaccurate and misleading output. For instance, in the long path problem developed by [53], the correlation coefficient obtained from a sample of the search space suggests an easy problem while if it was computed exactly over the entire search space then the value of the correlation coefficient would indicate a difficult problem. In situations where the correlation coefficient does not capture the relation between distance and fitness accurately, a scatter plot of the distance and fitness could provide more information [57]. However, it has been shown that even the insights from the scatter plot can be misleading sometimes [85].

2.1.7 Epistasis

Epistasis is one of the earliest attempts to measure problem difficulty. It is computed based on the fitness function only [25]. It studies the interaction between the solution components in an attempt to measure the amount of non-linearity in the fitness function. One of the first attempts to quantify epistasis was made by [25] when he proposed epistasis variance. A high epistasis means that the variables depend on each other while low epistasis means that the variables are independent of each other. Problems with high epistasis are assumed to be hard to optimise. This method, however, is found to be difficult to interpret and works only for limited number of cases. In addition, It has been shown to have a basic flaw in [92]. Other related proposed measures are epistasis correlation by [75] and bit-wise epistasis measure by [30]. Although these methods were devised to measure the epistasis in a fitness function, [76] point out that both epistasis variance and epistasis correlation only measure the absence of epistasis. The concept of epistasis is difficult to quantify and all the previously proposed measures are difficult to compute and have major shortcomings [55, 76, 85, 92, 77].

2.1.8 Information analysis

A different method to study the structure and ruggedness of the fitness landscapes is the information analysis by [120]. It is inspired by the concept that the information content of a system can be used as a measure of how difficult it is to describe that system. The main idea is to use the amount of information needed to describe a random walk in the fitness landscape of a problem as a difficulty measure of the problem [85]. Difficult problems are assumed to require more information to describe a random walk in their landscape. This measure is derived from a sample of the search space and assumes that the landscape is statistically isotropic. A related idea is the concept of information landscapes [9, 10, 139].

2.1.9 Evolvability

Evolvability involves studying the dynamic properties of certain meta-heuristics searching the landscape (e.g. evolving population). It studies the chances of improving a certain solution by measuring the correlation of successive solutions [85]. The general concept is that a higher degree of evolvability indicates that the problem is easier for the meta-heuristics. Some of the methods that have been proposed to measure evolvability are: evolvability portraits [101], fitness cloud [20, 119, 87], and fitness-probability cloud [63].

2.1.10 Phase transitions and backbones

Phase transition from easy to hard regions occurs in many NP-complete problems [45, 73]. The most notable example of problems with phase transition is the SAT problem which has an easy-hard-easy phase transition (easy-hard phase transition in MAX-SAT, the optimisation version of the problem). In phase transition, an order parameter partitions the problem instances into subsets with different degrees of difficulty [45, 124]. The change in problem difficulty is often accompanied with changes in the structure of the landscape which demonstrates that instances belonging to the same class could have very different landscape structures [80]. However, within theses partitions, there is often a considerable amount of variability in instance difficulty that cannot be explained by the phase transition [124]. Backbones are the values of solution components that are common in all optimal solutions [124]. Slaney and Walsh [99] show that for some problems the backbone size and the problem difficulty are correlated. For instance, it has been observed that the backbone size of random MAX 3-SAT problem increases as the problem transitions from easy to hard phase [140]. By Identifying the problems backbone, the focus of the search methods could be directed to find the correct assignment for the backbone components. Defining the backbone of a problem, however, can be difficult as noted by Slaney and Walsh [99].

2.1.11 Spectral landscape analysis

Stadler [103] was the first to introduce the study of isotropic fitness landscapes using Fourier Analysis, suggesting another approach to analyse fitness landscapes by decomposing the fitness landscape of an arbitrary problem into superpositions of elementary landscape [85]. An elementary landscape is a special type of landscape that can be described by Grover's wave equation [41]. This method has been used extensively to analyse many popular problems [103, 94, 104, 107, 130, 18, 128, 129].

2.1.12 Network measures

Recent attempts to develop predictive models of problem difficulty utilised the fact that a fitness landscape can be represented as a network, and thus used complex network analysis tools to analyse the fitness landscape. Local optima networks (LONs) was the first method that employed complex network analysis tools to combinatorial optimisation problems [78]. The proposed model is adapted from the inherent networks of energy landscapes in physical-chemistry. LON is a descriptive method (i.e. it does not give a number or summary statistic as an output) that reduces the fitness landscape into a sub-landscape consisting of local optima only. The model was extended later to also include the plateaux in the landscapes [121]. The proposed method involves exhaustively extracting the local optima of small problem instances. After that, the LON is constructed such that each vertex represents a local optimum and an edge between two vertices can be defined in different ways to represent: that the two optima have adjacent basins, the transition probability between the basins of the two optima, or the chances of escaping a local optimum to another after a controlled mutation [122, 78, 24, 121, 17]. Another network based measure is called motif difficulty (MD), which is based on a property used to analyse the structural design of complex networks called Network motifs [62]. Network motifs are patterns or subgraphs that occurs significantly frequently in the network and can be used to differentiate between different networks. The underlying idea of the motif difficulty measure is that the problem would be easy if high fitness regions are near the global optima and difficult otherwise. The output of this measure is a single number that falls into the range of -1.0 (easiest) to 1.0 (most difficult). As with the previous measures, this measure was found to work on some problems but for other problems it fails to predict their difficulty, e.g. the needle in a haystack problem. Some of the limitations of this measure are: it requires the knowledge of the global optima and imposes restrictions on the neighbourhood operators.

2.1.13 Landscape walks

One of the ways to obtain a sample over the landscape is through landscape walks [85]. A walk on the landscape can be viewed as a time series of fitness values. It starts from a random configuration and continuously moves to a neighbour solution, keeping a record of their fitness values. There are different types of walks on the landscape, each of which explores the landscape in a different way obtaining different information and thus providing different perspectives of the landscape [85]. A random walk chooses any of the neighbours randomly. An adaptive walk always chooses a better neighbour (the meaning of better can be defined as any or best improving move). A reverse adaptive walk always chooses a worse neighbour (the meaning of worse can be defined as any or worst improving move). An uphill-downhill walk is an adaptive walk followed by a reverse adaptive walk. A neutral walk explores plateaux where one continuously choose a neighbour with the same fitness and try to increase the distance from the starting configuration.

2.2 Analysis of Problem Classes

Several studies have been carried out to analyse the fitness landscape of certain problem classes, some of which have linked the studied features to the performance of the search heuristics while others went a step further and exploited the gained insight of the landscape structure to design a better search heuristic or to configure and tune the algorithm parameters. For example, Watson et al. [125] have demonstrated experimentally that the mean distance between random local minima and the nearest optimal solution in job-shop scheduling problem (JSP) is highly correlated with the cost of finding optimal solutions. They developed a model of problem difficulty for tabu search in the JSP that measures the cost of locating suboptimal solutions and explains the difference between the difficulty of two classes of JSP, square and rectangular. They also noted that their model has some limitations. First, the accuracy of the model is inversely correlated with the problem difficulty. Second, the model is significantly less accurate for the structured non-random instances of JSP. Qasem and Prügel-Bennett [89] proposed a hybrid algorithm that out-performs the state-of-the-art algorithms for finding good solution for large MAX-SAT problem instances. Their algorithm exploits the fact that the solutions cluster in small areas in the hard phase of MAX-SAT. The algorithm uses a search operator that clustered good solutions reached by hill-climbers and then restarts the search from the centroid of each cluster. They argued that the good performance of the algorithm is due to the algorithm learning the large scale structure of the fitness landscape during the search using the population. Tayarani-Najaran and Prügel-Bennett [88] extend the previous work done by [89] and provide more details about the landscape properties and the analysis of the algorithm. Fabio et al. [23] studied the impact of the landscape features of an extension of the multi-objective NK-landscapes on the performance of a global and a local multi-objective search heuristics. Tayarani-Najaran and Prügel-Bennett present an extensive fitness landscape analysis for a number of NP-hard problems, namely: graph-colouring, travelling salesman, maximum satisfiability, and quadratic assignment problems [114, 115, 113, 88]. They also provide a comparative analysis of the the landscape features of these problems in [112]. Yoshizawa and S. Hashimoto [138] provide a short statistical analysis of the fitness landscape of the uncorrelated binary knapsack problem with three different approaches of handling the constraint and they proposed a new algorithm based on that analysis. Yao et al. [136] proposed an improved fast evolutionary programming based on the understanding of the underlying mechanism of two mutation operators in relation to the modality of the search space. Examples of other studies are: NK landscapes [122, 78, 121], quadratic assignment problem [24], flow-shop scheduling problem [22], the multidimensional knapsack problem [111], generalised assignment problems [15], the number partitioning problem [105, 100], and travelling salesman problem [106]. A comprehensive survey of instance difficulty of six combinatorial optimisation problems: assignment problems, travelling salesman, knapsack problems, bin-packing, graph colouring, and timetabling is presented in [102].

2.3 Summary

In this chapter we reviewed most of the features and measures that have been proposed over the last few decades in the field of fitness landscape analysis. Many of these techniques have been proposed to predict general problem hardness, however, none of them has been very successful in achieving that. As shown by He et al. [46] a general measure that accurately predicts the difficulty of all the problems and that can be computed in polynomial-time cannot exist unless P = NP. Therefore, we argue that these measures should be viewed instead as a part of a toolbox of techniques to broadly characterise problems. Müller and Sbalzarini [74] used fitness distance correlation (FDC) to characterise the CEC 2005 benchmark test suite and they argued that "fitness-distance analysis can only provide one out of several useful landscape descriptors that need to be combined in order to form discriminative 'landscape fingerprints'". Indeed, there is an already existing body of work in this direction, where a group of measures is studied to characterise the landscape of several problem classes as shown in section 2.2 of this chapter. This thesis continues this line of research and studies a subset of features to characterise the landscape of three combinatorial NP-hard problems: the number partitioning problem (NPP), the binary knapsack problem (0-1KP), and the quadratic binary knapsack problem (0-1QKP).

CHAPTER 3

METHODOLOGY AND ESTIMATING THE NUMBER OF OPTIMA

In this thesis, we empirically study a number of landscape properties of randomly generated instances of some NP-hard problems with respect to local search. The properties include: types of search position, number of local and global optima and plateaux, quality of optima and plateaux, basin size and its correlation with fitness, time to local optima, and cost of finding the global solution. We study how these properties change with different problem parameters (e.g. phase transition parameter). Also, given a problem instance, we compare these properties across different landscapes (e.g. landscape induced by different neighbourhood operator, or different penalty functions). Most of the results were obtained through exhaustive enumeration of the fitness landscape, which was only computationally feasible for small problem sizes $n < 30$. For larger problem instances, we used sampling methods to obtain an estimate of these properties. The sampling methods are discussed in section 3.3.

3.1 Definitions

The following definitions will be used throughout this thesis:

Search Space The search space X is the finite set of all the candidate solutions. The fitness functions of all the studied problems in this thesis are pseudo-Boolean functions, hence the search space size is 2^n.

Neighbourhood A neighbourhood is a mapping $N : X \rightarrow P(X)$, that associates each solution with a set of candidate solutions, called neighbours, which can be reached by applying the neighbourhood operator once. The set of neighbours of x is called $N(x)$, and $x \notin N(x)$. We consider two different neighbourhood operators: the Hamming 1 operator (*H1*) and the 1+2 Hamming operator (*H1+2*) . The neighbourhood of the *H1* operator is the set of points that are reached by 1-bit flip mutation of the current solution x, hence the neighbourhood size is $|N(x)| = n$. The neighbourhood of the *H1+2* operator includes the Hamming one neighbours in addition to the Hamming two neighbours of the current solution x, which can be reached by 2-bits flip mutation. The neighbourhood size for this operator is $|N(x)| = (n^2 + n)/2$.

Suppose we lay out the search space in circles around a configuration x, so that x is placed in the centre and the configurations that are h Hamming distance away from it lie on the circle of radius h (see figure 3.1 for an illustrative example when $n = 10$). For a configuration in the h-th circle, its *H1* neighbours will be spread out as follows: h of them will reside in the $h - 1$ circle, the rest $(n - h)$ will reside in the $h + 1$ circle. Its *H1+2* neighbours will be spread out over the $h - 2$, $h - 1$, h, $h + 1$, $h+2$ circles as follows: $h(h-1)/2$, h, $h(n-h)$, $n-h$, $(n-h)(n-h-1)/2$ respectively.

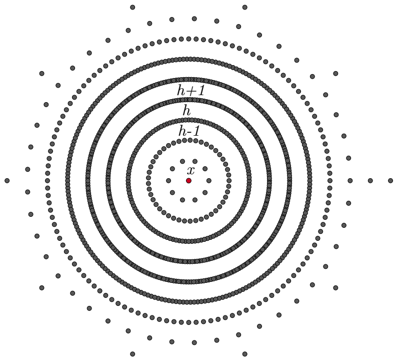


Figure 3.1: Illustration of the layout of the search space into circles of radius h around a configuration, so points that are h-Hamming distance away from it lie on the h-th circle.

Fitness Landscape. The fitness landscape of a combinatorial optimisation problem is a triple (X, N, f), where f is the objective function $f : X \to R$, X is the search space and N is the neighbourhood operator function [107].

Search Position Type For a given point $x \in X$ in the landscape, according to the topology and fitness values of its direct neighbourhood, it can belong to one of seven different types of search positions [51]. The types are:

- Strict local minimum (SLMIN): $\forall y \in N(x),\ f(y) > f(x)$.
- Non-strict local minimum (NSLMIN): $\forall y \in N(x),\ f(y) \geq f(x)$, and $\exists\ u, z \in N(x)$, such that $f(u) = f(x)$, and $f(z) > f(x)$.
- Interior plateau (IPLAT): $\forall\ y \in N(x),\ f(y) = f(x)$.
- Ledge (LEDGE): $\exists\ u, y, z \in N(x)$, such that $f(u) = f(x), f(y) > f(x)$, and $f(z) < f(x)$.
- Slope (SLOPE): $\forall y \in N(x),\ f(y) \neq f(x)$, and $\exists\ u, z \in N(x)$, such that $f(u) < f(x)$, and $f(z) > f(x)$.
- Non-strict local maximum (NSLMAX): $\forall y \in N(x),\ f(y) \leq f(x)$, and $\exists\ u, z \in N(x)$, such that $f(u) = f(x)$, and $f(z) < f(x)$.
- Strict local maximum (SLMAX): $\forall y \in N(x),\ f(y) < f(x)$.

An illustration of the position types is shown in figure 3.2. Table 3.1 shows the relation between the fitness value of a point $x \in X$ and the fitness value of its neighbours $y \in N(x)$ for each position types, where '+' denotes that one or more neighbour with the specified condition exists, and '-' denotes that no neighbour with the specified condition exists. For the purpose of this thesis, we use the term *local optimum* to refer to both strict and non-strict local optimum.

Global Optima Assuming maximisation, a point $x \in X$ is a *strict global maximum* if it is a strict local maximum and $\forall y \in X, f(x) \geq f(y)$, and a point $x \in X$ is a *non-strict global maximum* if it is a non-strict local maximum and $\forall y \in X, f(x) \geq f(y)$.

Plateaux A plateau is a set of connected non-strict local maxima, with or without interior plateau points. An exit is a neighbour of one or more configurations in the plateau, which shares the same fitness value of the plateau, but has an improving move. An exit could be a non-strict local minimum (maximum when minimising) or a ledge. We call a plateau open when it has at least one exit, otherwise we call it closed. We call a plateau of non-strict global maximum, a global plateau. Obviously, all global plateaux are closed. Following [88], we illustrated our definitions in Figure 3.3. Collecting information about the different plateaux types gives us an insight into the different plateaux regions in the problems and can help inform the algorithm design and the choice of search operators. For example, a problem with mostly open than closed plateaux motivates the use of plateaux moves. Algorithm 1 shows how a plateau is explored exhaustively starting from a non-strict local optimum, where U is the set of unvisited non-strict local optima, interior plateau and exits configurations, V is the set of visited non-strict local optimum and interior plateau configurations, and E is the set of exits founds. After exploring the entire plateau, the algorithm then returns the set V and E.

Local Search The local search algorithm used in this thesis is the steepest ascent (descent when minimising) with no plateau moves. Algorithm 2 shows steepest ascent with random restart to find the global solution.

Basin of Attraction The attraction basin $B(x^*)$ of an optimum $x^* \in X$ is the set of points that leads to it after applying local search to them, $B(x^*) = \{x \in X \mid \text{localsearch}(x) = x^*\}$. The basin of a plateau is the union of the basins of its configurations. The neighbours of a point x are evaluated in order from left to right, with respect to bit flips, in the case of having more than one neighbour with the best improving move, the first one is always selected. Of course, this deterministic way of choosing the improving move could introduce some bias to the size of the basin. However, there was only a small subset of such configurations in the landscapes of the instances we have studied. Thus, we speculate that the bias, if any, will be quite small. Another way to break the tie and avoid biasing the basin's size, is to choose randomly between the best improving configurations. This method, however, will cause the structure of the landscape to keep changing.

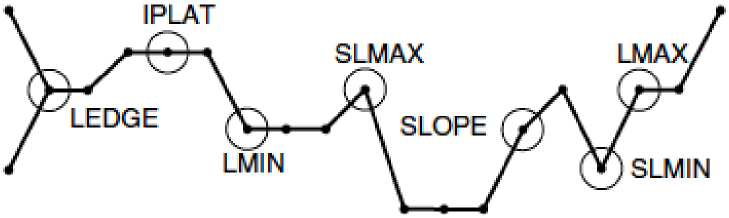


Figure 3.2: Illustration of the search position types (figure source [50]). For the purpose of this thesis, LMIN and LMAX, are called NSLMIN and NSLMAX respectively.

Algorithm 1 Exhaustive Plateau Exploring

```
start with $x$ , where $x$ is a NSLMAX
$c \leftarrow f(x)$
$V \leftarrow \phi$, $U \leftarrow \{x\}$, $E \leftarrow \phi$
while $|U| > 0$ do
    Choose $y \in U$
    $U \leftarrow U/\{y\}$
    if $c < \arg\max_{z \in N(y)} f(z)$ then
        $E \leftarrow E \cup \{y\}$
    else
        $V \leftarrow V \cup \{y\}$
        for all $z \in N(y)$ do
            if $z \notin V$ and $f(z) = c$ then
                $U \leftarrow U \cup \{z\}$
return (V,E)
```

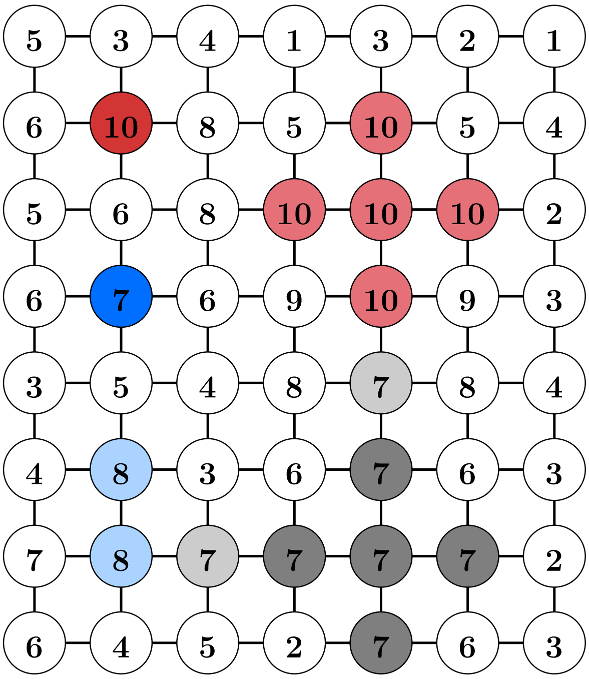


Figure 3.3: Schematic illustration (following [88]) of our definitions of strict global optima, strict local optima, and global, closed and open plateaux. Two points are neighbours if there is an edge between them. Assuming maximisation: a strict global optimum is shown by the single dark red point of fitness 10; A strict local optimum is shown by the single dark blue point of fitness 7; A global plateau is shown by the light red region of size 5 and fitness 10; A closed plateau is shown by the light blue region of size 2 and fitness 8; and an open plateau is shown by the grey region of size 5 and fitness 7, the open plateau has two exits (light grey), one (a non-strict local minimum) to the global plateau and one (a ledge) to the closed plateau.

	$f(y) < f(x)$	$f(y) = f(x)$	$f(y) > f(x)$
SLMIN	-	-	+
NSLMIN	-	+	+
IPLAT	-	+	-
LEDGE	+	+	+
SLOPE	+	-	+
NSLMAX	+	+	-
SLMAX	+	-	-

Table 3.1: Relation between neighbours fitness and search position types

Algorithm 2 Steepest Ascent with Random Restarts

```
repeat
    Choose $x \in \{0,1\}^n$, uniformly at random
    repeat
        Choose $x' \in N(x)$, such that $f(x') = \arg\max_{y \in N(x)} f(y)$
        Replace $x$ with $x'$ if $f(x') > f(x)$
    until $f(x) \geq f(x')$
until $f(x)$ is the optimal solution
```

3.2 Random Instance Generation

We are interested in studying if and how the landscape properties of instances generated randomly from different distributions vary. Problem instances were generated with integers drawn randomly from five different discrete probability distributions: uniform, normal, negatively skewed, positively skewed and bimodal distribution with peaks at both ends, figure 3.4 shows an illustration of the distributions pmfs[1]. Note that some of the ranges we studied were very large. In these cases, we used arbitrary-precision arithmetic [2].

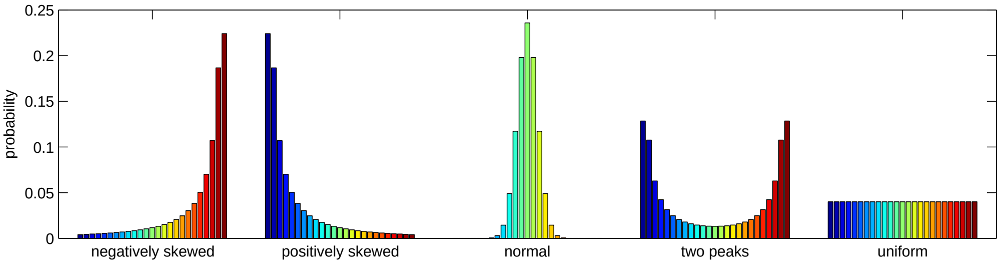


Figure 3.4: Illustration of the probability mass functions (pmfs) of the studied distributions of the weights.

[1] The random numbers were generated using boost::random::discrete_distribution from Boost C++ libraries collection.

[2] For that we used the GNU multiple precision arithmetic library (GMP).

3.3 Estimating the Number of Local Optima

In the last two decades, a number of approaches have been proposed for estimating the number of local optima in combinatorial optimisation problems (COPs) [16, 90, 36, 37, 27, 91, 88]. Most of these methods start from a random sample of different configurations and apply local search to them until a local optimum is reached. Some of the methods are non-parametric estimators such as jackknife and bootstrap [27], while others assume some parametric distribution of the basin sizes (e.g. gamma distributions) [36, 37]. However, each of these methods has its particular limitations and none of them provide a good estimate in all scenarios (e.g. when the basin sizes are different or when the number of optima is small). For example, the jackknife method [27] requires the sample size to increase as the number of optima increases, which is impractical since the number of optima grows exponentially or sub-exponentially with the problem size in most problems [65, 88]. One drawback of the bootstrap method is its computational demands to carry out the re-samplings [27]. The approach proposed by [36] models the basin sizes using gamma distribution and requires an estimate of the parameter value of the distribution, which may not be practical. Another possible limitation of all the methods that apply local search to an initial random sample is the time needed to converge to a local optimum. In many cases, this time is linear or superlinear in problem size [117, 88], but it can be exponential in other cases [26]. A review and an evaluation for several of these methods and other methods from the statistical literature can be found in [47].

The problem of estimating the number of local optima in COPs can be considered as the classical problem of estimating a population proportion in statistics. However, the use of this method to estimate the number of local optima is seldom found in the literature and even if it was used, the sample size and the confidence interval are usually not mentioned. In [43], it has been used to estimate the number of optima in the feasible region of the multidimensional assignment problem. It has been used in [115, 112] to estimate the number of optima in the quadratic assignment problem. [16] mentioned the attractiveness of the simplicity and the unbiased estimate provided by estimating the proportion, but they argued against using it as the required sample size can be very large when the proportion is exceedingly small. They also criticised that in such a case, the method is more likely to provide an upper bound estimate on the number of local minima rather than a lower bound estimate. [47] recommends using it only when all or most of the sampled optima have been seen once, after applying local search to an initial sample of points. We argue that this method is the best for estimating the number of local optima in terms of simplicity, accuracy and computational requirement when the studied landscape has a large proportion of local optima. As mentioned before, the required sample size for an accurate estimate increases as the proportion decreases, which makes obtaining an accurate estimate of the number of optima very expensive. However, an upper bound on the number of local optima in such landscapes, e.g. MAX-SAT [137, 88], can still be obtained with smaller sample sizes, which can give some useful information about the studied landscapes.

In the rest of this thesis, we refer to estimating the number of local optima by estimating their proportion as simple random sampling (SRS). To provide a baseline, we compare the performance of SRS with the performance of the jackknife method on estimating the number of optima in some instances of two of the problems studied in this thesis. In the rest of this section, we describe SRS and jackknife, and discuss different choices of confidence intervals for SRS. We then evaluate their performance and finally finish with some concluding remarks [1].

3.3.1 Estimation methods

Simple Random Sampling

Suppose that a random sample of size s is taken from the search space, and that Y optima has been observed in the sample ($0 \leq Y \leq s$), and p is the unknown proportion of the optima in the search space. Since the sample size is fixed, and the sampled configurations

[1] The work presented in this section is published in the proceedings of PPSN 2016 [7].

are independent and have a constant probability of being an optimum given by p, then Y has a Binomial distribution, $\mathrm{B}(s, p)$, with s trials and p success probability. The unbiased point estimate of the population proportion is given by $\hat{p} = Y/s$ and the estimated number of local optima can then be directly calculated by multiplying $\hat{p}$ by the search space size $S = |X|$.

There are several methods for computing confidence interval estimates for p; the most referred ones are based on the approximation of the binomial distribution by the normal distribution [81]. A rule of thumb, that is frequently mentioned, is that the binomial distribution is suitable for approximation by normal distribution as long as $sp \geq 5$ and $s(1-p) \geq 5$ [118, 12]. Figure 3.5 shows how, as a result of following this rule, the sample size grows when the population proportion declines. The most widely used confidence interval for p is the standard Wald confidence interval ($\mathrm{CI_s}$) [118, 12, 81]:

$$\mathrm{CI_s} = \hat{p} \pm z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{s}} \tag{3.1}$$

Where $z_{\alpha/2}$ is the z-score for $(1-\alpha)100\%$ confidence level and $z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{s}}$ is the error margin e . The error margin can be corrected for a finite population of size S to be equal to $e = z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{s}}\sqrt{\frac{S-s}{S-1}}$, where the value $\sqrt{\frac{S-s}{S-1}}$ is the finite population correction (fpc) factor [118]. The value of fpc is approximately one when the population size S is large compared to the sample size s, and, obviously, is equal to zero when the sample size is equal to the population size. The sample size for a desired confidence level and a desired margin of error can be determined for an infinite population by:

$$s_0 = \frac{z_{\alpha/2}^2\hat{p}(1-\hat{p})}{e^2} \tag{3.2}$$

If no prior information about p or no initial estimate of $\hat{p}$ is available, then $\hat{p}$ can conservatively be set to 0.5 where the expression $\hat{p}(1-\hat{p})$ is maximised. This will ensure that the sample size is at its maximum for the desired e. However, the proportion of optima is typically much smaller than that, thus it might be more wise to set p to a smaller value

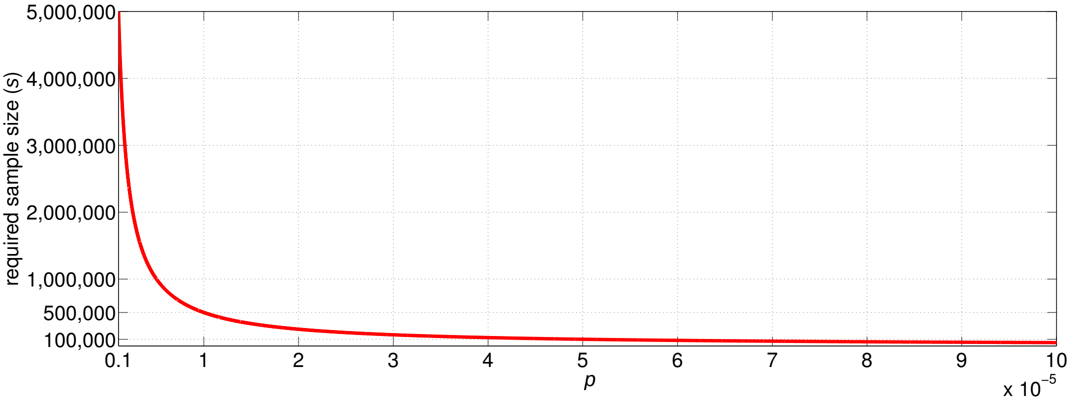


Figure 3.5: The population proportion p against the required sample size s, for the binomial distribution to be suitable for approximation by normal distribution.

and set e to a much smaller value. The sample size can be corrected for a finite population by the following formula:

$$s_1 = \frac{s_0 S}{s_0 + (S - 1)} \tag{3.3}$$

From eq.(3.2) we can see that the sample size does not depend on the population size but only on the desired confidence level, the desired margin of error, and the known estimate of p.

The behaviour of Wald interval is poor when p is close to 0 or 1, and when $Y = 0$ or $Y = s$, the length of the Wald interval is zero [2, 12, 81] . The exact Clopper-Pearson interval is an alternative method to consider in such cases [141]. However, and because of the inherent conservativeness of exact methods, other approximate methods are more useful [2]. The *Agresti-Coull* confidence interval ($\mathrm{CI}_{\mathrm{AC}}$) is recommend for correcting the Wald interval. It recentres the Wald interval by adding the value $z^2_{\alpha/2}/2$ to Y so it becomes $\tilde{Y} = Y + z^2_{\alpha/2}/2$ and adding the value $z^2_{\alpha/2}$ to s to become $\tilde{s} = s + z^2_{\alpha/2}$. When the z-score for the 95% confidence level ($z^2_{0.05/2} = 1.96$) is approximated to 2, the *Agresti-Coull* interval is equivalent to adding two successes and two failures to the sample [2, 12] . The corrected point estimate is $\tilde{p} = \tilde{Y}/\tilde{s}$ and the confidence interval is given by:

$$\mathrm{CI}_{\mathrm{AC}} = \tilde{p} \pm z_{\alpha/2}\sqrt{\frac{\tilde{p}(1-\tilde{p})}{\tilde{s}}} \tag{3.4}$$

Using *Agresti-Coull* confidence interval, the SRS estimation of the number of local optima is given by:

$$\hat{v}^{SRS} = \tilde{p}S \tag{3.5}$$

Jackknife

Jackknife is a non-parametric method based on the idea of re-sampling to reduce the bias of the estimate. The use of jackknife to estimate the number of local optima was first proposed by [27]. We selected jackknife method as a comparison baseline for two reasons: jackknife has an attractive simple and fast closed-form computation, and it is recommend to be used when the size of the sample is adequate with respect to the number of local optima [27, 47].

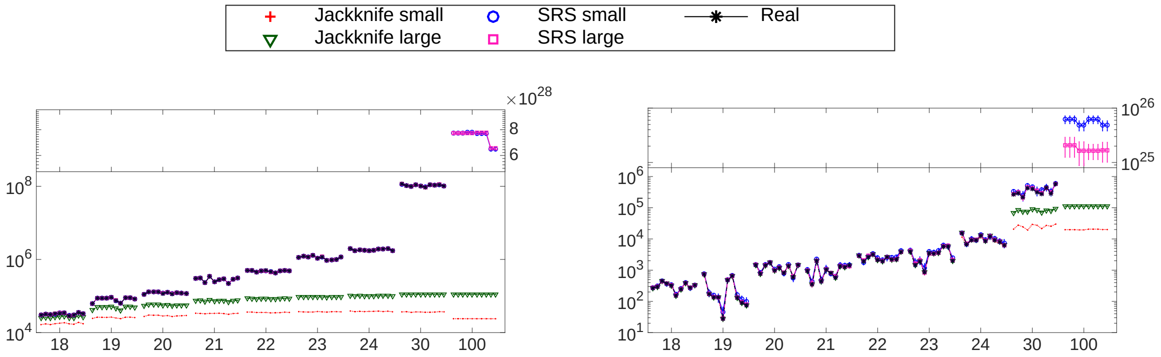
Starting from s different randomly sampled configurations and after applying local search to each one of them, the jackknife estimate of the number of local optima is given by:

$$\hat{v}^{JK} = \beta + \frac{s-1}{s}\beta_1 \tag{3.6}$$

Where β_1 is the number of optima that have been seen once and $\beta = \sum_{i=1}^{r} \beta_i$ is the number of distinct optima seen.

3.3.2 Evaluation

We obtain statistical estimates of the number of optima in randomly generated instances of the number partitioning problem and the 0-1 knapsack problem. The aim of the experiments is twofold: compare the estimates of SRS with that of jackknife, and examine the effect of the sample size on the accuracy of the SRS estimation. We compare the performance of the two methods using two sample sizes to allow for a fair comparison, since SRS uses at most $s(|N(x)|+1)$ number of fitness evaluations compared to $s(|N(x)|+1)+t|N(x)|$ fitness evaluations used by jackknife, where t is the total number of steps taken when descending(ascending) from each initial configuration. We describe the settings of the two sample sizes in more details below.



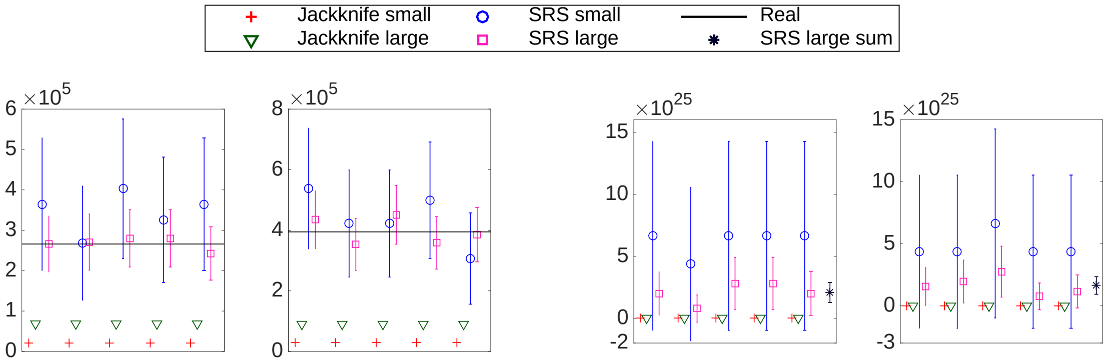
(a) *H1* landscape

(b) *H1+2* landscape

Figure 3.6: Simple random sampling (SRS) and Jackknife estimation of the number of optima (in log scale) as the problem size grows. Each data point represents the average estimate of 10 samples from a single instance. The error bars show the standard deviations. The results are for the 0-1 knapsack problem with weights drawn from the normal distribution and with 10 different instances for each problem size.

The mean estimates of the number of optima in the two landscape of the 0-1 knapsack problem is shown as n grows in figure 3.6 (note that some data points lie on top of each other). The estimates were obtained by the jackknife and SRS, and were averaged over 10 samples for each sample size. The sample sizes are set as follows: first we obtained the sample size s for each n from eq.(3.2) and eq.(3.3) by setting $e = 0.005$, $\hat{p} = 0.3$ and $z_{\alpha/2} = 2.576$. Note that the sample size, only changes slightly as the problem size increases, starting from $s = 45,701$ when $n = 18$, until it reaches $s = 55,351$ when $n = 100$. After obtaining s, we then set the small sample size of SRS to s and the small sample size of jackknife to $s-t+t/(|N(x)|+1)$ (i.e. we subtract the fitness evaluations used when ascending from the sample budget). We set the large sample size of jackknife to s and the large sample size of SRS to $s+t-t/(|N(x)|+1)$, where t is the total number of steps taken by jackknife with the large sample. The samples are drawn without replacement for small problem sizes $n \leq 24$. The figure shows that SRS using both small and large sample sizes accurately estimates the real proportions in both landscapes, apart from $n = 100$ in the *H1+2* landscape. The discrepancy between estimates of the large and small samples in this case, in addition to the larger standard deviations, indicate that the proportion is small and that the sample size, in particular the small one is probably inadequate. As for the jackknife, both sample sizes quickly become inadequate as the number of optima seen once quickly grows with n until all the optima that have been seen were only seen once. Thus, the method fails to provide accurate estimates and grossly underestimates the number of optima. This is more noticeable in the *H1* landscape where the number of local optima is large. The confidence intervals of SRS estimates are very small in *H1* landscape across all n, but they get wider as n increases in the *H1+2* landscape. In figure 3.7, we look closely at the results of four instances of size $n = 30, 100$ from figure 3.6. The figure shows the confidence interval around 5 estimates of each method with each sample size. The width of the confidence interval decreased with the large sample size as expected. The SRS large sample size for $n = 30$ is around 2×10^5 and around 3×10^5 for $n = 100$. Obtaining the real number of optima was infeasible for $n = 100$, therefore we show the estimate of SRS with a larger sample size by setting Y to the sum of the number of optima found in all the large samples and s to the sum of the large sample sizes. The outcome estimates of the proportion of both instances are around 10^{-5}. The very wide confidence intervals with negative lower bounds around the small sample size estimates of SRS in $n = 100$ indicate that the proportion is much smaller than what SRS can precisely estimate with this sample size. In such a case, the estimates of SRS can only provide an upper bound to the number of optima. However, we suggest combining the results of the two methods in such cases by using the result of the jackknife method for a better lower bound than just zero.

Figures 3.8 and 3.9 show how the accuracy of SRS estimates increases as the desired error margin e decreases. Decreasing e consequently increases the sample size. The figures also show how SRS is able to accurately estimate the fraction of local optima with relatively small sample sizes. As we mentioned before, the required sample size does not directly depend on n, but since the fraction of optima usually declines as the problem size grows [29], the required sample size will increase with the problem size as shown in



(a) $n = 30$

(b) $n = 100$

Figure 3.7: Each figure shows the estimates of the number of optima in a single instance of 0-1KP, and each data point shows the estimate of a single sample. The error bars around SRS estimates are the 95% $\mathrm{CI}_{\mathrm{AC}}$.

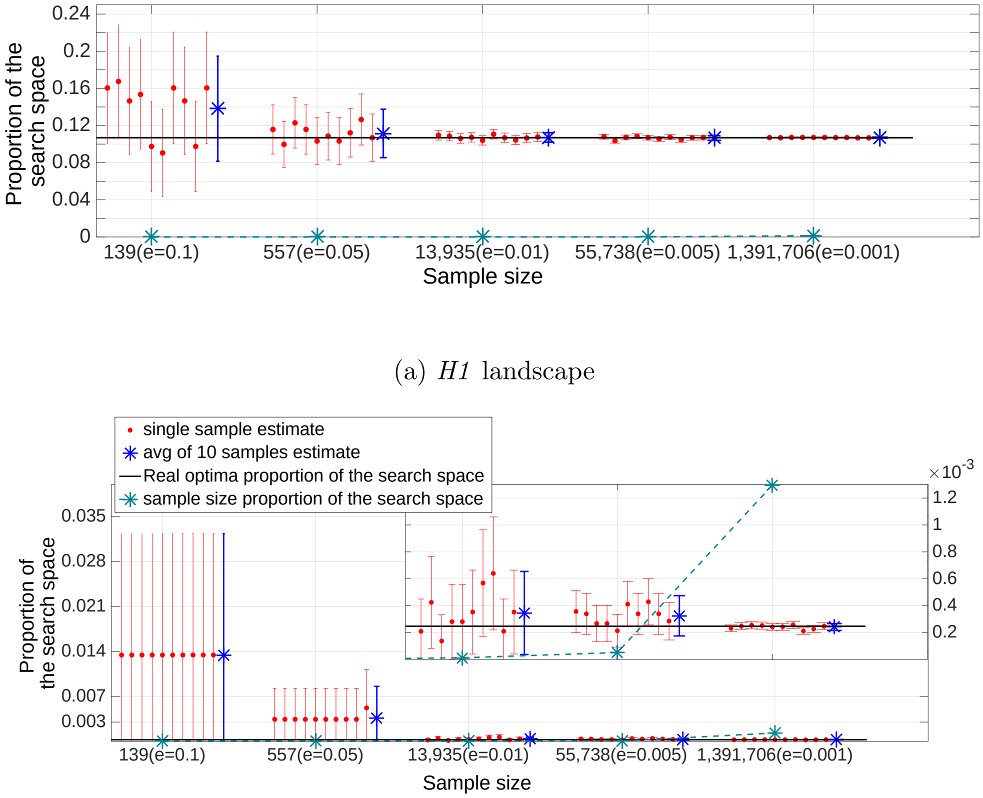
table 3.2. The sample sizes in table 3.2 are obtained from eq.(3.2) and eq.(3.3) by setting $\hat{p} = \langle p \rangle^{\mathrm{NPP}}$ (obtained from eq.(4.4)), $z_{\alpha/2} = 2.576$ and e as shown in the table. In both problems and in both landscapes, most of the optima have small basin sizes with only very few having large basin sizes.

Table 3.2: NPP sample size

	n	**24**	**30**	**100**	**1000**
s	$e = \langle p \rangle^{\mathrm{NPP}}$	276	388	2,395	75,915
	$e = \frac{\langle p \rangle^{\mathrm{NPP}}}{5}$	6,889	9,697	59,855	1,897,856
	$e = \frac{\langle p \rangle^{\mathrm{NPP}}}{10}$	27,520	38,785	239,420	7,591,421

3.3.3 Conclusions

Simple random sampling with $\mathrm{CI}_{\mathrm{AC}}$ provides a simple way to obtain an unbiased statistical estimate of the number of local optima. The accuracy of the obtained estimate depends on the sample size s, which can be determined for a desired margin of error e. A wide $\mathrm{CI}_{\mathrm{AC}}$ or a negative lower bound indicates that the proportion is smaller than the desired e. In such a case, s can be increased considering that it only costs at most $|N(x)| + 1$ fitness

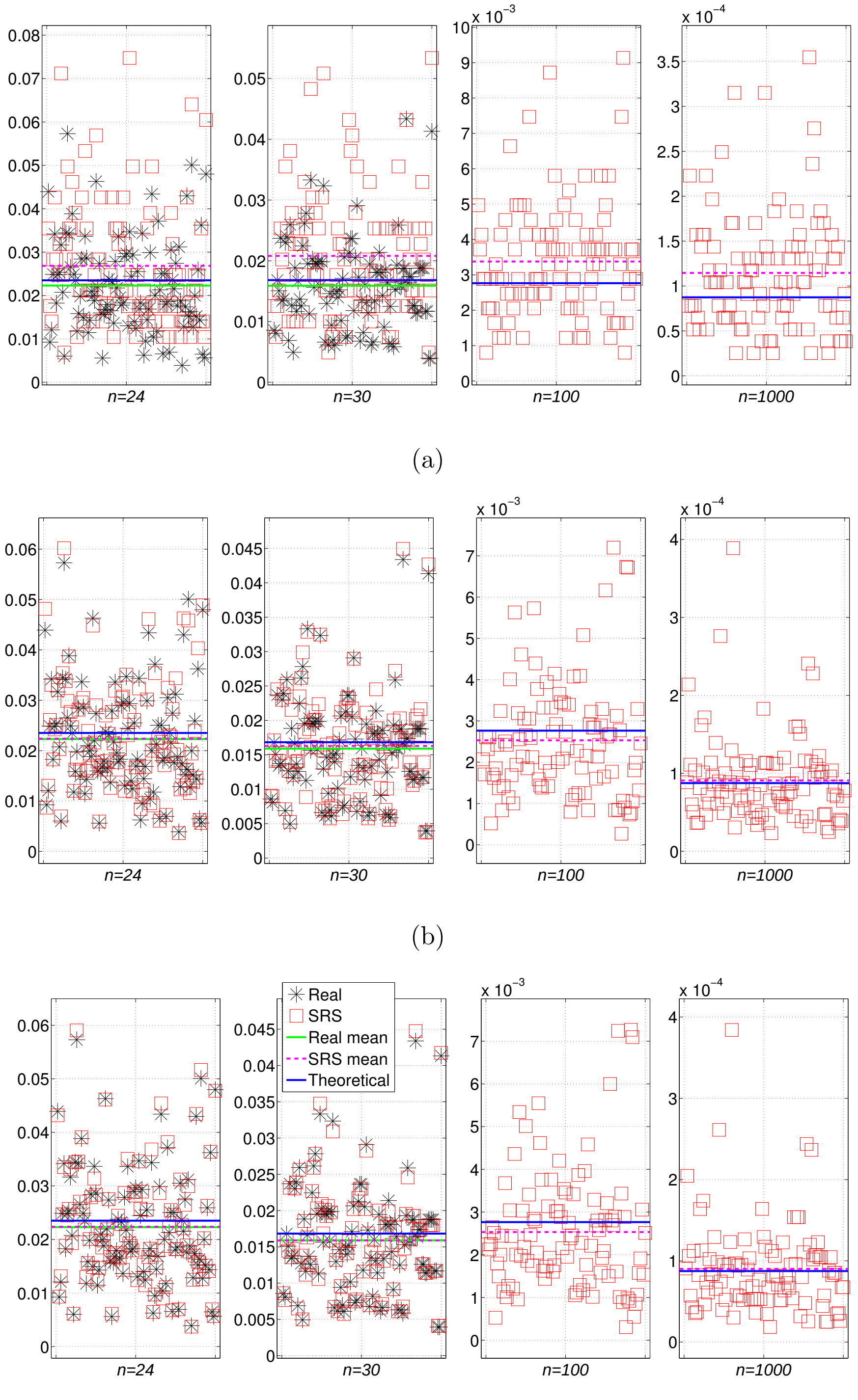


(a) *H1* landscape

(b) *H1+2* landscape

Figure 3.8: Simple random sampling estimation of the optima proportion versus sample size. The sample sizes are obtained from eq.(3.2) and eq.(3.3) by setting $\hat{p} = 0.3$ and $z_{\alpha/2} = 2.576$ (corresponding to 99% confidence level). The results are for a single instance of knapsack problem of size $n = 30$ and weights drawn from normal distribution. The error bars are the 95% *Agresti-Coull* confidence intervals.

evaluations per configuration. This is practical as long as the proportion is not exceedingly small. Alternatively, the estimate of SRS can be used as an upper bound, combined with the estimate of another method that applies local search to an initial sample, for a lower bound other than zero. We recommend that SRS should be the first method to use for estimating the number of optima, especially when no prior information is available about the problem being studied. In this thesis, we only use SRS for estimating the number of optima, this is motivated by the findings of this evaluation and the fact that the optima proportions in the exhaustively studied small problem sizes were found to be mainly large.



(a)

(b)

(c)

Figure 3.9: Proportion of the strict optima in the *H1* landscape of NPP for different problem sizes. SRS estimates are shown when the sample size is obtained with three different desired error margins (a) $e = \langle p \rangle^{\mathrm{NPP}}$, (b) $e = \frac{\langle p \rangle^{\mathrm{NPP}}}{5}$, (c) $e = \frac{\langle p \rangle^{\mathrm{NPP}}}{10}$. The sample size for each problem size is shown in Table 3.2. The results are for 100 random instances with uniform weights and $k_c > 1$. Obtaining the real proportion was only computationally feasible for $n = 24$ and $n = 30$. The theoretical mean proportions are obtained from eq.(4.4).

3.4 Basin of Attraction Shape: Return Probability

The size of the basin of attraction can be estimated using the return probability concept introduced in [88]. The return probability to an optimum starting from a Hamming sphere of radius h around it is given by $p_r(h)$. We mainly use this method to estimate the shape of the attraction basin. We randomly sample s configurations that are h Hamming distance away from an optimum, we then apply local search to them and calculate the fraction that led to the starting optimum. The sample size s is obtained from eq.(3.2) and eq.(3.3) by setting $\hat{p} = 0.5$, $e = 0.005$, and $z_{\alpha/2} = 1.645$ (corresponding to 90% confidence level). The probability of finding an optimum starting from a random configuration can then be calculated as follows: $P(\text{fining an optimum}) = \frac{1}{2^n} \sum_{h=0}^{n} \binom{n}{h} p_r(h)$. In this thesis, we continue to sample configurations even after no configuration was found to be in the optimum's basin in the previous sphere. We continue sampling until the last configuration in sphere n, however, and especially for the *H1* operator, it might be more efficient when studying larger problem sizes to stop sampling after no configuration was found to be in the optimum's basin.

3.5 Local Search Performance

We study a number of aspects of the performance of the steepest ascent/descent with random restarts (algorithm 2) when each of the two neighbourhood operators is employed. The first is the cost of finding the global optimum, where we measure the cost by the number of fitness evaluations used. Note that we treat the objective function as a black-box, hence the number of times the objective function is queried for each step taken by the algorithm equals the size of the neighbourhood. We compare the results of the two operators and determine the statistical significance between the two performances using Wilcoxon rank-sum test at the 5% level. We also study the scaling behaviour of the cost as the problem size grows. Usually the question of finding the optimal solution is irrelevant for larger problem sizes as it is typically infeasible to find it, therefore, we study the quality of the local optima obtained using a fixed budget of fitness evaluations. We also study the time the algorithm takes until a local optimum is found starting from a random configuration. Note that the nature of the *H1+2* operator can allow it to take fewer number of steps by hopping over spheres to reach the local optimum. Note also that although the local search algorithm under study here is called steepest descent/ascent, it does not necessarily guarantee that the path, starting from a random configuration x until a local optimum x^* is found, will be the shortest path (i.e. the number of steps taken from x until x^* is reached is at most equal to the Hamming distance between x and x^*). From the perspective of the Hamming spheres around an optimum x^*, this can occur when the best improving move of a configuration x^i in the path, which resides in the Hamming sphere h^i, is in a Hamming sphere $h^{i+1} \geq h^i$.

3.6 Summary

We discussed the definitions of the properties and the methods to measure them that we used to carry out the fitness landscape analysis of the problems studied in this thesis. We also discussed how we randomly generate instances of these problems but we left the parameters that are problem-specific to be discussed in relevant chapters. To study the landscape of larger problem sizes we need to use some sampling methods. We presented a brief overview of the sampling methods used to estimate the number of local optima. One method that is overlooked in the literature and rarely used is the simple random sampling, where a sample of points is chosen uniformly at random from the search space, and then examined to determine the proportion of them that are local optima (without the application of local search, only examining if the condition of local optimality is met). Not only this method is rarely used in the literature, but also only the point estimate is reported without the confidence interval and usually with no mention of the used sample size. We provided an evaluation of this method in section 3.3 and discussed the choices of different confidence intervals and the effect of the sample size on the estimate accuracy.

CHAPTER 4

NUMBER PARTITIONING PROBLEM

In this chapter we study the number partitioning problem (NPP), a classical problem in theoretical computer science and one of Garey and Johnson's six basic NP-complete problems [35]. We study various landscape features of a large number of randomly generated instances. The instances were generated with different values of problem parameters to study what effects they have on the *H1* and *H1+2* landscapes and the consequence of that on the performance of local search [2].

4.1 Problem Definition

Given a set $W = \{w_1, \ldots, w_n\}$ of m-bit positive integers (weights) drawn at random from the set $\{1, 2, \ldots, M\}$ with $M = 2^m$, the goal is to partition W into two disjoint subsets S, S' such that the discrepancy between them $|\sum_{w_i \in S} w_i - \sum_{w_i \in S'} w_i|$ is minimised. A partition is called perfect, if the discrepancy between the two subsets is 0 when the sum of the original set is even, or 1 when the sum is odd. Equivalently, the problem can be viewed as minimising: $\max\left\{\sum_{w_i \in S} w_i,\ \sum_{w_i \in S'} w_i\right\}$, the maximum sum over the two subsets. Let $x \in \{0, 1\}^n$, the fitness function to be minimised can be defined as:

[2] Part of the work presented in this chapter was published in the proceedings of EvoCOP 2014 [5] and PPSN 2014 [4].

$$f(x) = \left| \sum_{i=1}^{n} w_i x_i - \sum_{i=1}^{n} w_i (1 - x_i) \right| \tag{4.1}$$

The binary representation of NPP creates a symmetry in the search space, in the sense that a solution and its bitwise complement have the same fitness value. Thus, the number of unique solutions is $\leq 2^{n-1}$.

The NPP is NP-hard in the weak sense [35], that is, there exists an algorithm that can solve it in pseudo-polynomial time through dynamic programming. The complexity of such an algorithm, $\mathcal{O}(n2^{\log_2 \sum_{i=1}^{n} w_i})$, is polynomial in the number of weights and the sum of the weights but exponential in the number of bits required to represent the sum. As Garey and Johnson [35] note, such an algorithm will display an exponential behaviour only when extremely large input numbers are allowed. The running time of such an algorithm would thus exhibit an exponential behaviour as M grows large.

4.1.1 Phase transition

NPP undergoes a sudden phase transition from solvability (a perfect partition exist) to insolvability (a perfect partition doesn't exist), determined by the control parameter $k = \log_2(M)/n$, which corresponds to the number of the bits required to encode the numbers in the set divided by the size of the set. For $\log_2(M)$ and n tending to infinity, the transition occurs at the critical value of $k_c = 1$, such that for $k < 1$, there are many perfect partitions with probability tending to 1, whereas for $k > 1$, the number of perfect partitions drops to zero with probability tending to 1 [11]. A more detailed parameterisation of the critical value of the control parameter is given by the following [1] [68]:

$$k_c = 1 - \frac{\ln(\frac{\pi}{6} n)}{2n \ln(2)} \tag{4.2}$$

The transition between the two phases appears in the size of the problem backbone. The pairs of weights that are placed in the same subset or in opposite subsets in all optimal

[1] A more rigorous derivation of the transition point can be found in [11].

solutions of an NPP instance, form the backbone of that instance. There is a very sharp increase in the backbone size of the optimal solutions in the NPP as one approaches the phase transition boundary, after which the backbone tends to be complete giving a unique optimal solution [100]. Gent and Walsh [38] were the first to verify the existence of a phase transition in NPP. They have shown an empirical evidence of its existence in their paper through numerical simulations. They introduced the control parameter k and estimated the transition point to occur around $k_c = 0.96$. Previously, Fu [33] used statistical mechanics to analyse the problem and concluded incorrectly that NPP does not undergo a phase transition. Mertens [68, 67] used the same method from statistical physics and the parameterisation of Gent and Walsh to obtain non-rigorous analytical results of the phase transition in NPP. Borgs et al. [11] then performed a rigorous analysis of the problem and showed the mathematical proofs for the existence of the phase transition.

In the literature, the effect of this phase transition has been shown in the computational complexity of some exact solvers such as the complete Karmarkar-Karp differencing algorithm [68]. Where instances with $k < k_c$ were "easy-to-solve" and the ones with $k > k_c$ were "hard-to-solve". In this chapter we are interested to see whether the landscape properties of number partitioning problem change with the phase transition. We are also interested to see if similar changes in the computational complexity occur in the performance of the local search algorithm we study. Note that through out this chapter we refer to the phase with $k < k_c$ as the "easy" phase and the phase with $k > k_c$ as the "hard" phase. Our use of hard and easy here does not correspond to the computational complexity but to the probability of having a perfect partition.

4.1.2 Distribution of the weights

Most of the existing studies of the NPP assume that the weights are drawn at random from a uniform distribution [105, 68, 67, 11, 60, 29, 133]. Only very few consider different distributions, for example the exponential distribution in [133]. We study instances generated by drawing the weights at random from various distributions as shown in subsection 3.2. We found that the number of strict local optima and the cost of local search to find the global, vary greatly between some of the distributions. The variation is most noticeable in the *H1* landscape as shown in figures 4.2 and 4.25, respectively, for the number of strict local optima and the cost of local search. In particular, the negatively skewed and the normal distributions have the largest number of strict local optima in the *H1* landscape and the highest cost of local search, while the positively skewed and the two peaks distributions have the fewest and lowest. We believe that most of this behaviour can be explain by the variability of the weights. To capture this with a single parameter, that does not require the knowledge of the underlying distribution of the weights, we suggest using the coefficient of variation (CV). The CV provides a measure of relative variability or dispersion. It is defined as the ratio of the standard deviation σ to the mean μ:

$$CV = \frac{\sigma}{\mu} \tag{4.3}$$

4.2 Search Position Types

The search positions found in randomly generated instances of NPP with different values of the CV and the phase transition control parameter k, are shown as proportion of the search space in tables A.1, A.2 and A.3, in the appendix, for the *H1* and the *H1+2* landscapes. When the objective function is the square of the discrepancy, NPP has an elementary landscape under the *H1* operator [42]. This has an implication on the types of plateaux and search positions that can exist on elementary landscapes [131]. The first implication is that configurations of type IPLAT can only exist when the entire landscape is flat, meaning that every configurations in the search space belongs to the same plateau. The second implication is that exits of open plateaux can only be ledges. In our results, no configuration of type IPLAT or NSLMAX has been found in either landscapes. On both landscapes, there are always two configurations of type SLMAX: the all zeros solution $x = (0, \cdots, 0)$ and its bitwise complement. The clear difference between instances from the easy and hard phase is in the number of NSLMIN. We examine this further, along side the number of strict local and global optima, in the next section.

4.3 Optima and Plateaux

4.3.1 Number of optima and plateaux

Figure 4.1 shows the number of global optima found in randomly generated instances of NPP. The figure shows that for all the distributions, the number decreases as we approach the phase transition point and keeps decreasing as we cross the phase transition until we have only two optimal solutions. There are some variations in the number of global optima between the different distributions in the easy phase, with instances drawn from the positively skewed distribution having the highest number of global optima and instances drawn from the negatively skewed distribution having the lowest number of global optima. Similar behaviour has been observed for the number of non-strict local optima as figure 4.3 shows. The figure shows that the number starts to decrease as we approach the phase transition until it becomes zero in the hard phase.

Figure 4.2 shows the number of strict local optima found in the randomly generated instances of NPP. There is a very clear difference in the number of local optima across instances generated from the different distributions. In the landscapes induced by the H1 operator, instances drawn from normal distribution have the highest number of strict local optima (around 15% of the search space). Instances generated from negatively skewed distribution have a quite high number of local optima as well (around 8% of the search space) but the number varies a lot between the randomly drawn instances from this distribution. Instances drawn from the uniform distribution have less number of local optima (around 3% of the search space), while the lowest number of local optima is seen in instances drawn from both positively skewed and two peaks distributions (representing around 1% of the search space). Figure 4, also, shows that the number of strict local optima does not change very much between the easy and the hard phase regardless of the distribution from which the weights are chosen. The number of local optima can be used as a measure of the ruggedness of the landscape [107]. The found results indicate that instances with weights drawn from normal and negatively skewed distributions have more rugged landscapes than instances drawn from uniform, positively skewed and two peaks distributions.

For the landscapes induced by *H1+2* operator, the number of strict local optima drops for all the different distributions compared to the *H1* landscapes. It seems that the largest drop occurs in instances drawn from normal and negatively skewed distributions. As in the *H1* landscapes, the number of strict local optima does not seem to change much between the easy and the hard phase, apart from very small values of k (0.4 and 0.5). We believe that this is due to the slightly higher number of global and non-strict local optima in the H1+2 landscapes of such instances.

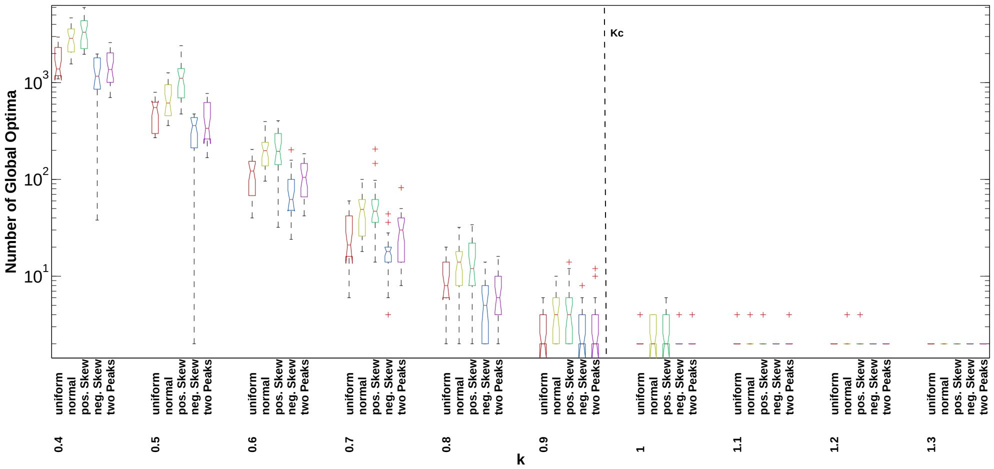


Figure 4.1: Number of global optima versus the phase transition control parameter k, for all the different distributions of the weights. Each box represents the number found in 30 random instances of size $n = 20$. The dotted line is given by k_c from eq.(4.2).

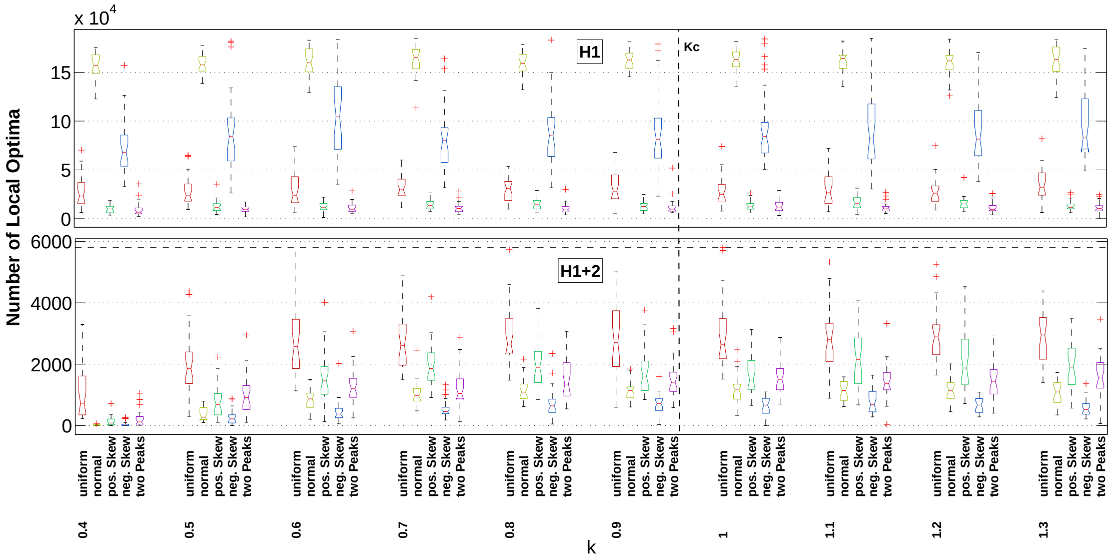


Figure 4.2: Number of strict local optima versus the phase transition control parameter k, for all the different distributions of the weights. Each box represents the number of strict local optima found in the 30 random instances of size $n = 20$. This is shown for both neighbourhood operators H1 and H1+2. The dotted line is given by k_c from eq.(4.2).

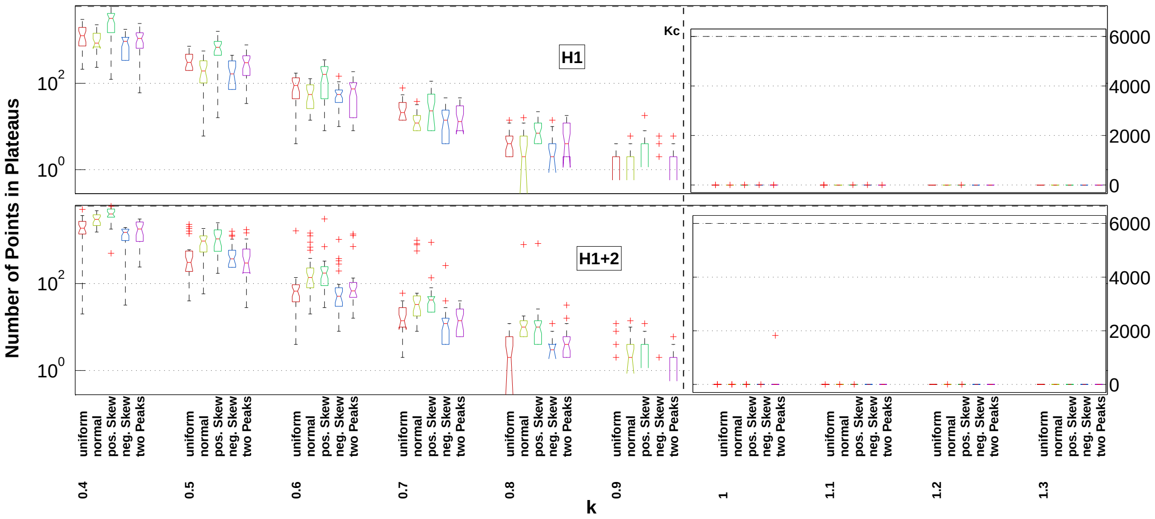
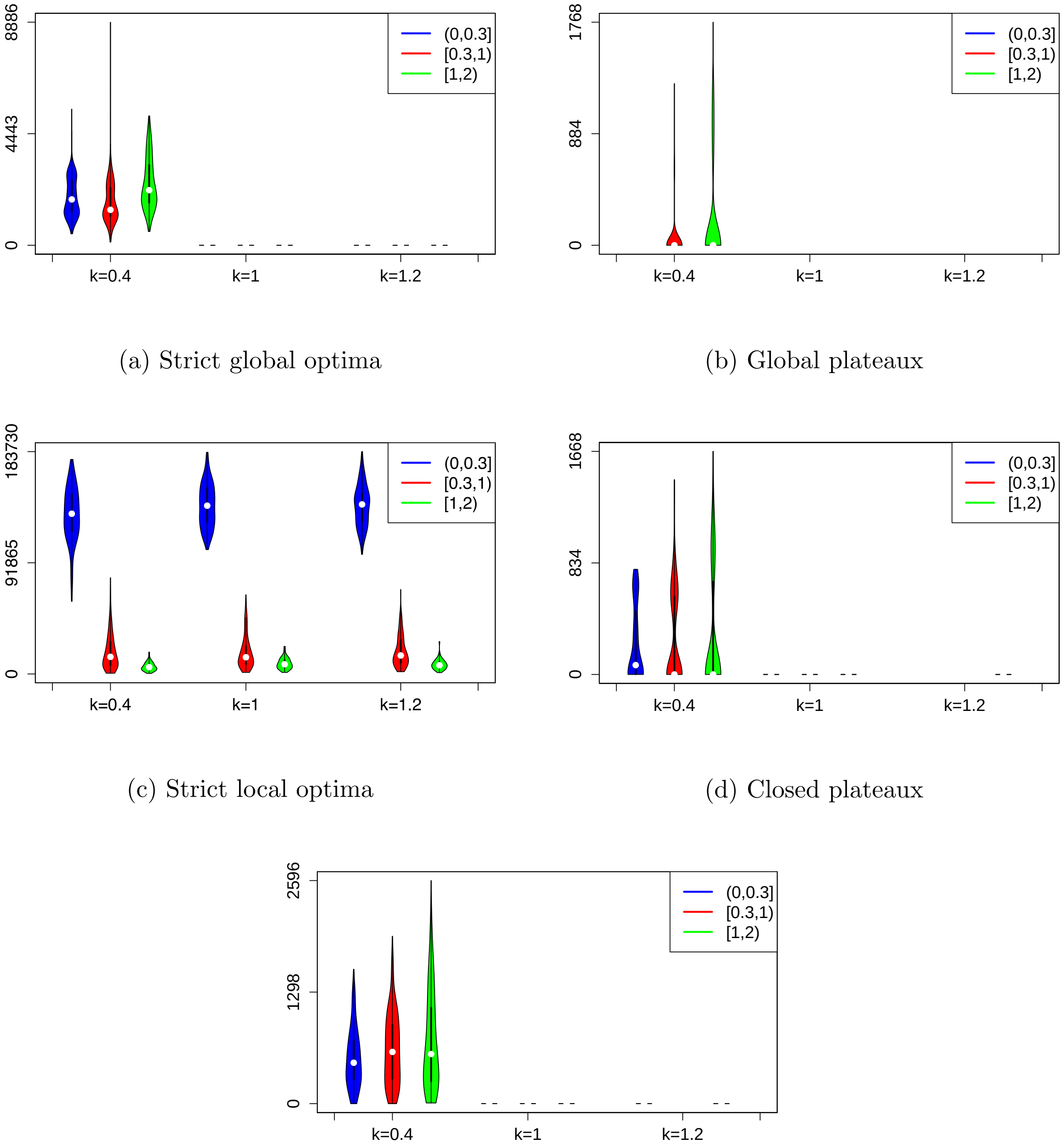


Figure 4.3: Number of non-strict local optima versus the phase transition control parameter k, for all the different distributions of the weights. Each box represents the number found in 30 random instances of size $n = 20$. This is shown for both neighbourhood operators H1 and H1+2. The dotted line is given by k_c from eq.(4.2).



(a) Strict global optima

(b) Global plateaux

(c) Strict local optima

(d) Closed plateaux

(e) Open plateaux

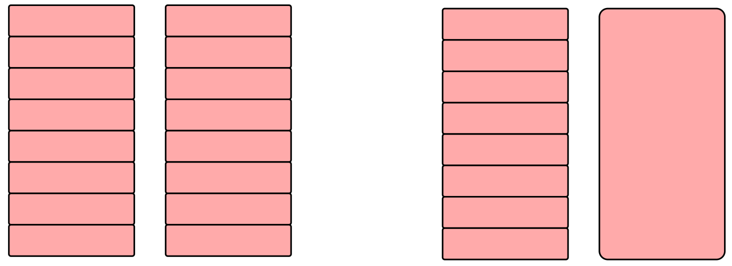
Figure 4.4: Number of the different optima and plateaux found in the *H1* landscape against k. The results are for 600 instances of $n = 20$ for each value of k. The colours show the different ranges of CV values.

Similar results are shown in figures 4.4 and 4.8 [1], but now with the instances being

[1] A violin plot is a mixture of a box plot and a kernel density plot. In addition to the usual four main features shown by a box plot (i.e. centre, spread, asymmetry and outliers), violin plot adds an estimated density trace (smoothed histogram), which reveals the shape of the data distribution that would not have been obvious in a box plot otherwise [48].

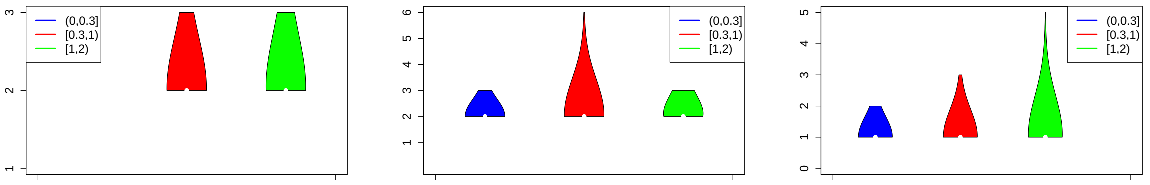
described by their CV values instead of the distribution they are drawn from. Figures 4.11 and 4.12 show how instances drawn from the different distributions map to different regions of the CV. Roughly speaking, normal and negatively skewed instances map to the region ≤ 0.3, uniform instances map to the region between 0.4 to 0.7, positively skewed and two peaks instances map to the region > 0.8. The density of the region ≤ 0.3 is high because both the normal and the negatively skewed map to this small region. The CV seems to capture most of the variation in the number of strict local optima in the *H1* landscape as the two correlate very strongly and negatively across different values of n and k. To explain the intuition behind this strong correlation, we show an example of two extreme cases in figure 4.9; when the CV is small there are $\binom{2n}{n} \sim \frac{4^n}{\sqrt{\pi n}}$ possible ways to split the weights into two piles, while for the larger value there are only two ways to do that.

In the easy phase where plateaux exists, the majority of the plateaux in the *H1* landscape were open plateaux with very few global ones. Also, the number of configurations in each plateau is very small as shown in figure 4.6. The size of the global and closed plateaux for example was found to be only 2 to 3 non-strict local optima (all the plateaux are composed of NSLMIN only, as there is no configuration of type IPLAT). All the open plateaux we found are composed of only one non-strict local optimum. The majority of open plateaux have only one exit, although the number of exits seems to increase slightly as the the CV value increases.



(a) Small CV, $\binom{2n}{n} \sim \frac{4^n}{\sqrt{\pi n}}$ optima (b) Large CV, only two optima

Figure 4.5: Two extreme examples of small and large CV values of the weights and the resulting number of optima when using the *H1* operator.



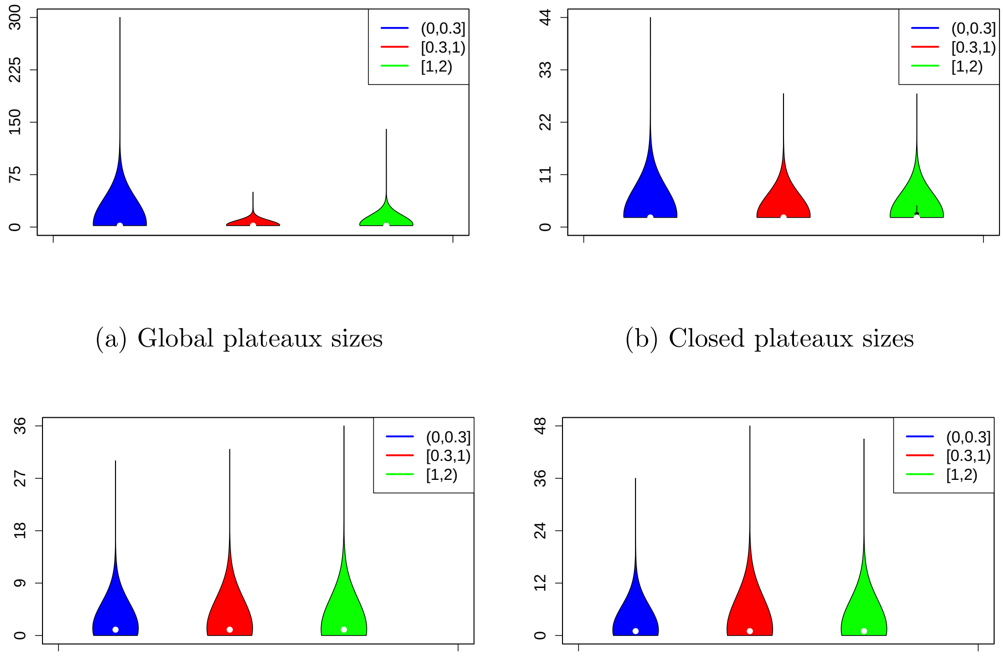
(a) Global plateaux sizes

(b) Closed plateaux sizes

(c) Open plateaux exit numbers

Figure 4.6: Sizes of the different plateaux and the number of exits in open plateaux found in the *H1* landscape. The results are for 600 instances of $n = 20$ for each value of k. The colours show the different ranges of CV values. All open plateaux found in the *H1* landscape are composed of a single configuration.

The number of strict local optima decreases in the *H1+2* landscape compared to the *H1* as shown in figure 4.8. The slightly higher number of local optima in instances with CV values between $0.3 < CV < 1$, captures the same behaviour that was observed in instances drawn from uniform distribution (see figure 4.2). The number of strict global optima decreases in the *H1+2* landscape as many of them become part of global plateaux. There are more global plateaux in the *H1+2* landscape, but fewer number of open and closed plateaux. The number of exits from open plateaux is larger in this landscape as expected. The size of all the plateaux in this landscape is also much larger than that in the *H1* landscape as figure 4.7 shows. This may seem counter-intuitive, as applying a larger neighbourhood operator as opposed to a smaller one, usually has the positive effect of reducing the number of optima and plateaux, but it can also have an effect of introducing new larger plateaux. A schematic illustration of this mechanism is shown in figure 4.10. The figure shows how after applying the larger neighbourhood operator, the same-fitness strict optima became connected forming a closed plateau. If the green triangle-shaped optimum had a better fitness value than the red diamond-shaped optima, then the figure shows how two open plateaux, sharing the same exit and each of size one, can be formed. An example of the different types of optima and plateaux found in a small instance of size $n = 12$ is shown in figure 4.9.



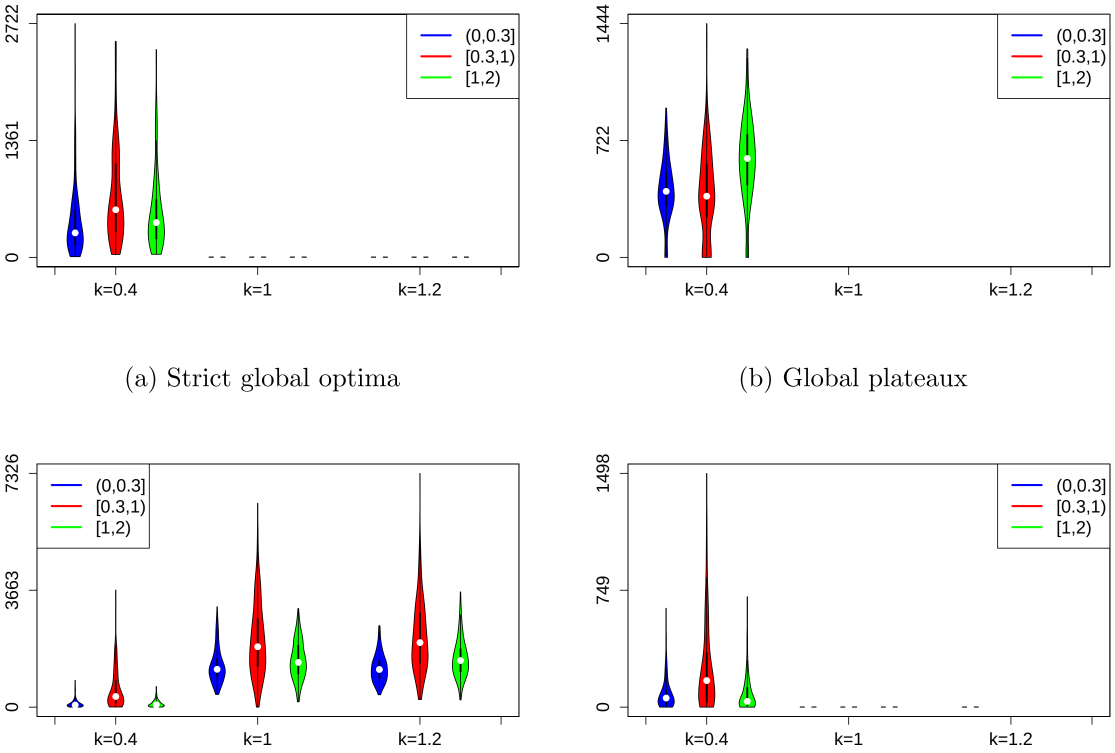
(a) Global plateaux sizes

(b) Closed plateaux sizes

(c) Open plateaux sizes

(d) Open plateaux exit numbers

Figure 4.7: Sizes of the different plateaux and the number of exits in open plateaux found in the *H1+2* landscape. The results are for 600 instances of $n = 20$ for each value of k. The colours show the different ranges of CV values.

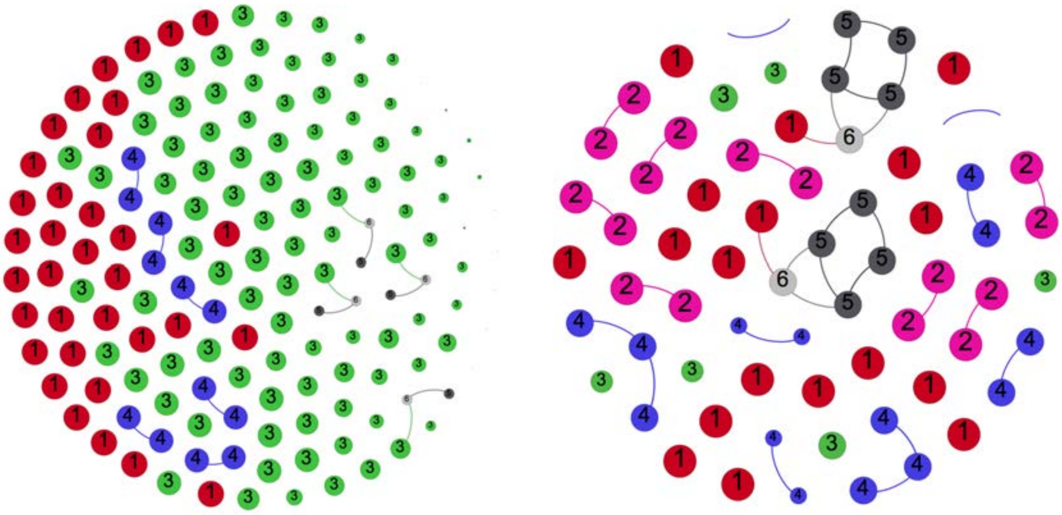


(a) Strict global optima

(b) Global plateaux

(c) Strict local optima

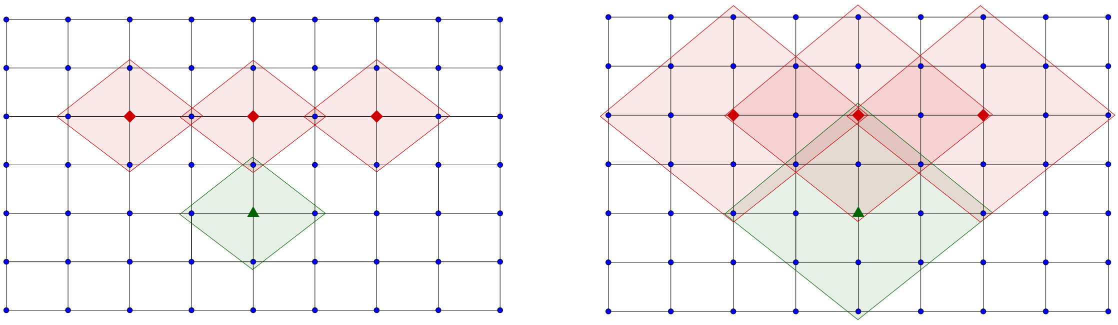
(d) Closed plateaux



(a) *H1* landscape

(b) *H1+2* landscape

Figure 4.9: The different types of optima and plateaux found in an easy NPP instance ($k = 0.5$) of size $n = 12$ and $CV = 0.61$. The colours of the nodes and their labels correspond to their type as follows: (1, red) a strict global optimum, (2, pink) a global plateau, (3, green) a strict local optimum; (4, blue) a closed plateau; (5, dark grey) an open plateau; and (6, light grey) an exit. The node size is scaled proportional to its fitness (larger means fitter). An edge between two nodes can either indicate that they are neighbours, or if it is between an exit and a node with a better fitness, it indicates that the exit leads to the basin of that optimum or plateau.



(a) Small neighbourhood

(b) Large neighbourhood

Figure 4.10: A schematic illustration showing how applying a larger neighbourhood operator as opposed to a smaller one, can reduce the number of optima, but also can introduce plateaux. (a) Assuming minimisation and under the small neighbourhood operator: the red diamond-shaped nodes are strict optima with the same fitness value, and the green triangle-shaped node is a strict optimum with a higher fitness value than the red optima. Every other node is either a local maximum, a slope or a ledge at a higher fitness value than the four optima. The shaded areas indicate the neighbourhood of the optimum at its centre. (b) After applying the larger neighbourhood operator, the green optimum is no longer an optimum but a slope, however, the red optima have now formed a closed plateau.

4.3.2 Average number of strict local optima

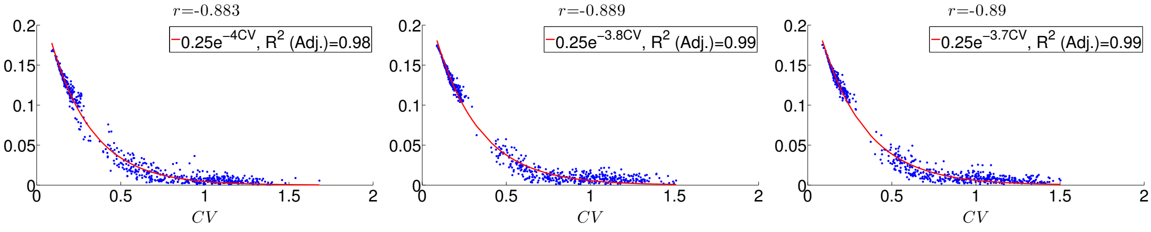
When the weights of NPP are drawn from uniform distribution, the average proportion of the strict local optima in the *H1* landscape is given by the following formula, which was derived by Ferreira and Fontanari [29] using statistical mechanics analysis:

$$\langle p \rangle^{\mathrm{NPP}} = \sqrt{\frac{24}{\pi}} n^{-3/2} \tag{4.4}$$

Based on the data we observed, we propose a generalized formula for estimating the average proportion of strict local optima in the *H1* landscape. The formula does not require the knowledge of the distribution from which the weights are drawn and only depends on the CV of the weights and the size of the problem.

$$\frac{\langle v \rangle}{2^n} = a\, e^{-b\,CV} \tag{4.5}$$

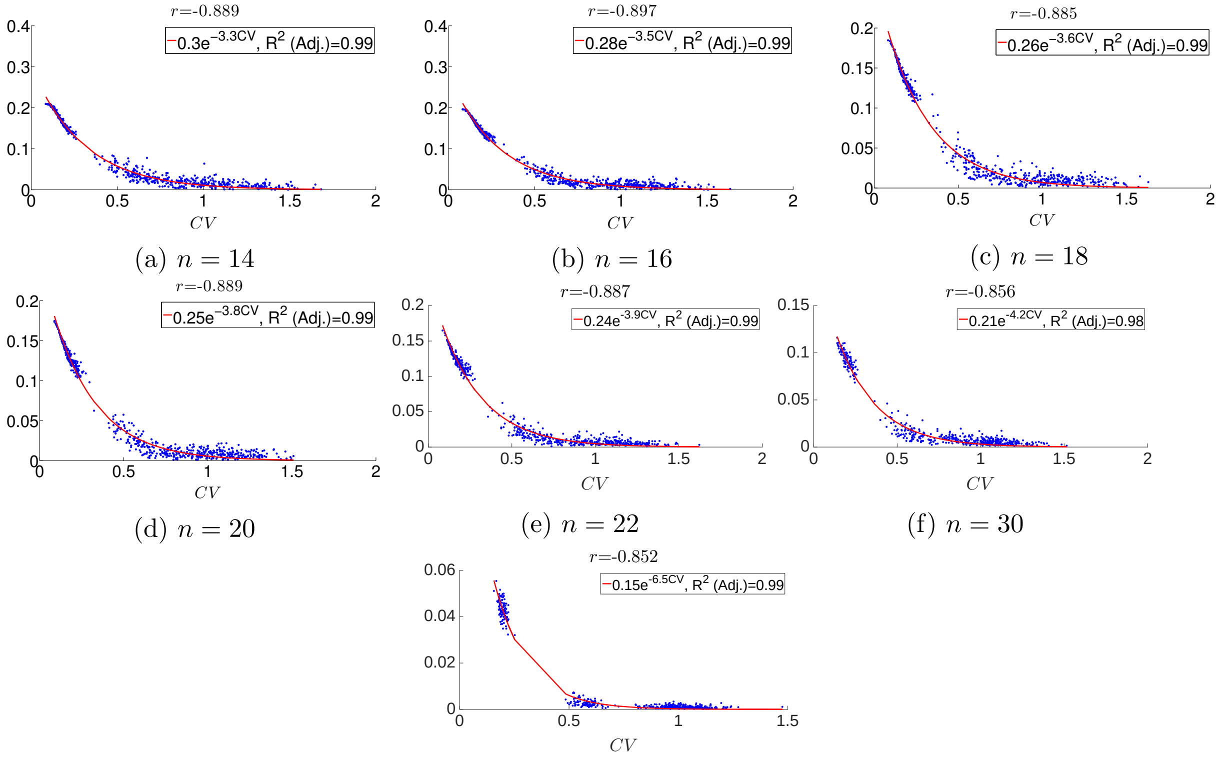
Where the values of the coefficients a and b depend on n. Figures 4.11 and 4.12 shows the estimation of the fraction of the strict local optima using this formula. The values of a and b were determined by least-squares regressions; the goodness-of-fit is reported using R-squared (R^2) [1]. In the case of $n = 100$, the randomly sampled instances are not very well spread across the CV values. This is believed to be due to the decrease in the statistical fluctuation of the CV values of the sampled weights, since they tend towards the theoretical values of the distributions CV as n increases.



(a) $k = 0.4$ (b) $k = 1$ (c) $k = 1.2$

Figure 4.11: The fraction of strict local optima versus CV. The results are for 600 instances of size $n = 20$ for each value of k. The solid lines were obtained using least-squares fit. Pearson's correlation coefficient r between the two quantities is shown for each plot.

[1] $R^2 = 1 - \mathsf{SS}_{\mathsf{residual}}/\mathsf{SS}_{\mathsf{total}}$.



(a) $n = 14$

(b) $n = 16$

(c) $n = 18$

(d) $n = 20$

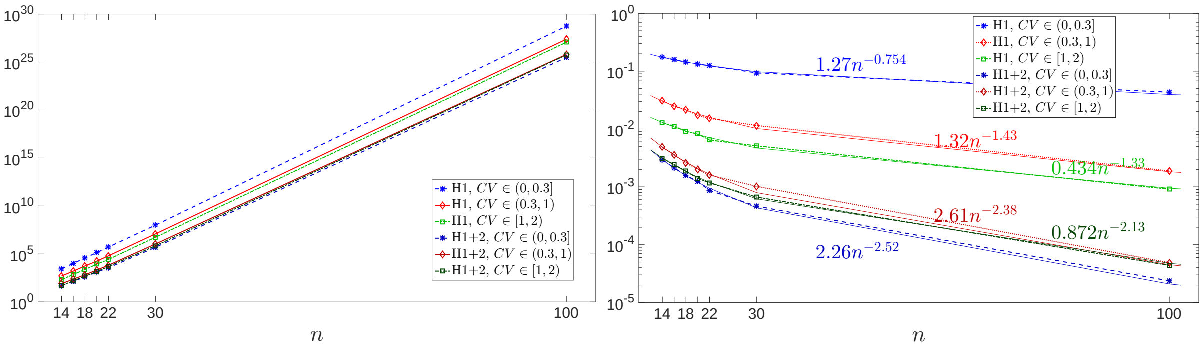
(e) $n = 22$

(f) $n = 30$

(g) $n = 100$

Figure 4.12: The fraction of strict local optima versus the CV values. The results are for 600 instances for each $n = 14, 16, 18, 20, 22$ and for 500 instances for $n = 30, 100$ ($k = 1$). The number of strict optima is estimated for $n = 30, 100$ using SRS, the sample sizes are $s = 10^5, 5 \times 10^5$ respectively. The solid lines were obtained using least-squares fit. Pearson's correlation coefficient r between the two quantities is shown for each plot.

To easily study the growth behaviour of the number of strict local optima as the problem size increases, we grouped the instances based on their CV values into three intervals: $(0, 0.3]$, $(0.3, 1)$, and $[1, 2)$. Figure 4.13 shows the growth of the number of strict local optima and the decay of its proportion against n. The number of strict local optima seems to grow exponentially with n in both landscapes and across all the intervals. The results for $n = 30, 100$ are the SRS estimates obtained with the sample sizes $s = 10^5, 5 \times 10^5$ respectively. All the proportions seems to decrease polynomially with n in the form an^{-b}. The largest decay happens in the landscape of *H1+2* and the smallest in the *H1* landscape of the interval $(0, 0.3]$. The proportion of the strict local optima appears to decay faster in the *H1+2* landscape compared to the *H1* across all the intervals.



(a) Log of mean number of strict optima.

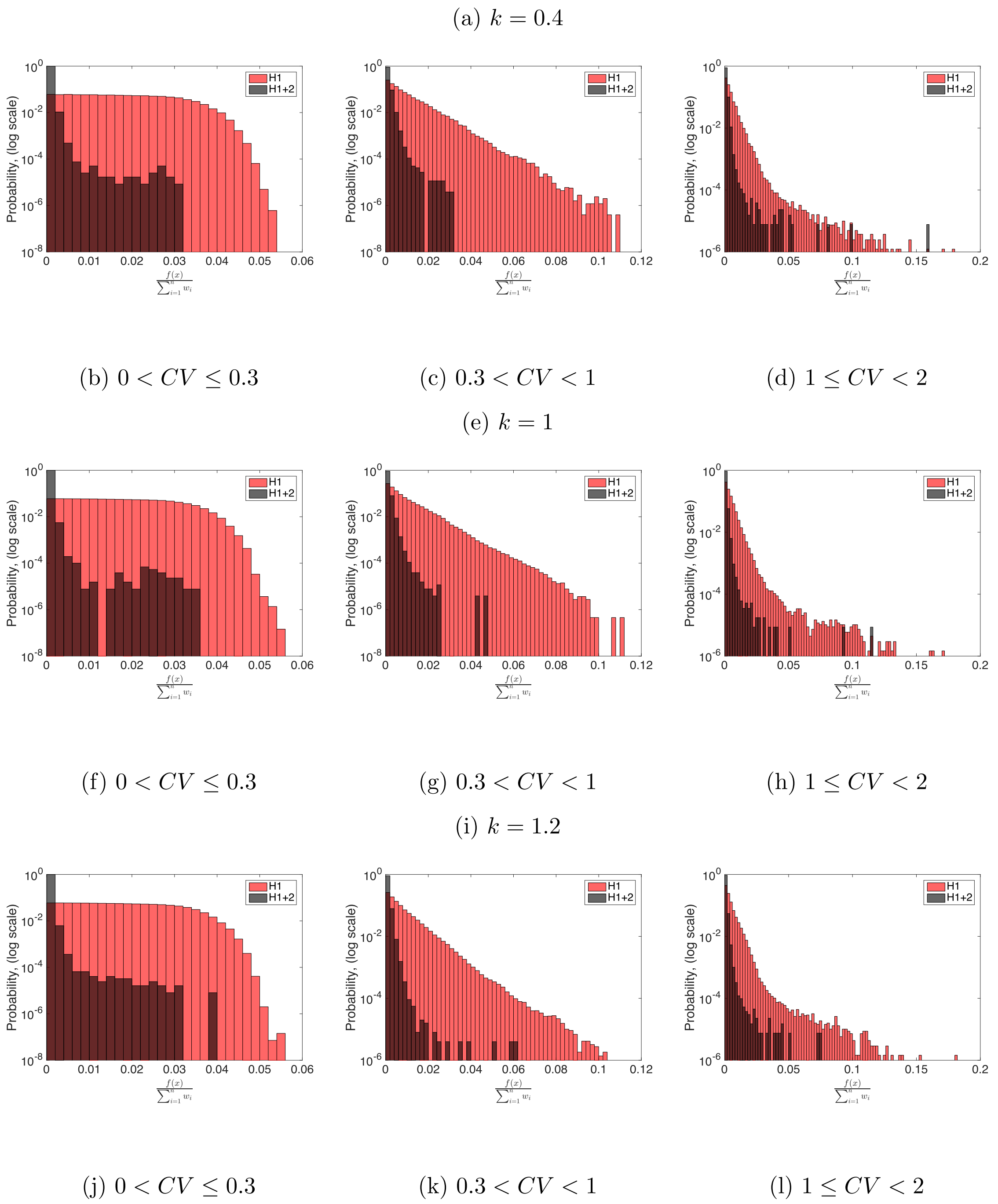
(b) Log of mean strict optima proportion.

Figure 4.13: The growth of the number of strict local optima and the decay of its proportion as the problem size n grows ($k = 1$). The results are averaged over 600 instances for each $n = 14, 16, 18, 20, 22$ and over 500 instances for $n = 30, 100$. The number of strict optima is estimated for $n = 30, 100$ using SRS, the sample sizes are $s = 10^5, 5 \times 10^5$ respectively. The solid lines in (b) were obtained using least-squares fit. Note that the proportion of the strict local optima in the *H1+2* landscape is always lower and decays faster in comparison to the *H1* landscape.

4.3.3 Quality of optima and plateaux

Here we examine the quality of the optima and plateaux between the two landscapes. Obviously, every optima in the *H1+2* landscape is an optima in the *H1* landscape. Thus, the quality of the optima in in the *H1+2* landscape is better or at least equal to that in the *H1*. However, we want to examine how the difference in quality between the two landscape changes across the CV values. To obtain a measure of quality that is independent of the problem instance and that does not require the knowledge the optimal solution, we measure the quality of an optimum x in a given instance as $f(x)/\sum_{i=1}^{n} w_i$. The quality of all the found optima and plateaux in different instances of size $n = 14, 16, 18$ are shown in figure A.1 and A.2. The quality of the optima in the *H1* gets better with very large CV values, apart from very few bad optima. The number of optima with bad quality in the *H1* landscape of instance with very small CV is very high. In figure 4.14 we grouped the instances again into the three intervals of CV values: $(0, 0.3]$, $(0.3, 1)$, and $[1, 2)$. We can see that the number of optima with bad quality decreases as the CV values increases. The apparent similarity of the histograms across the values of the control parameter k

indicates that this behaviour occurs in both easy and hard phases of the problem.


(a) $k = 0.4$

(b) $0 < CV \leq 0.3$

(c) $0.3 < CV < 1$

(d) $1 \leq CV < 2$

(e) $k = 1$

(f) $0 < CV \leq 0.3$

(g) $0.3 < CV < 1$

(h) $1 \leq CV < 2$

(i) $k = 1.2$

(j) $0 < CV \leq 0.3$

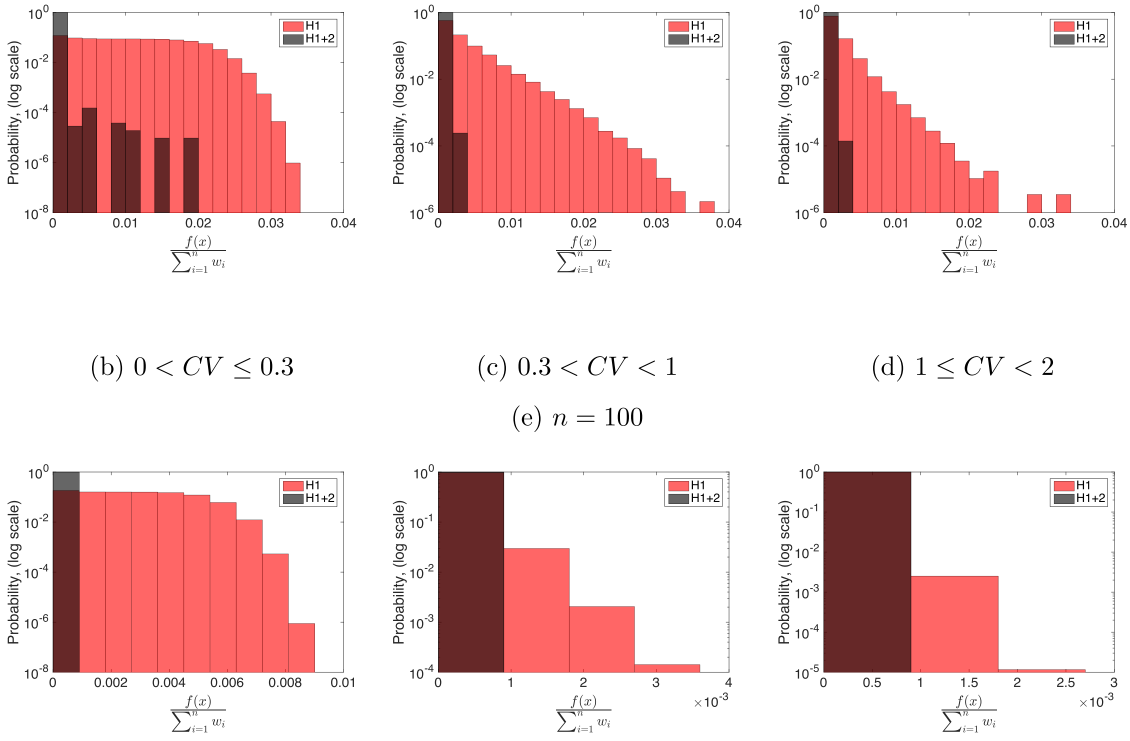
(k) $0.3 < CV < 1$

(l) $1 \leq CV < 2$

Figure 4.14: The quality of optima and plateaus in the *H1* and *H1+2* landscapes. The x-axis shows the fitness value divided by $\sum_{i=1}^{n} w_i$. The data includes all optima and plateaux found in 600 instances for each k value of problem size $n = 20$.

We continue to observe this phenomenon in sampled optima from larger instances as shown in figure 4.15 for $n = 30, 100$. The optima obtained for each instance are the collection of optima sampled by 1000 random steepest descents and SRS with sample sizes of $s = 10^5, 5 \times 10^5$ for $n = 30, 100$ respectively.

(a) $n = 30$



(b) $0 < CV \leq 0.3$ (c) $0.3 < CV < 1$ (d) $1 \leq CV < 2$

(e) $n = 100$

(f) $0 < CV \leq 0.3$ (g) $0.3 < CV < 1$ (h) $1 \leq CV < 2$

Figure 4.15: The quality of optima in the *H1* and *H1+2* landscapes ($k = 1$). The x-axis shows the fitness value divided by $\sum_{i=1}^{n} w_i$. The data includes all sampled optima from 500 instances for each n. The sampling for each instance includes 1000 steepest descents and SRS of size $s = 10^5, 5 \times 10^5$ for $n = 30, 100$ respectively.

4.4 Basins of Attraction

The basin of attraction size, shape and the correlation between the size and the optimum fitness represent important aspects of the fitness landscape [85, 104]. In this section we try to examine most of these properties across the different problem parameters and compare the results of the two landscapes. We exhaustively calculate the basin sizes, thus we were limited to studying small problem sizes $n \leq 22$ only.

4.4.1 Basin size

As with the number of strict optima, the basin sizes do not seem to change much between the easy and hard phase for all the different distributions and for both landscapes. Figures 4.17 and 4.16 show the average basin sizes for the different distributions across k. The figures show the average sizes of basins associated with each strict and non-strict local optima (i.e in case of a plateau, the basin of each of its configuration is counted as a single observation when calculating the average, instead of counting their union as a single observation). There is a large increase in the average basin size in the *H1+2* landscape in comparison to the *H1* landscape. This increase is in accordance with the decrease in the number of local optima for all the different distributions and for all the different values of k.

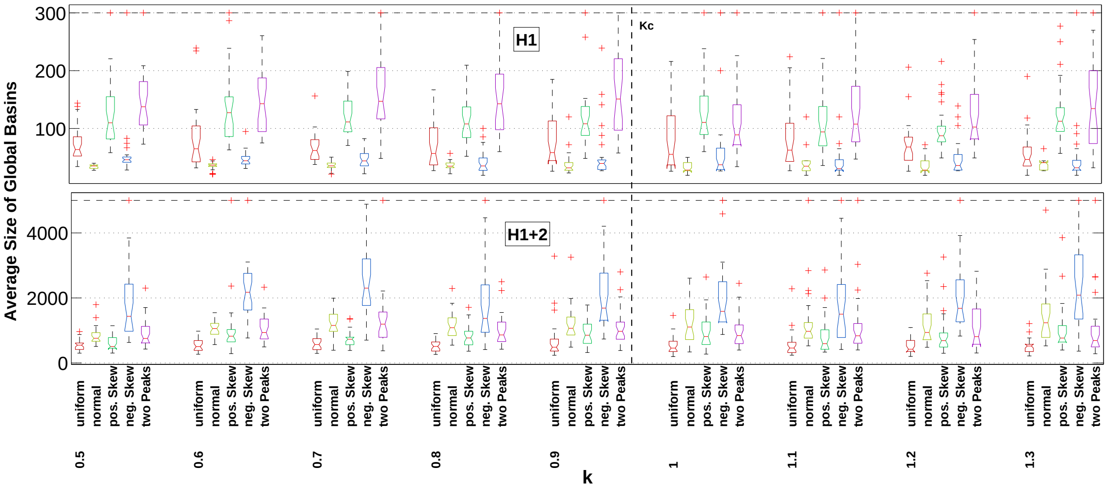


Figure 4.16: Average basin size of global optima versus the phase transition control parameter k, for all the different distributions of the weights. Each box represents data from the 30 random instances of size $n = 20$. This is shown for both neighbourhood operators *H1* and *H1+2*. The dotted line is given by k_c from eq.(4.2).

The distribution of the basin sizes in all the instances we studied was found to be highly skewed to the right, with many small basins and only few large ones. This is true for both landscapes. Similar skewness in the distribution of basin sizes was reported in other combinatorial problems, for example in the flow-shop scheduling problem where the

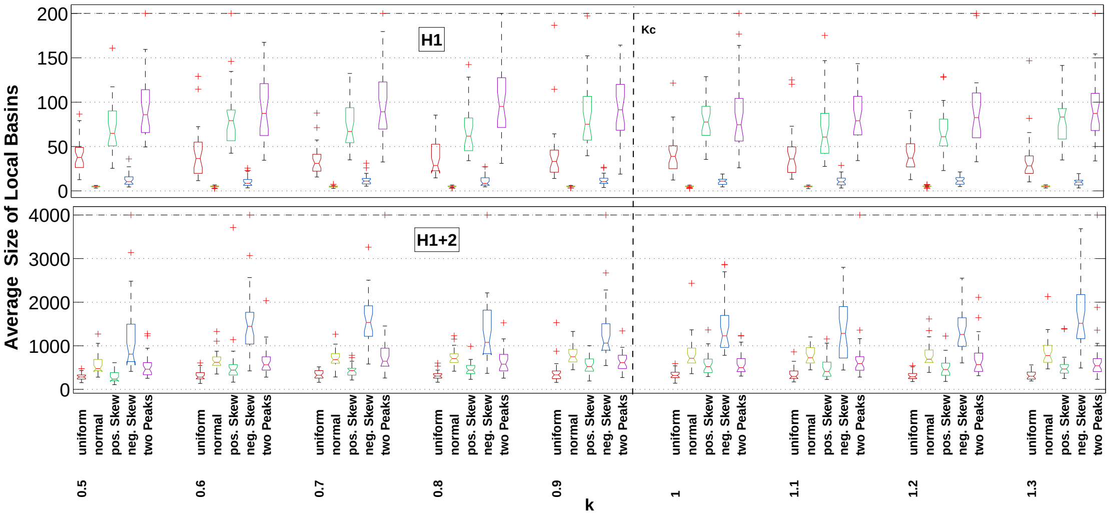
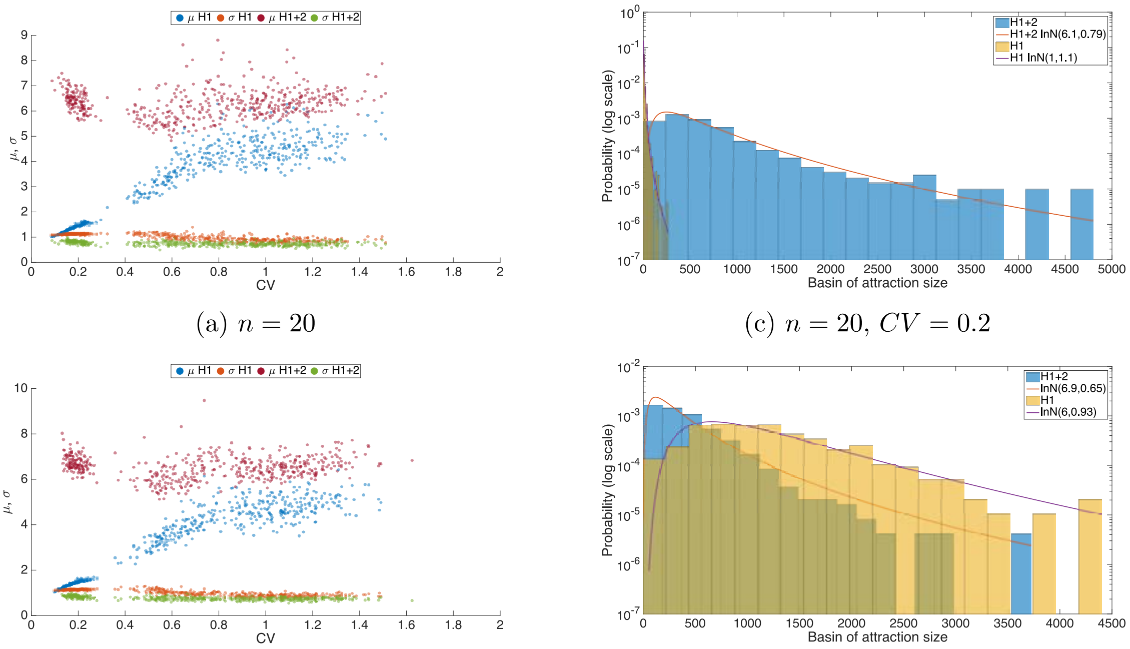


Figure 4.17: Average basin size of local optima versus the phase transition control parameter k, for all the different distributions of the weights. Each box represents data from the 30 random instances of size $n = 20$. This is shown for both neighbourhood operators *H1* and *H1+2*. The dotted line is given by k_c from eq.(4.2).

log-normal distribution was found to be a plausible model of the basin sizes [91]. We tried to fit the log-normal distribution to the instances we studied, to examine if it is also a plausible model for the basin sizes in NPP. Figure 4.18 shows the fitted log-normal distribution on two instances of small and large CV values. The figure also shows how the estimated distribution parameters (i.e. μ and σ) change with the CV. There is almost no change in the *H1+2* landscape, but in the *H1* landscape μ increases with the CV, while σ decreases. This increase in the basin sizes can be attributed to the decrease in the number of optima as the CV increases. However, in almost all the instances, the null hypothesis was rejected (at the 5% significance level) when chi-squared and Kolmogorov-Smirnov goodness-of-fit tests were used to test the plausibility of the log-normal distribution. We tried to fit what we believed to be other possible good models, namely the following distributions: inverse Gaussian, exponential, Poisson, gamma, negative binomial and power law (we followed the method in [19] to test the goodness-of-fit of a power law). The best visual fits we found were gamma, log-normal, and negative binomial. But again in almost all the instances, all the tested distributions failed the goodness-of-fit tests.

This can be due to a number of reasons: statistical fluctuation (especially in large CV instances where the number of optima is small), the presence of outliers, different degree of freedom should have been used, or simply because the basin sizes do not actually follow any of those distributions. Identifying and removing the outliers and examining the use of different degrees of freedom could be investigated in the future.



(a) $n = 20$

(c) $n = 20$, $CV = 0.2$

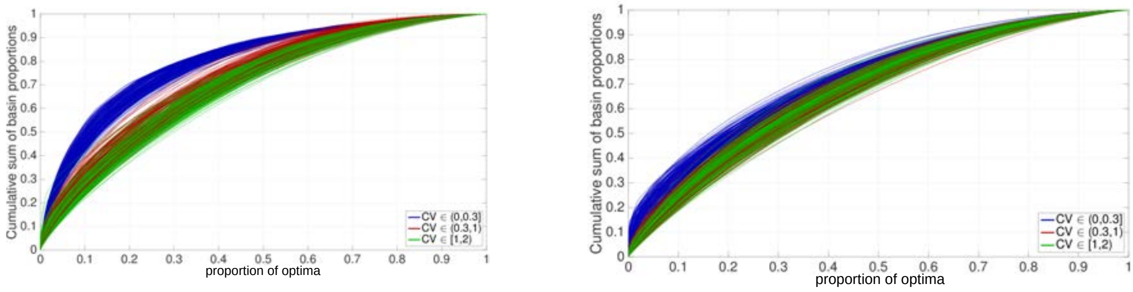
(b) $n = 22$

(d) $n = 20$, $CV = 1.1$

Figure 4.18: (Left) The mean and the standard deviation of the fitted log-normal distributions to the basin sizes against the CV. The results are for both landscape of 600 instances for each $n = 20, 22$ and $k = 1$. (Right) Histograms of the basin sizes of two instances with small and large CV values; the fitted log-normal distribution is shown for each landscape.

In the *H1* landscape, 20% of the basins cover around 60-70% of the search space in instances with $CV \leq 0.3$, while they cover around 35-50% of the search space in instances with $CV > 1$. This discrepancy between the two CV intervals seems to continue to exist in the *H1+2* landscape, though at a lower level. Figure 4.19, shows the cumulative sum of the basin proportions, after being sorted in descending order, against the percentage of the basins. The search space is covered by fewer basins in the *H1* landscape compared to the *H1+2* one, in instances with $CV \leq 0.3$, in particular. For instance, around 90% of the search space is covered by half of the basins in the *H1* landscape, while it takes around

70% of the basins to cover the same amount of the search space in the *H1+2* landscape.



(a) *H1* landscape

(b) *H1+2* landscape

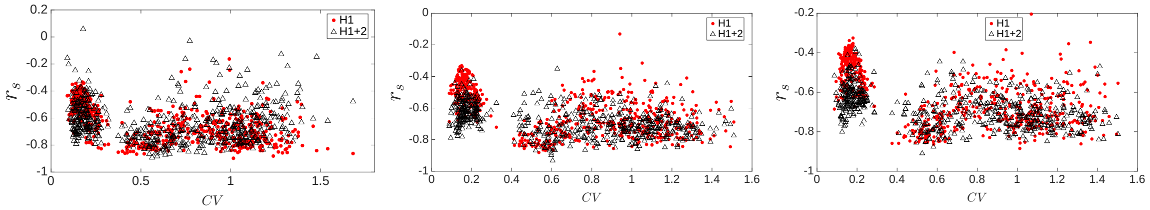
Figure 4.19: To examine how quickly the largest basins cover the search space we plot the cumulative sum of the basin proportions going from the largest to smallest (i.e. we plot the cumulative sum of the basin proportions after sorting them in descending order) against the proportion of the optima. Each line shows the results of a single instance. The results are for 600 instances of $n = 20$ and $k = 1$. You can see that in instances with $CV \leq 0.3$, the search space is covered by fewer basins in the *H1* landscape compared to the *H1+2* one.

4.4.2 Basin size and fitness

Another important aspect of the fitness landscape is the correlation between the basin size and the fitness of the optimum. Previous studies have shown that in general, fitter optima have larger basins [104, 112], and landscapes with this kind of feature usually tend to be easier to search. In general, the correlation between the fitness and the basin size in both landscapes of NPP (see figures 4.20 and 4.21) was found to be moderately negative $(0.4 - 0.6)$ to strongly negative (> 0.6). This indicates that fitter optima do indeed tend to have bigger basins in NPP. We measured the correlation between the two quantities using Spearman's correlation coefficient instead of the traditional Pearson's correlation coefficient. The reason for that is, Pearson's method assumes that both variables are drawn from normal distribution, while Spearman's method is non-parametric; as we have seen in the previous sections, the distribution of the fitness values and the basin sizes are highly skewed and far from being normally distributed in NPP [1], thus Spearman's correlation coefficient is more suitable in this case.

Like with the basin size, the correlation between the fitness and basin size does not seem to change much between the easy and hard phases, at least for instances with $CV > 0.3$ in the *H1* landscape. However, some instances with $CV \leq 0.3$ seem to have stronger correlations in the easy phase. In the *H1+2* landscape, a small proportion of instances have weak negative correlation in the easy phase. This changes in the hard phase where we see that all the instances have moderate to strong correlation. The degree of the negative correlation seems to remain more or less the same across the CV values in the *H1+2* landscape. The same applies to the *H1* landscape, except again for instances with $CV \leq 0.3$, where they seem to have weaker correlation than the rest, particularly in the hard phase. As a reminder, in the context of the weight distributions, instances with $CV \leq 0.3$ are generated from normal and negatively skewed distributions.

Figure 4.21 shows the correlation between the fitness and basin size as n grows. In general, the correlation seems to get slightly stronger as n increases. Again, however, the correlation of instances with $CV \leq 0.3$ in the *H1* landscape seems to get weaker with larger n. Of course, as a consequence of studying only small instances, we cannot know with certainty if this trend will continue to show in n larger than 22.



(a) $k = 0.4$

(b) $k = 1$

(c) $k = 1.2$

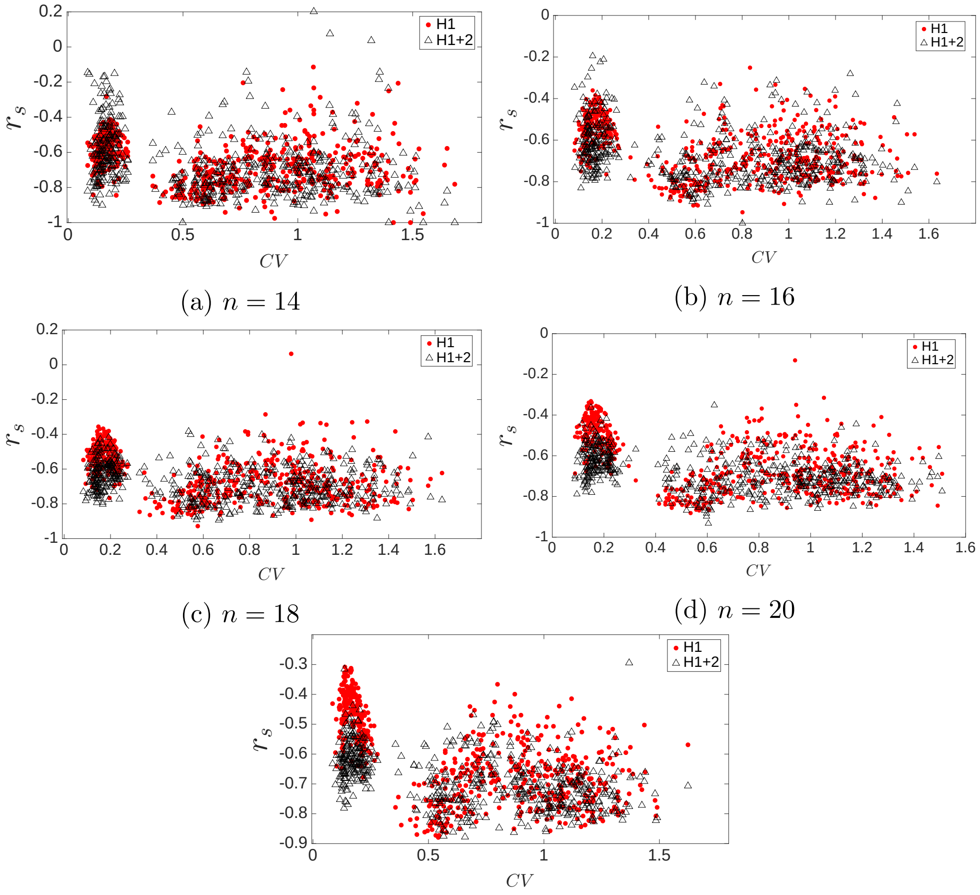
Figure 4.20: Spearman's rank correlation coefficient between basin size and fitness versus CV. The results are for 600 instances of size $n = 20$ for each value of k.

In summary, if we excluded the *H1* with $CV \leq 0.3$, the correlation appears to remain more or less the same across k and CV apart from few cases in the *H1+2* landscape. It

[1] You can also see figures A.3, A.4, and A.5 in the appendix, which show plot-matrices of the basin size proportions and optima fitness values for three instances of different CV values.

also appears to get slightly stronger with larger n. For the *H1* landscape of instances with $CV \leq 0.3$, the correlation seems to be slightly stronger in the easy phase and it seems to get weaker with larger n in the hard phase.

Figures A.6, A.7, and A.8 in the appendix give an overview of the relation between the basin size and fitness in all the instances we studied. They show the skewness of the distributions of the fitness values and the basin sizes, namely that most of the basins are small and that most of the optima are good (apart from the ones in the *H1* landscape with $CV < 0.3$). They also show how the basin sizes in the *H1* landscape increases with the CV until their sizes become similar to those of the *H1+2* landscape.



(a) $n = 14$

(b) $n = 16$

(c) $n = 18$

(d) $n = 20$

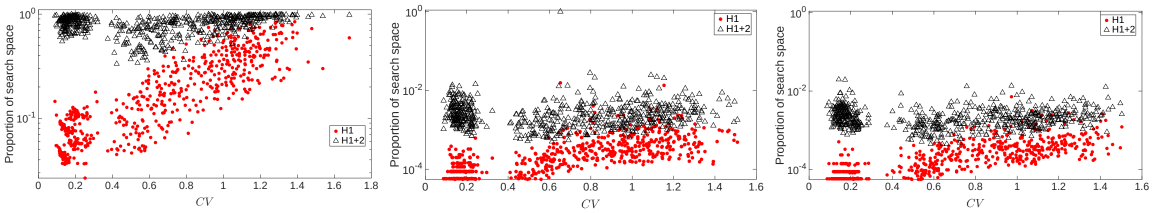
(e) $n = 22$

Figure 4.21: Spearman's rank correlation coefficient between basin size and fitness versus CV. The results are for 600 instances for each problem size ($k = 1$).

4.4.3 Global basin

As we have seen before, the number of global solutions drops down as we cross the phase transition point. To examine how much the probability of finding the global changes between the two phases, we plot the total sum of all the global basin proportions found in instances of $n = 20$ against the control parameter k in figure 4.22. In general, the probability of finding the optimal solution is always higher in the *H1+2* landscape than in the *H1* one. In the *H1+2* landscape, the probability of finding the global optima drops down from almost around ~ 1 in the easy phase to around $\sim 10^{-2}$ in the hard phase. Like the previously studied features, the probability does not seem to change much across the CV values in this landscape, unlike the *H1* landscape, where the probability of finding the global increases with the CV in both the easy and hard phases. Again, in *H1* landscape the probability of finding the global decreases from between around ~ 0.1 in small CV instance and ~ 0.8 in large CV instances in the easy phase to be around $\sim 10^{-4}$ and $\sim 15 \times 10^{-3}$ in the hard phase. Figure 4.23 shows that the probability of finding the global in the hard phase, in both landscapes, decreases as the problem size grows.

In an attempt to study the shape of the global basin in the hard phase, we plot in figure 4.24, the proportion of the configurations that are part of its basin in every Hamming sphere of radius h around it. The proportions were estimated as described in subsection 3.4. The plot shows the result for one of the two global found, as the same result applies to the other global due to the symmetry of the search space. The results are shown for three instances of size $n = 22$. Note that the global basin is not the largest in all of these instances. For example, in the instance with the smallest CV value, the global basin proportion in the *H1* landscape is 1.98×10^{-05} while the largest basin proportion is 1.86×10^{-04}. Similarly in the *H1+2* landscape the global basin proportion is 0.001 while the largest basin proportion is 0.009. The probability of return estimated using this method was very close to the true probability that was obtained by exhaustive calculation of the basins. From the figure we can see that in both landscapes the configurations in the global basin are concentrated in the immediate Hamming spheres around it.

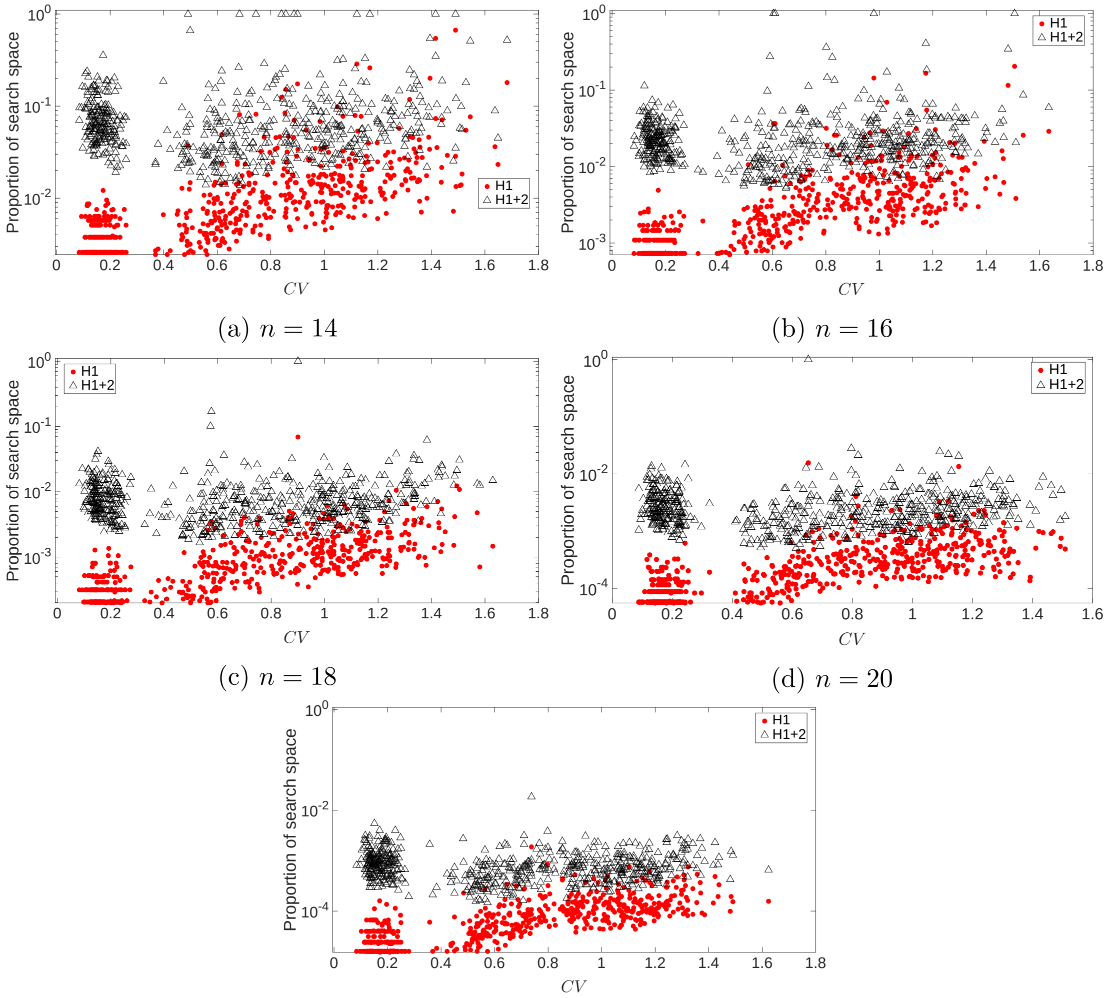


(a) $k = 0.4$

(b) $k = 1$

(c) $k = 1.2$

Figure 4.22: The proportion (in log scale) of the basin size of all the global optima found in an instance for each landscape against the CV. The results are for 600 instances of size $n = 20$ 0for each value of k. Notice how the probability of finding the global optimum increases with the CV in the *H1* case.



(a) $n = 14$

(b) $n = 16$

(c) $n = 18$

(d) $n = 20$

(e) $n = 22$

Figure 4.23: The proportion (in log scale) of the basin size of all the global optima found in an instance for each landscape against the CV. The results are for 600 instances of each problem size ($k = 1$). Notice how in all of the sub-figures the probability of finding the global optimum increases with the CV in the *H1* case.

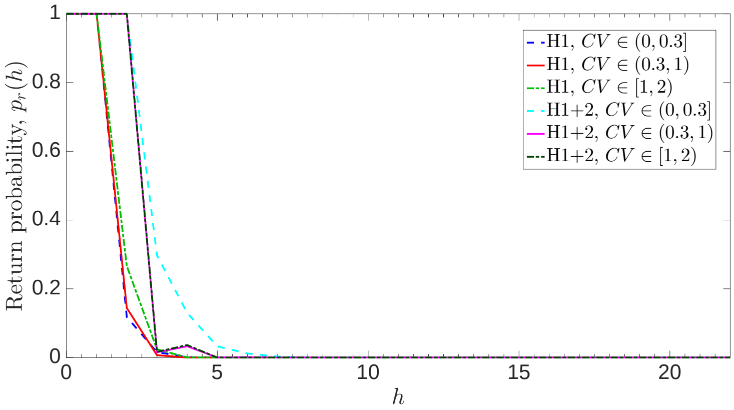


Figure 4.24: Return probability to the global optimum starting from a Hamming sphere of radius h versus h. The results are for 3 instances of size $n = 22$ and $k = 1$. Notice how the probability of return approaches zero slightly faster in the *H1* case compared to *H1+2*.

4.5 Local Search

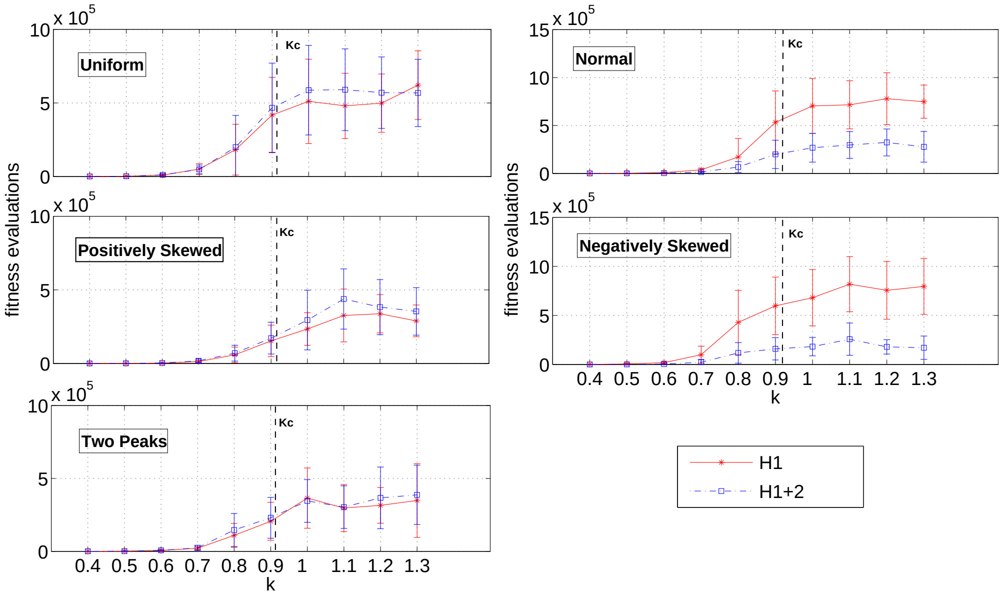


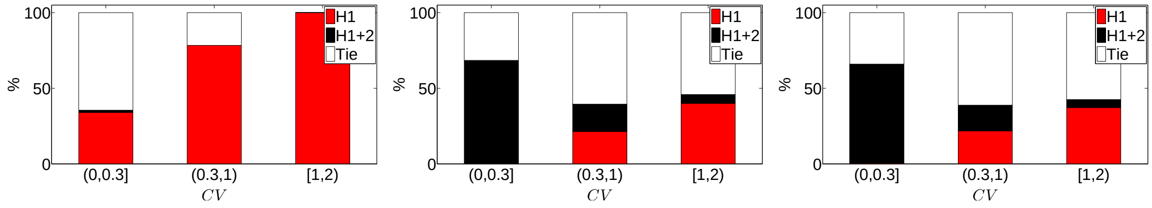
Figure 4.25: Average number of fitness evaluations used to find the global optimum, plotted against the phase transition control parameter k. This is shown for all the considered distributions of the weights and for both neighbourhood operators *H1* and *H1+2*. Each data point represents the average over the 30 instances of size $n = 20$ and 100 runs of the steepest descent algorithm per instance. The dotted line is given by k_c from eq.(4.2).

We study in this section the performance of local search, namely steepest descent with random restart algorithm, using the two neighbourhood operators. We carry out the analysis of the algorithm performance from the perspective of the previously studied landscape features in the earlier sections.

4.5.1 Cost of finding the global

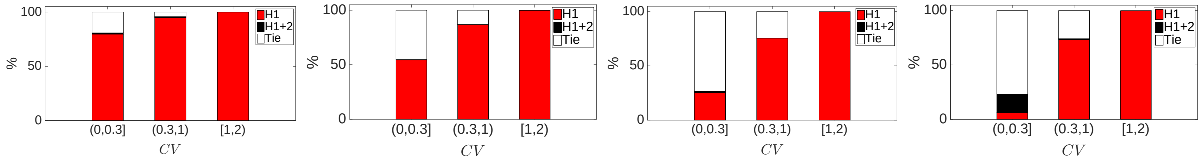
To examine how the cost of finding the optimal solutions varies from the easy phase to the hard phase, we ran the algorithm with the two neighbourhood operators for 100 times for each instance. The cost of finding the global optima is then calculated using the number of used fitness evaluations. Note that we treat the objective function as a black-box here, hence the number of times the objective function is queried for each step taken by the algorithm equals the size of the neighbourhood. Also note that treating the objective function of this problem as a white-box would achieve significantly better results in terms of the cost. Figure 4.25 shows the average cost of finding the global against k for each distribution of the weights. For all the different distributions, the figure shows that the average number of fitness evaluations used to find the global optima increases as we approach the phase transition point and keep increasing as we cross the phase transition. This is expected due to the drastic decrease in the number of global optima in the hard phase. As we have seen before, the probability of finding the global is higher in the easy phase, the algorithm quickly finds one of the many global optima while it struggles to find the single (two if we considered the symmetry) global optimum in the hard phase. The number of used fitness evaluations varies across the different distributions. Instances drawn from positively skewed and two peaks distributions have the lowest number of fitness evaluations, which is unsurprising due to the low number of local optima in the landscape of both cases. For instances drawn from normal and negatively skewed distributions the performance of the *H1+2* operator was much better than the performance of the *H1* operator. This can be explained by the very big difference between the number of local optima in the *H1+2* landscape compared to the *H1* landscape which has far more local optima, suggesting that the algorithm probably had to do far less restarts when using the *H1+2* operator. For the rest of the distributions, the *H1* operator seems to have a better performance even though the number of local optima is less in the landscapes induced by the *H1+2* operator. This perhaps can be explained by the number of fitness evaluations needed to explore the much larger neighbourhood of the *H1+2* operator, which might have offset the advantage of having lower number of local optima.

To examine further this relation between the number of fitness evaluations needed to explore the neighbourhood and the difference between the number of local optima between the two landscapes: we compare the performance of the two operators in figure 4.26, where we determine the statistical significance between the two performances using Wilcoxon rank-sum test at the 5% level. Note that now the instances are mapped into the three CV intervals. We can see that in the hard phase, the *H1* operator performs better when the CV is large ≥ 1. This behaviour continues to show as n grows, as shown in figure 4.28. In this interval, although the number of local optima is higher in the *H1* landscape and the probability of finding the global is lower than that in the *H1+2*, the number of local optima is still small enough for the algorithm with the *H1* operator to carry out a number of restarts until the global is found and still use lower number of fitness evaluations than that used by the *H1+2* operator. In the easy phase, the *H1* operator performs better across all the CV intervals, despite the fact that the *H1+2* landscape is always smoother and has a higher probability of finding the global across all the values of the CV. This can be explained by the presence of many global optima in the easy phase which mitigates the ruggedness of the *H1* landscape, even in the very rugged landscape of the $(0, 0.3]$ CV interval. However, this behaviour seems to fade away as n grows as shown in figure 4.27, where the *H1+2* operator starts to win more instances. This can be attributed to the growth of the number of local optima in the *H1* landscape in the $(0, 0.3]$ interval, which is the fastest growth rate out of all the CV intervals, while the same interval has the lowest growth rate in the *H1+2* landscape.



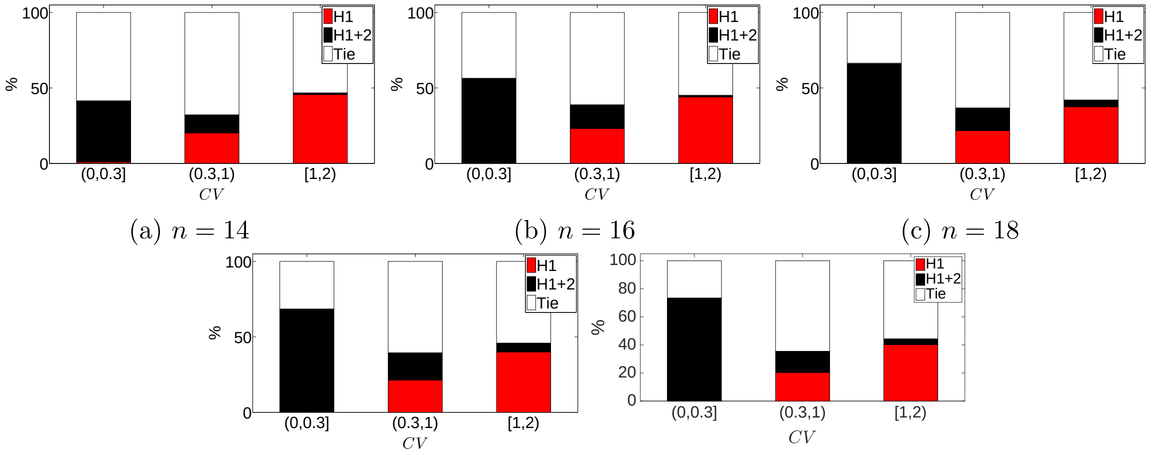
(a) $k = 0.4$ (b) $k = 1$ (c) $k = 1.2$

Figure 4.26: Number of fitness evaluations used to find the global optimum averaged over 30 runs. The results show the percentage of instances where each operators performed significantly better and the percentage where no significance difference was found (Tie). Significance determined using Wilcoxon rank-sum ($p-$value ≤ 0.05). The results are for 600 instances of size $n = 20$ for each k.



(a) $n = 14$ (b) $n = 16$ (c) $n = 18$ (d) $n = 22$

Figure 4.27: Number of fitness evaluations used to find the global optimum averaged over 30 runs. The results show the percentage of instances where each operators performed significantly better and the percentage where no significance difference was found (Tie). Significance determined using Wilcoxon rank-sum ($p-$value ≤ 0.05). The results are for 600 instances for each n and $k = 0.4$.

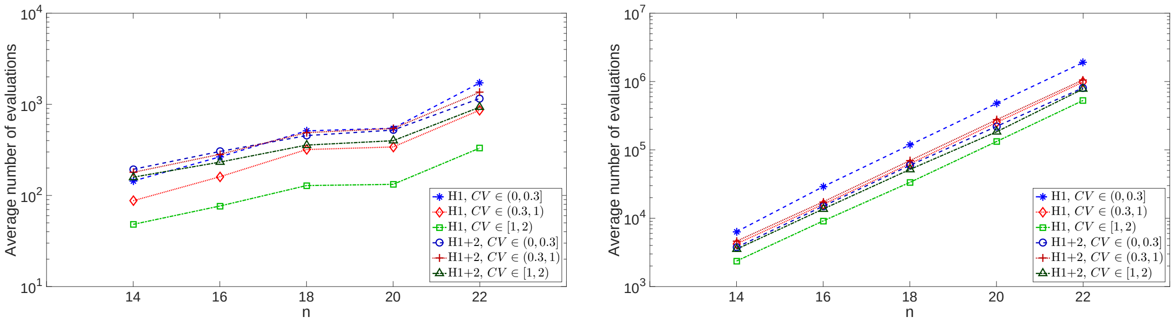


(a) $n = 14$ (b) $n = 16$ (c) $n = 18$

(d) $n = 20$ (e) $n = 22$

Figure 4.28: Number of fitness evaluations used to find the global optimum averaged over 30 runs. The results show the percentage of instances where each operators performed significantly better and the percentage where no significance difference was found. Significance determined using Wilcoxon rank-sum ($p-$value ≤ 0.05). The results are for 600 instances for each n ($k = 1$).

The cost of finding the global optimum grows exponentially with n in the hard phase as figure 4.29 depicts. The growth in the easy phase seems to be much slower but we are unable to comment on its growth type as the trend is not very clear from the data in this case. Bear in mind that the average number of evaluations used to find the optimal is reported in log scale in the figure. You can see the big difference in the cost finding the global optimum (almost one order of magnitude) between instances with small $CV < 0.3$ and instances with large $CV \geq 1$ when *H1* operator is used.



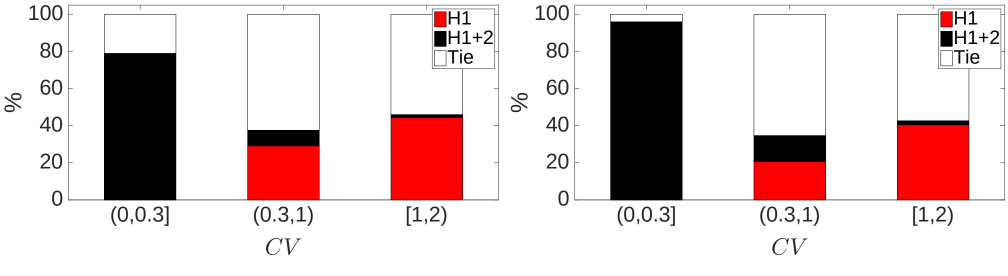
(a) $k = 0.4$

(b) $k = 1$

Figure 4.29: Average number of fitness evaluations used to find the global optimum (y-axis) against the problem size n (x-axis). The y-axis is in log scale. Each data point is an average of 30 runs of steepest descent, averaged over the number of instances in each CV interval. The results are for 600 instances for each n. Notice the large difference in the average number of evaluations (almost one order of magnitude) between instances with $CV < 0.3$ and $CV \geq 1$ when *H1* operator is used.

4.5.2 Quality of optima obtained with fixed budget search

The trend of the *H1+2* operator performing better in the $(0, 0.3]$ CV interval and the *H1* operator performing better in the $[1, 2)$ CV interval continues in larger problem sizes of $n = 30, 100$ as shown in figure 4.30. Now the results show the quality of optima obtained by a fixed arbitrarily selected budget of fitness evaluations. As we have seen before, the difference in the quality between the two landscape decreases as CV grows, which explains the results for the wining case of the *H1* operator, of course alongside the lower difference in the number of local optima in this interval between the two landscape. Note that these results are specific to the budget we selected, whether the same trends will continue to occur with other budget values remains an open question.



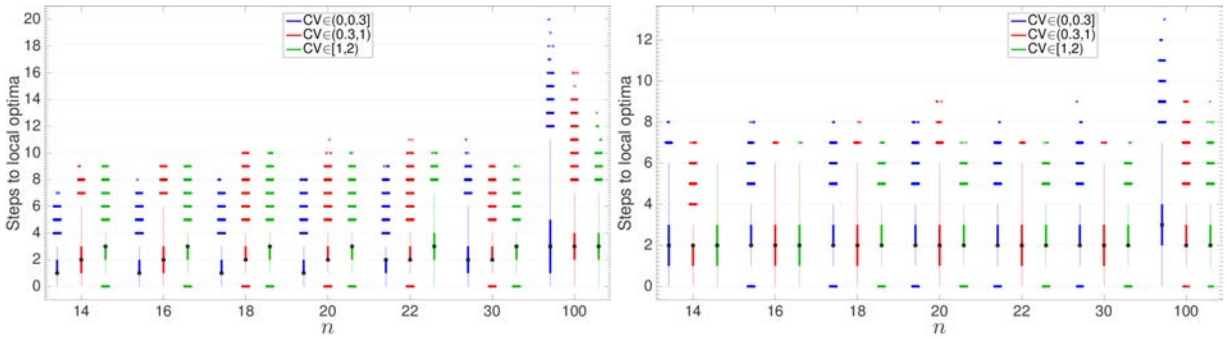
(a) $n = 30$

(b) $n = 100$

Figure 4.30: The quality of the solution found averaged over 30 runs of local search with fixed budget of 10^5 fitness evaluations. The results are for 500 instance per problem size ($k = 1$).The results show the percentage of instances where each operators performed significantly better and the percentage where no significance difference was found (Tie). Significance determined using Wilcoxon rank-sum ($p-$value ≤ 0.05).

4.5.3 Time to local optima

Here we study the time it takes steepest descent, starting from a random configuration until a local optimum is found. As shown in figure 4.31, this was found to be very small in both landscapes. In the *H1* landscape, we believe that this is due to the large number of local optima in this landscape. In the *H1+2* landscape, although the number of optima is much smaller than that in the *H1*, its neighbourhood is larger, which explains the small number of steps in this landscape. Also, the small number of steps can be explained by the fact that the attraction basin sizes in both landscapes were found to be mainly small. The number of steps grows very slowly with n in both landscapes and across all the CV values. This can be attributed to the exponential growth of the number of local optima in both landscapes and across all the CV values. Note that the number of steps taken in the *H1* landscape was found to be always equal to the Hamming distance between the initial random configuration and the found local optimum. In the *H1+2* landscape this was found to be almost always smaller or equal to the Hamming distance between the initial random configuration and the found local optimum. However, in extremely few cases, it was found to be one or two steps larger than the Hamming distance.



(a) *H1* landscape

(b) *H1+2* landscape

Figure 4.31: Number of steps starting from a random configurations until an optimum is reached. The results are for 1000 steepest descents per instance and 600 instances for each $n = 14, 16, 18, 20, 22$ and 500 instances for $n = 30, 100$ ($k = 1$).

4.6 Summary

In this chapter, we empirically studied various properties of two fitness landscapes of random instances of the NPP, with a focus on how these properties change with the phase transition and the CV value. The following is a summary of the main results:

- No configuration of type IPLAT or NSLMAX has been found in either landscapes.
- The only two properties that were found to change when the problem crosses the phase transition, apart from slight changes in the correlation between the basin size and fitness in instances when $CV \leq 0.3$, are the number of global optima (and consequently the probability of finding the global) and the number of plateaux, the rest of the properties remained oblivious to the phase transition. This result is in agreement with the results obtained by Stadler et al. [105], in which they found that the features of the uniform NPP landscape, that have been mapped into barriers trees, are insensitive to the phase transition.
- In general, the sizes of plateaux in the *H1+2* landscape is larger than that in *H1*.
- The number of strict local optima seems to grow exponentially with the problem size in both landscape, with a faster rate in the *H1* landscape.

- We proposed a formula to estimate the average number of local optima in the *H1* landscape that depends only on the problem size and the CV of the weights, exploiting the strong correlation between the CV and the number of local optima in this landscape.

- The quality of optima is always better in the *H1+2* landscape, but the difference in quality between the two landscape decreases as the CV increases.

- In general, the correlation between the size of the basin of attraction of a local optimum and its fitness was found to be strong and negative, indicating that fitter optima tend to have larger basins.

- The distribution of the basin sizes was found to be skewed with many small basins and only few large ones.

- The performance of local search algorithms was found to be affected by the phase transition in NPP, as shown by the considerable increase in the cost of locating the optimal solution when $k > k_c$.

- The *H1* operator performs better in instances with $CV \geq 1$, while the *H1+2* operator performs better in instances with $CV \leq 0.3$. This shows that the CV of the weights has a potentially useful application in guiding the choice of the move operator of local search heuristics.

- The number of steps until an optimum is found starting from a random configuration grows very slowly with n in both landscapes.

CHAPTER 5

0-1 KNAPSACK PROBLEM

In this chapter we study the landscape properties of another NP-hard problem, the binary knapsack problem (0-1KP) [2]. We study 11 different problem types of the problem, which vary in the relation between the problem coefficients (i.e. the profits and the weights). One of these problem types is a generalisation of the NPP. As in the NPP chapter, we study various landscape features of a large number of randomly generated instances with different values of problem parameters. One of these parameters is the weights distribution. Instances were generated by drawing weights from the five different distributions shown in subsection 3.2. However, in this chapter, we abandon the use of the underlying distribution of the weights to describe the problem instance and only use the weights CV to do so. We also carry out grouping the instances based on their CV values into the three intervals: $(0, 0.3]$, $(0.3, 1)$, and $[1, 2)$.

5.1 Problem Definition

Given a knapsack of capacity C and a set of n items each with associated weight w_i and profit p_i, the aim is to find a subset of items that maximises

$$f(x) = \sum_{i=1}^{n} x_i p_i \tag{5.1}$$

[2] Initial work presented in this chapter was published in GECCO'15 companion proceedings [6]

subject to:

$$\sum_{i=1}^{n} x_i w_i \leq C, \ x \in \{0,1\}^n \tag{5.2}$$

where

$$C = \lambda \sum_{i=1}^{n} w_i, \ 0 \leq \lambda \leq 1 \tag{5.3}$$

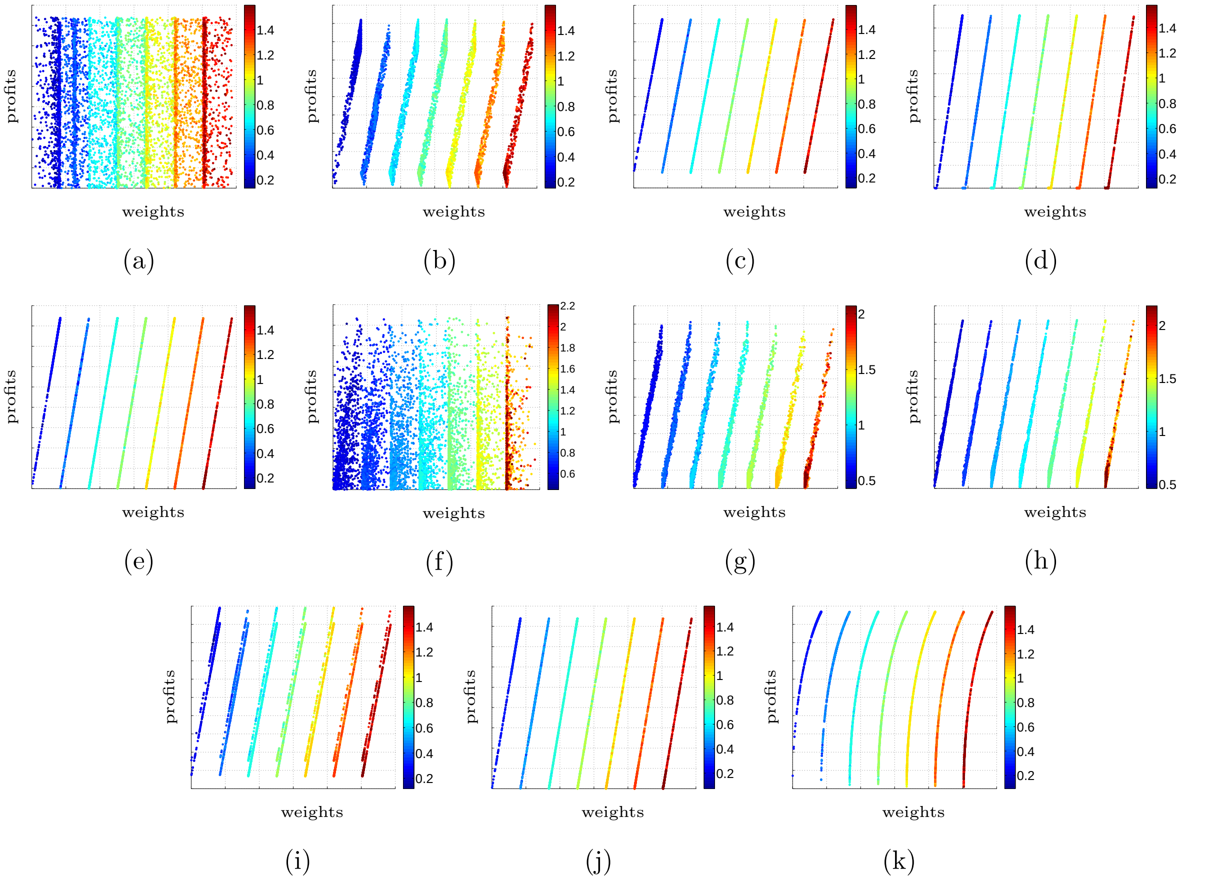
The binary vector $x = (x_1, \ldots, x_n)$ represents the decision variable where $x_i = 1$ when item i belongs to the subset and $x_i = 0$ otherwise. We study instances where p_i and w_i are positive integers drawn from the set $\{1, 2, \ldots, M\}$.

The 0-1KP is NP-hard in the weak sense [35], that is, there exists an algorithm that can solve it in pseudo-polynomial time through the use of dynamic programming. The complexity of such an algorithm, $\mathcal{O}(n2^{\log_2 C})$, is polynomial in the number of items and the capacity of the knapsack but exponential in the number of bits required to represent the capacity. The running time of such an algorithm would exhibit an exponential behaviour as M grows large.

Note that the 0-1KP search space, $X = \{0,1\}^n$, is partitioned into a feasible region $F = \{x \in X \mid \sum_{i=1}^{n} x_i w_i \leq C\}$ and an infeasible region $\mathit{INF} = X \setminus F$. For $\lambda = 1$ there are no infeasible solutions and as the value of λ decreases the size of the infeasible region increases until $\mathit{INF} = X$ when $\lambda = 0$. We define the boundary between feasible and infeasible regions as the set of feasible configurations that have at least one infeasible neighbour, $B = \{x \in X \mid x \in F \wedge \exists y : (y \in N(x) \wedge y \in \mathit{INF})\}$.

5.1.1 Problem types

Randomly generated instances of the 0-1 KP can be classified into different types based on the relation between the item's profit and weight. We study 11 types, which have been the focus of several studies in the literature, each with different properties that could influence the performance of problem solvers [82, 83, 64, 14]. Given a positive integer a and w_i drawn at random from a given data range $[1, M]$, the profit p_i can be expressed as a function of w_i yielding the following instance types:



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k)

Figure 5.1: The relation between weights (x-axis) and profits (y-axis) in the different types of knapsack instances. The different colours indicate different CV values of the weights. The weights have been shifted along the x-axis for better visibility. (a) Uncorrelated. (b) Weakly correlated. (c) Strongly correlated. (d) Inverse strongly correlated. (e) Subset sum. (f) Uncorrelated spanner $\texttt{span}(2, 10)$. (g) Weakly correlated spanner $\texttt{span}(2, 10)$. (h) Strongly correlated spanner $\texttt{span}(2, 10)$. (i) Multiple strongly correlated $\texttt{mstr}(\texttt{3M}/10, \texttt{2M}/10, 6)$. (j) Profit ceiling $\texttt{pceil}(3)$. (k) Circle $\texttt{circle}(2/3)$.

Uncorrelated **`ucorr`:** there is no correlation between the profit and weight of an item; p_i is uniformly random in $[1, M]$.

Weakly Correlated **`wcorr`:** despite the label of this instance type, the profit and weight of an item are *highly* correlated; p_i is chosen uniformly at random from $[w_i - M/a, w_i + M/a]$ such that $p_i \geq 1$.

Strongly Correlated **`scorr`:** the profit of an item is linearly related to its weight $p_i = w_i + M/a$.

Inverse Strongly Correlated **`invscorr`:** like strongly correlated instances, the profit of an item is linearly related to its weight but with a negative fixed charge; $p_i = w_i - M/a$, and w_i is drawn at random from $[M/a + 1, M(M/a)]$. In the original definition of this instance, the weights were assigned accordingly after the profits have been sampled. We changed the definition slightly by sampling the weights first, to preserve the CV value of the weights.

Subset Sum `sbstsum`**:** the item's profit and weight are equal $p_i = w_i$. Obtaining a filled knapsack is thus the only aim when solving instances of this type.

The previous types are standard instances in the literature of the 0-1KP. The following instance types were proposed by Pisinger in [83]. They are constructed in such a way to make them difficult for the branch-and-bound algorithms.

Spanner $\texttt{span(v, d)}$: a set called the spanner set is generated with v items each with a profit and a weight. The spanner type is characterised by v, the spanner set size, and d, a multiplier limit. The weight of each item is drawn at random from a given range $[1, M]$. The profit of each item is then generated according to the distribution of the spanner problem type to be: *uncorrelated* (`uspan`), *weakly correlated* (`wspan`), or *strongly correlated* (`sspan`) with the item's weight. The items in the spanner set are then normalised by dividing both profits and weights by $d + 1$. The last step is to construct the n items by randomly selecting an item (w_i, p_i) from the spanner set and a multiplier b drawn from the interval $[1, d]$ such that the constructed item has the following profit and weight (bw_i, bp_i). All items in a spanner instance are multiples of the spanner set. For all the spanner problem types, v and d are set to equal 2 and 10. Note that because of the way the spanner problem types are generated, it was difficult to generate instances of this type with $CV \leq 0.3$.

Multiple Strongly Correlated $\texttt{mstr(k}_1\texttt{, k}_2\texttt{, d)}$: if the weight w_i is divisible by d, then $p_i := w_i + k_1$, otherwise, $p_i := w_i + k_2$. Since the weights in the first group are all multiples of d, using only these weights will fill at most $d\lfloor C/d \rfloor$ of the capacity [83]. Hence, the need to use some of the items from the second distribution to obtain a completely filled knapsack. In both groups the profits and weights are strongly correlated. Setting the values of (k_1, k_2, d) to $(3M/10, 2M/10, 6)$ has been shown to generate very difficult instances according to computational experiments in [83].

Profit Ceiling `pceil(d)`: all profits are multiples of a given parameter d, $p_i = d\lceil w_i/d \rceil$. We set d to 3 , since this setting produces difficult instances according to [83].

Circle `circle(d)`: the item's profit is a function of the weights from an arc of an ellipsis, $p_i := d\sqrt{4M^2 - (w_i - 2M)^2}$ and $d = 2/3$.

5.1.2 Phase transition

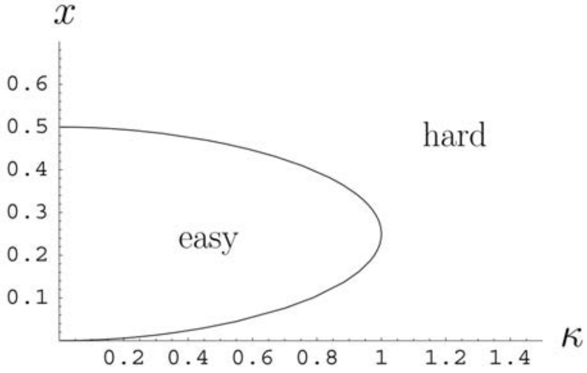


Figure 5.2: Plot taken from [98] that shows the phase transition in subset sum ($x = C/(nM)$ and $K = k$). The area labelled easy is the region where exponentially many perfect solutions (i.e. solutions with $f(x) = C$) are expected to exist, while the area labelled hard is the region where the probability of finding a perfect solution is zero.

The problem type, subset sum, is a generalisation of the NPP. It has a similar phase transition determined by the same control parameter $k = log_2 M/n$ [98, 97]. Through the application of the statistical mechanics framework, Sasamoto et al. [98] obtained an asymptotic expression of the number of perfect solutions in the subset sum problem (i.e. solutions with $f(x) = C$). They identified the easy and hard regions of this problem type to be as shown in figure 5.2. In the easy phase, exponentially many perfect solutions are expected to exist, while in the hard phase the probability of finding a perfect solution decreases sharply to zero. The parametrisation of the critical value k_c that was obtained for subset sum by Sasamoto et al. is given in [98]. The parametrisation depends on the value of $x = \frac{C}{nM}$. The method of statistical mechanics has been applied to investigate some properties of the knapsack problem and some of its variations, such as an upper bound to the optimal profit in the multi-knapsack problem [54, 31, 61]. However, to the best of our knowledge, no phase transition has been identified other than for the subset sum problem type. Nevertheless, we examine the effect of setting k to 0.4 and 1, which respectively map to the easy and hard regions of the subset sum, for all the 11 problem types. Note that like in the NPP chapter, our use of "hard" and "easy" here does not correspond to the computational complexity but to the probability of having a perfect solution. In this chapter we are interested to see if the computational complexity of finding the optimal using local search algorithm changes with the phase transition.

5.1.3 Constraint handling

We use penalty as a constraint handling method. Various penalty functions have been proposed for the 0-1KP [79, 72] and its generalisation the multiple knapsack problem [40, 59]. An infeasible solution x that violates the given constraint is penalised by a value $\mathrm{Pen}(x) > 0$, while $\mathrm{Pen}(x) = 0$ for a feasible solution x. The fitness functions after adding the penalty term is as follows:

$$f(x) = \sum_{i=1}^{n} x_i p_i - \mathrm{Pen}(x) \tag{5.4}$$

Allowing infeasible solutions to be part of the searchable space leads to a smoother fitness landscape when a suitable penalty function is used [70]. Penalising the infeasible solutions proportional to the the degree of violation of the constraint allows them to contribute to the search process. Therefore, the choice of an appropriate penalty function is very critical for inducing smoother landscapes and guiding the search to good feasible regions. For the 0-1KP, Olsen [79] found that weak penalties prevent the search from finding any feasible solution for highly constrained instances. Also, Gottlieb [40] note that the local optima in all covering and packing problems lie in the boundary of the feasible region. Hence, assigning a lower fitness value to infeasible solutions than all feasible solutions is important for successful penalty-based search.

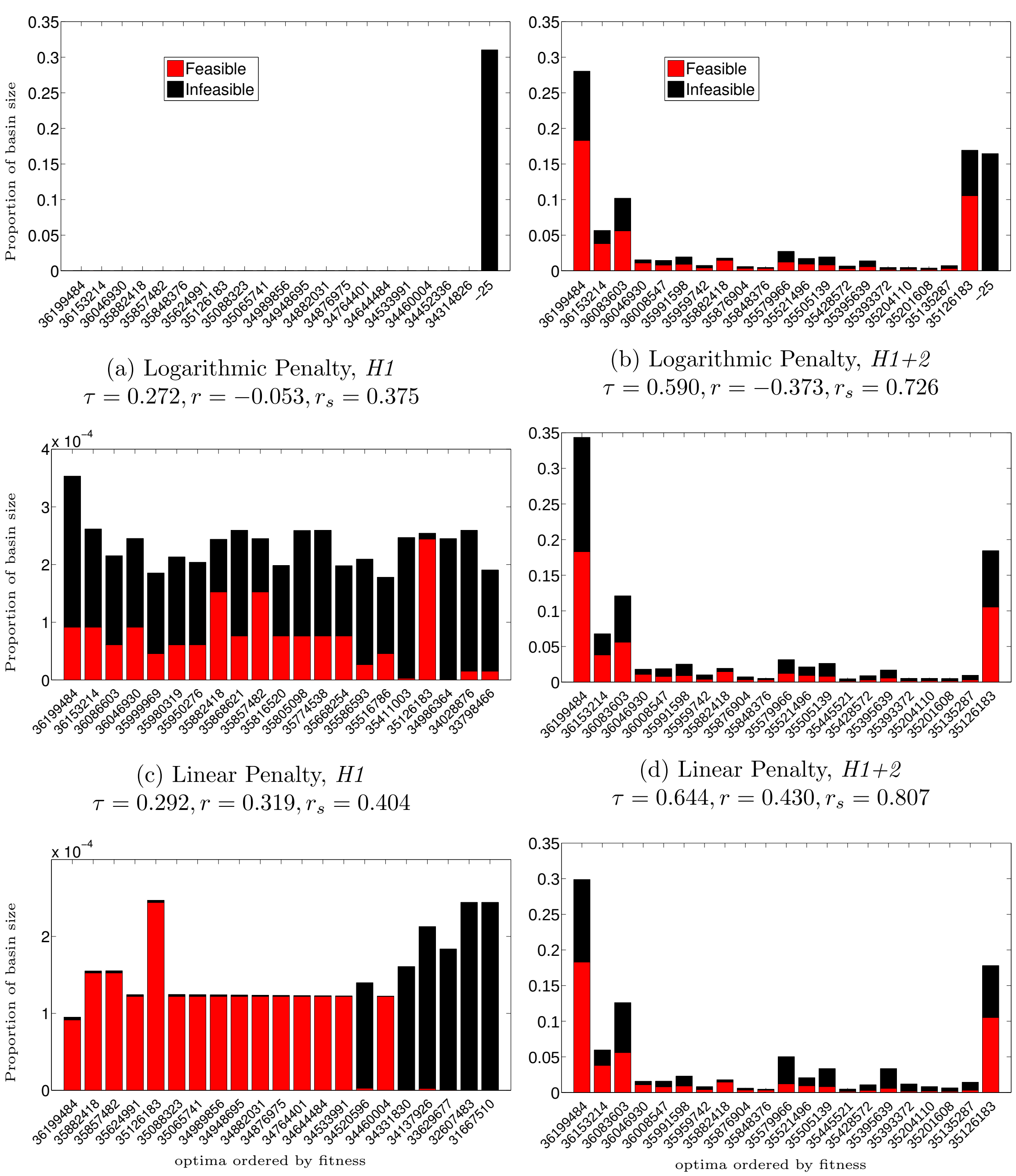
We studied the effect of using three types of penalty functions, which have been proposed in [72], on the *H1* and *H1+2* landscapes. The functions differ in the growth of the penalty value with respect to the degree of constraint violation, namely the three types are: logarithmic, linear and quadratic. We also add the term $\sum_{i=1}^{n} p_i$ to the penalty function as an offset term that insures that all infeasible solutions achieve lower fitness values than all feasible solutions [40]. Otherwise, in the logarithmic case for instance, the entire search space becomes part of the basin of the all ones solution $x = (1, \cdots, 1)$. The penalty functions, in order from weak to strong, are as follows:

$$\mathrm{Pen}(x) = \log_2 \left(1 + \rho \left(\sum_{i=1}^{n} x_i w_i - C \right) \right) + \sum_{i=1}^{n} p_i \tag{5.5}$$

$$\mathrm{Pen}(x) = \rho \left(\sum_{i=1}^{n} x_i w_i - C \right) + \sum_{i=1}^{n} p_i \tag{5.6}$$

$$\mathrm{Pen}(x) = \left(\rho \left(\sum_{i=1}^{n} x_i w_i - C \right) \right)^2 + \sum_{i=1}^{n} p_i \tag{5.7}$$

where $\rho = \max_{i=1,\ldots,n} \{p_i\} / \min_{i=1,\ldots,n} \{w_i\}$. In figure 5.3, we show an example instance that demonstrates how the basin sizes change with the use of the different penalty functions. It is clear that using the logarithmic penalty function is a bad choice since it creates a strict local optimum, the all ones solution $x = (1, \cdots, 1)$, in the infeasible region. While both the linear and quadratic functions do not create any local optima in the infeasible region, the strong penalty enforced by the quadratic function seems to direct the infeasible configurations to be part of the basins of optima with lower quality, as opposed to the linear penalty function. This is also shown in the correlation coefficient between the basin size and fitness when using the two penalty functions, where the linear function has stronger correlations in both landscapes.



(a) Logarithmic Penalty, *H1*
$\tau = 0.272, r = -0.053, r_s = 0.375$

(b) Logarithmic Penalty, *H1+2*
$\tau = 0.590, r = -0.373, r_s = 0.726$

(c) Linear Penalty, *H1*
$\tau = 0.292, r = 0.319, r_s = 0.404$

(d) Linear Penalty, *H1+2*
$\tau = 0.644, r = 0.430, r_s = 0.807$

(e) Quadratic Penalty, *H1*
$\tau = 0.236, r = 0.212, r_s = 0.331$

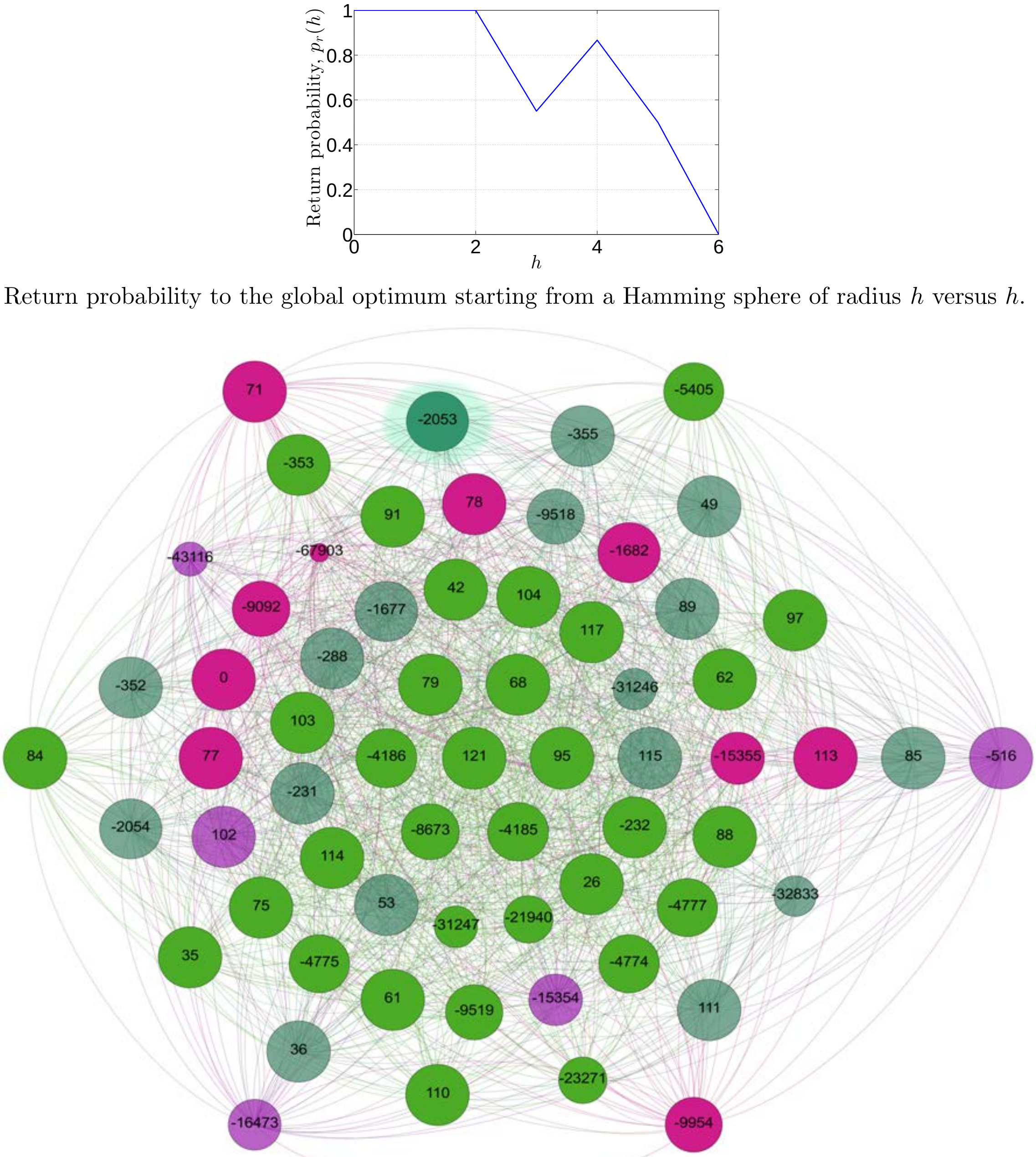
(f) Quadratic Penalty, *H1+2*
$\tau = 0.533, r = 0.431, r_s = 0.717$

Figure 5.3: The basin size fractions of the search space using different penalty functions. The red colour corresponds to the feasible configurations that are part of the basin and the black corresponds to the infeasible ones. The figures show the global optimum basin and the largest 20 local optima basins. The optima are ordered according to fitness (x-axis) starting from the global optimum in the far left. The results are for an instance of weakly correlated knapsack of size $n = 22$, $k = 1$, $CV = 0.12$, and $\lambda = 0.5$. The fraction of the number of optima in the *H1* landscape is 0.115 and in the *H1+2* landscape is 15×10^{-6}. The correlation between fitness and basin size of all the optima for each landscape is shown by Kendall's τ, Pearson's r, and Spearman's r_s correlation coefficients. Obviously changing the penalty function only affects the infeasible part of the basin (the black region of the bars).

Figure 5.5 shows the return probability to the global optimum starting from a Hamming sphere of radius h, where we can see that the linear penalty function has the highest return probability in both landscapes. One interesting observation is that the basin configuration proportions continue to increase and decrease between consecutive spheres in the *H1+2* landscape. This can be attributed to the very small number of optima (61 when using either linear and quadratic penalty functions and 62 when using logarithmic penalty function) and the nature of the *H1+2* neighbourhood, as the neighbours of a configuration in sphere h would be spread over more spheres using this neighbourhood operator compared with the *H1* operator. Figure 5.4 presents an example illustrating this oscillating behaviour in the *H1+2* landscape. Figure 5.6 shows the performance of local search to find the global when all the three penalties are used, where again, the best performance was achieved with the linear penalty function. The previous results provide some evidence that the linear penalty function appears to be the best choice, in terms of local search performance and correlation of basin size and fitness, out of the three investigated functions . Therefore, in the rest of this chapter we only use this linear penalty function to handle the constraint.

We use the linear penalty function (eq. 5.6) with all instance types except for the subset sum. Since applying this penalty function to infeasible solutions in a subset sum instance assigns equal fitness values for all infeasible solution, thus creating large plateaus in the landscape. Instead, we simply set the fitness of an infeasible solution in a subset sum instance to the negative of the amount it exceeded the knapsack capacity by. The fitness function of subset sum instances is, thus, as follows:

$$f(x) = C - \sum_{i=1}^{n} x_i w_i \tag{5.8}$$



(a) Return probability to the global optimum starting from a Hamming sphere of radius h versus h.

(b) Each node represents a configuration and edges indicate neighbourhood relation. The fitness is shown for each node, also the node size is scaled proportional to fitness. The graph has been laid out such that the global optimum (with fitness 121) is placed in the centre and the configurations that are h Hamming distance away from the global lie on the h-th circle. The colours dark green and light green indicate that a configuration is in the global's basin, while the colours pink and purple indicate that a configuration is not in the global's basin. The semi-transparent nodes with the colours dark green and purple are neighbours of the highlighted node (with fitness -2053). In addition to the global, there are two strict local optima with fitness 102 and 113.

Figure 5.4: Example of an *H1+2* landscape where the return probability to the global does not decrease monotonically as h increases. The landscape is for an instance of weakly correlated knapsack of size $n = 6$, $k = 1$, $CV = 0.25$, and $\lambda = 0.5$.

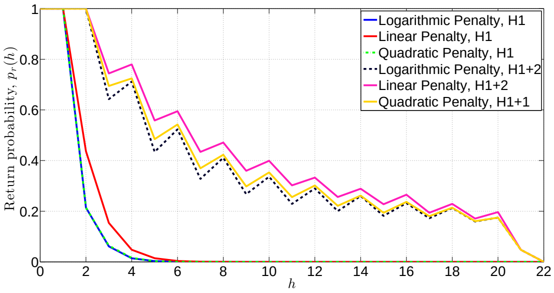
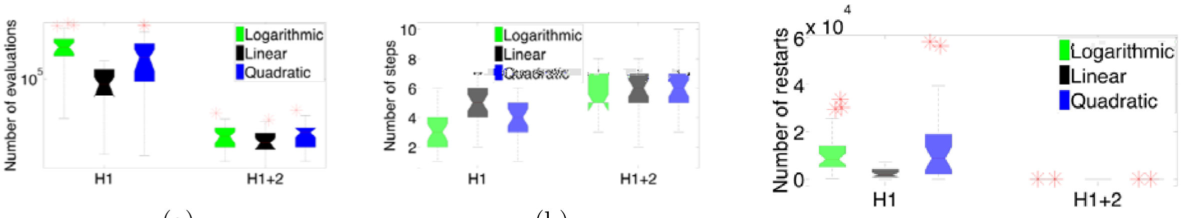


Figure 5.5: Return probability to the global optimum starting from a Hamming sphere of radius h versus h. The results are for the same knapsack instance in figure 5.3. Notice how the probability of return approaches zero faster in the *H1* case compared to the *H1+2* which only approaches zero in the last sphere.



(a)

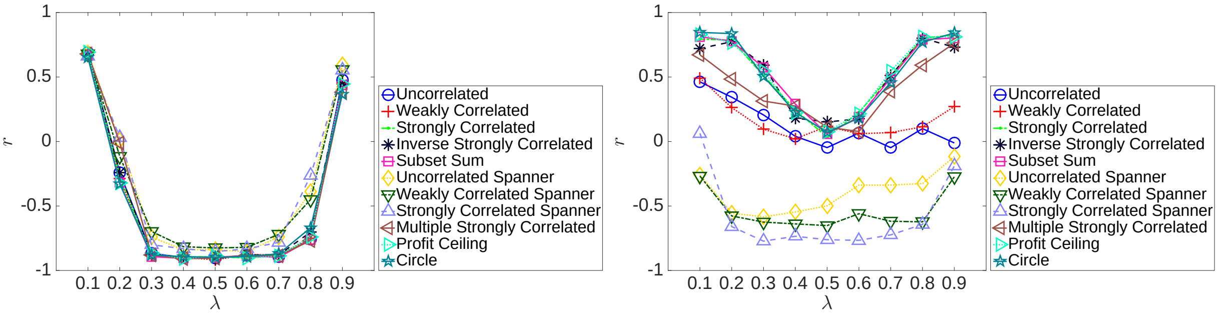
(b)

(c)

Figure 5.6: Number of fitness evaluations, restarts and steps to find the global optimum. Each box-plot shows the data distribution of 30 runs of local search to find the global optimum. The results are for the same knapsack instance in figure 5.3.

5.1.4 Constraint level

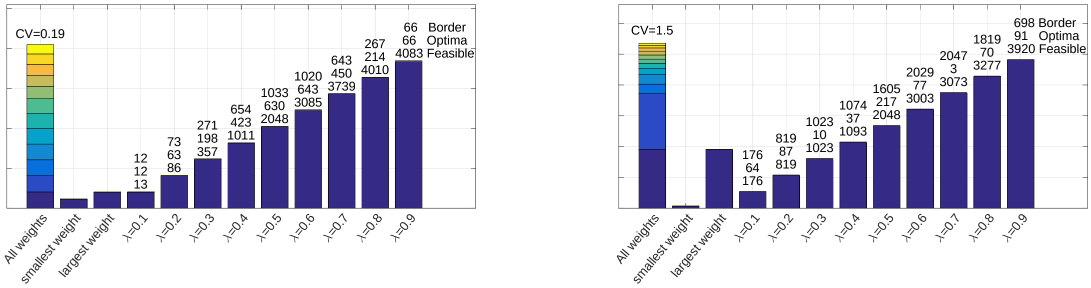
Perhaps the most interesting observation about the landscape and the *CV* of the weights in the NPP was the strong negative correlation between the number of local optima and the *CV* in the *H1* landscape. We show here that it is also the case in the 0-1KP. Figure 5.7 shows the correlation between the number of strict local optima and the *CV* of the weights in the *H1* and *H1+2* landscapes against λ. As in NPP, the correlation was found to be strong and negative in the *H1* landscape, apart from highly constrained instances ($\lambda \leq 0.2$) and weakly constraint instances ($\lambda > 0.8$).



(a) *H1* landscape

(b) *H1+2* landscape

Figure 5.7: Correlation between the *CV* of the weights and the number of strict local optima in the *H1* and *H1+2* landscapes versus λ. The results are shown for all instance types of problem size $n = 20$. Each data point shows Pearson's r correlation coefficient calculated over 180 instances.



(a) Small *CV*

(b) Large *CV*

Figure 5.8: The number of optima in the *H1* landscape against the different values of λ. The results are for two uncorrelated KP instances of size $n = 12$. The bar on the far left in each plot shows the set of all weights in the instance, the *CV* of the weights is shown over the bar. The following two bars show the smallest and the largest weight respectively (to facilitate comparison). The rest of the bars illustrate the knapsack capacity C with each λ setting. The boundary size $|B|$, the number of optima (all of which are strict and with only one global), and the feasible region size $|F|$ for each λ setting are shown over the corresponding bar. For example in (a): when $\lambda = 0.3$, there are 357 feasible configurations, of which 271 lie on the boundary between the feasible and the infeasible regions, and out of these 271 configurations 198 of them are optima. The rest of the configurations are infeasible

In the following we attempt to explain the positive correlation between the *CV* and the number of strict local optima when the problem is highly or weakly constrained. In small *CV* instances, most of the solutions are infeasible when highly constrained and feasible when weakly constrained. This consequently decreases the boundary size and thus the number of strict local optima. On the other hand, the larger weights in large *CV* instances, prevent many solutions from becoming feasible when weakly constrained and allow many solutions to be feasible when highly constrained. This makes the boundary size in large CV instances relatively larger in both cases, causing the number of strict optima to be higher than that in small CV instances. An illustrative example is shown in figure 5.8. In fact, when highly constrained, the mechanism of solving large CV instances becomes similar to that of solving small CV instances, in that the problem becomes about fitting the small and similar weights into the knapsack.

In the *H1+2* landscape, the strong positive correlation, in some of the problem types, when highly and weakly constrained could be attributed to the same above explanation. The case in moderately constrained instances is similar to that in NPP, in that, the correlation between the CV and the number of local optima in this landscape is weak. Generally in this landscape, the number of local optima is slightly higher in the middle CV interval $(0.3, 1)$ than in the other two CV intervals ((0,0.3],[1,2)). The strong negative correlation in the spanner problem types, when moderately constrained, can be explained by the lack of instances with small CV values, as it was difficult to generate instances of these types with $CV \leq 0.3$. Thus as the CV ranged between middle to large, the number of local optima ranged from slightly higher to slightly lower values.

Moderately constrained instances have the largest boundary sizes and thus the largest number of optima. We believe, for this reason, that they have the most interesting landscapes to study. This was the motivation behind setting λ to 0.5 in the rest of the instances studied in this chapter.

5.2 Search Position Types

The search position types found in randomly generated instances of all the problem types with different values of CV and k are shown in appendix B. The tables B.1 to B.6 show results of the feasible region, and tables B.7 to B.12 show the infeasible region results. The positions in both the infeasible and feasible regions are shown as proportions of the search space for both the *H1* and the *H1+2* landscapes. As in NPP, no configuration of type IPLAT has been found in either landscapes or regions. In the infeasible region, all the found configurations were of the following types: LEDGE, SLOPE, SLMIN and NSLMIN. No strict optima or plateaux were found in the infeasible region. In the *H1* landscape of the infeasible region, the configurations were mainly of type LEDGE with both values of k. In the *H1+2* landscape of the infeasible region, the configurations were mainly of types LEDGE and SLOPE when $k = 0.4$, and mainly LEDGE when $k = 1$.

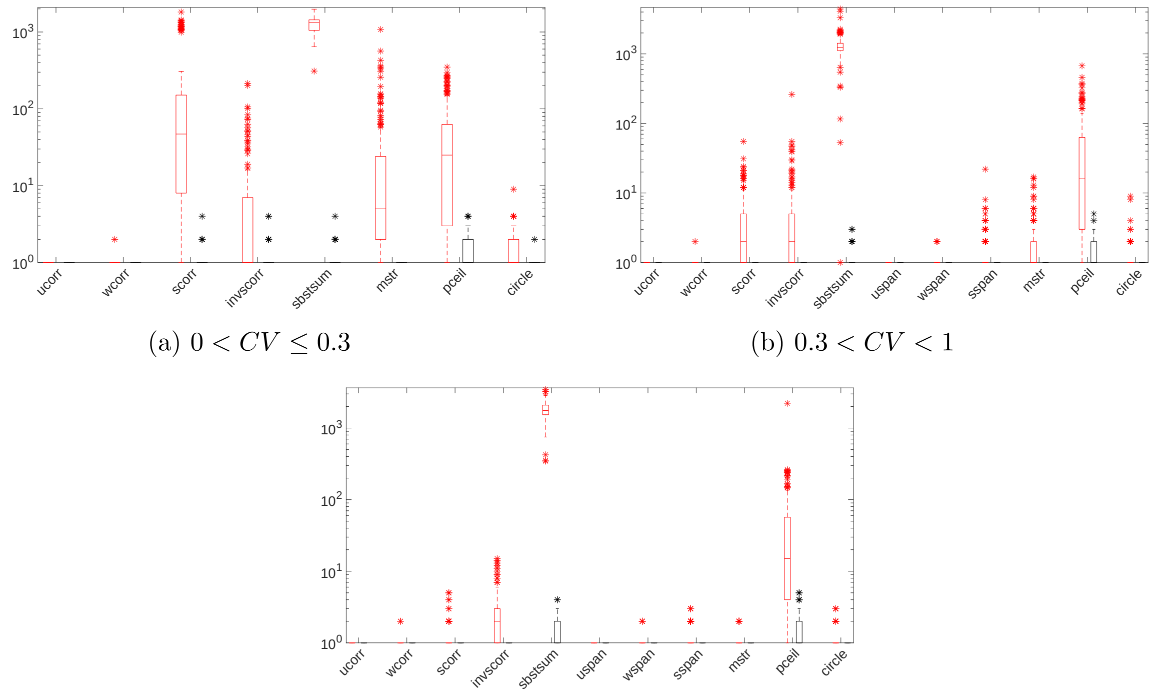
In the feasible region of the *H1+2* landscape, when $k = 0.4$, the found configurations were mainly of types LEDGE, SLOPE , NSLMAX and SLMAX in all the problem types. Apart from the uncorrelated and uncorrelated spanner instances, where no configurations of type NSLMAX or SLOPE were found. When $k = 1$, the SLOPE and NSLMAX types disappear, aside from very few configurations in small CV instances ($CV \leq 0.3$) of the profit ceiling and circle problem types. The feasible region of the *H1* landscape is similar to its infeasible region, in that, no apparent changes were observed between the values of k. In general, the configurations were of type SLMAX and LEDGE in this region.

5.3 Optima and Plateaux

5.3.1 Number of optima and plateaux

Figure 5.9 shows the number of global optima found in randomly generated instances of all the problem types and across the different values of k and CV, while figure 5.10 shows the number of local optima. Note that all the global are strict local optima in the *H1* landscape, but some of them become connected forming global plateaux in the *H1+2* landscape. In fact, all the global and local optima in the *H1* landscape are strict. Plateaux were only found in the *H1+2* landscape and mainly when $k = 0.4$ as shown in the previous section. In general, only one optimal solution was found in all the problem types when $k = 1$. There is a clear difference in the number of optimal solutions between instances of subset sum that were drawn from the easy and the hard phase. The number of the optimal solutions is around 1000 in the easy phase ($k = 0.4$), and it drops down to only one in the hard phase ($k = 1$). This is true across all the CV values, apart from the largest CV interval where the number of optimal solutions was found to be slightly more than one sometimes but always less than ten. The number of global when $k = 0.4$ in the strongly correlated and multiple strongly correlated instances seems to decrease as the CV increases, until it reaches around only one in the interval $[1, 2)$. The number of global in the profit ceiling is slightly higher when $k = 0.4$ compared to that when $k = 1$. For the rest of the problem types, the number of global appears to not change much over the values of k.

In general and across all the different problem types, the number of local optima in the *H1* landscape is the highest in the CV interval $(0, 0.3]$ and it starts decreasing as the CV gets larger. This does not seem to change much between the two values of k. Also, the variation in the number of local optima in this landscape between the different problem types was found to be very small as shown in figure 5.10. As in the NPP, the difference in the number of local optima between the two landscape is very large, and it is the largest in the small CV interval $(0, 0.3]$. In the *H1+2* landscape, the number of local optima seems to vary between the different problem types. The lowest number of local optima was found in the uncorrelated, weakly correlated, uncorrelated and weakly correlated spanner instances. Note that the number of local optima is very low in the weakly correlated (when $k = 0.4$) and the uncorrelated instances with medians < 10. The number of local optima in the weakly correlated and the weakly and strongly correlated spanner problems is higher in instances with $k = 1$ than that with $k = 0.4$. The opposite is true for the profit ceiling and circle instances where the number of local optima is slightly lower when $k = 1$ than that when $k = 0.4$.



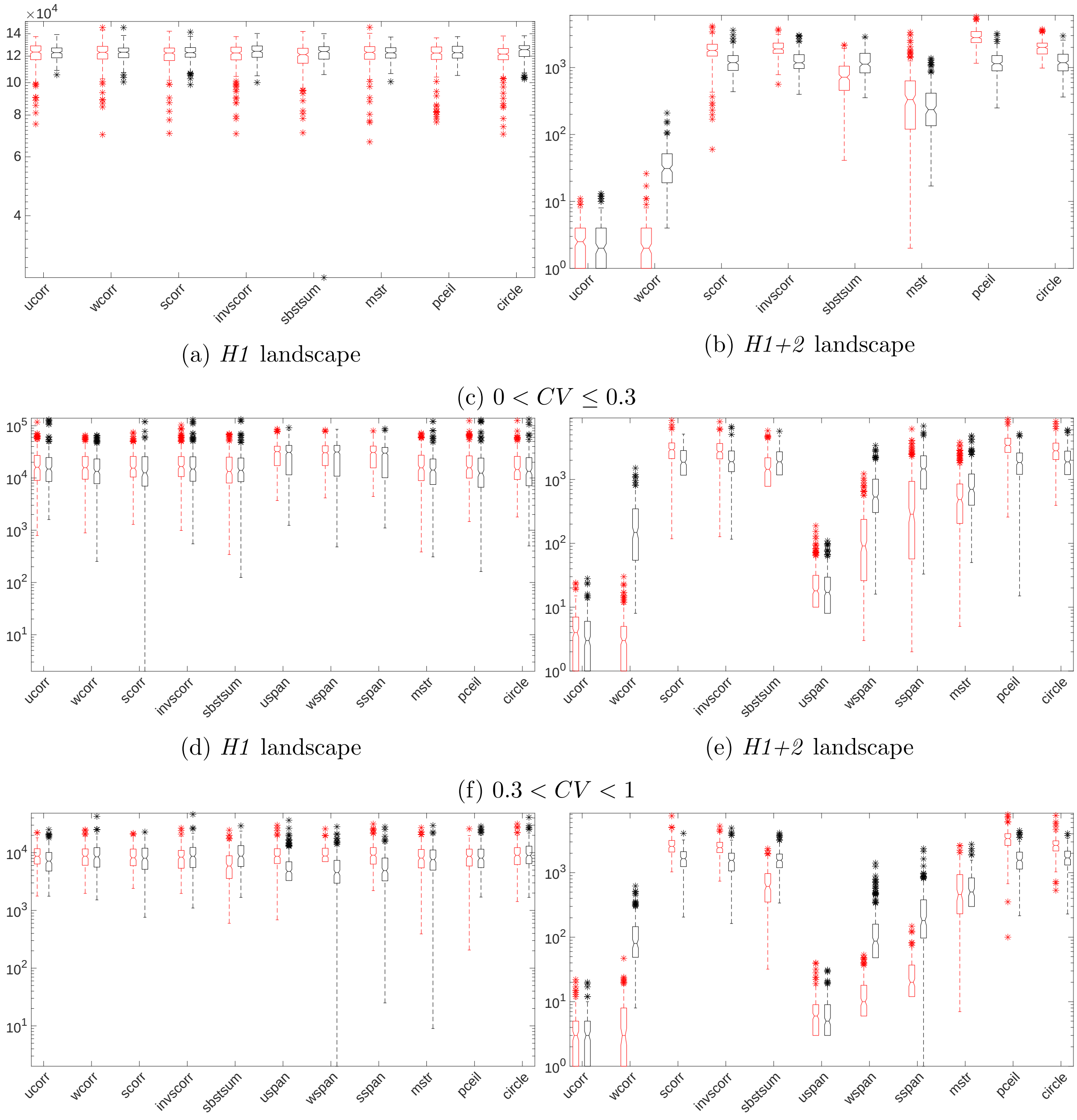
(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

Figure 5.9: Number of both strict and no-strict global optima (in log scale) found in 600 instances of size $n = 20$ of each problem types for each k value. The red boxes show the $k = 0.4$ results and the black ones show the $k = 1$ results.

As we have seen before, plateaux were only found in the *H1+2* landscape. Aside from very few configurations in small CV instances ($CV \leq 0.3$) of the profit ceiling and circle problem types, these plateaux were found only when $k = 0.4$. Both uncorrelated and uncorrelated spanner instances did not have any plateaux. The weakly correlated and weakly correlated spanner instances were found to have very few plateaux of all the types but mainly open plateaux. For the rest of problem types, the majority of the found plateaux were closed plateaux. For the subset sum instances, the majority of plateaux were global and closed plateaux. Most of the plateaux found in the profit ceiling instances were open and closed plateaux.



(a) *H1* landscape

(b) *H1+2* landscape

(c) $0 < CV \leq 0.3$

(d) *H1* landscape

(e) *H1+2* landscape

(f) $0.3 < CV < 1$

(g) *H1* landscape

(h) *H1+2* landscape

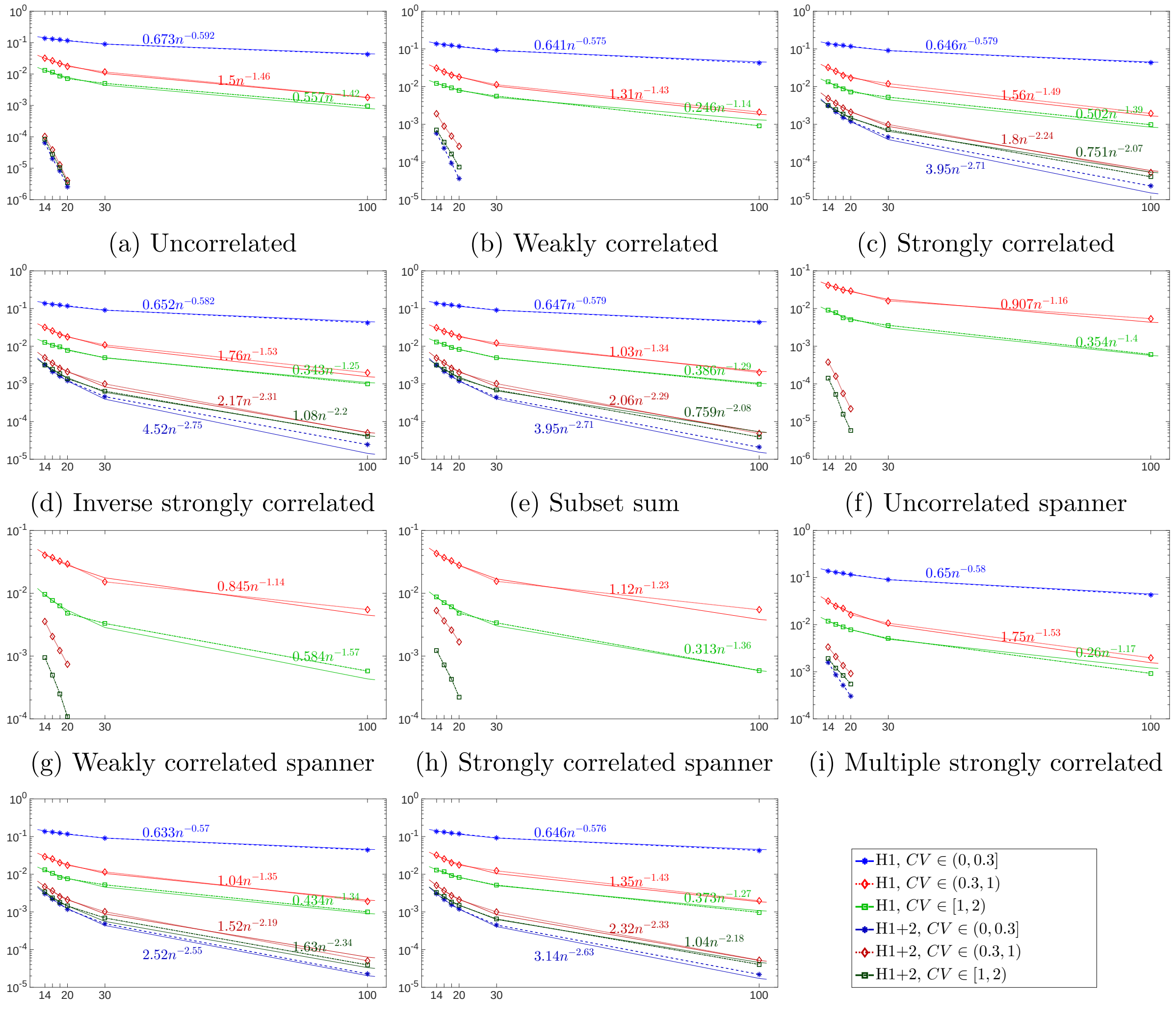
(i) $1 \leq CV < 2$

Figure 5.10: Number of both strict and no-strict local optima (in log scale) found in 600 instances of size $n = 20$ of each problem types for each k value. The red boxes show the $k = 0.4$ results and the black ones show the $k = 1$ results.

In all the problem types, most of the found plateaux have very small sizes, around two or three configurations. The largest found plateaux were less than 10 configurations in all the instances, apart from the profit ceiling and circle instances where the largest plateaux were found to be composed of around 30 configurations. However, these large plateaux were rarely found. The number of exits in open plateaux was found to be also quite small in all the problem types; it was found to be mainly between 1 and 3 exits.

5.3.2 Average number of strict local optima

Figure 5.11 shows the mean decay of the strict local optima proportion against n. The results for $n = 30, 100$ are the SRS estimates obtained with the sample sizes $s = 10^5, 5 \times 10^5$ respectively. Most of the proportions seems to decrease polynomially with n in the form an^{-b}. The proportion in the *H1+2* landscape of some of the problem types, namely: uncorrelated, weakly correlated, uncorrelated spanner, weakly correlated spanner, strongly correlated spanner, and multiple strongly correlated, seems to be smaller than what the SRS with the above sample sizes can detect. This was evident by the negative lower bound of the 95% $\mathrm{CI}_{\mathrm{AC}}$ of the obtained estimates indicating that the point estimates are greatly overestimating the real proportions. Therefore, we did not include these estimates in figure 5.11. We also did not fit the decay of the proportions with the form an^{-b}, since we are only left with four close data points. In general, and as in the NPP, the largest decay happens in the landscape of *H1+2* and the smallest in the *H1* landscape of the interval $(0, 0.3]$. The proportion of the strict local optima appears to decay faster in the *H1+2* landscape compared to the *H1* across all the CV intervals in all the problem types. We are unable to comment on the growth of the number of strict local optima in the *H1+2* landscapes of the following problem types: uncorrelated, weakly correlated, uncorrelated spanner, weakly correlated spanner, strongly correlated spanner, and multiple strongly correlated. However, the growth of the number of strict local optima in the *H1+2* landscape of the rest of the problem types and the growth in all the *H1* landscapes seems to be exponential with n.



(a) Uncorrelated (b) Weakly correlated (c) Strongly correlated

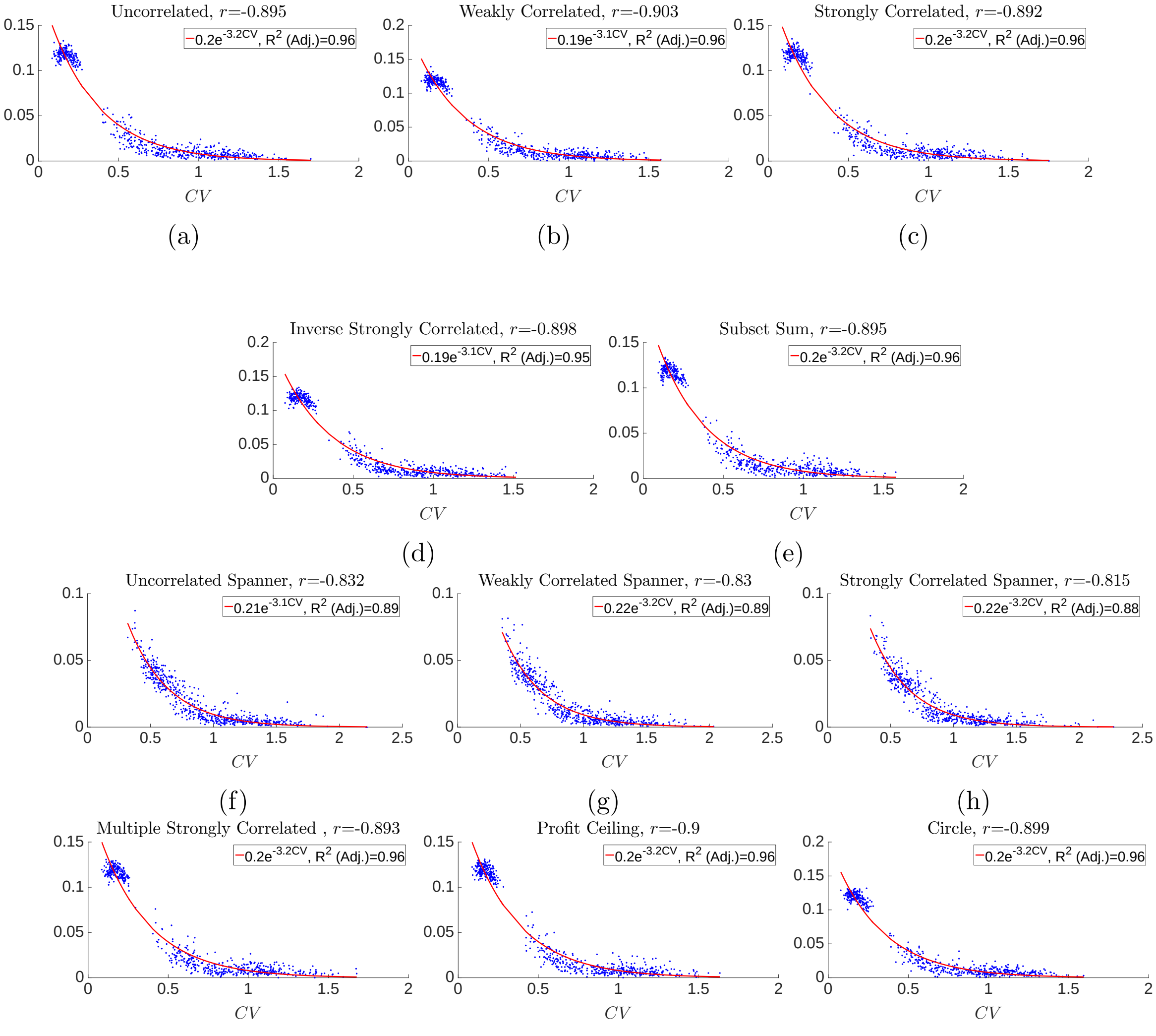
(d) Inverse strongly correlated (e) Subset sum (f) Uncorrelated spanner

(g) Weakly correlated spanner (h) Strongly correlated spanner (i) Multiple strongly correlated

(j) Profit ceiling (k) Circle

Figure 5.11: The decay of the strict optima proportion (in log scale) as the problem size n grows ($k = 1$). The results for each problem type are averages over 600 instances for each $n = 14, 16, 18, 20$ and over 500 instances for $n = 30, 100$. The number of strict optima is estimated for $n = 30, 100$ using SRS, the sample sizes are $s = 10^5, 5 \times 10^5$ respectively. The solid lines were obtained using least-squares fit. Note that the proportion of the strict local optima in the *H1+2* landscape is always lower and decays faster in comparison to the *H1* landscape.

As in the NPP, the number of strict local in the *H1* landscape was found to be strongly and negatively correlated with the CV across all the problem types (see figure 5.12). We tried to fit the same used formula in NPP for the average number of strict optima ($a\, e^{-b\, CV}$) to the optima found in all the problem types (the values of the coefficients a and b depend on n). In general and as in NPP, this seems to be a good approximate fit of the average number of strict optima. However, it seems to be noisier in the 0-1KP especially around the small CV values ($CV \leq 0.3$). The estimated number of local optima in the *H1* landscape of $n = 30, 100$ was also found to follow this trend.



(a) (b) (c)

(d) (e)

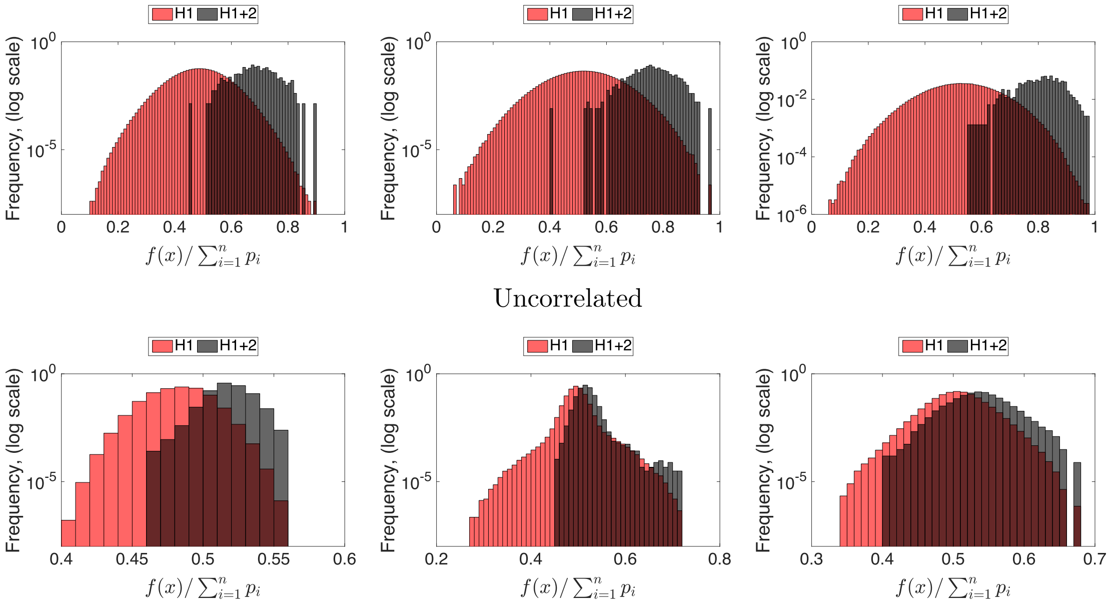
(f) (g) (h)

(i) (j) (k)

Figure 5.12: The fraction of strict local optima in the *H1* landscape versus CV. The results are for 600 instances of size $n = 20$ for each problem type and $k = 1$. The solid lines were obtained using least-squares fit. Pearson's correlation coefficient r between the two quantities is shown for each plot.

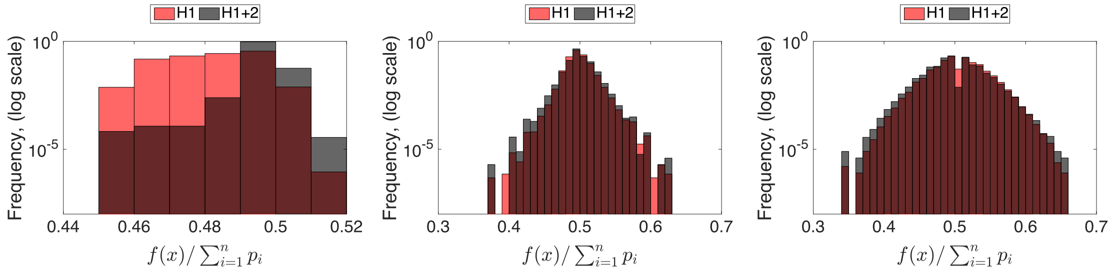
5.3.3 Quality of optima and plateaux

We examine here how the difference in the optima quality between the two landscape changes across the CV values. Again, obviously every optimum in the *H1+2* landscape is also an optimum in the *H1* landscape. Thus, the quality of the optima in the *H1+2* landscape is at least equal to or better than that in the *H1*. As in the previous chapter, we want to obtain a measure of quality that is independent of the problem instance and that does not require the knowledge the optimal solution. Thus, we measure the quality of an optimum x in a given instance as $f(x)/\sum_{i=1}^{n} p_i$. Figure 5.15 shows the quality of optima across the three CV intervals for all the problem types. There is a clear difference among the various problem types in terms of the overall quality of the optima and in the difference between the quality of the optima in the two landscapes. For example, in the uncorrelated and the uncorrelated spanners instances there is a large difference in the quality of optima between the two landscapes and that difference does not seem to change much across the CV intervals. On the other hand, the quality of the optima in the two landscapes seems to be similar in the circle instances and that similarity seems to increase as the CV value increases. The case in the subset sum instances is perhaps the most similar to that in the NPP, in that the quality of the optima in the *H1* landscape gets better as the CV value increases, which in turn decreases the difference in the quality between the two landscapes. In general, and apart from the uncorrelated and the uncorrelated spanner instances, the difference in the quality of optima between the two landscapes appears to decrease as the CV value increases.

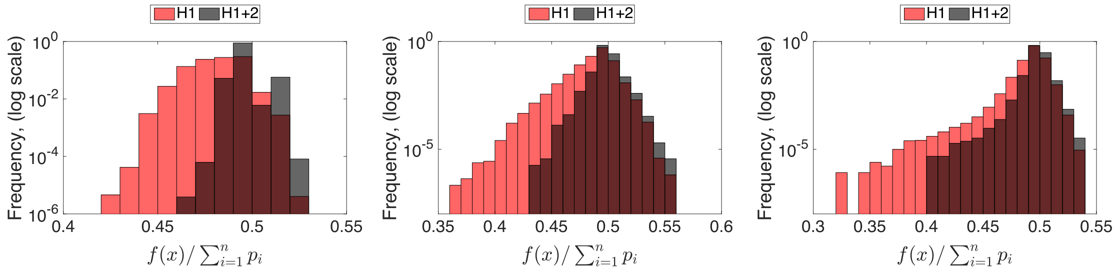


Uncorrelated

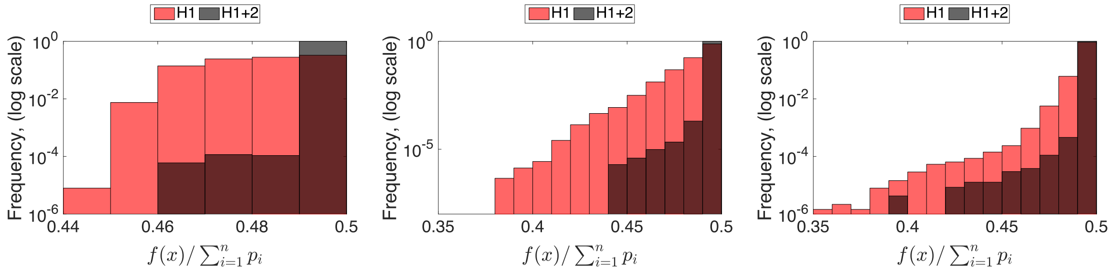
Weakly correlated



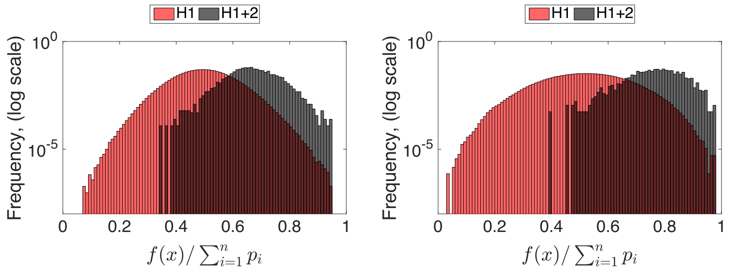
Strongly correlated



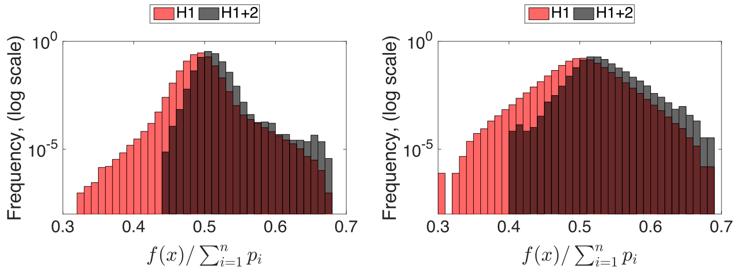
Inverse strongly correlated



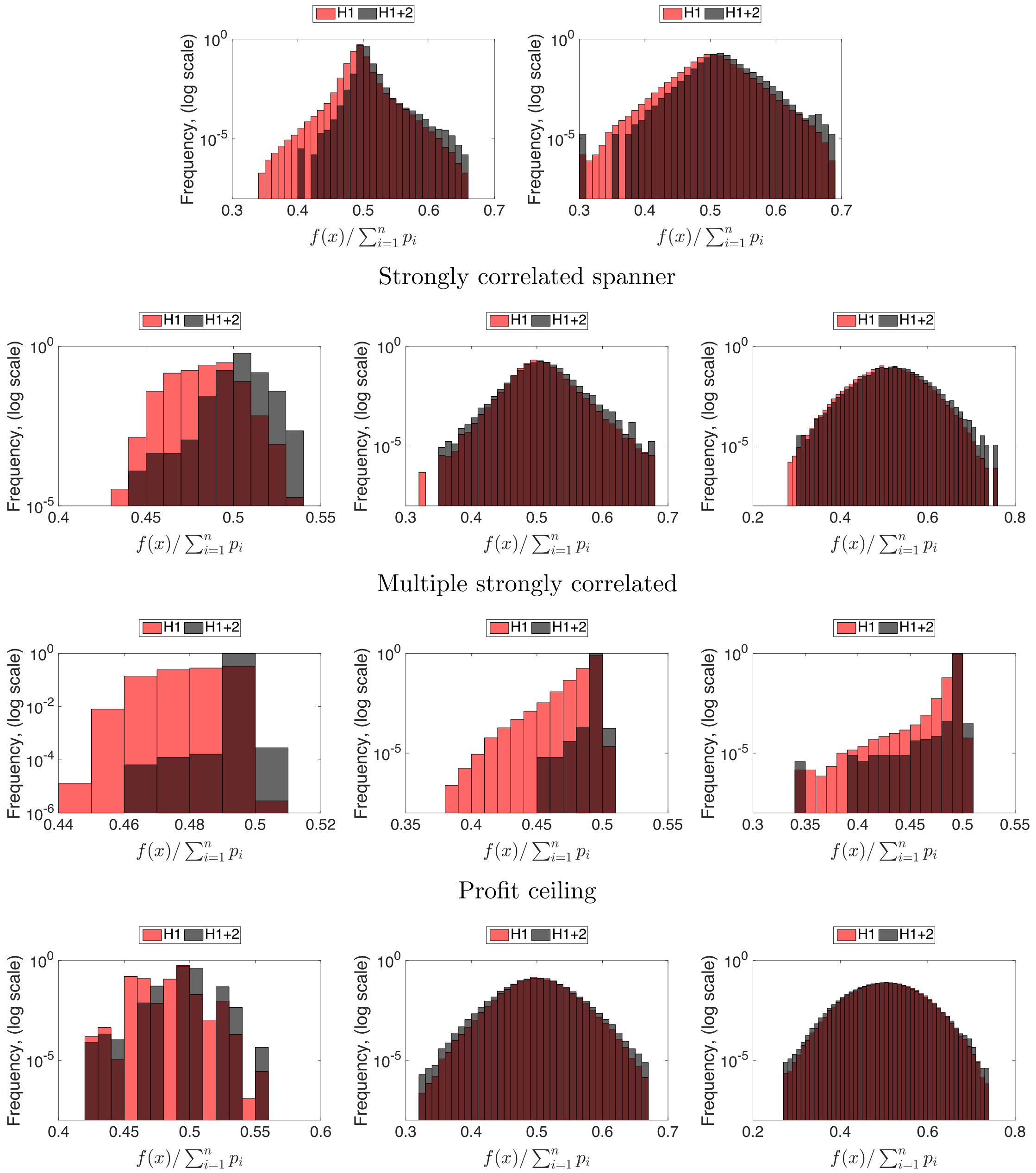
Subset sum



Uncorrelated spanner



Weakly correlated spanner



Strongly correlated spanner

Multiple strongly correlated

Profit ceiling

Circle

Figure 5.15: The quality of optima and plateaus in the *H1* and *H1+2* landscapes across the different values of CV: $0 < CV \leq 0.3$ (left), $0.3 < CV < 1$ (middle), $1 \leq CV < 2$ (right). The x-axis shows the fitness value divided by $\sum_{i=1}^{n} p_i$. The data includes all optima and plateaux found in 600 instances for each problem type of problem size $n = 20$ and $k = 1$.

Remember that the results in figure 5.15 were obtained by dividing the fitness of the optima by the sum of all the profits in that given instance ($f(x)/\sum_{i=1}^{n} p_i$). The shape of the obtained distributions can give us an idea about the ratio between the fitness of the optima and the total possible profit. You can see that in the subset sum case for example this cannot exceed 0.5 since we are studying instances where the constraint is set to half of the total weights $\lambda = 0.5$ (remember that in subset sum $p_i = w_i$). The distributions in the figure cannot be used directly to infer the relation between the quality of the local and global optima since the distributions are calculated over all the instances studied.

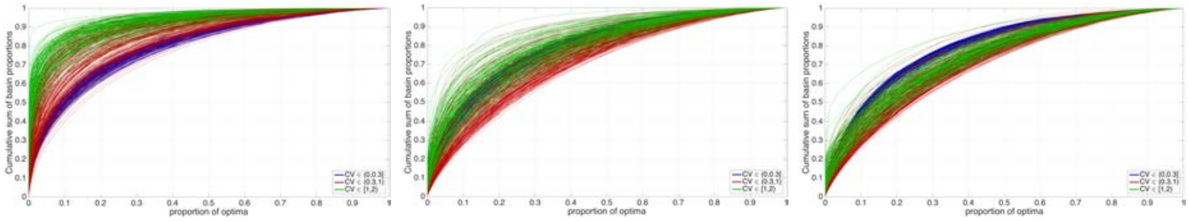
5.4 Basins of Attraction

As in the NPP, in this section we study the following features of the attraction basins: the basin size, shape and the correlation between the size and the optimum fitness. We try to examine most of these properties across the different problem parameters and compare the results of the two landscapes. We exhaustively calculate the basin sizes, thus we were limited to studying small problem sizes $n = 20$ only.

5.4.1 Basin size

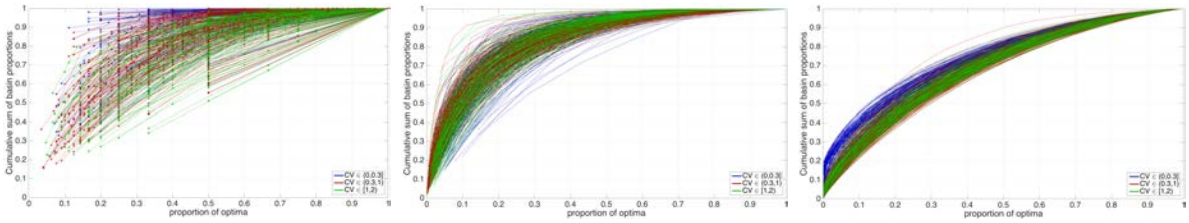
Figures B.1 to B.11 in the appendix give an overview of the relation between the basin size and fitness in all the instances we studied for each problem type. They show the general right skewness of the distribution of the basin sizes (i.e. most of the basin sizes are small with very few large ones). They also show how the basin sizes in the *H1* landscape increase with the CV until their sizes become similar to those of the *H1+2* landscape.

Figure 5.16, shows the cumulative sum of the basin proportions, after being sorted in descending order, against the percentage of the basins in the *H1* landscape for uncorrelated, weakly correlated and subset sum problem types. The results of the uncorrelated spanner are similar to the results of the uncorrelated problem type. The results of the weakly correlated spanner are similar to the results of the weakly correlated problem type. The rest of the problem types have similar results to that of subset sum. The figure shows that the largest basins in the uncorrelated problem type quickly covers large area of the search space, where around 60% to 90% of the search space is covered by only 20% of the basins. The same percentage of the basins covers around 40% to 70% of the search space in the weakly correlated problem type, and around 40% to 60% in the subset sum problem type. Figure 5.17 shows the same results for the *H1+2* landscape. Note that, in the uncorrelated problem type, we show each data point as it has a very few number of optima in this landscape. The results of the cumulative sum of the basin proportions in all the problem types are more or less similar between the values of k apart from weakly correlated problem type. When $k = 0.4$, the results of the *H1+2* landscape of the weakly correlated instances are similar to the shown results of the uncorrelated problem type.



(a) Uncorrelated (b) Weakly correlated (c) Subset Sum

Figure 5.16: To examine how quickly the largest basins cover the search space in the *H1* landscape, we plot the cumulative sum of the basin proportions going from the largest to smallest (i.e. we plot the cumulative sum of the basin proportions after sorting them in descending order) against the proportion of the optima. Each line shows the results of a single instance. The results are for 600 instances of $n = 20$ and $k = 1$. You can see that in the uncorrelated instances around 1% of the basins covers more than half of the search space.



(a) Uncorrelated (b) Weakly correlated (c) Subset Sum

Figure 5.17: To examine how quickly the largest basins cover the search space in the *H1+2* landscape, we plot the cumulative sum of the basin proportions starting from the largest to smallest (i.e. we plot the cumulative sum of the basin proportions after sorting them in descending order) against the proportion of the optima. Each line shows the results of a single instance. The results are for 600 instances of $n = 20$ and $k = 1$. Note that in the uncorrelated instances the landscape has far fewer number of optima compared to the rest of the problem types. You can see that some lines have only two data points which corresponds to having only two optima in those instances, with the largest basin of out of the two covering between half to almost 90% of the search space.

5.4.2 Basin size and fitness

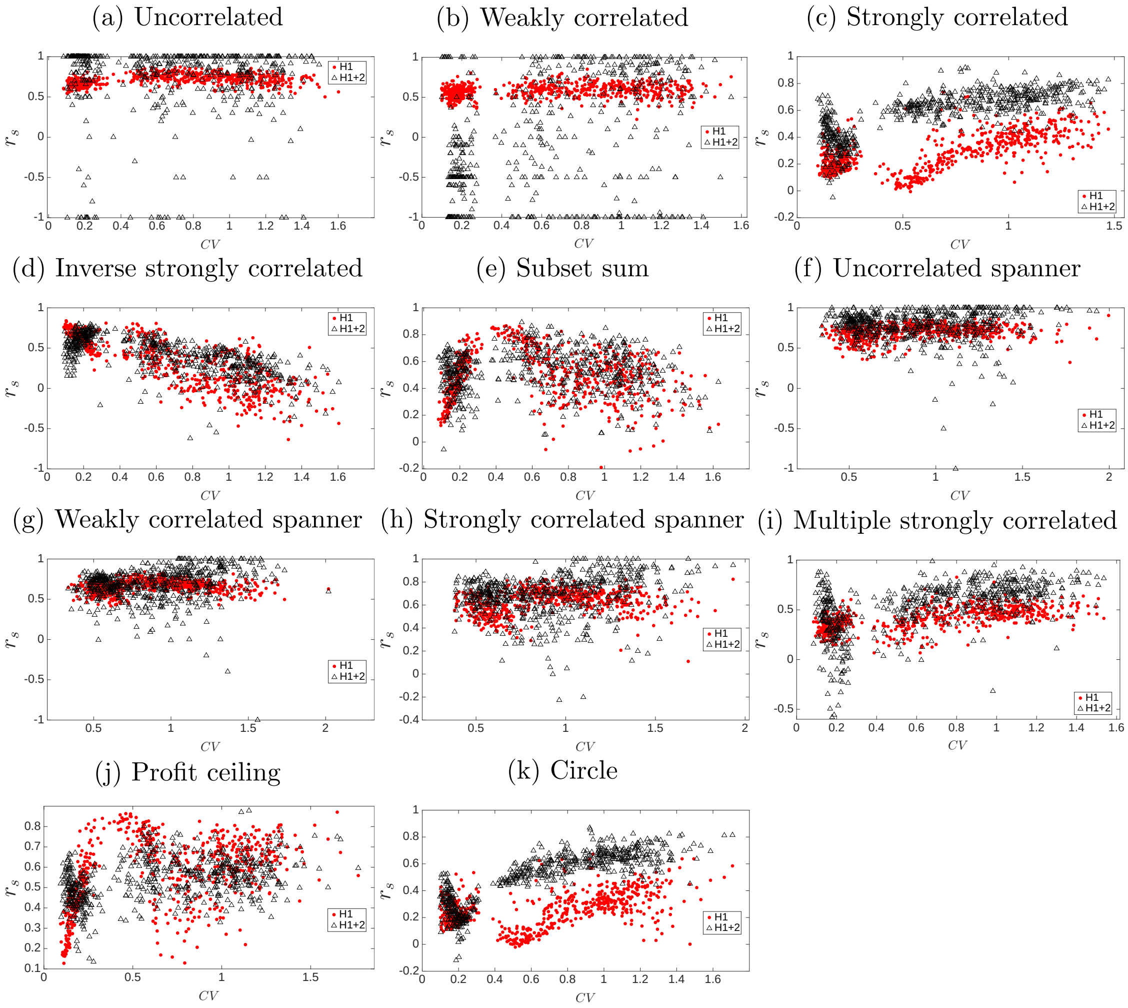
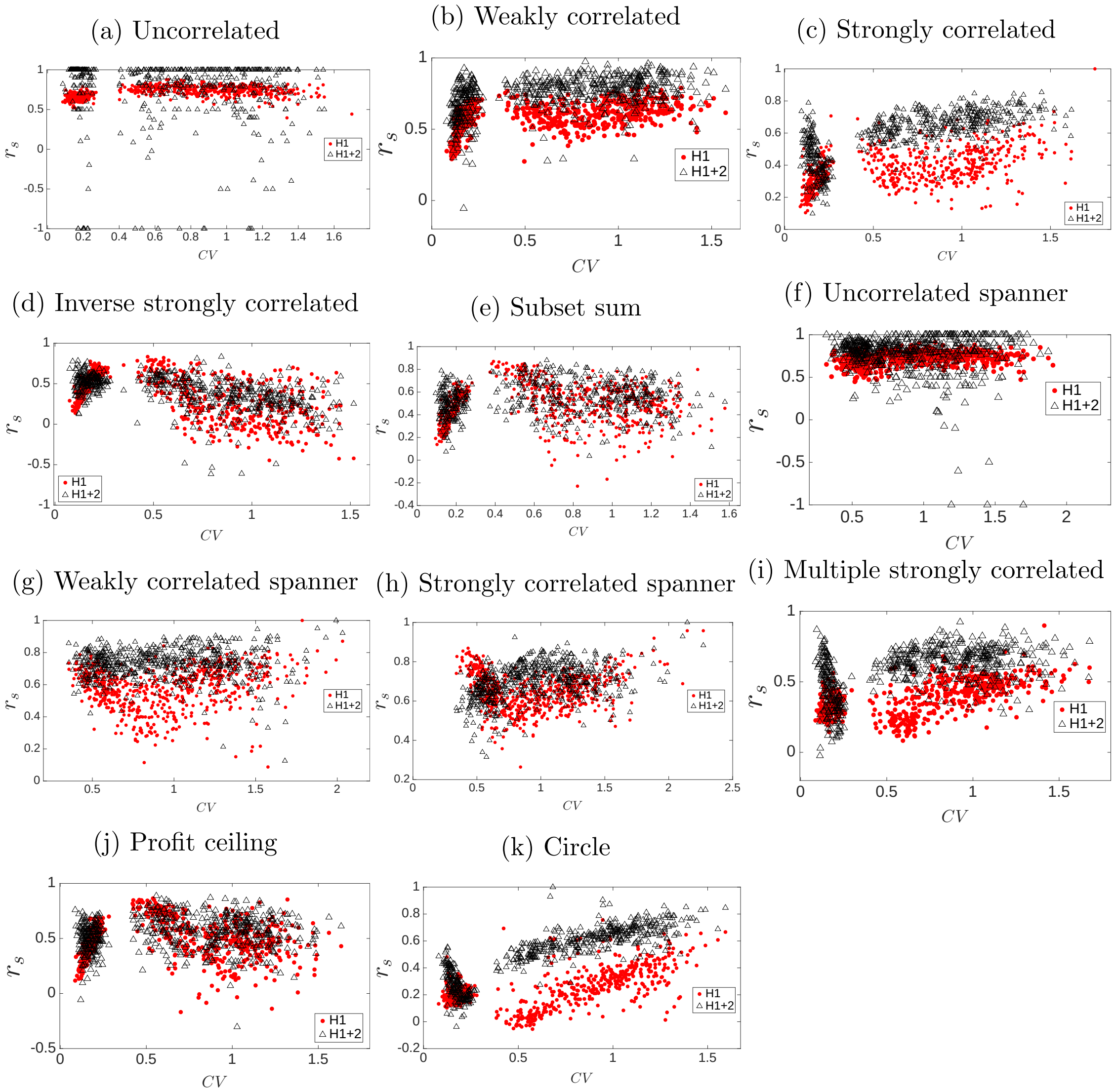

(a) Uncorrelated (b) Weakly correlated (c) Strongly correlated (d) Inverse strongly correlated (e) Subset sum (f) Uncorrelated spanner (g) Weakly correlated spanner (h) Strongly correlated spanner (i) Multiple strongly correlated (j) Profit ceiling (k) Circle

Figure 5.18: Spearman's rank correlation coefficient between basin size and fitness versus CV.The results are for 600 instances of size $n = 20$ and $k = 0.4$ for each problem type.

Figures 5.18 and 5.19 show the correlation between the basin size and the optimum fitness when $k = 0.4, 1$ respectively. The uncorrelated and uncorrelated spanner problem types have very strong positive correlation in both landscapes. This does not seem to change much across the values of k or CV. The correlation in the weakly correlated and weakly correlated spanner, does not seem to change very much between the values of CV, but they seem to change with the values of k. In the weakly correlated, when $k = 0.4$, the correlation in the *H1+2* landscape varies between strong positive and strong negative, while it is almost always strong positive when $k = 1$. The correlation of the *H1* landscape of this problem type however does not seem to be affected much by the values of k. In the weakly correlated spanner, when $k = 0.4$, the correlation in both landscapes is strong positive, but when $k = 1$, the correlation in the *H1* landscape seems to get weaker but never negative.


(a) Uncorrelated

(b) Weakly correlated

(c) Strongly correlated

(d) Inverse strongly correlated

(e) Subset sum

(f) Uncorrelated spanner

(g) Weakly correlated spanner

(h) Strongly correlated spanner

(i) Multiple strongly correlated

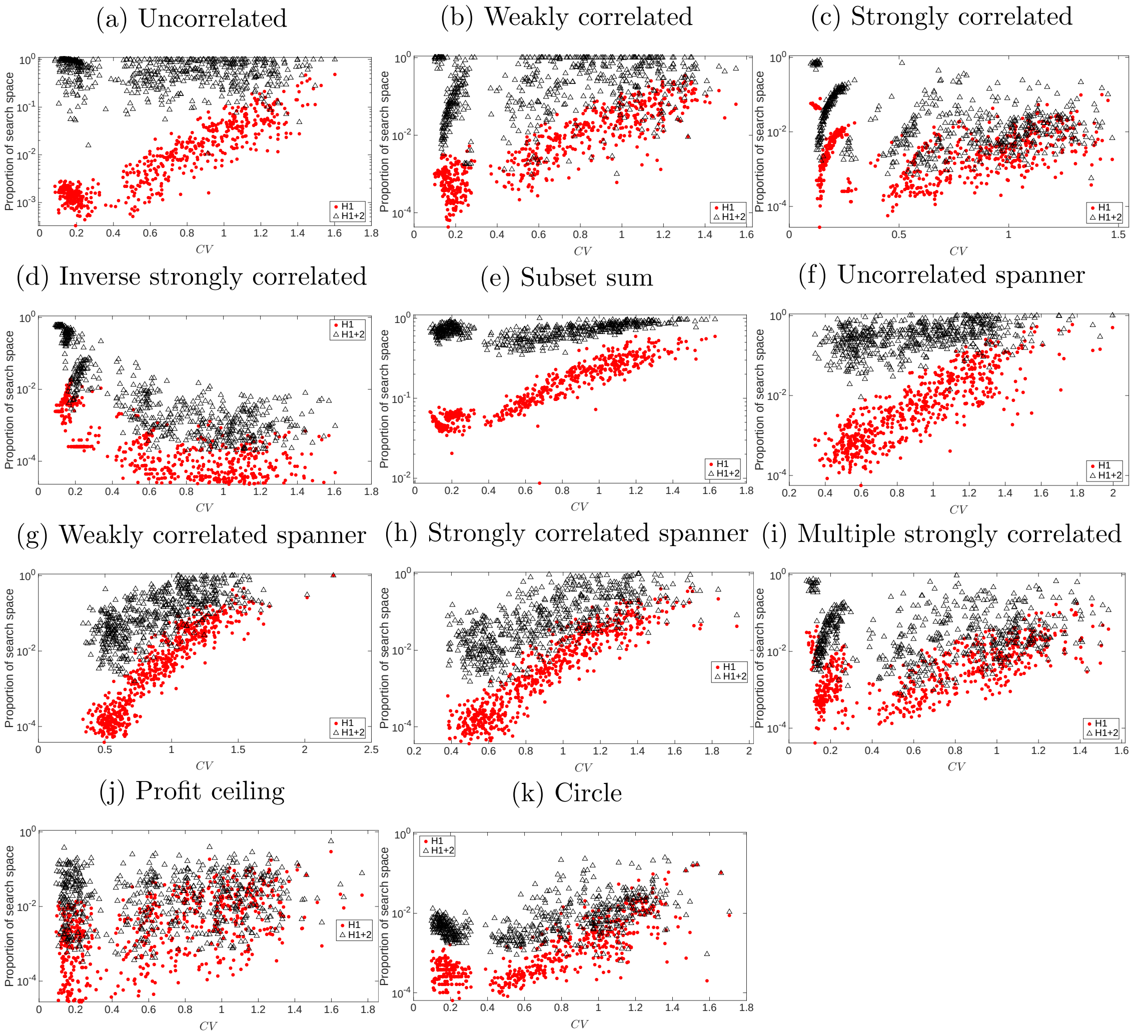
(j) Profit ceiling

(k) Circle

Figure 5.19: Spearman's rank correlation coefficient between basin size and fitness versus CV.The results are for 600 instances of size $n = 20$ and $k = 1$ for each instance type.

The correlation in the strongly correlated instances seems to increase slightly as the CV increases, in both values of k. The correlation in both landscapes of the inverse strongly correlated type seems to decrease (to sometimes strong negative correlation) as the CV increases, and this does not seem to change much between the values of k. The correlation in the subset sum, multiple strongly correlated, and profit ceiling instances does not seem to change very much between the values of k or CV, and generally it was found to vary between weak to strong positive. Similar results were seen in the strongly correlated spanner instances, but the correlation in this type was found to vary between moderate to strong positive correlation. In the circle problem type, the correlation in both landscapes seems to increase as the CV increases, and this does not seem to change much between the values of k.

5.4.3 Global basin


(a) Uncorrelated

(b) Weakly correlated

(c) Strongly correlated

(d) Inverse strongly correlated

(e) Subset sum

(f) Uncorrelated spanner

(g) Weakly correlated spanner

(h) Strongly correlated spanner

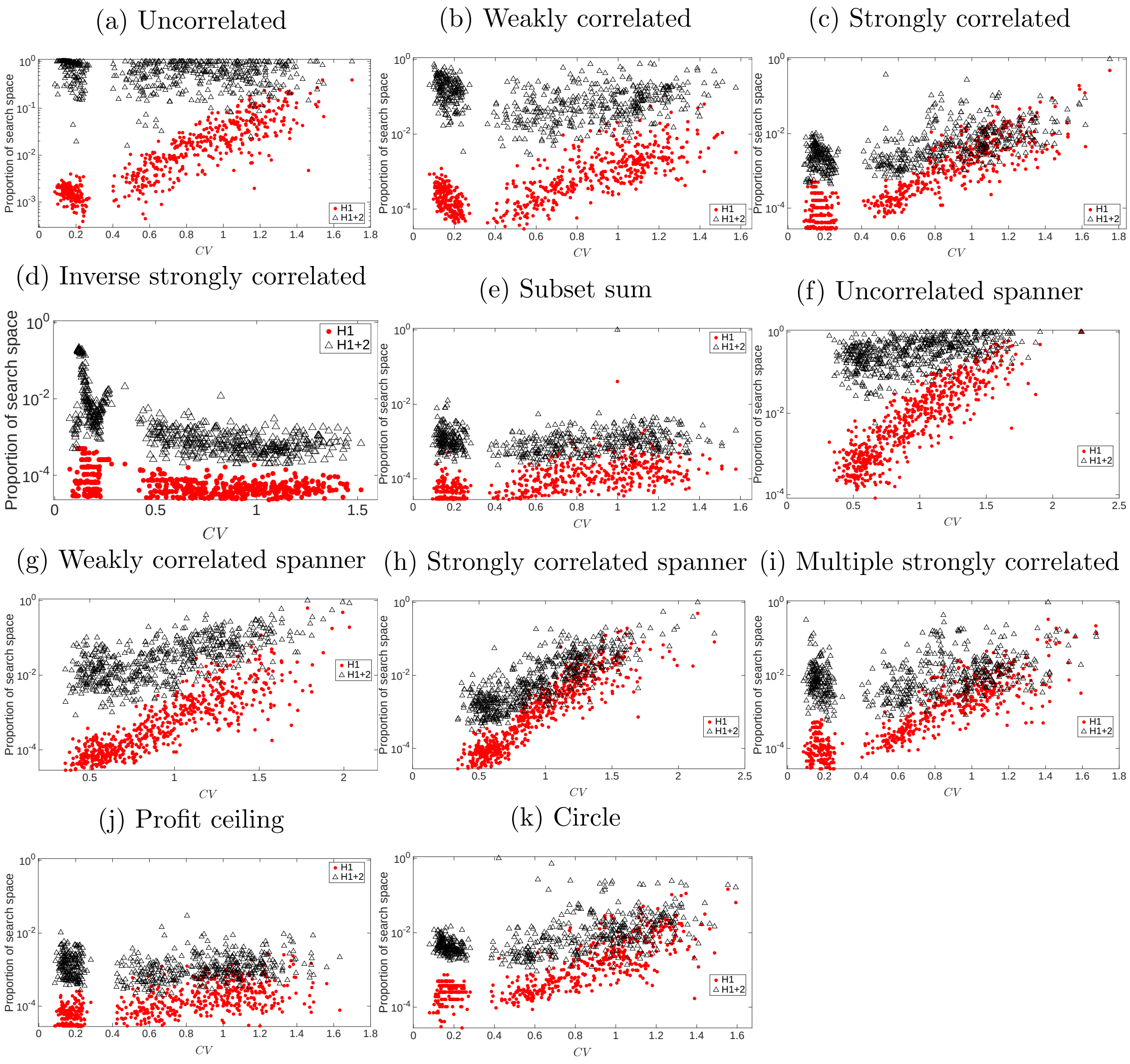
(i) Multiple strongly correlated

(j) Profit ceiling

(k) Circle

Figure 5.20: The proportion (in log scale) of the basin size of all the global optima found in an instance for each landscape against the CV. The results are for 600 instances of size $n = 20$ and $k = 0.4$ for each problem type. Notice how the probability of finding the global optimum increases with the CV in the *H1* landscape in all of the problem types apart from the inverse strongly correlated.

In a few of the problem types and especially in the subset sum, number of global solutions drops down as we cross the phase transition point (from $k = 0.4$ to $k = 1$). To examine how much the probability of finding the global is affect by this, we plot the total sum of all the global basin proportions found in instances of $n = 20$ against the CV for all the problem types in figures 5.20 and 5.21 for the values of $k = 0.4, 1$ respectively.



(a) Uncorrelated

(b) Weakly correlated

(c) Strongly correlated

(d) Inverse strongly correlated

(e) Subset sum

(f) Uncorrelated spanner

(g) Weakly correlated spanner

(h) Strongly correlated spanner

(i) Multiple strongly correlated

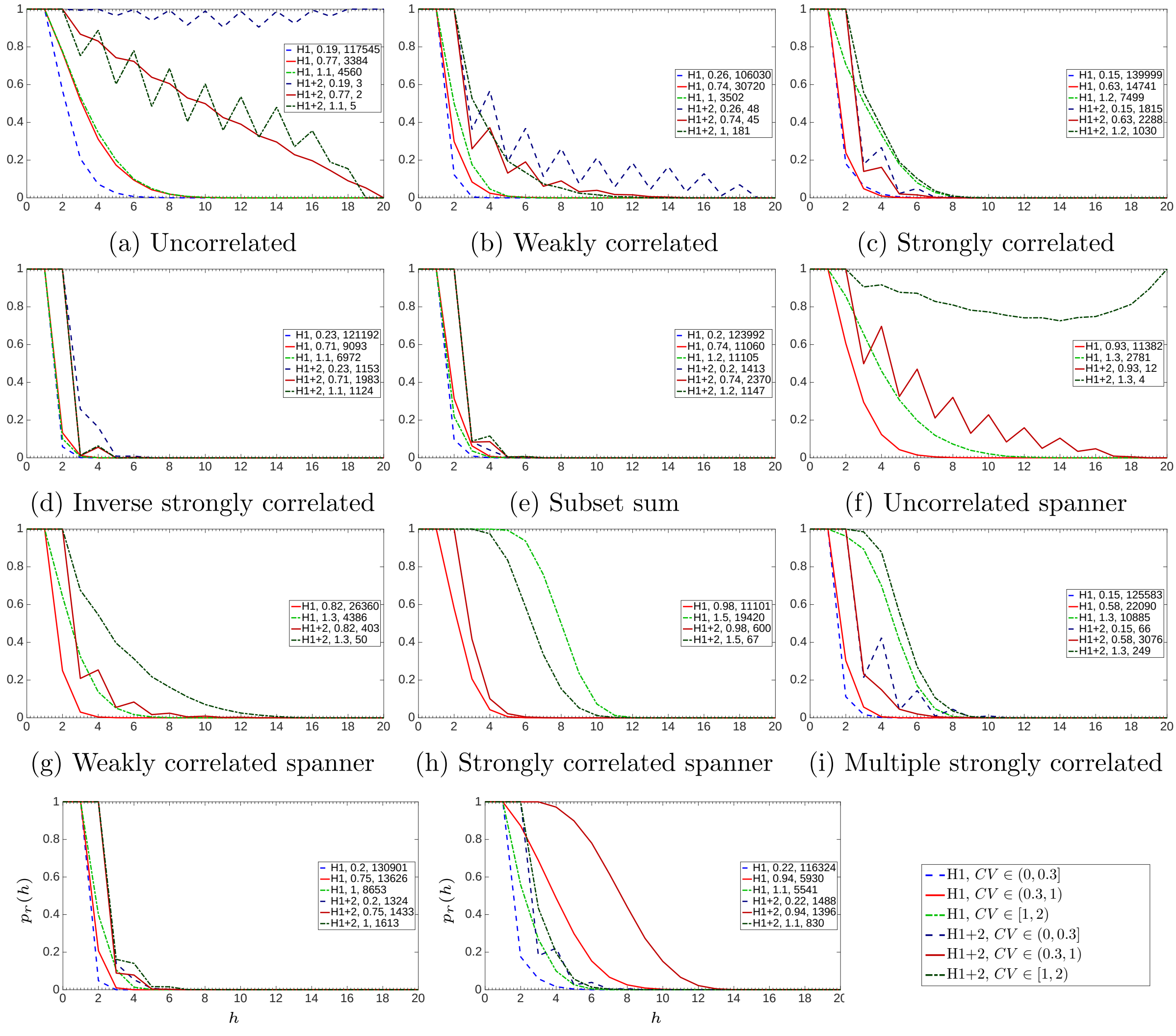
(j) Profit ceiling

(k) Circle

Figure 5.21: The proportion (in log scale) of the basin size of all the global optima found in an instance for each landscape against the CV. The results are for 600 instances of size $n = 20$ and $k = 1$ for each problem type. Notice how the probability of finding the global optimum increases with the CV in the *H1* landscape in all of the problem types apart from inverse strongly correlated.

Generally, the probability of finding the optimal solution is always higher in the *H1+2* landscape than that in the *H1*. In general, and especially in the *H1* landscape, the probability of finding the global increases as the CV increases. This can be attributed to the decrease in the number of local optima in this landscape as the CV increases, and the positive correlation between the basin size and fitness. In the inverse strongly correlated instances the opposite happens and the probability decreases as the CV increases, despite the fact that the number of local optima decreases as the CV increases. This reflects the results of the correlation between the basin size and the fitness in this problem type, where the correlation was found to be moderately to strongly positive in the small CV interval but it starts to decease as CV increases to a strong negative sometimes. We continue to see the effect of this feature on the performance of local search to find the global in this problem type in the next section. As in NPP, the probability of finding the global decreases in the hard phase ($k = 1$) of the subset sum problem. Note that the probability of finding the global is very high in the *H1+2* landscape of the uncorrelated instances.

In an attempt to study the shape of the global basin (one of the global basins if there is more than one), we plot in figure 5.22, the proportion of the configurations that are part of its basin in every Hamming sphere of radius h around it. The proportions were estimated as described in subsection 3.4. The results are shown for three instances of size $n = 20$ for each problem type. From the figure we can see that in the *H1* landscape the configurations in the global basin are concentrated in the immediate Hamming spheres around it, similar to what we have seen in the NPP. This is also the case in the *H1+2* landscape of most of the problem types. In the *H1+2* landscape of the uncorrelated, weakly correlated, and uncorrelated spanner types, the probability of returning to the global continues until the last sphere sometimes. This can be attributed to the very small number of optima in these instances. Again we continue to see the oscillating behaviour of the probability of return over the spheres in the *H1+2* landscape of some of the instances. This can be attributed to the very small number of optima in this landscape (e.g. only 3 in the instance shown for the uncorrelated type with $CV = 0.19$) and the nature of the *H1+2* neighbourhood, as the neighbours of a configuration in a given sphere h would be spread over five spheres using this neighbourhood operator compared to only two spheres when using the *H1* operator.



(a) Uncorrelated

(b) Weakly correlated

(c) Strongly correlated

(d) Inverse strongly correlated

(e) Subset sum

(f) Uncorrelated spanner

(g) Weakly correlated spanner

(h) Strongly correlated spanner

(i) Multiple strongly correlated

(j) Profit ceiling

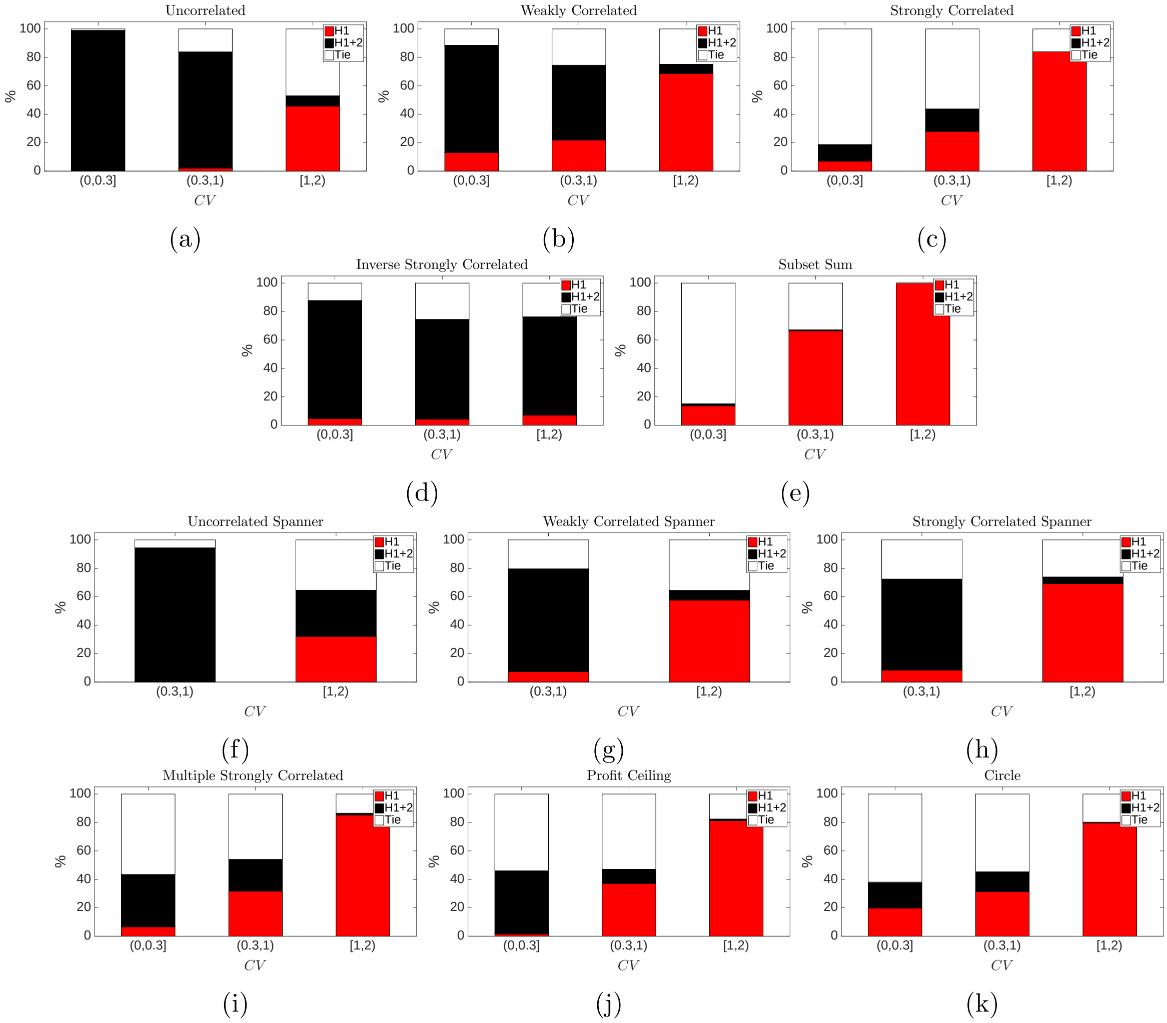
(k) Circle

Figure 5.22: Return probability $p_r(h)$ to the global optimum starting from a Hamming sphere of radius h (y-axis) versus h (x-axis). The results are for 3 instances of size $n = 20$ and $k = 1$ for each problem type. Additional information about each instance is shown in legends of every sub-figure, where each legend entry shows respectively: the landscape type, the instance *CV* value, and the number of optima in that landscape of that instance. Notice how the probability of return approaches zero faster in the *H1* case compared to *H1+2*.

5.5 Local Search

We study in this section the performance of local search, namely steepest ascent with random restart algorithm, using the two neighbourhood operators. We carry out the analysis of the algorithm performance from the perspective of the studied landscape features in the earlier sections.

5.5.1 Cost of finding the global



(a) (b) (c)

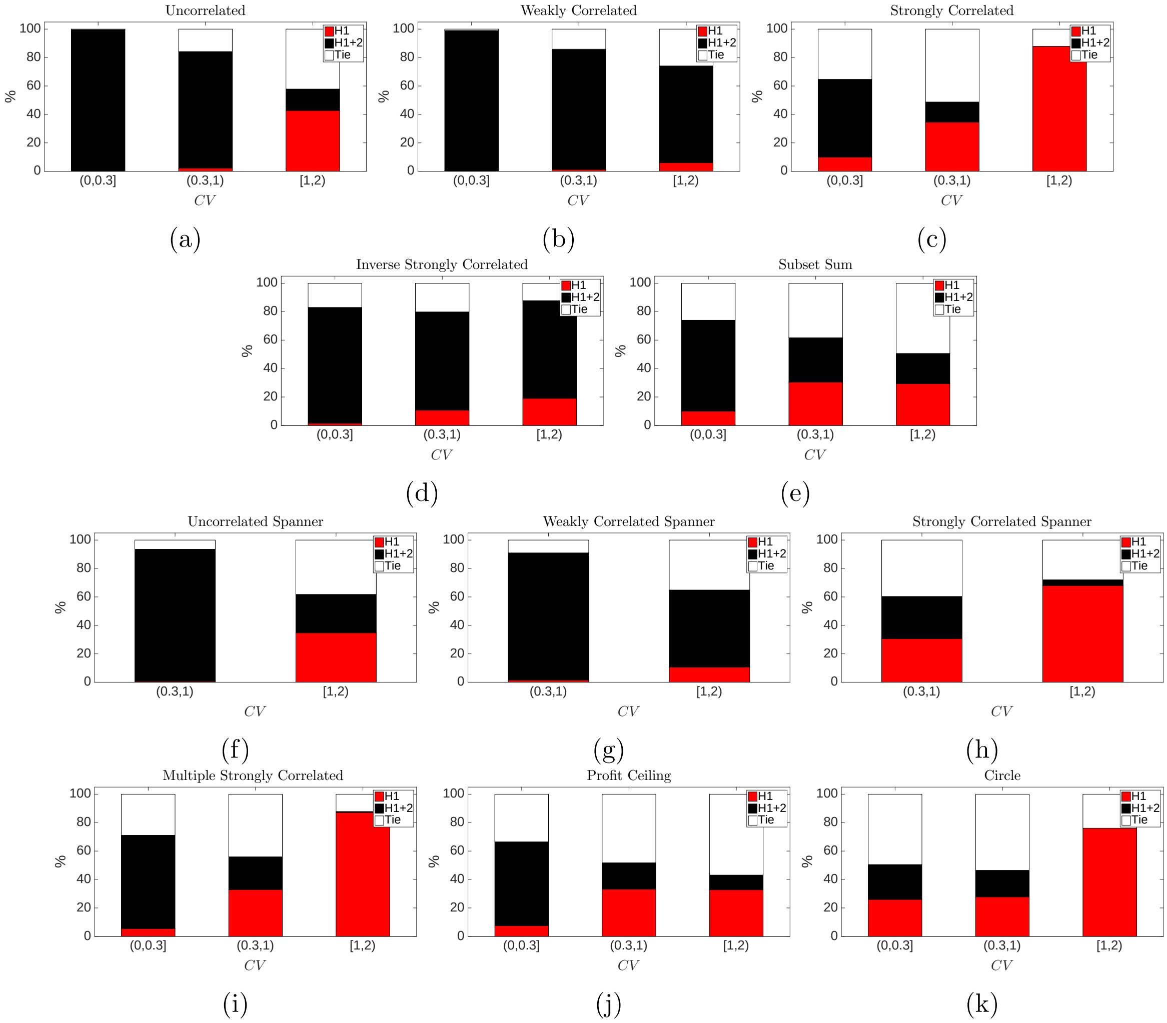
(d) (e)

(f) (g) (h)

(i) (j) (k)

Figure 5.23: Number of fitness evaluations used to find the global optimum averaged over 30 runs. The results show the percentage of instances where each operators performed significantly better and the percentage where no significance difference was found. Significance determined using Wilcoxon rank-sum ($p-$value ≤ 0.05). The results are for 600 instances for each problem type of size $n = 20$ and $k = 0.4$.

Figures 5.23, and 5.24 compare the performances of the two operators in terms of the number of fitness evaluations used to find the global when $k = 0.4$ and $k = 1$ respectively. The results for the uncorrelated, uncorrelated spanner, inverse strongly correlated, multiple strongly correlated and circle instances do not seem to be affected by the different values of k. In the subset sum, the *H1* operator was found to perform better in instances from the easy phase with $CV > 0.3$, while the *H1+2* operator was found to perform better in instances from the hard phase with $CV < 0.3$.



(a) (b) (c)

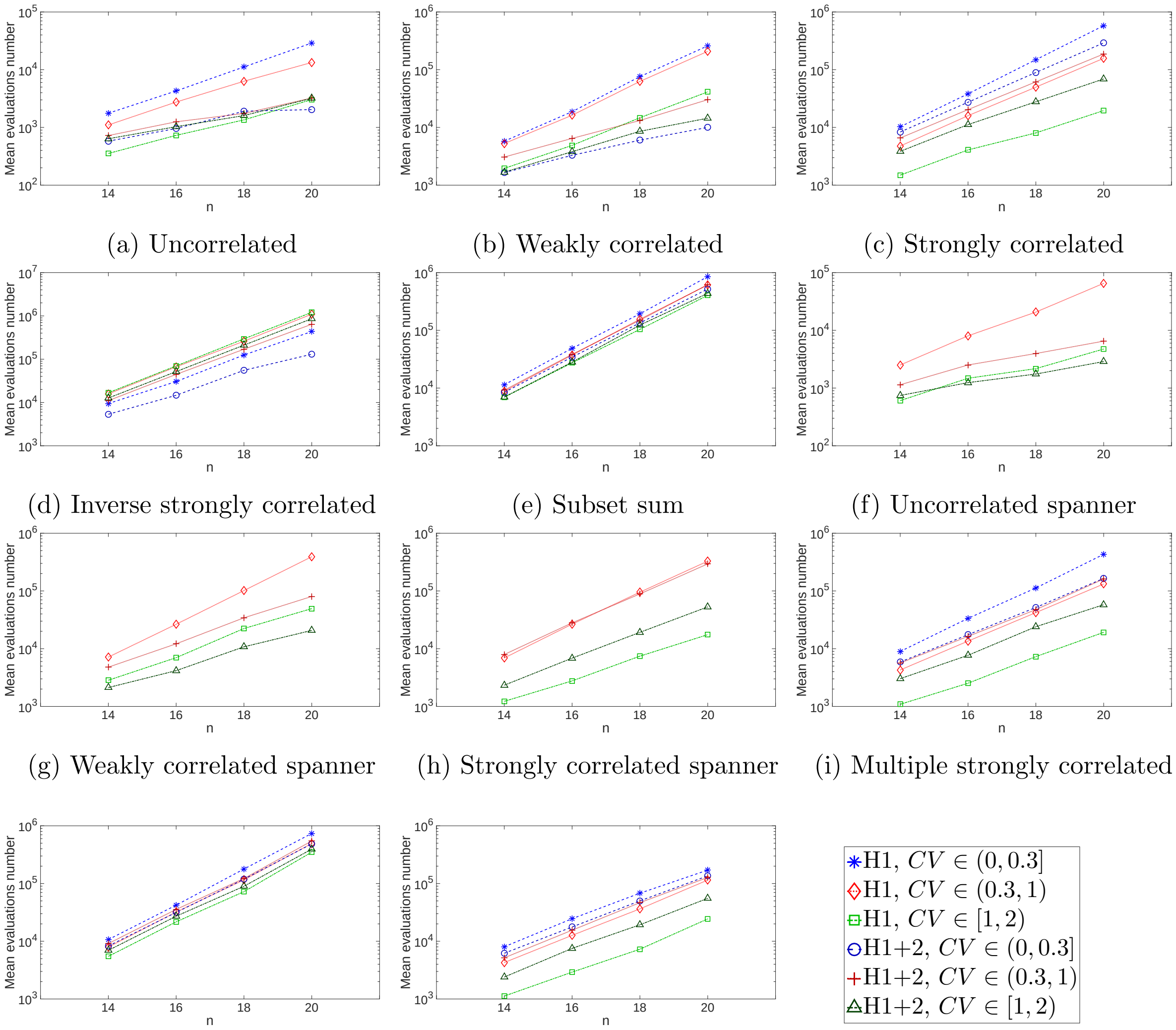
(d) (e)

(f) (g) (h)

(i) (j) (k)

Figure 5.24: Number of fitness evaluations used to find the global optimum averaged over 30 runs. The results show the percentage of instances where each operators performed significantly better and the percentage where no significance difference was found. Significance determined using Wilcoxon rank-sum ($p-$value ≤ 0.05). The results are for 600 instances for each problem type of size $n = 20$ and $k = 1$.

In the weakly correlated and the weakly correlated spanner, when $k = 0.4$, the *H1* operator was found to perform better in instances with $CV > 1$ and the *H1+2* was found to perform better in the rest of instances. However, when $k = 1$, the *H1+2* was found to perform better across all the CV values. This reflects the results we obtained about the probability of finding the global optima in these instances, where the probability in the *H1* landscape of the weakly correlated and weakly correlated spanner instances with large CV were higher when $k = 0.4$.



(a) Uncorrelated

(b) Weakly correlated

(c) Strongly correlated

(d) Inverse strongly correlated

(e) Subset sum

(f) Uncorrelated spanner

(g) Weakly correlated spanner

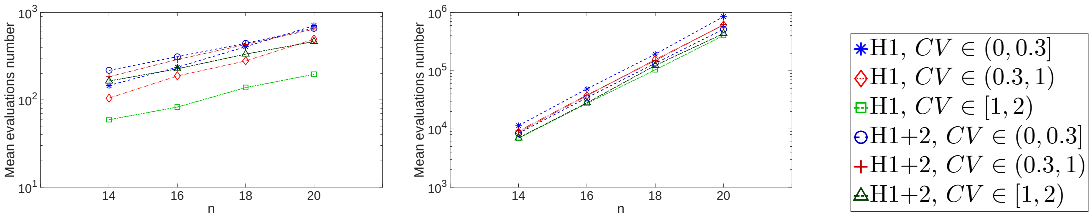
(h) Strongly correlated spanner

(i) Multiple strongly correlated

(j) Profit ceiling

(k) Circle

Figure 5.25: Number of fitness evaluations used to find the global (in log scale) against n. Each data point is an average of 30 runs of steepest ascent, averaged over the number of instances in each CV interval. The results for each n are for 600 instances with $k = 1$. Notice that in most of the problem types, there is the large difference in the average number of evaluations (almost one order of magnitude) between instances with $CV < 0.3$ and $CV \geq 1$ when *H1* operator is used. Also notice that the uncorrelated instances have the lowest cost of finding the global while inverse strongly correlated and subset sum have the highest.



(a) $k = 0.4$

(b) $k = 1$

Figure 5.26: Number of fitness evaluations used to find the global (in log scale) against n. Each data point is an average of 30 runs of steepest ascent, averaged over the number of instances in each CV interval. The results for each n are for 600 instances of subset sum. Notice the large difference in the average number of evaluations (almost one order of magnitude) between instances with $CV < 0.3$ and $CV \geq 1$ when *H1* operator is used. Also notice in (a) that the cost using *H1* operator is always lower or equal to the cost using the *H1+2*.

Figure 5.25 shows the mean number of fitness evaluations used to find the global as n grows for each problem type. Note that uncorrelated and uncorrelated spanner instances have the lowest cost of finding the global in both landscapes. This is a reflection of the very strong positive correlation between the basin size and fitness, and the higher probability of returning to the global in theses instances as we have seen in the previous section. Instances of type inverse strongly correlated have the highest mean cost of finding the global, apart from the cost of the $CV \in (0, 0.3]$ interval. The increase in the cost in the instances with $CV > 0.3$ can be attributed to the sometimes strongly negative correlation between the basin size and fitness, which in turn resulted in a lower probability of finding the global in these instances. In fact, we can see that the cost of finding the global using the *H1* operator in this problem type is the lowest in the CV interval $(0, 0.3]$. Despite the fact that the number of local optima is the highest in this interval, which translates in all the other problem types to having the highest cost of locating the global out of all the CV intervals. This again goes to show the importance of the correlation between the basin size and fitness (remember the correlation in `invscorr` was found to be moderately to strongly positive in $CV \in (0, 0.3]$ but it starts to decease as the CV increases to a strong negative sometimes). The straight lines in almost all the observations in each problem type indicate that the cost of finding the global seems to grow exponentially with n.

Figure 5.26 shows the cost of finding the optimal solution in the easy and hard phase of the subset sum. As we have seen before, the probability of finding the global is higher in the easy phase, the algorithm quickly finds one of the many global optima while it struggles to find the single global optimum in the hard phase. This explains why the cost of finding the global is much lower in the easy phase. Also, and like in the NPP, the growth in the hard phase is much faster than that in the easy phase.

5.5.2 Quality of optima obtained with fixed budget search

The previous results of the growth of the cost of finding the global give an indication that finding the global seems to be irrelevant as the problem size grows. Therefore, we look here at the quality of the optima obtained by a fixed budget of fitness evaluations. Figures 5.27 and 5.28 compare the performances of the two operators in terms of the quality of the obtained optima for $n = 30, 100$ respectively. In some of the problem types, the number of tie cases seems to decrease as n increases, and a clear winner emerges. In general, and from the results of $n = 100$, the *H1* operator was found to perform better in the strongly correlated, inverse strongly correlated and circle problem types, across all the CV intervals. The *H1+2* operator was found to perform better in the uncorrelated, weakly correlated, uncorrelated spanner, and weakly correlated spanner problem types, across all the CV values. In the multiple strongly correlated, profit ceiling, subset sum and strongly correlated spanner problem types, the *H1+2* operator was found to perform clearly better in the small CV ($(0, 0.3]$) interval ((0.3,1) for the strongly correlated spanner). The *H1* operator was found to clearly perform better in the $[1, 2)$ CV interval of the strongly correlated spanner and multiple strongly correlated problem types. In the multiple strongly correlated, the *H1* operator was found to also perform better in the $(0.3, 1)$ CV interval. Theses results can be explained by the difference in the quality of the optima between the two landscapes that we have seen in section 5.3.3. Note that these results are specific to the budget we selected, whether the same trends will continue to occur with other budget values remains an open question.

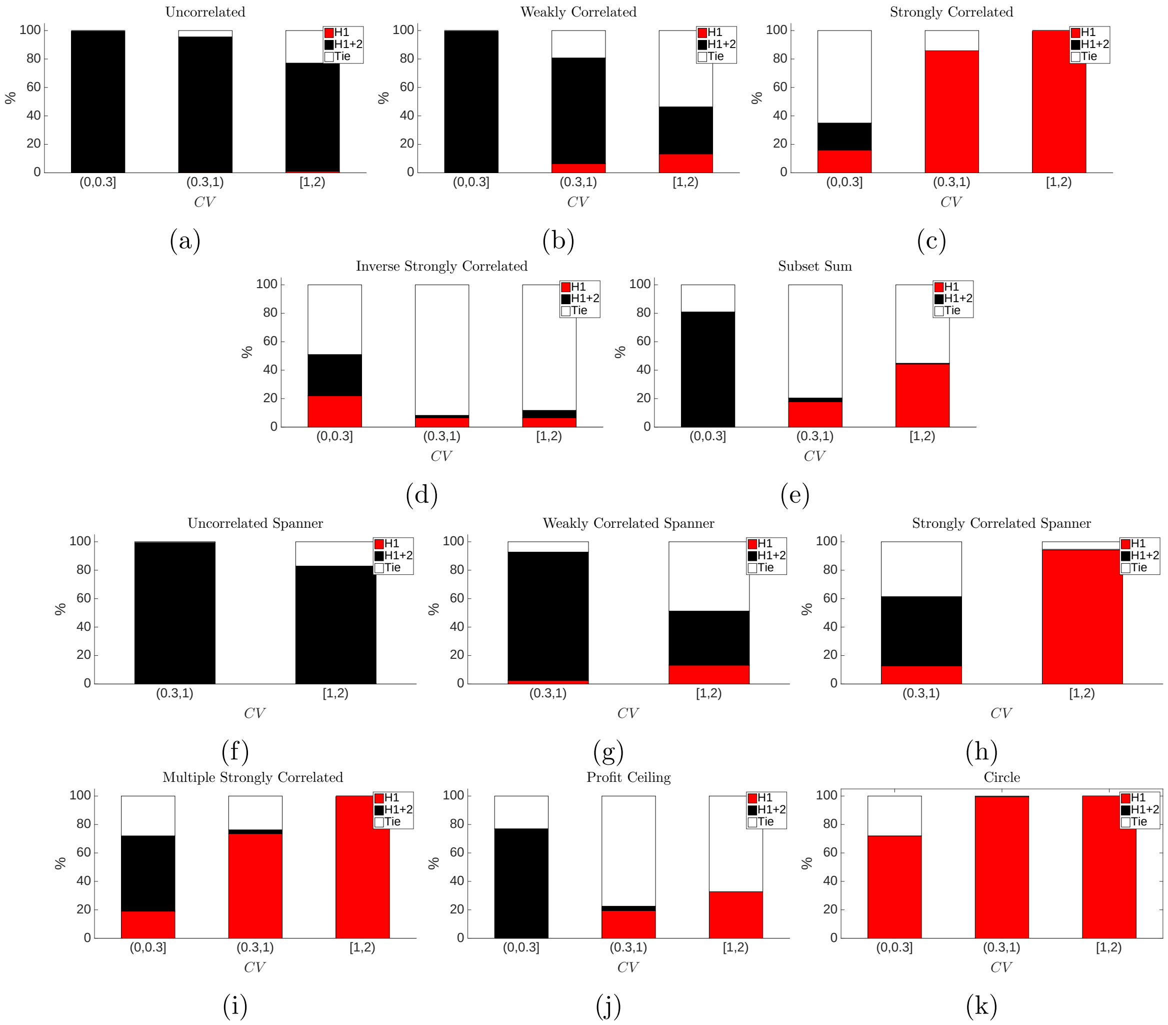
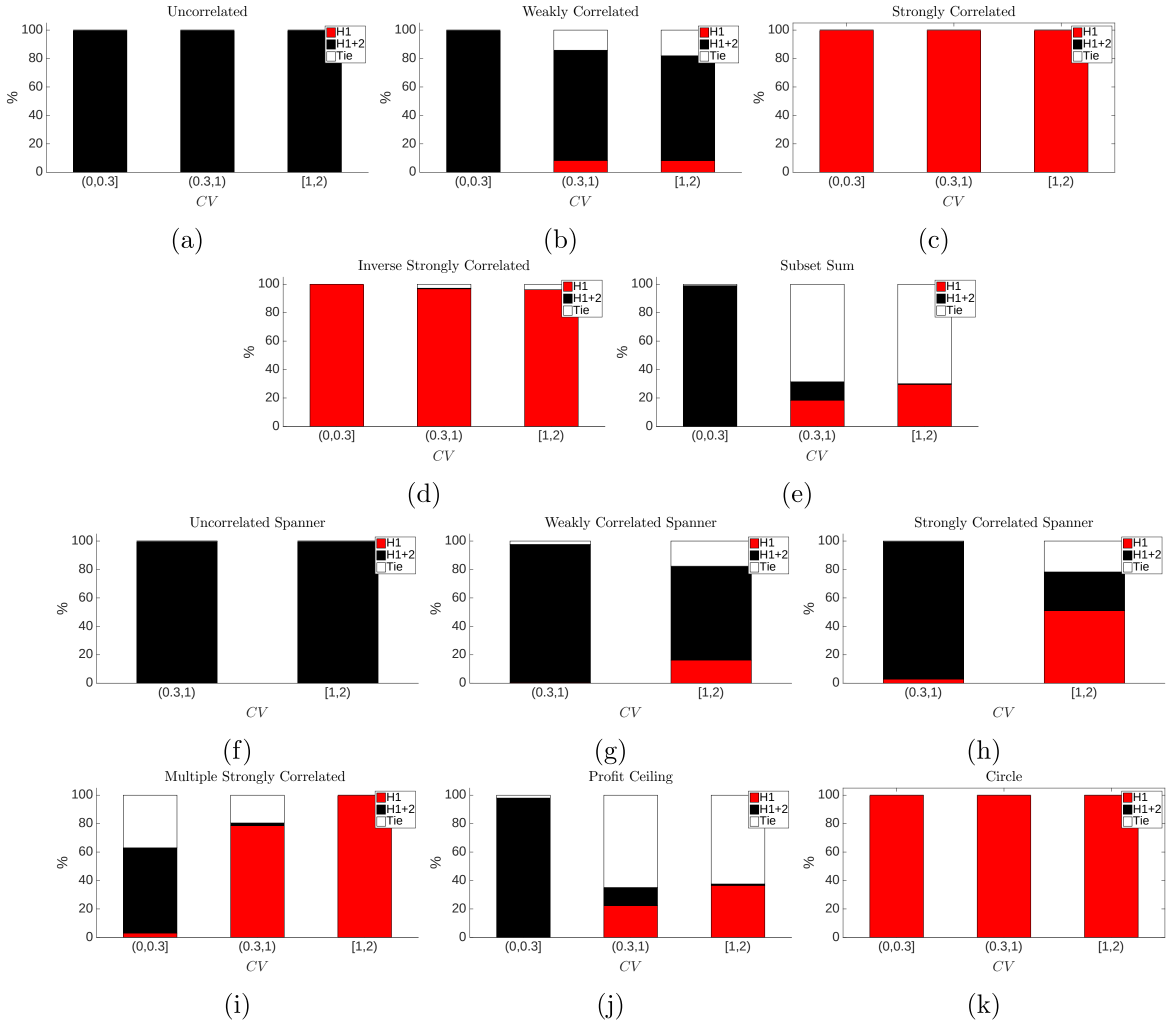


Figure 5.27: The quality of the solution found averaged over 30 runs of local search with fixed budget of 10^5 fitness evaluations. The results are for 500 instance for each instance type of size $n = 30$ and $k = 1$. The results show the percentage of instances where each operators performed significantly better and the percentage where no significance difference was found (Tie). Significance determined using Wilcoxon rank-sum ($p-$value ≤ 0.05).



(a) (b) (c)

(d) (e)

(f) (g) (h)

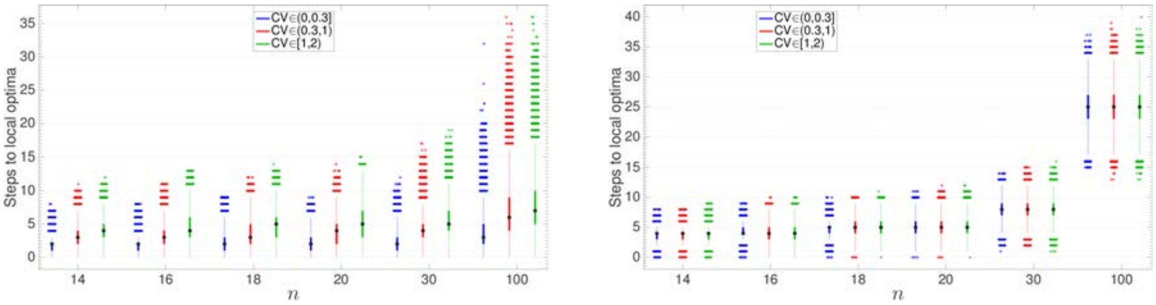
(i) (j) (k)

Figure 5.28: The quality of the solution found averaged over 30 runs of local search with fixed budget of 10^5 fitness evaluations. The results are for 500 instance for each problem type of size $n = 100$ and $k = 1$. The results show the percentage of instances where each operators performed significantly better and the percentage where no significance difference was found (Tie). Significance determined using Wilcoxon rank-sum (p—value ≤ 0.05).

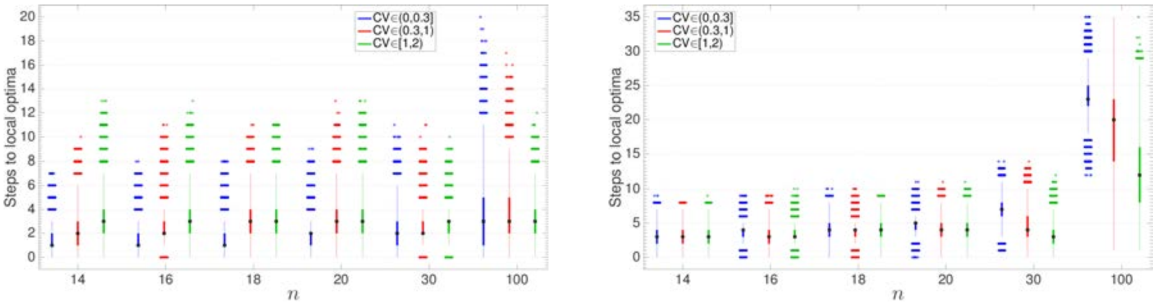
5.5.3 Time to local optima

The time it takes steepest ascent, starting from a random configuration until a local optimum is found, is shown as n grows in figure 5.32. In the *H1* landscape, and in most of the problem types, this was found to be similar to the case in NPP, in that, the number of steps is very small and grows slowly with n. Again, this is believed to be due to the large number of optima in this landscape and the exponential growth of the number of optima with the problem size. Note that the number of steps in the uncorrelated and the uncorrelated spanner instances is slightly higher, which is believed to be due to the larger basin sizes in these instances. In the *H1+2* landscape, the number of steps is still small and grows slowly for most problem types apart from uncorrelated, weakly correlated, uncorrelated spanner, weakly correlated spanner, strongly correlated spanner, and multiple strongly correlated instances. The number of steps in these problem types seems to grow faster with n. This is clearly shown when $n = 100$. This supports our observation that the decay of the number of local optima in the *H1+2* landscape of these problem types is faster than that in the other types. Note that the number of steps taken in the *H1* landscape was found to be always equal to the Hamming distance between the initial random configuration and the found local optimum. In the *H1+2* landscape this was found to be almost always smaller or equal to the Hamming distance between the initial random configuration and the found local optimum. However, in very few cases it was found to be one or two steps larger than the Hamming distance.

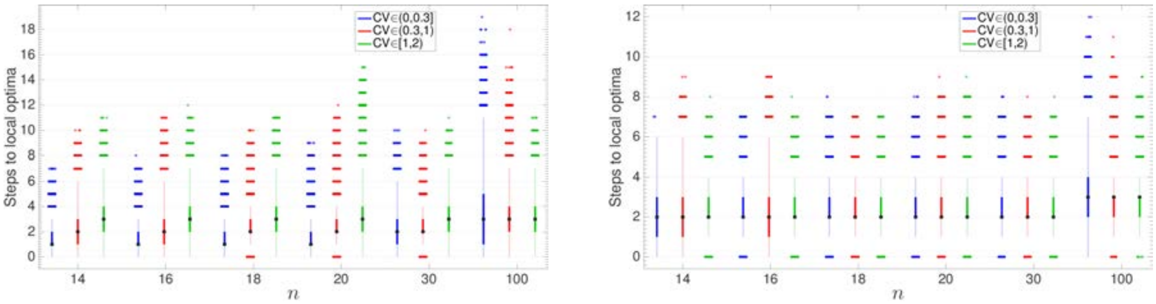
Uncorrelated



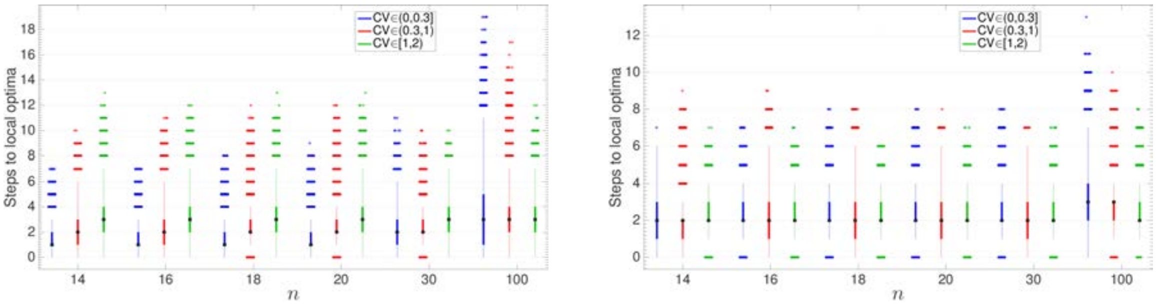
Weakly correlated



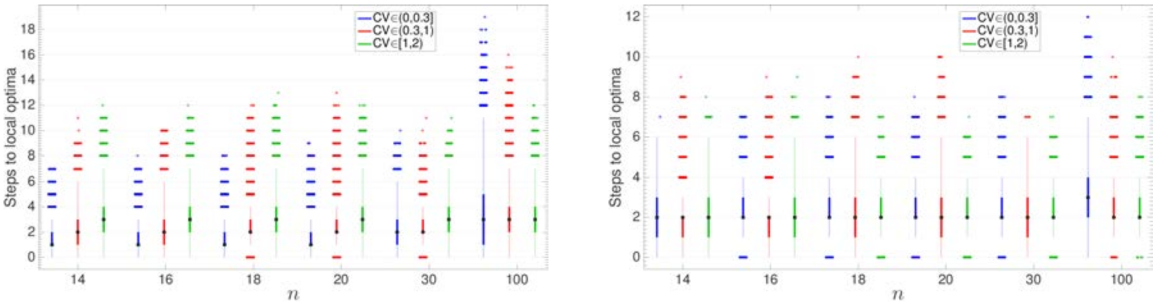
Strongly correlated



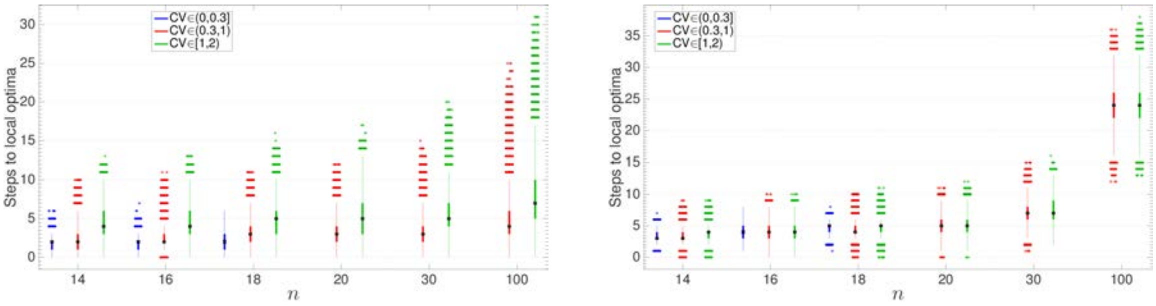
Inverse strongly correlated



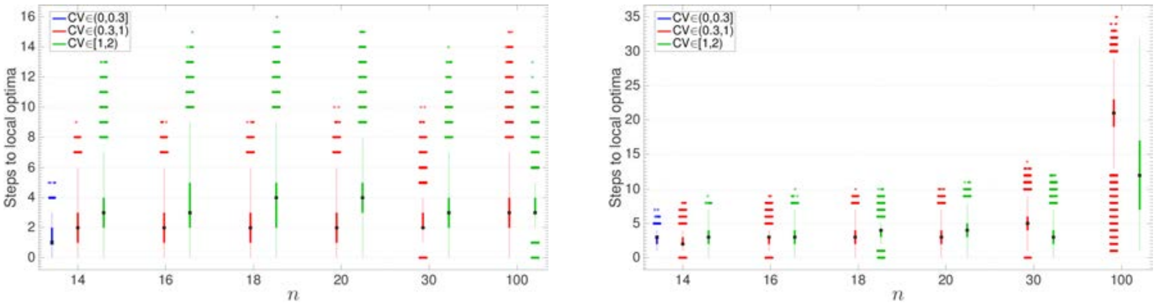
Subset sum

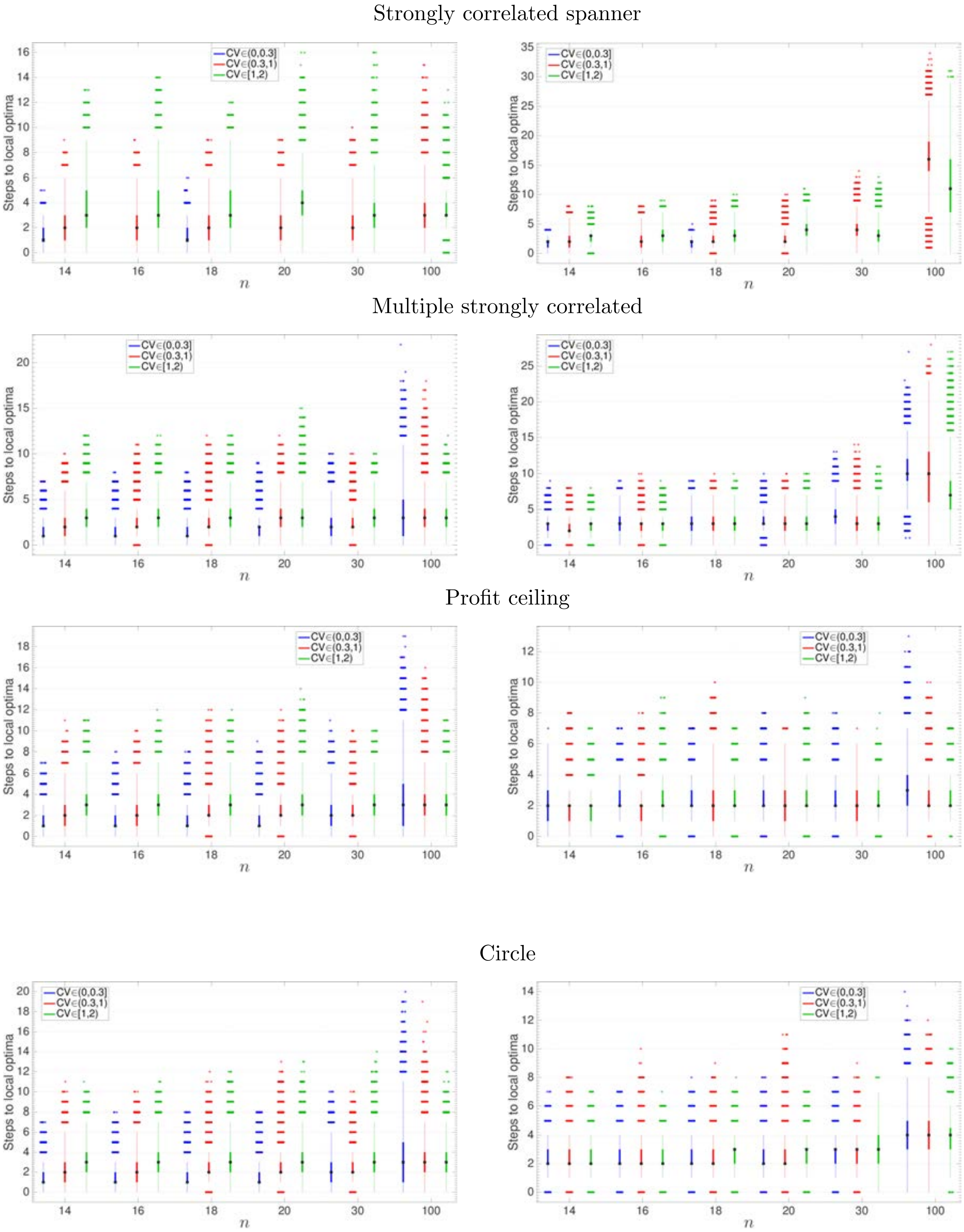


Uncorrelated spanner



Weakly correlated spanner





Strongly correlated spanner

Multiple strongly correlated

Profit ceiling

Circle

Figure 5.32: Number of steps starting from a random configuration until an optimum is reached when using *H1* (left) and *H1+2* operators (right). The results for each instance types are for 1000 steepest ascents per instance and 600 instances for each $n = 14, 16, 18, 20$ and 500 instances for $n = 30, 100$ ($k = 1$).

5.6 Summary

In this chapter, we empirically studied various properties of two fitness landscapes of random instances of 11 different problem types of the binary knapsack problem. We focused on how these properties change with k and the CV of the weights. The following is a summary of the main results:

- The logarithmic penalty function was found to create a strict local optimum in the infeasible region.
- The quadratic penalty function seems to direct the infeasible configurations to be part of the basins of lower quality optima as opposed to the linear penalty function.
- In subset sum, the only properties that were found to change when the problem crosses the phase transition is the number of global optima (and consequently the probability of finding the global), and the number of plateaux in the *H1+2* landscape.
- No configuration of type IPLAT has been found in either landscapes of all the problem types.
- Plateaux were only found in the *H1+2* landscape and mainly when $k = 0.4$.
- The number of local optima in the *H1+2* of the weakly correlated (when $k = 0.4$) and uncorrelated problem types was found to be very low, as low as only one or two optima sometimes.
- In all the problem types, there is a very strong and negative correlation between the CV and the number of local optima in the *H1* landscape of moderately constrained instances.
- The average number of local optima in the *H1* landscape of all the problem types seems to be well approximated by the same formula used to estimate the average number of local optima in the *H1* landscape of NPP.

- The number of local optima seems to grow exponentially in the *H1* landscape of all the problem types.

- The quality of optima and the difference in the quality between the two landscapes were found to vary across the problem types.

- In all the problem types, the distribution of the basin sizes was found to be skewed with many small basins and only few large ones, particularly in the *H1* landscape.

- In general, the correlation between the basin size and fitness was found to vary between weak to strong positive in both landscapes, apart from instances of type inverse strongly correlated with $CV > 0.3$ and few other cases where the correlation was found to be negative. This negative correlation is in fact unusual in the combinatorial optimisation problems studied in the literature, where in general fitter optima were found have larger basins [104, 112]. We can see that in the case of the inverse strongly correlated this negative correlation between fitness and basin size has totally changed the complexity of finding the global using local search between instances with small $CV \leq 0.3$ and instances with large $CV > 0.3$. The instances with small CV have more number of optima than the instances with large CV (by almost more than one order of magnitude). However, because the correlation between fitness and basin size in the instances with large CV is negative that translated into an increase in the cost of finding the global in these instance to be almost one order of magnitude more than the cost of finding the global in the instances with small CV.

- The performance of local search algorithms was found to be affected by the phase transition in subset sum, as shown by the considerable increase in the cost of locating the global solution when $k > k_c$.

- Unlike in the NPP, the trends of the winner operator were found to change in some of the problem types between when searching for the global and the fixed budget search. They also were found to change with n. In terms of the quality of the obtained optima with fixed budget search, the *H1* operator performs better in the strongly correlated, inverse strongly correlated and circle problem types, across all the CV intervals. The *H1+2* operator performs better in the uncorrelated, weakly correlated, uncorrelated spanner, and weakly correlated spanner problem types, across all the CV values. In the multiple strongly correlated, profit ceiling, subset sum and strongly correlated spanner problem types, the *H1+2* operator was found to perform clearly better in the small CV $(0, 0.3]$ interval ((0.3,1) for the strongly correlated spanner). The *H1* operator clearly performs better in the multiple strongly correlated problem type in instance with $CV > 0.3$.

- The time it takes steepest ascent, starting from a random configuration until a local optimum is found, grows very slowly with n in the *H1* landscape. The time in the *H1+2* landscape seems to grow faster for the uncorrelated, weakly correlated, uncorrelated spanner, weakly correlated spanner, strongly correlated spanner, and multiple strongly correlated problem types.

Table 5.1: Overview of main differences between the problem types

Problem Type	**Number of local optima**		**Quality of optima with fixed budget**	**Time to optima using H1+2**
	H1 landscape	**H1+2 Landscape**		
Uncorrelated	The number of optima is more or less the same across all the types and it increases exponentially with n. The number is the highest in the CV interval $(0, 0.3]$ and it starts decreasing as the CV increases.	Has the lowest number of local optima. The optima proportion decays faster than in the rest of types.	H1+2	Grows faster with n compared to the rest of types.
Weakly Correlated		Has a lower number of local optima than the rest. The optima proportion decays faster than in the rest of types.	H1+2	
Strongly Correlated			H1	
Inverse Strongly Correlated			H1	
Subset Sum			H1+2 when $CV < 0.3$	
Uncorrelated Spanner		Generally has a lower number of local optima than the rest. The optima proportion decays faster than the rest.	H1+2	Grows faster with n compared to the rest of types.
Weakly Correlated Spanner			H1+2	
Strongly Correlated Spanner			H1+2 when $0.3 < CV < 1$	
Multiple Strongly Correlated			H1+2 when $CV \leq 0.3$ H1 when $CV > 0.3$	
Profit Ceiling			H1+2 when $CV \leq 0.3$	
Circle			H1	

CHAPTER 6

QUADRATIC 0-1 KNAPSACK PROBLEM

In this chapter we study the landscape properties of another NP-hard problem, the quadratic binary knapsack problem (0-1QKP). This problem is a variant of the 0-1KP, where the profit associated with an item depends also on the other selected items. As in the previous two chapters, we study various landscape features of a large number of randomly generated instances with different values of problem parameters. One of these parameters is the weights distribution. Instances were generated by drawing weights from the five different distributions shown in subsection 3.2. However, in this chapter and as with the Knapsack chapter, we abandon the use of the underlying distribution of the weights to describe the problem instance, and only use the weights CV to do so. We also carry out grouping the instances based on their CV values into the three intervals: $(0, 0.3]$, $(0.3, 1)$, and $[1, 2)$.

6.1 Problem Definition

Given a knapsack of capacity C and a set of n items each with associated weight w_i, in addition to an $n \times n$ non-negative integer matrix $P = p_{ij}$, where p_{jj} is the profit achieved if item j is selected and $p_{ij} + p_{ji}$ is the profit achieved if both items i and j are selected (for $i < j$) [34]. The aim of the 0-1QKP is to find a subset of items that maximises the profit without exceeding the knapsack capacity. The density of the profit matrix, that is the percentage of non-zero elements, is given by Δ. The quadratic fitness function to be maximised is as follows:

$$f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij} x_i x_j \tag{6.1}$$

subject to the linear constraint

$$\sum_{i=1}^{n} w_i x_i \leq C, \; x \in \{0,1\}^n \tag{6.2}$$

where

$$C = \lambda \sum_{i=1}^{n} w_i, \; 0 \leq \lambda \leq 1. \tag{6.3}$$

As with any constrained optimisation problem, the 0-1QKP search space, $X = \{0,1\}^n$, is partitioned into a feasible region $F = \{x \in X \mid \sum_{i=1}^{n} x_i w_i \leq C\}$ and an infeasible region $\mathit{INF} = X \setminus F$. For $\lambda = 1$, there are no infeasible solutions and as the value of λ decreases, the size of the infeasible region increases until $\mathit{INF} = X$ when $\lambda = 0$. We define the boundary between feasible and infeasible regions as the set of feasible configurations that have at least one infeasible neighbour, $B = \{x \in X \mid x \in F \wedge \exists y : (y \in N(x) \wedge y \in \mathit{INF})\}$. Note that all the optima in this problem reside in the boundary, as pointed out by Gottlieb [40] about the optima of all covering and packing problems.

We only consider instances where the profit matrix is symmetric, i.e. $p_{ij} = p_{ji}$. We study instances where p_{ij} and w_i are positive integers drawn at random from the set $\{1, 2, \ldots, M\}$. In this chapter, we only study instances where the profits and the weights are uncorrelated. The weights are drawn at random from the five different distributions shown in subsection 3.2. The profits are drawn at random from the uniform distribution. We also continue to investigate setting k ($k = log_2 M/n$) to 0.4 and 1 as in the previous chapters. In addition, we explore the effect of varying the density of the profit matrix on the landscape by studying instances with $\Delta = 0.1$, 0.25, 0.5, 0.75, 0.95, and 1. As with the 0-1KP, we study instances where λ is set to 0.5, this is motivated by the fact that moderately constrained instances have the largest boundary sizes and thus the largest number of optima.

The 0-1QKP is NP-hard in the strong sense [35, 84], it cannot be solved by a pseudo-polynomial time algorithm unless P=NP [35]. Wide spectrum of real world problems can be formulated as instances of the 0-1QKP. One example is the allocation of airports or railway stations where the global traffic between the stations needs to be maximised under a constrained budget [95, 34, 84].

6.1.1 Constraint Handling

As with the 0-1KP, we use a penalty-based approach to handle the constraint. An infeasible solution x that violates the given constraint is penalised by a value $\text{Pen}(x) > 0$, while $\text{Pen}(x) = 0$ for a feasible solution x. The fitness functions after adding the penalty term is as follows:

$$f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij} x_i x_j - \text{Pen}(x) \tag{6.4}$$

The choice of an appropriate penalty function is very critical. Gottlieb [40] notes that some penalty-based algorithms suffer from the feasibility problem, that is they often terminate with completely infeasible solutions, due to inappropriate choice of the penalty function. As with the 0-1KP, we want to allow the infeasible solutions to be part of the searchable space and we want to penalise them proportional to the degree of violation of the constraint. Also, we want all infeasible solutions to have lower fitness values than all the feasible solutions. To ensure that, we added the offset term $\sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}$ to the penalty function. We ruled out the logarithmic penalty function as it creates a strict local optimum in the infeasible region as shown in the 0-1KP chapter. We also ruled out the use of the linear one as it was found to create a strict local optimum (the all ones solution $x = (1, \cdots, 1)$) in the infeasible region of some instances with highly dense profit matrix. It was also found to create some open and closed plateaux in the infeasible region of some instances with various values of Δ. The quadratic penalty function was found to induce a landscape with a smooth infeasible region that does not have any local optima or plateaux. Therefore, we use the quadratic penalty function to handle the constraint in this problem. The function is defined as follows:

$$\text{Pen}(x) = \left(\rho\left(\sum_{i=1}^{n} x_i w_i - C\right)\right)^2 + \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij} \tag{6.5}$$

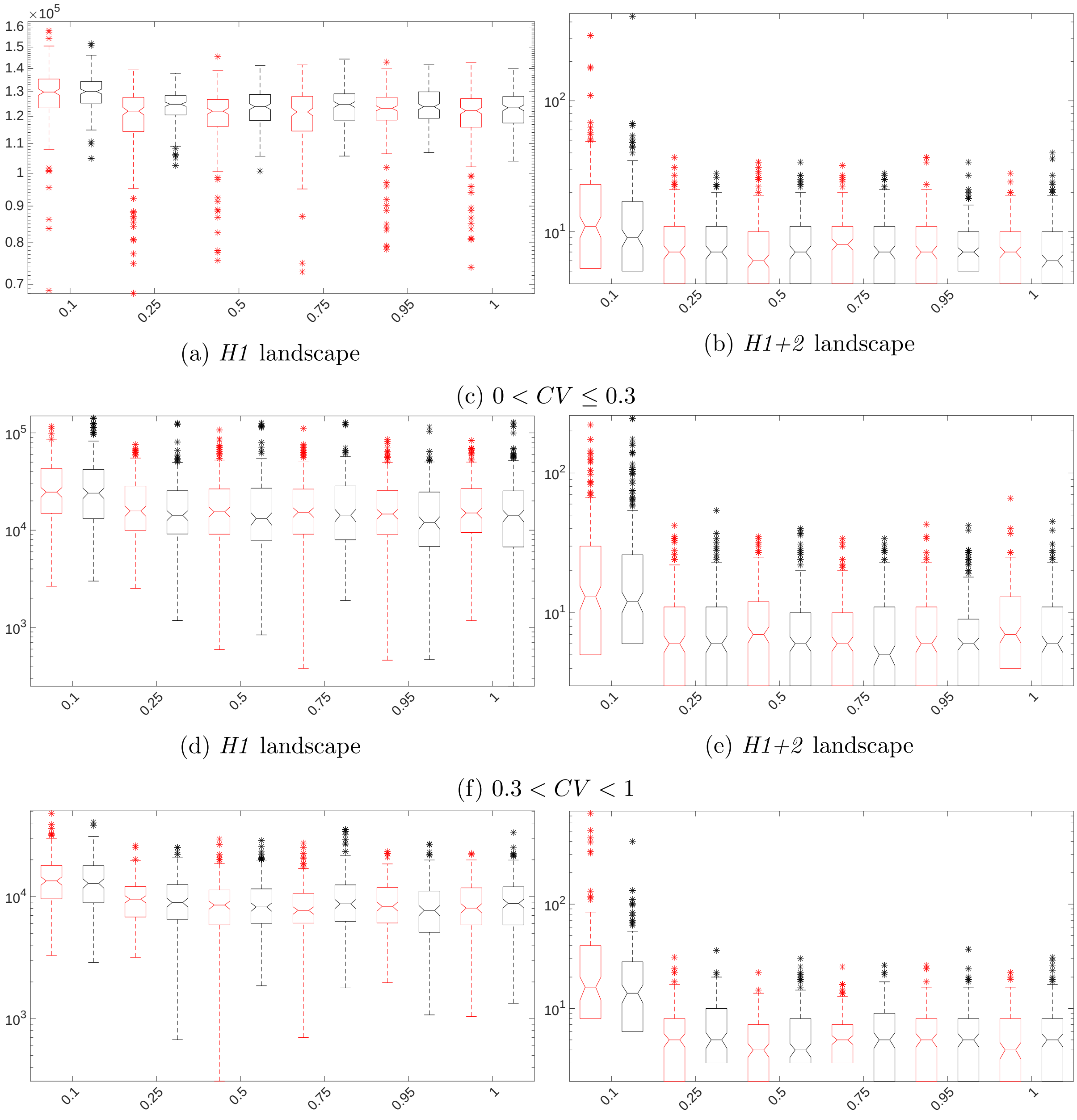
where $\rho = \max_{i,j=1,\ldots,n}\{p_{ii} + p_{ij} + p_{ji}\} / \min_{i=1,\ldots,n}\{w_i\}$.

6.2 Search Position Types

The tables in appendix C show the search position types found in randomly generated instances of the 0-1QKP with different values of the CV and k for all values of Δ. Tables C.1 to C.6, show the types proportions in the feasible regions of the *H1* and the *H1+2* landscapes. Tables C.7 to C.12, show the types proportions in the infeasible regions of both landscapes. Very few configurations of type IPLAT were found in the feasible region of the *H1* landscape in instances with very small $\Delta \leq 0.25$. The existence of IPLAT types in such instances is not surprising, since the very low density of the profit matrix in these instances results in many solutions sharing similar fitness values. The *H1* landscape of the rest of the instances and all the *H1+2* landscapes have no configuration of type IPLAT, which is similar to the findings in the previous two chapters. The configurations in the infeasible region were of types: SLMIN, LEDGE, and SLOPE. The SLOPE configurations seem to disappear in the infeasible region of the *H1+2* landscape when k goes from 0.4 to 1. Apart from that, not much difference is found between the two values of k across all the different parameters and for both landscapes and regions. In the feasible region of both landscapes, there are, as expected, more plateaux in instances with sparse profit matrix. As the density of the matrix increases, the number of plateaux decreases. We can clearly see that in the number of configurations of type NSLMAX, which was found to decrease as Δ increases.

6.3 Optima and Plateaux

6.3.1 Number of optima and plateaux



(a) *H1* landscape

(b) *H1+2* landscape

(c) $0 < CV \leq 0.3$

(d) *H1* landscape

(e) *H1+2* landscape

(f) $0.3 < CV < 1$

(g) *H1* landscape

(h) *H1+2* landscape

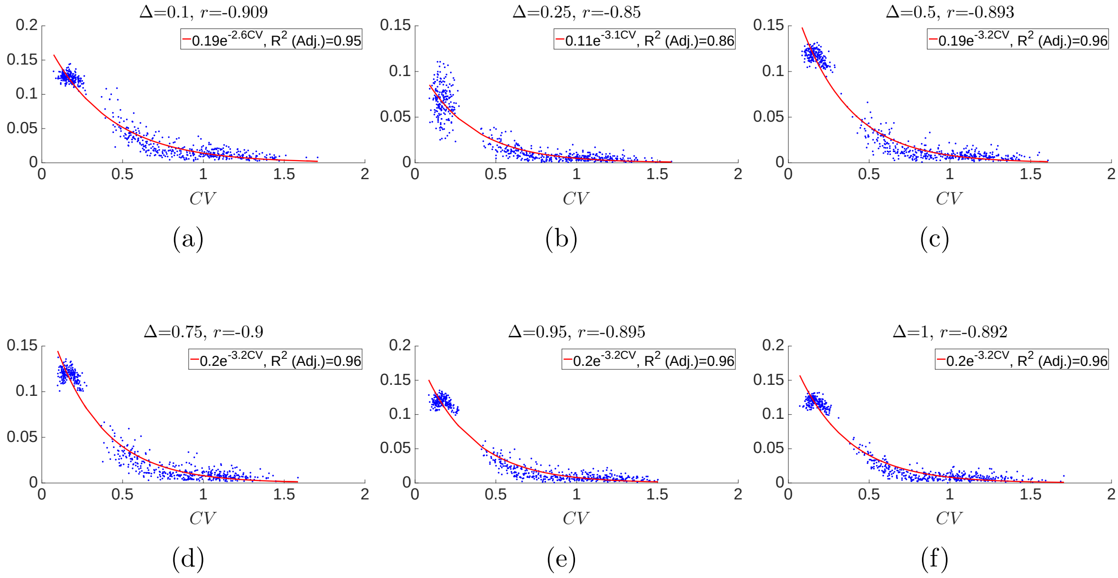
(i) $1 \leq CV < 2$

Figure 6.1: Number of both strict and non-strict local optima (in log scale) found in 600 instances of size $n = 20$ against Δ for each k value. The red boxes show the $k = 0.4$ results and the black ones show the $k = 1$ results.

The number of global optima was found to be the same across the values of k and CV. In general, there is only one optimal solution (the number was found to be slightly higher in instances with very sparse profit matrix $\Delta = 0.1$). Figure 6.1, shows the number of both strict and non-strict local optima. Again in both landscapes and across all the CV intervals, there is no difference in the number of optima between the values of k. As in the uncorrelated problem types in the 0-1KP, the number of local optima in the *H1+2* landscape is very small, less than ~ 100, and with median ≤ 10. Also, and as in NPP and the 0-1KP, the number of optima in this landscape does not seem to change much across the CV values. Similar to the previously studied problems, the number of optima in the *H1* landscape is the highest in the small CV interval and it starts decreasing as the CV increases. In both landscapes, the number of local optima seems to be slightly higher in instances with very sparse profit matrix $\Delta = 0.1$, apart from that the number of optima does not seem to change much between the values of Δ. Note that the number of non-strict local optima decreases as Δ increases until almost all of them become strict optima in instances with very dense profit matrix. In both landscapes, the non-strict local optima form either closed or open plateaux of very small sizes (less than five configurations). The number of exits in an open plateau of an instance with very sparse profit matrix ($\Delta = 0.1$) can reach up to 14 (for $n = 20$). For the rest of Δ, the number of exits drops to only one or two.

6.3.2 Average number of strict local optima

As with the NPP and the 01-KP, the number of strict local in the *H1* landscape was found to be strongly and negatively correlated with the CV across all values of Δ as shown in figure 6.2. In general and as in the previous chapters, the formula $a\,e^{-bCV}$ seems to be a good approximate of the average number of strict optima in the 01-QKP (the values of the coefficients a and b depend on n). However, and as in the 0-1KP, this seems to be nosier, especially in the small CV interval $(0, 0.3]$. The estimated number of local optima in the *H1* landscape of $n = 30, 50, 100$ was also found to follow this trend.

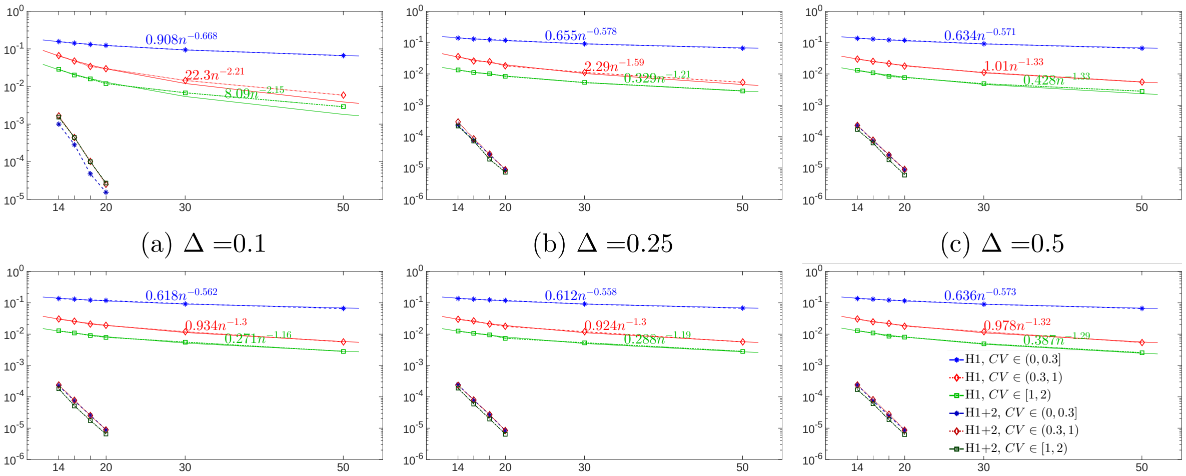


(a) (b) (c)

(d) (e) (f)

Figure 6.2: The fraction of strict local optima in the *H1* landscape versus CV. The results are for 600 instances of size $n = 20$ for each Δ and $k = 1$. The solid lines were obtained using least-squares fit. Pearson's correlation coefficient r between the two quantities is shown for each plot.

Figure 6.3 shows the mean decay of the local optima proportion against n. The results for $n = 30, 50$ are the SRS estimates obtained with a sample size $s = 10^5$. Again, and as with the uncorrelated problem types in the 0-1KP, the SRS with this sample size greatly overestimates the real proportion in the *H1+2* landscape. This was evident by the negative lower bound of the 95% CI_{AC} of the obtained estimates. Therefore, we did not include these estimates in figure 6.3. We also did not fit the decay of the proportions with the form an^{-b}, since we are only left with four close data points. The proportion of the local optima appears to decay faster in the *H1+2* landscape compared to the *H1* across all the CV intervals in all values of Δ. In general, and as in the previous two problems, the largest decay happens in the landscape of *H1+2*. Note that the decay in the *H1+2* landscape of this problem is similar across the CV intervals. The smallest decay occurs in the *H1* landscape of the interval $(0, 0.3]$. We are unable to comment on the growth of the number of local optima in the *H1+2* landscapes, however, their growth in the *H1* landscapes seems to be exponential with n.



(a) Δ =0.1

(b) Δ =0.25

(c) Δ =0.5

(d) Δ =0.75

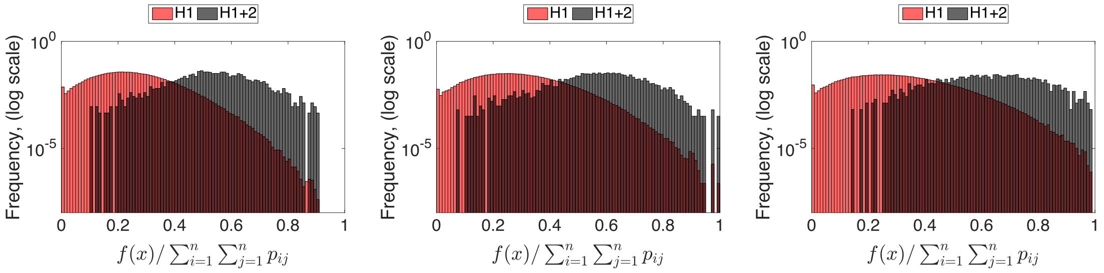
(e) Δ =0.95

(f) Δ =1

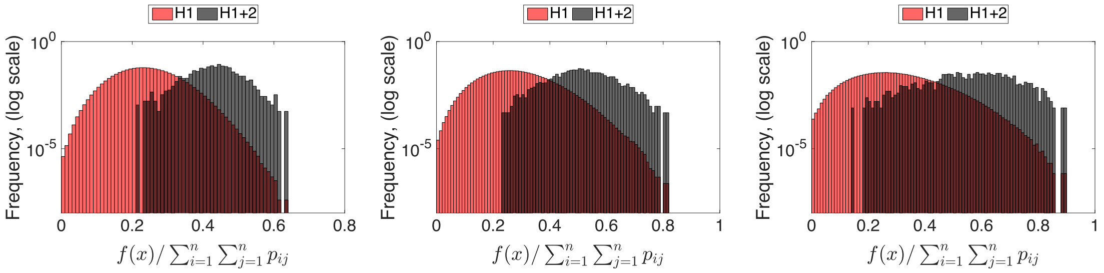
Figure 6.3: The decay of the optima proportion as the problem size n grows ($k = 1$). The results for each Δ are averaged over 600 instances for each $n = 14, 16, 18, 20$ and over 500 instances for $n = 30, 50$. The number of strict optima is estimated for $n = 30, 50$ using SRS with a sample size $s = 10^5$. The solid lines were obtained using least-squares fit. Note that the proportion of the optima in the *H1+2* landscape is always lower and decays faster in comparison to the *H1* landscape.

6.3.3 Quality of optima and plateaux

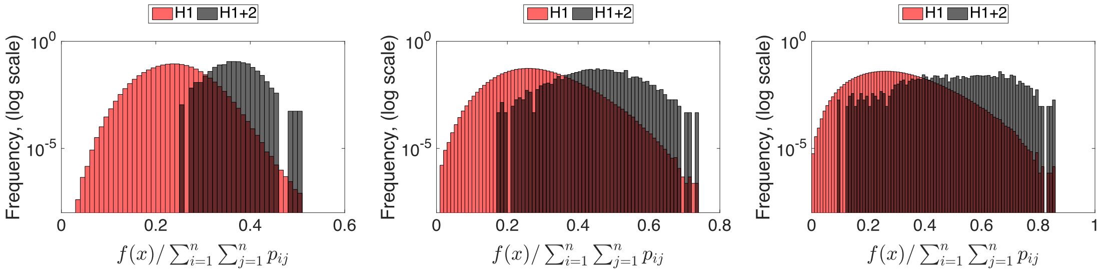
We examine here how the difference in the optima quality between the two landscape changes across the CV values. Obviously, the quality of the optima in the *H1+2* landscape is at least equal to or better than that in the *H1* as every optimum in the *H1+2* landscape is also an optimum in the *H1* landscape. As in the previous chapter, we want to obtain a measure of quality that is independent of the problem instance and that does not require the knowledge of the optimal solution. Thus, we measure the quality of an optimum x in a given instance as $f(x)/\sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}$. Figure 6.5 shows the quality of optima across the three CV intervals for every Δ. In general, and across all values of Δ, the quality of optima in the *H1+2* landscape is better than that in the *H1*, and this difference in the quality does not seem to change much across the CV values. This is again similar to the uncorrelated problem types in the 0-1KP.



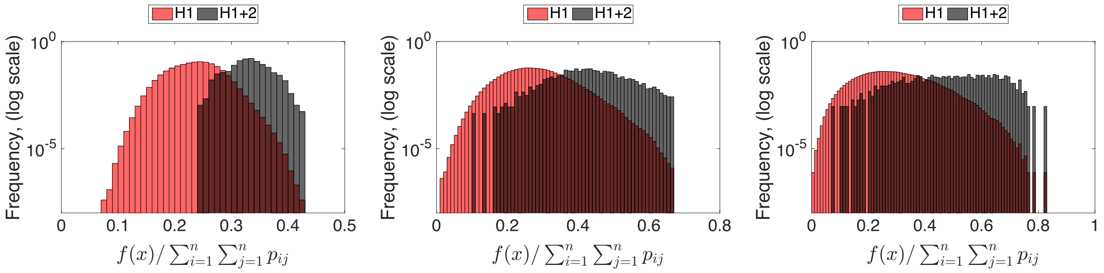
Δ =0.1



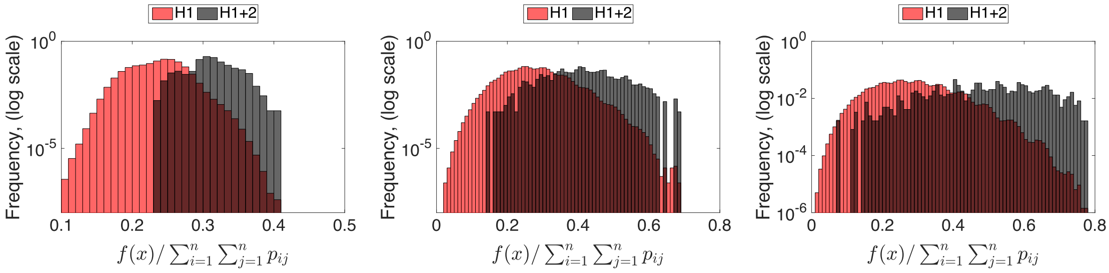
Δ =0.25



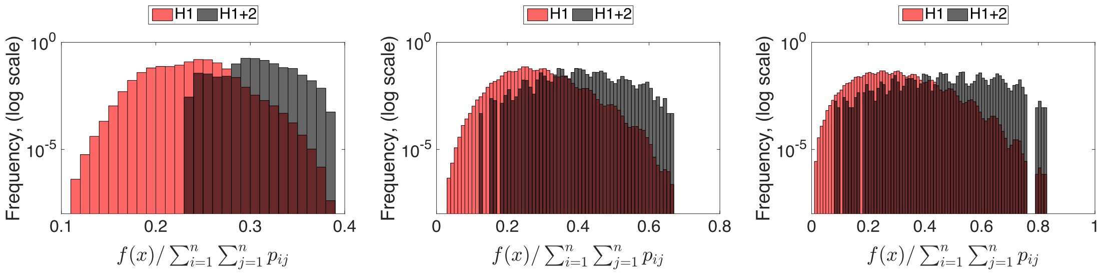
Δ =0.5



Δ =0.75



Δ =0.95

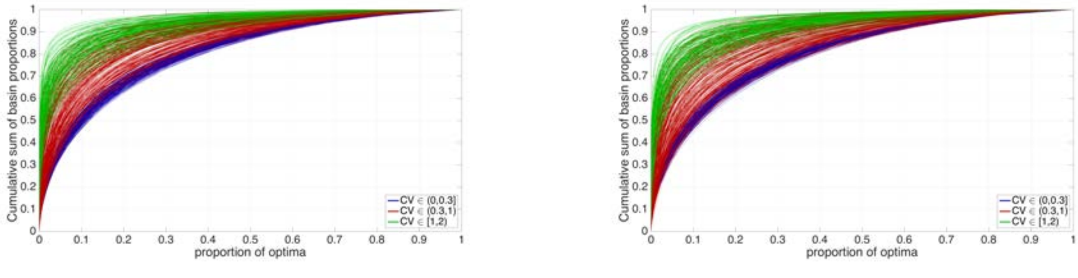


Δ =1

Figure 6.5: The quality of optima and plateaus in the *H1* and *H1+2* landscapes across the different values of CV: $0 < CV \leq 0.3$ (left), $0.3 < CV < 1$ (middle), $1 \leq CV < 2$ (right). The x-axis shows the fitness value divided by $\sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}$. The data includes all optima and plateaux found in 600 instances for each Δ of problem size $n = 20$ and $k = 1$.

6.4 Basins of Attraction

As in the previous chapters, in this section we study the following features of the attraction basins: the basin size, shape and the correlation between the size and the optimum fitness. We try to examine most of these properties across the different problem parameters and compare the results of the two landscapes. We exhaustively calculate the basin sizes, thus we were limited to studying small problem sizes $n = 20$ only.



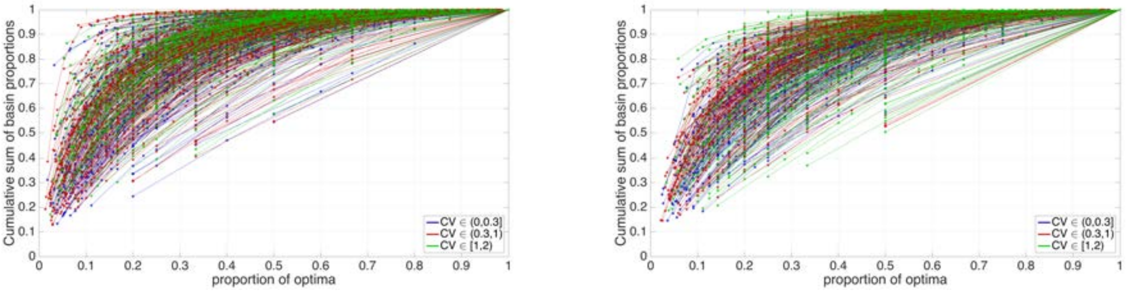
(a) $\Delta = 0.1$

(b) $\Delta = 1$

Figure 6.6: To examine how quickly the largest basins cover the search space in the *H1* landscape, we plot the cumulative sum of the basin proportions starting from the largest to the smallest (i.e. we plot the cumulative sum of the basin proportions after sorting them in descending order) against the proportion of the optima. Each line shows the results of a single instance. The results are for 600 instances of $n = 20$ and $k = 1$. These results are similar to results of the uncorrelated 01KP instances, in that only a very small percentage of the optima covers most of the search space.

6.4.1 Basin size

Figures C.2 to C.6 in the appendix give an overview of the relation between the basin size and fitness in all the instances we studied for each Δ. They show the general right skewness of the distribution of the basin sizes, in the *H1* landscape in particular (i.e. most of the basin sizes are small with very few large ones). The figures, also, show how the basin sizes in the *H1* landscape increases with the CV until their sizes become similar to those in the *H1+2* landscape.



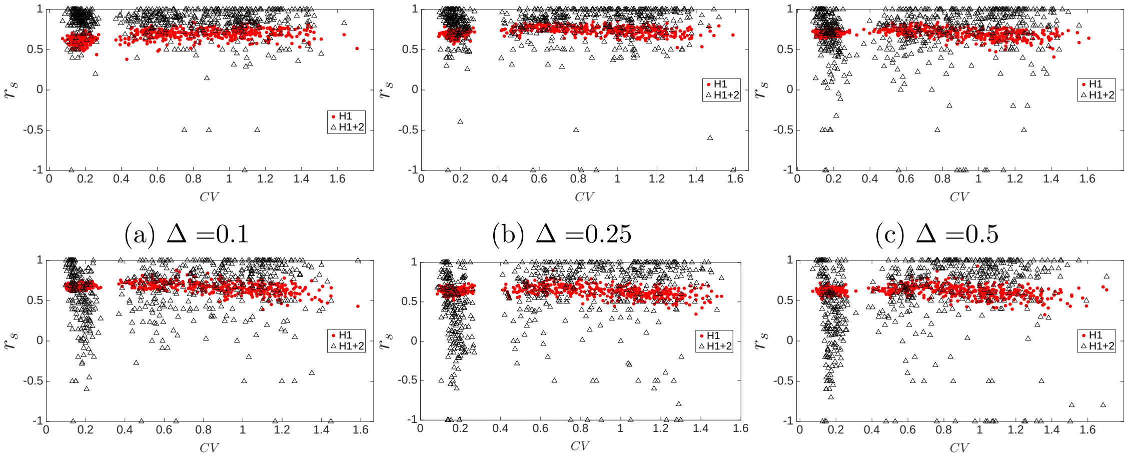
(a) $\Delta = 0.1$

(b) $\Delta = 1$

Figure 6.7: To examine how quickly the largest basins cover the search space in the *H1+2* landscape, we plot the cumulative sum of the basin proportions starting from the largest to smallest (i.e. we plot the cumulative sum of the basin proportions after sorting them in descending order) against the proportion of the optima. Each line shows the results of a single instance. The results are for 600 instances of $n = 20$ and $k = 1$. As with the *H1* landscape, these results are similar to the results of the *H1+2* landscape of uncorrelated 01KP instances.

Figure 6.6 shows the cumulative sum of the basin proportions, after being sorted in descending order, against the percentage of the basins in the *H1* landscape for $\Delta = 0.1, 1$. In general, this was found to be similar across all values of Δ for both landscapes. The figures show that the largest basins in this problem quickly covers large part of the search space. In instances from the large CV interval $[1, 2)$, large portion of the search space gets covered by very few basins, where we can see that only 10% of the basins cover between 70% to 90% of the search space. The same percentage of the basins cover around 50% to 70% of the search space in instances with $CV \in (0.3, 1)$, and cover around 50% of the search space in instances with $CV \in (0, 0.3]$. Figure 6.7 shows the same results for the *H1+2* landscape. Note that we now show each data point as there are far fewer number of optima in this landscape. In general, we still see that the largest basins cover a large part of the search space. In most cases, around 80% of the search space gets covered by half of the basins.

6.4.2 Basin size and fitness



(a) Δ =0.1

(b) Δ =0.25

(c) Δ =0.5

(d) Δ =0.75

(e) Δ =0.95

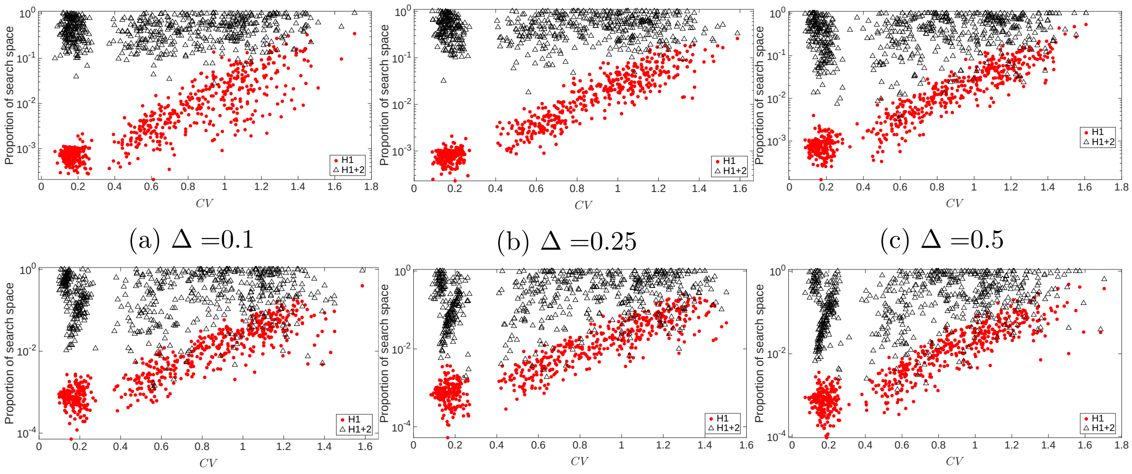
(f) Δ =1

Figure 6.8: Spearman's rank correlation coefficient between basin size and fitness versus CV. The results are for 600 instances of size $n = 20$ and $k = 1$ for each Δ. Notice how in the *H1+2* landscape more negative correlations occur as the value of Δ increases.

Similar to the uncorrelated 0-1KP problem types, the correlation between the attraction basin size and the optimum fitness in both landscape is very strong and positive, indicating that indeed in this problem the fitter optima tend to have larger basins (see figure 6.8). This does not seem to change much across the CV values for all the values of Δ. However, as Δ gets larger the correlation in the *H1+2* landscapes of some of the instances seems to get weaker and sometimes even very strong negative (this is more noticeable in the small CV interval $(0, 0.3]$). This indicates that, in such instances, fitter optima have smaller basins.Usually this feature means that these landscapes are more difficult to search particularly for local search, as the fitter optima has less probability of being found with a hill climber.

6.4.3 Global basin

We plot the total sum of all the global basin proportions found in instances of $n = 20$ against the CV for all the values of Δ in figure 6.9. In general, the probability of finding the optimal solution is always higher in the *H1+2* landscape than that in the *H1*. Like the previously studied features, the probability does not seem to change much across the CV values in the *H1+2* landscape, unlike the *H1* landscape, where the probability of finding the global increases with the CV across all values of Δ. The probability of finding the global in the *H1+2* landscape seems to decrease slightly as Δ increases. This reflects the results of the correlation between the basin size and fitness that we have seen in the previous subsection. The correlation was found to be strongly negative in some of these instances, which explains the decrease in the global basin size in these cases.



(a) Δ =0.1

(b) Δ =0.25

(c) Δ =0.5

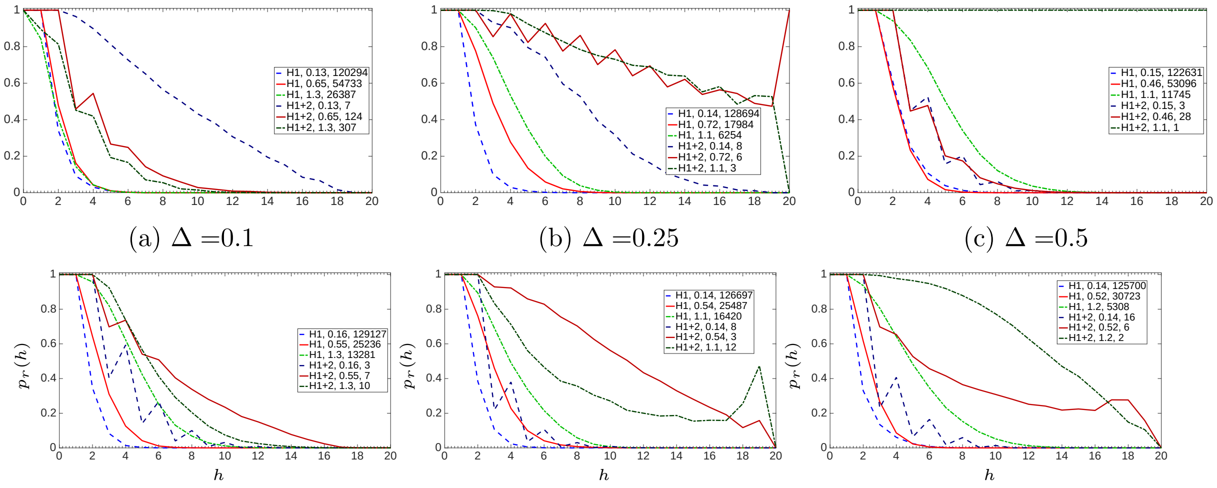
(d) Δ =0.75

(e) Δ =0.95

(f) Δ =1

Figure 6.9: The proportion (in log scale) of the basin size of all the global optima found in an instance for each landscape against the CV. The results are for 600 instances of size $n = 20$ and $k = 1$ for each Δ. Notice how the probability of finding the global optimum increases with the CV in the *H1* case.

In an attempt to study the shape of the global basin, we plot in figure 6.10 the proportion of the configurations that are part of its basin in every Hamming sphere of radius h around it. The proportions were estimated as described in subsection 3.4. The results are shown for three instances of size $n = 20$. From the figure we can see that in the *H1* landscape the configurations in the global basin are concentrated in the immediate Hamming spheres around it. In the *H1+2* landscape, the number of optima is much smaller, and can actually be the only optimum in the landscape as in the instance with $CV = 1.1$ and $\Delta = 0.5$. This explains why the probability of returning to the global continues until the last sphere sometimes. Again we continue to see the oscillating behaviour of the probability of return over the spheres in the *H1+2* of some of the instances. Also, we can see that the return to the global in the *H1+2* landscape of instances with small CV values approaches zero faster in larger values of Δ, where the return probability is almost zero in configurations different than the global in half or more of the dimensions. This agrees with the results obtained previously about the proportion of the global basins and the correlation between the fitness and the basin size.



(a) Δ =0.1

(b) Δ =0.25

(c) Δ =0.5

(d) Δ =0.75

(e) Δ =0.95

(f) Δ =1

Figure 6.10: Return probability $p_r(h)$ to the global optimum starting from a Hamming sphere of radius h (y-axis) versus h (x-axis). The results are for 3 instances of size $n = 20$ and $k = 1$ for each Δ. Each legend entry shows respectively: the landscape type, the instance CV value, and the number of optima in that landscape of that instance. Notice how the probability of return approaches zero faster in the *H1* case compared to *H1+2*.

6.5 Local Search

We study in this section the performance of local search, namely steepest ascent with random restart algorithm, using the two neighbourhood operators. We carry out the analysis of the algorithm performance from the perspective of the studied landscape features in the earlier sections.

6.5.1 Cost of finding the global

Figures 6.11 and 6.12 compare the performances of the two operators in terms of the number of fitness evaluations used to find the global when $k = 0.4$ and $k = 1$ respectively. As with the previous properties, this does not seem to change with the values of k. Similar to the trend observed in the previous problems, the *H1* operator was found to perform better in the $[1, 2)$ *CV* interval, and the *H1+2* operator was found to perform better in the rest of the *CV* intervals. However, there is no clear winner in the *CV* interval $[1, 2)$ of instances with very sparse profit matrix.

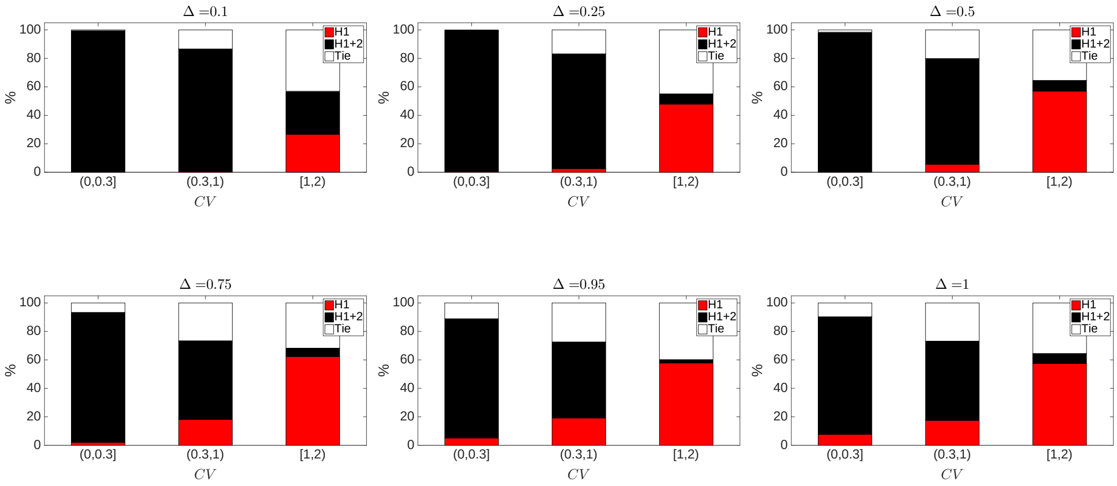


Figure 6.11: Number of fitness evaluations used to find the global optimum averaged over 30 runs. The results show the percentage of instances where each operators performed significantly better and the percentage where no significance difference was found. Significance determined using Wilcoxon rank-sum ($p-$value ≤ 0.05). The results are for 600 instances for each Δ of size $n = 20$ and $k = 0.4$.

In general, the cost of finding the global optimum seems to grow exponentially with n as shown in figure 6.13. We can see that for $\Delta \geq 0.5$, the *H1* operator has the lowest mean number of used fitness evaluations to find the global in instances with $CV \in [1, 2)$. The cost of the *H1+2* operator increases slightly as Δ increases. This can be attributed to the results of the global basin proportion of the search space and how it was found to decrease slightly as Δ increases.

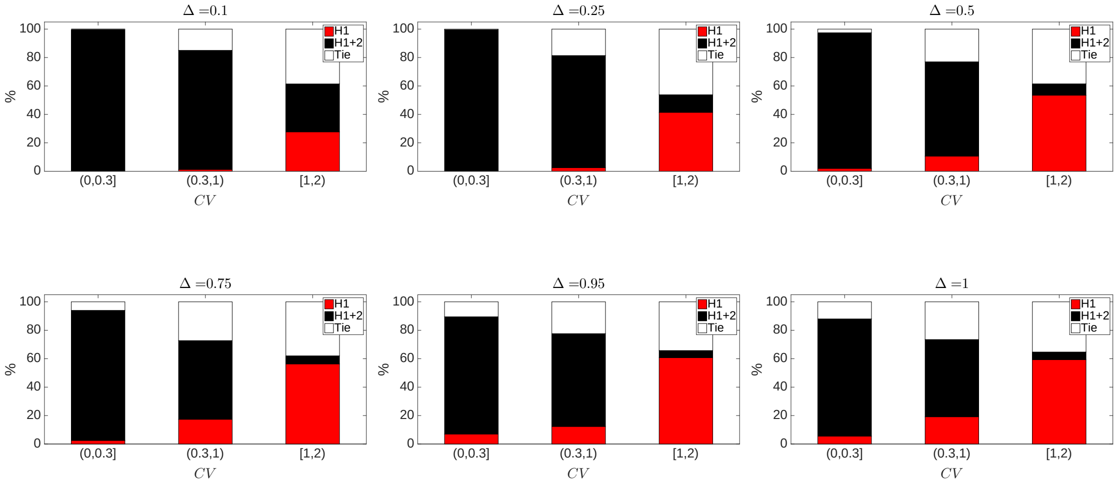
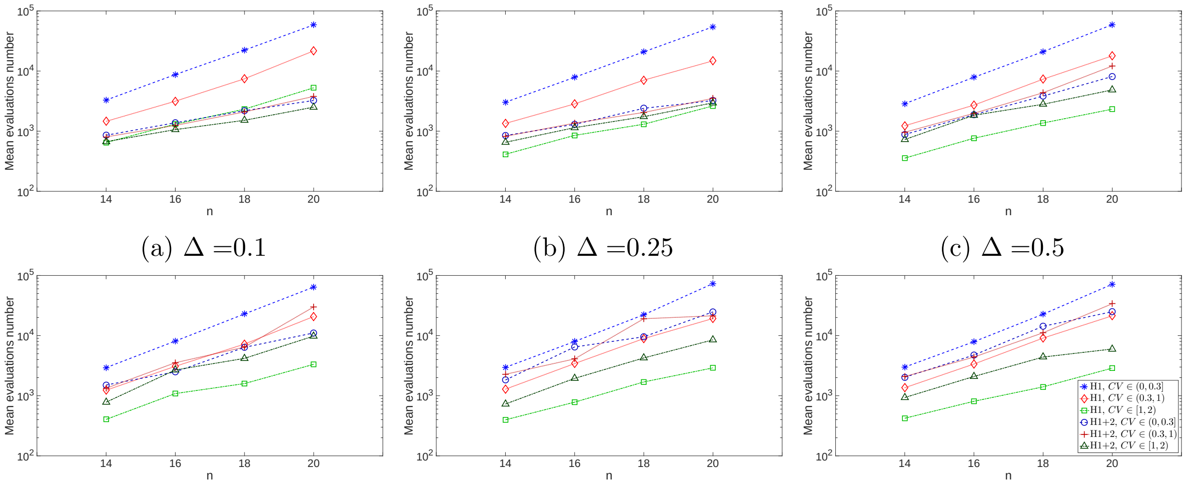


Figure 6.12: Number of fitness evaluations used to find the global optimum averaged over 30 runs. The results show the percentage of instances where each operators performed significantly better and the percentage where no significance difference was found. Significance determined using Wilcoxon rank-sum ($p-$value ≤ 0.05). The results are for 600 instances for each Δ of size $n = 20$ and $k = 1$.



(a) Δ =0.1

(b) Δ =0.25

(c) Δ =0.5

(d) Δ =0.75

(e) Δ =0.95

(f) Δ =1

Figure 6.13: Number of fitness evaluations used to find the global (in log scale). The results are averaged over 30 runs of steepest ascent. The results for each n are for 600 instances with $k = 1$. Notice the large difference in the average number of evaluations (almost one order of magnitude) between instances with $CV < 0.3$ and $CV \geq 1$ when *H1* operator is used.

6.5.2 Quality of optima obtained with fixed budget search

The previous results of the growth of the cost to find the global give an indication that finding the global seems to be irrelevant as the problem size increases. Therefore, we look here at the quality of optima obtained by a fixed budget of fitness evaluations. Figures 6.14 and 6.15 compare the performances of the two operators in terms of the quality of the obtained optima for $n = 30, 50$ respectively. Now we can see that clearly the *H1+2* operator performs better in all the CV intervals and across all values of Δ. This can be explained by the small number of optima in this landscape and the faster decay of their proportions. Also, as we have seen before the quality of the optima in the *H1+2* is always better than that in the *H1* in all the CV intervals. This result is again similar to the uncorrelated problem types of the 0-1KP. Note that as in the previous two chapters, these results are specific to the budget we selected, whether the same trends will continue to occur with other budget values remains an open question.

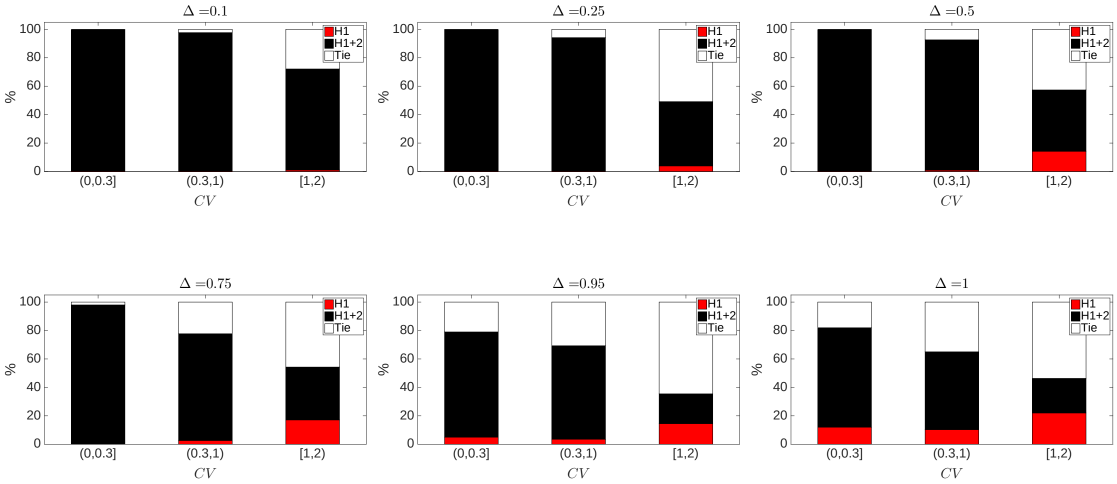


Figure 6.14: The quality of the solution found averaged over 30 runs of local search with fixed budget of 10^5 fitness evaluations. The results are for 500 instances for each Δ of size $n = 30$ and $k = 1$. The results show the percentage of instances where each operators performed significantly better and the percentage where no significance difference was found (Tie). Significance determined using Wilcoxon rank-sum ($p-$value ≤ 0.05).

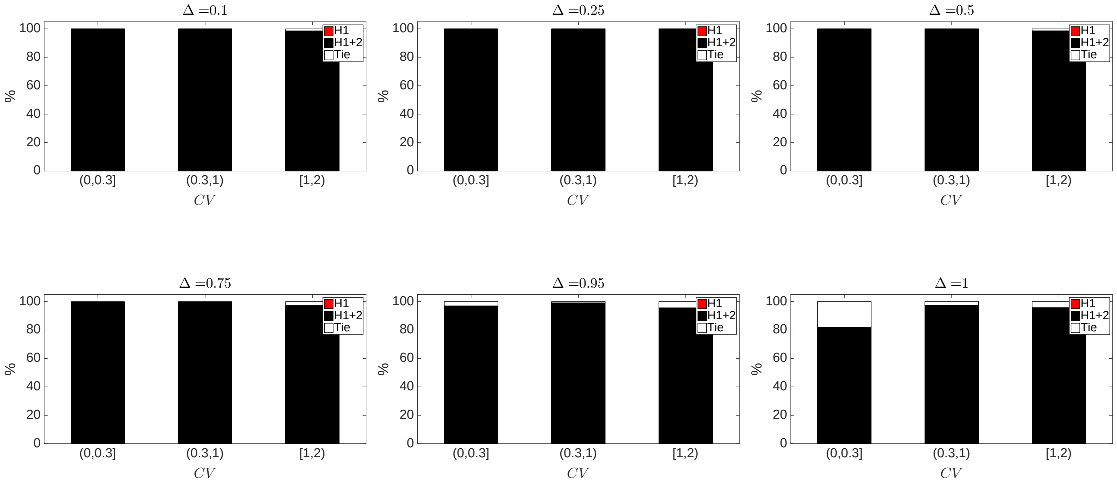
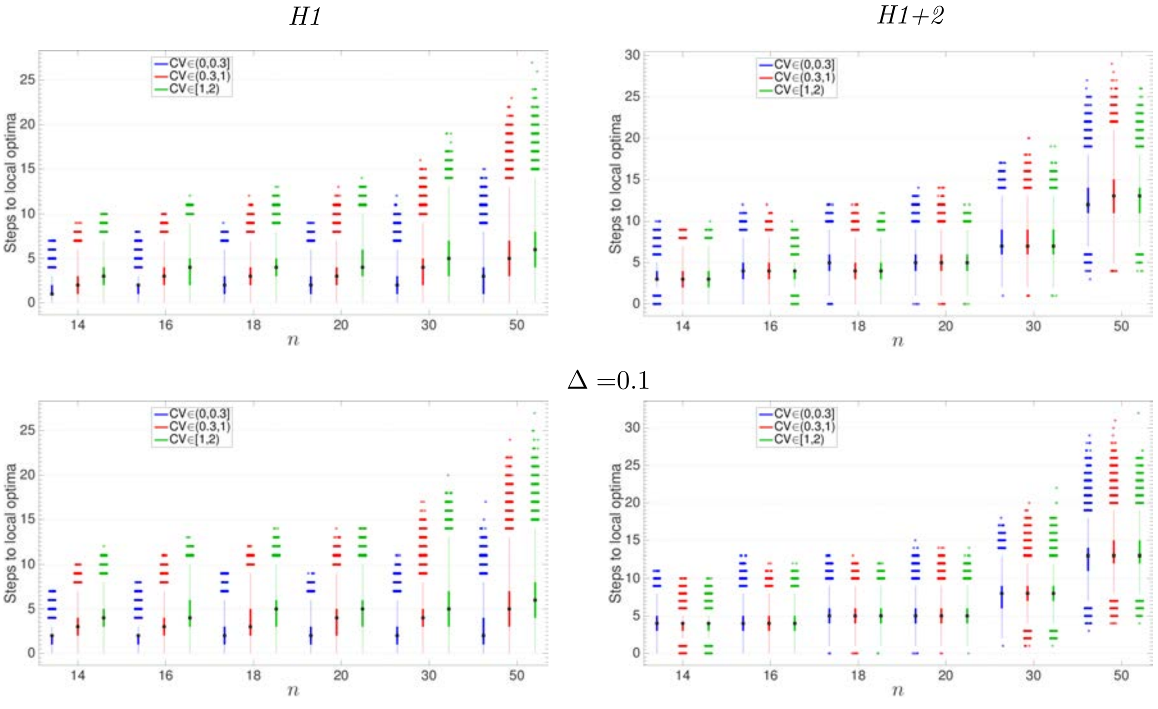


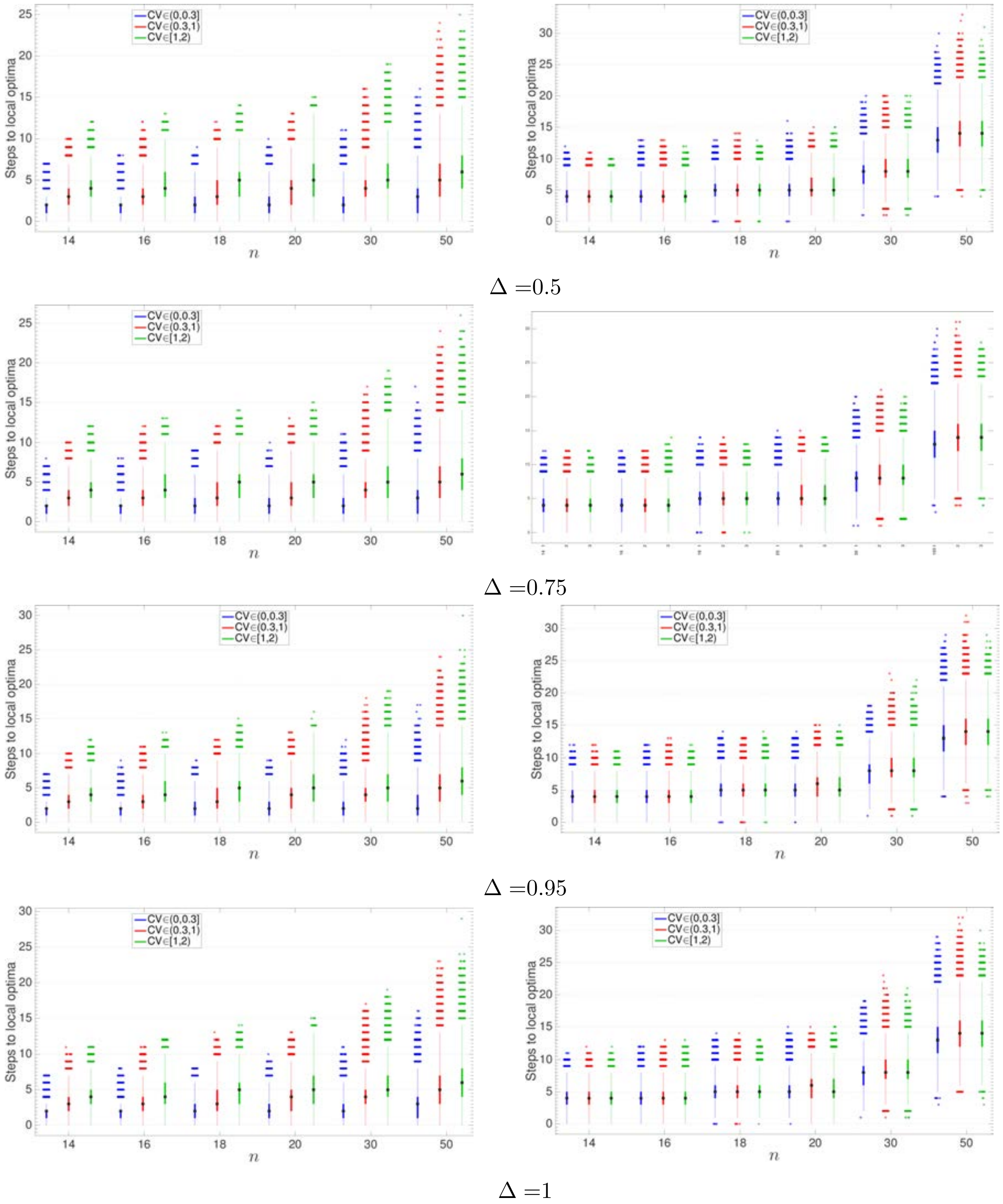
Figure 6.15: The quality of the solution found averaged over 30 runs of local search with fixed budget of 10^5 fitness evaluations. The results are for 500 instances for each Δ of size $n = 50$ and $k = 1$. The results show the percentage of instances where each operators performed significantly better and the percentage where no significance difference was found (Tie). Significance determined using Wilcoxon rank-sum ($p-$value ≤ 0.05).

6.5.3 Time to local optima



Δ =0.1

Δ =0.25



Δ =0.5

Δ =0.75

Δ =0.95

Δ =1

Figure 6.17: Number of steps starting from a random configuration until an optimum is reached (*H1* on the left and *H1+2* on the right). The results for each Δ are for 1000 steepest ascents per instance and 600 instances for each $n = 14, 16, 18, 20$ and 500 instances for $n = 30, 50$ ($k = 1$).

The time it takes steepest ascent, starting from a random configuration until a local optimum is found, is shown as n grows in figure 6.17. Again, this was found to be similar to the uncorrelated instances of the 0-1KP. In the *H1* landscape, the number of steps was found to be relatively small and has a small growth rate. In the *H1+2* landscape, the number of steps is still relatively small but it has a faster growth rate. This supports our observation that the decay of the number of local optima in the *H1+2* landscape of these instances appears to be fast. Note that the number of steps taken in the *H1* landscape was found to be always equal to the Hamming distance between the initial random configuration and the found local optimum. In the *H1+2* landscape this was found to be almost always smaller or equal to the Hamming distance between the initial random configuration and the found local optimum. However, and like in the two previously studied problems, the number of steps was found to be larger than the Hamming distance in very few cases. But unlike the previous problems, the difference between the steps and the Hamming distances was found to be slightly larger here, ranging from one to five (this is for $n = 22$).

6.6 Summary

In this chapter, we empirically studied various properties of two fitness landscapes of moderately constrained random instances of the quadratic binary knapsack problem with uncorrelated weights and profits. We focused on how these properties change with these problem parameters: k, Δ and the CV. The following is a summary of the main results:

- The logarithmic penalty function was found to create a strict local optimum in the infeasible region. The linear penalty function was also found to create a strict local optimum in the infeasible region of some of the instances with highly dense profit matrix.

- Configurations of type IPLAT were only found in instances with very sparse profit matrix.

- In both landscapes, the number of plateaux decreases as Δ increases until they almost disappear in instances with very dense profit matrix.

- As with the uncorrelated knapsack problem types, in general the properties were found not to be affected by the different values of k.

- The number of local optima in the *H1+2* was found to be very low, as low as only one or two optima sometimes.

- There is a very strong and negative correlation between the CV and the number of local optima in the *H1* landscape.

- The same formula used to estimate the average number of local optima in the *H1* landscape of the previous problems seems to be a good approximate here as well.

- The number of local optima seems to grow exponentially in the *H1* landscape.

- The quality of optima is always better in the *H1+2* landscape, and the difference in quality between the two landscape does not seem to be affected by the CV.

- The distribution of the basin sizes was found to be skewed with many small basins and only few large ones, especially in the *H1* landscape.

- In general, the correlation between the basin size and fitness was found to be very strong and positive in both landscapes, which indicates that fitter optima tend to have larger basins.

- The number of steps until an optimum is found, starting from a random configuration, grows very slowly with n in the *H1* landscape, but seems to have a faster growth rate in the landscape induced by the *H1+2* operator.

- The *H1* operator was found to have a better performance in instances with $CV \geq 1$ only when searching for the global solution only (this was examined in small problem sizes only).

- The *H1+2* operator performs better in instances with $CV < 1$ when searching for the global solution. But it was found to always perform better, in terms of the quality of the obtained optima with fixed budget search, across all the CV values.

- In general the landscape properties and the performance of local search in the uncorrelated 0-1QKP was found to be similar to that of the uncorrelated 0-1KP.

CHAPTER 7

CONCLUSION AND FUTURE WORK

In this thesis, we empirically studied a number of landscape features of three NP-hard problems: the number partitioning problem (NPP), the binary knapsack problem (0-1KP), and the quadratic binary knapsack problem (0-1QKP). All of the three problems are similar in nature, in the sense that all of them fall into a class of NP-hard binary packing problems related to the 0-1 knapsack problem. The subset sum problem, which is a special case of the 0-1KP and one of its 11 problem types studied in this thesis, is a generalisation of the NPP. The 0-1QKP is a variant of the 0-1KP where the profit of an item depends also on the the other selected items. The 0-1KP and the 0-1QKP are constrained optimisation problems and their search space is partitioned into a feasible and an infeasible region. The NPP and the 0-1KP are both NP-hard in the weak sense but the 0-1QKP is NP-hard in the strong sense [35]. Only the NPP and the subset sum problems have an identified phase transition determined by the control parameter k, which corresponds to the number of the bits required to encode the weights divided by the size of the problem [11, 98]. Most of the existing studies of these problems only consider instances where the weights are drawn at random from a uniform distribution. We studied instances generated by drawing the weights at random from various distributions. In all of the three problems, we found that the number of strict local optima and the cost of local search to find the global, vary greatly (most noticeably in the *H1* landscape) between some of the distributions. We proposed and demonstrated that the use of the CV of the weights, a single parameter that is easy to calculate and does not require the knowledge of the underlying distribution of the weights, captures most of this variability.

We studied the fitness landscapes induced by two neighbourhood operators, the *H1* operator with a neighbourhood that grows linearly with the problem size and the larger neighbourhood operator *H1+2*, which has a neighbourhood that grows quadratically with the problem size. We also studied the landscape structure induced by different penalty functions in the 0-1KP and the 0-1QKP. We studied a set of properties that we believe to be representative of the problems difficulties (with respect to local search) and to give an insight into structure of the problem landscape. The set includes: types of search position, number of local and global optima and plateaux, quality of optima and plateaux, basin size and its correlation with fitness, time to local optima, cost of finding the global solution, and quality of optima obtained with a fixed budget search. Our work focuses on studying how these properties vary with different values of problem parameters. The parameters are, for the NPP: k and the CV; for the 0-1KP: k, λ, CV, and the correlation between w_i and p_i; for the 0-1QKP: k, Δ, and the CV. We also studied the scaling behaviour of some of these properties as the problem size grows. Our approach was mainly exhaustive and thus we were limited to studying problem sizes with enumerable search spaces only. We did sample some of these properties from medium sized instances. One of these properties is the number of local optima. We evaluated the performance of estimating the number of local optima by estimating their proportion using simple random sampling (SRS) and discussed the choice of confidence intervals in chapter 3. Of course, as a consequence of studying mainly small instances, we cannot know with certainty if the observed trends will continue to show as n gets larger, but we have shown that they still occur in medium problem sizes ($n = 100$).

In the following we outline and discuss some of the main observations and trends found:

- One observation that may seem counter-intuitive is that the landscape induced by the larger neighbourhood has sometimes more plateaux and usually has larger plateaux sizes. In fact, in the 0-1KP, plateaux were only found in this landscape. This perhaps can be understood when knowing the fact that no configuration of type IPLAT was found (apart from instances with a very sparse profit matrix in the 0-1QKP), and that all the found plateaux were composed of connected non-strict optima. Plateaux in the NPP and the 0-1KP were only found in the easy region ($k < 1$). In the 0-1QKP, plateaux were mainly found in instances with a sparse profit matrix. This fact also goes to explain the previous observation as in such cases more strict optima share the same fitness value in the *H1* landscape and they get connected forming plateaux with the application of the *H1+2* neighbourhood operator.

- For the NPP and subset sum, the only two properties, in both landscapes, that were found to change when the problem crosses the phase transition are the number of global optima (and consequently the probability of finding the global) and the number of plateaux (note plateaux in subset sum were found only in the *H1+2* landscape). The rest of the properties remained oblivious to the phase transition. The performance of local search algorithms was found to be affected by the phase transition in both NPP and subset sum, where there was a considerable increase in the cost of locating the global solution in instances with $k > k_c$.

- The different values of k did not have any effect on the landscape of the 0-1QKP, apart from the disappearance of SLOPE configurations in the *H1+2* landscape of the infeasible region when $k = 1$. Some properties of a few problem types in the 0-1KP seemed to be affected by the values of k. One property that changed with the values of k across almost all of the problem types is the existence of plateaux when $k = 0.4$.

- In all the problems investigated, there is a very strong and negative correlation between the CV and the number of local optima in the *H1* landscape. We continued to see this trend in all the problem sizes we studied. The average number of local optima in this landscape, in all the problems, seemed to be well approximated by the formula $\frac{a}{e^{bCV}}2^n$ (where a and b for each problem depend on n). We believe that this phenomenon is particular to the binary packing problems related to the 0-1 knapsack problem. From some preliminary experiments on the weighted not all equal 3-SAT problem and the generalised assignment problems, there was no such strong correlation between the CV and the number of local optima in the *H1* landscape of these problems.

- In all the problems, the number of local optima in the *H1+2* landscape was found to be much smaller than that in the *H1* landscape. The largest difference between them occurs in the small CV interval $(0, 0.3]$. The number of local optima seems to grow exponentially with the problem size in both the landscapes, with a faster rate in the *H1* landscape. However, we are unable to comment on the growth behaviour of the the number of local optima in the *H1+2* landscape of the 0-1QKP, and the following types of the 0-1KP: uncorrelated, weakly correlated, uncorrelated spanner, weakly correlated spanner, strongly correlated spanner, and multiple strongly correlated. The local optima proportion in the *H1+2* landscape of these instances appears to decay faster than what the sample size we used for SRS can detect with accuracy. The faster growth in the time it takes steepest ascent, starting from a random configuration until a local optimum is found, in these instances seems to support this observation.

- The landscape properties and the performance of local search in the uncorrelated 0-1KP and the uncorrelated 0-1QKP were found to be similar to that of the uncorrelated 0-1KP. The number of local optima in their *H1+2* was found to be very low, as low as only one or two optima sometimes.

- The distribution of the basin sizes in both landscapes of all the problems was found to be skewed to the right, with many small basins and only few large ones.

- The CV of the weights has a potentially useful application in guiding the choice of the move operator of local search heuristics, particularly in the NPP and some of the problem types of the 0-1KP.

7.1 Future Work

The work presented in this thesis provides fingerprints of the studied problems landscapes. This should be of interest for anyone who wants to design or select an algorithm for these problems or similar problems. The work also revealed some of the mechanism behind the changes in the landscape of a problem when a different search operator is used, which again should be of interest for algorithm designers. A specific example of possible use of the presented findings in the process of selecting an algorithm or a component of an algorithm is that the number of optima in the *H1+2* landscape of the uncorrelated 0-1KP and the uncorrelated 0-1QKP is very small, as small as one or two sometimes. This finding raises the question if other strategies of local search, which do not exhaustively explore the whole neighbourhood, such as next ascent would perform better than steepest ascent, by using less number of fitness evaluations to find the global (through avoiding the cost of exploring the large neighbourhood). Another interesting finding that is worth further investigation is that the empirical results in this thesis show no difference between the landscape of NP-hard and NP-weak problems. This raises an important question if there is no difference between NP-hard and NP-weak from the point of view of search heuristics. Additional direction of future work is the identification of other problem that have similar landscape features as the ones studied in this thesis. Finally, another obvious direction for future work is to continue investigating the studied properties for larger problem sizes and examine which of the trends continue and which of them disappear.

APPENDIX A

NUMBER PARTITIONING PROBLEM SUPPLEMENTARY RESULTS

The following tables are used to support the observations/conclusions made in section 4.2.

CV	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
(0,0.3]	1.28e-01	9.54E-04	0.00E+00	8.63E-01	8.48E-03	0.00E+00	1.91E-06
(0.3,1)	1.94e-02	1.24E-03	0.00E+00	9.69E-01	1.08E-02	0.00E+00	1.91E-06
[1,2)	8.10e-03	1.67E-03	0.00E+00	9.75E-01	1.55E-02	0.00E+00	1.91E-06

(a) *H1* Landscape

CV	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
(0,0.3]	5.00e-04	1.99E-03	0.00E+00	3.74E-01	6.24E-01	0.00E+00	1.91E-06
(0.3,1)	1.22e-03	2.11E-03	0.00E+00	5.06E-01	4.91E-01	0.00E+00	1.91E-06
[1,2)	6.05e-04	2.87E-03	0.00E+00	3.28E-01	6.69E-01	0.00E+00	1.91E-06

(b) *H1+2* Landscape

Table A.1: Proportion of the search position types, averaged over 600 instances of NPP with $k = 0.4$ and $n = 20$.

CV	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
(0,0.3]	1.33e-01	3.05E-07	0.00E+00	8.67E-01	2.16E-06	0.00E+00	1.91E-06
(0.3,1)	1.72e-02	2.67E-07	0.00E+00	9.83E-01	2.84E-06	0.00E+00	1.91E-06
[1,2)	8.24e-03	3.89E-07	0.00E+00	9.92E-01	3.96E-06	0.00E+00	1.91E-06

(a) *H1* Landscape

CV	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
(0,0.3]	1.24e-03	3.05E-07	0.00E+00	9.99E-01	3.15E-05	0.00E+00	1.91E-06
(0.3,1)	2.01e-03	3.30E-07	0.00E+00	9.98E-01	3.23E-05	0.00E+00	1.91E-06
[1,2)	1.41e-03	6.07E-07	0.00E+00	9.99E-01	4.95E-05	0.00E+00	1.91E-06

(b) *H1+2* Landscape

Table A.2: Proportion of the search position types, averaged over 600 instances of NPP with $k = 1$ and $n = 20$.

CV	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
(0,0.3]	1.33e-01	1.91E-08	0.00E+00	8.67E-01	1.72E-07	0.00E+00	1.91E-06
(0.3,1)	1.83e-02	0.00E+00	0.00E+00	9.82E-01	1.51E-07	0.00E+00	1.91E-06
[1,2)	7.61e-03	3.31E-08	0.00E+00	9.92E-01	2.98E-07	0.00E+00	1.91E-06

(a) *H1* Landscape

CV	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
(0,0.3]	1.18e-03	2.86E-08	0.00E+00	9.99E-01	1.80E-06	0.00E+00	1.91E-06
(0.3,1)	2.11e-03	1.68E-08	0.00E+00	9.98E-01	1.68E-06	0.00E+00	1.91E-06
[1,2)	1.47e-03	1.10E-08	0.00E+00	9.99E-01	2.57E-06	0.00E+00	1.91E-06

(b) *H1+2* Landscape

Table A.3: Proportion of the search position types, averaged over 600 instances of NPP with $k = 1.2$ and $n = 20$.

The following figures are used to support the observations/conclusions made in subsection 4.3.3.

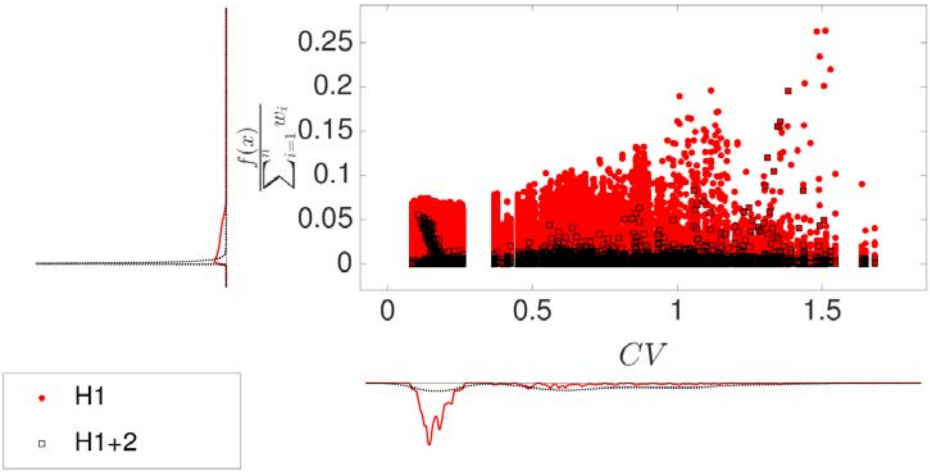
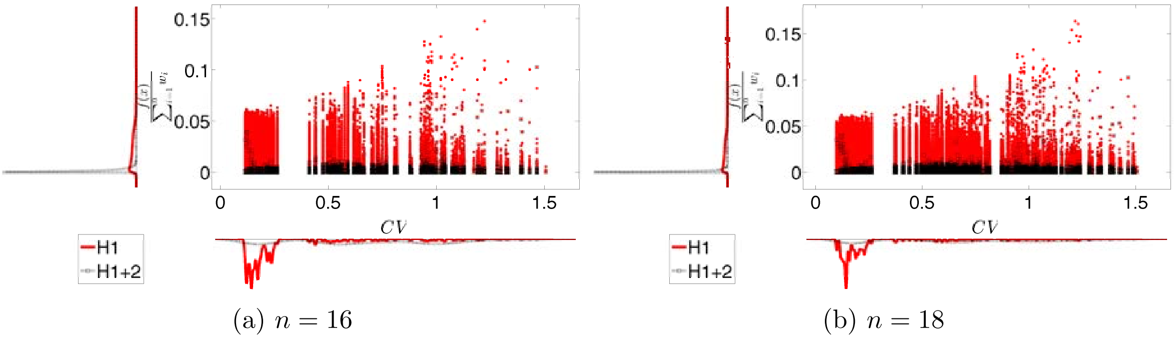


Figure A.1: The quality of optima in the *H1* and *H1+2* landscapes versus the CV value of the weights. The y-axis shows the fitness value divided by $\sum_{i=1}^{n} w_i$. The data includes both global and local optima, and each data point represents a single optimum. The results are for 600 instances of $n = 14$ and $k = 1$. Smooth histograms of the number of optima of both *H1* and *H1+2* landscapes are shown in the horizontal and vertical axes of the plot.



(a) $n = 16$

(b) $n = 18$

Figure A.2: The quality of optima in the *H1* and *H1+2* landscapes versus the CV value of the weights. The y-axis shows the fitness value divided by $\sum_{i=1}^{n} w_i$. The data includes both global and local optima, and each data point represents a single optimum. The results are for 126 instances per problem size and $k = 1$. Smooth histograms of the number of optima in both *H1* and *H1+2* landscapes are shown in the horizontal and vertical axes of the plot.

The following figures are used to support the observations/conclusions made in subsection 4.4.2.

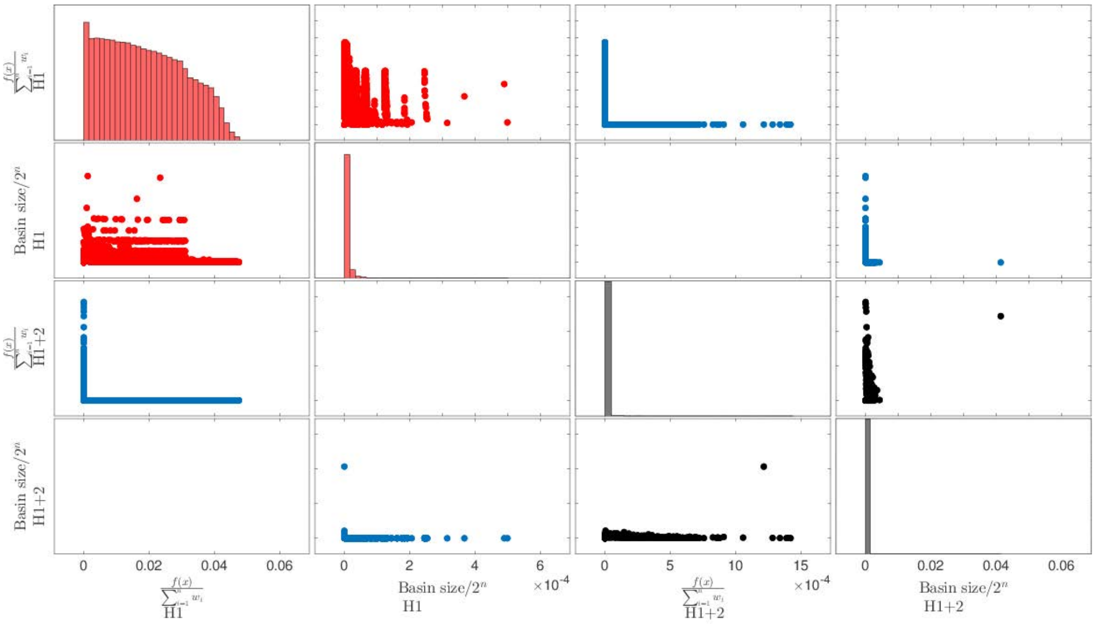


Figure A.3: Plot-matrix of the basin size proportion and optima fitness $(f(x)/\sum_{i=1}^{n} w_i)$ of both *H1* and *H1+2* landscapes of a single instance of size $n = 20$, $CV = 0.14$ and $k = 1$. Histograms of the basin size and the fitness for each landscape are shown along the diagonal.

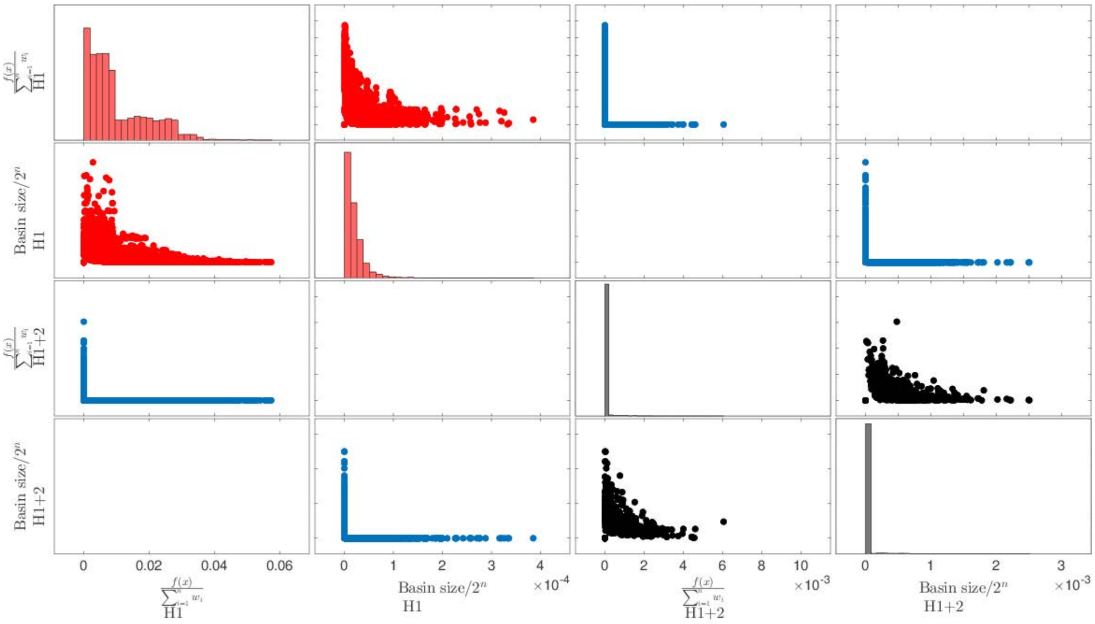


Figure A.4: Plot-matrix of the basin size proportion and optima fitness $(f(x)/\sum_{i=1}^{n} w_i)$ of both *H1* and *H1+2* landscapes of a single instance of size $n = 20$, $CV = 0.41$ and $k = 1$. Histograms of the basin size and the fitness for each landscape are shown along the diagonal.

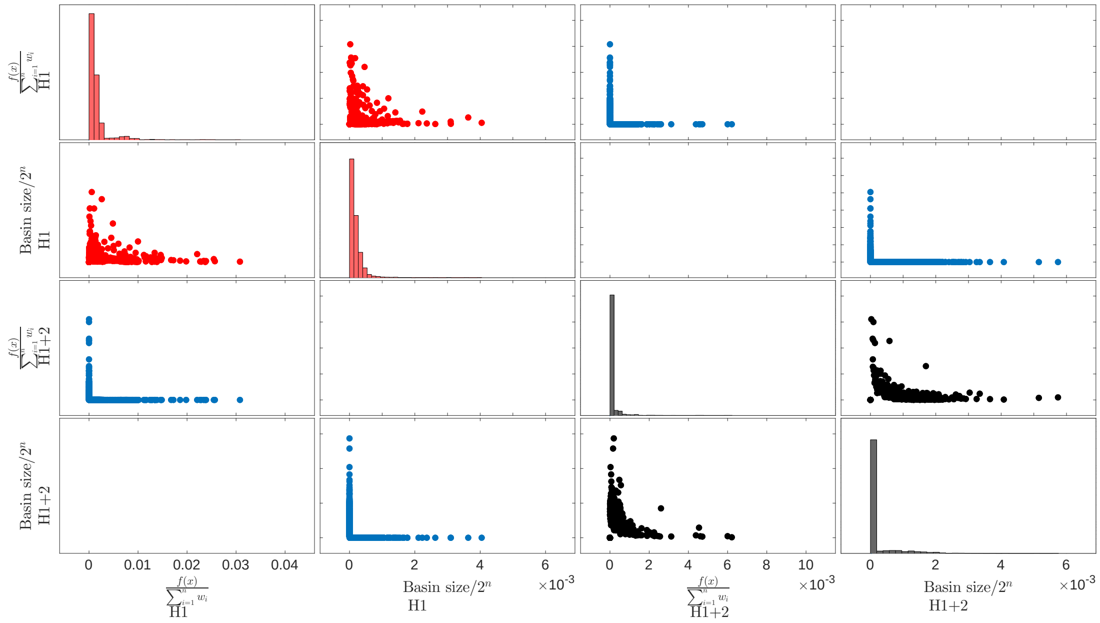
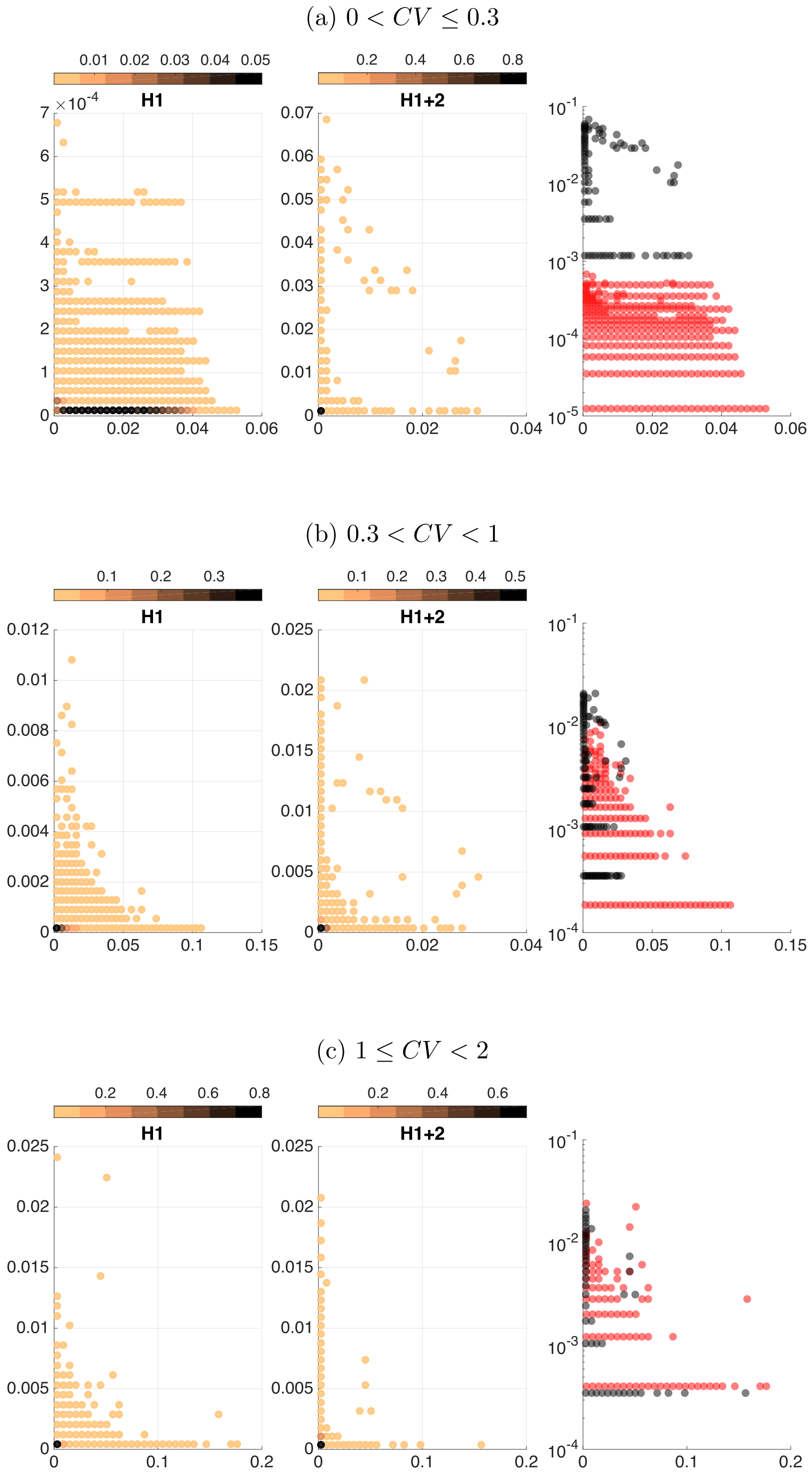


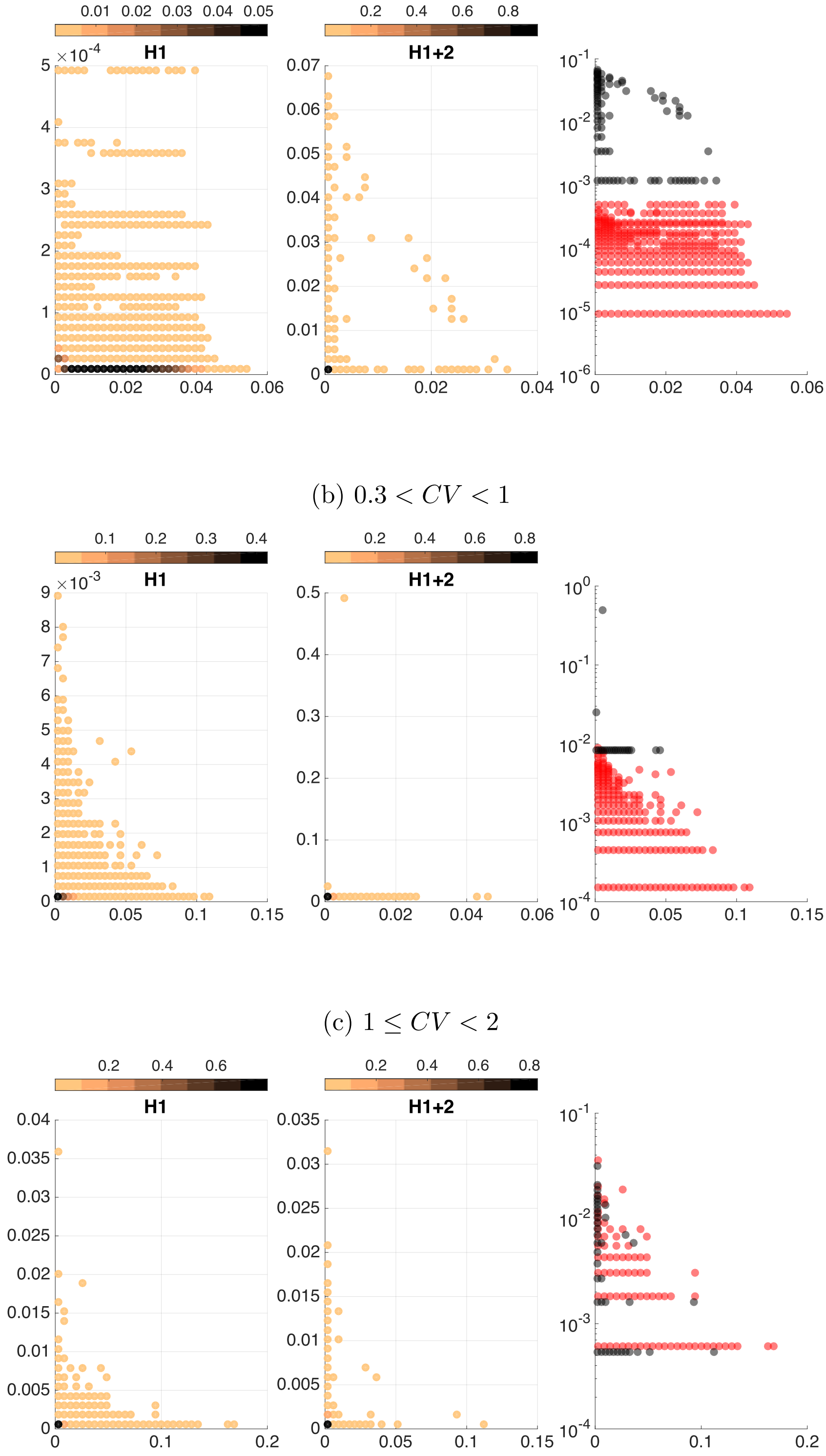
Figure A.5: Plot-matrix of the basin size proportion and optima fitness $(f(x)/\sum_{i=1}^{n} w_i)$ of both *H1* and *H1+2* landscapes of a single instance of size $n = 20$, $CV = 1.4$ and $k = 1$. Histograms of the basin size and the fitness for each landscape are shown along the diagonal.


(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

Figure A.6: Basin size/2^n (y-axis) against $f(x)/\sum_{i=1}^{n} w_i$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the strict optima and plateaux found in 600 instances of $n = 20$ and $k = 0.4$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.

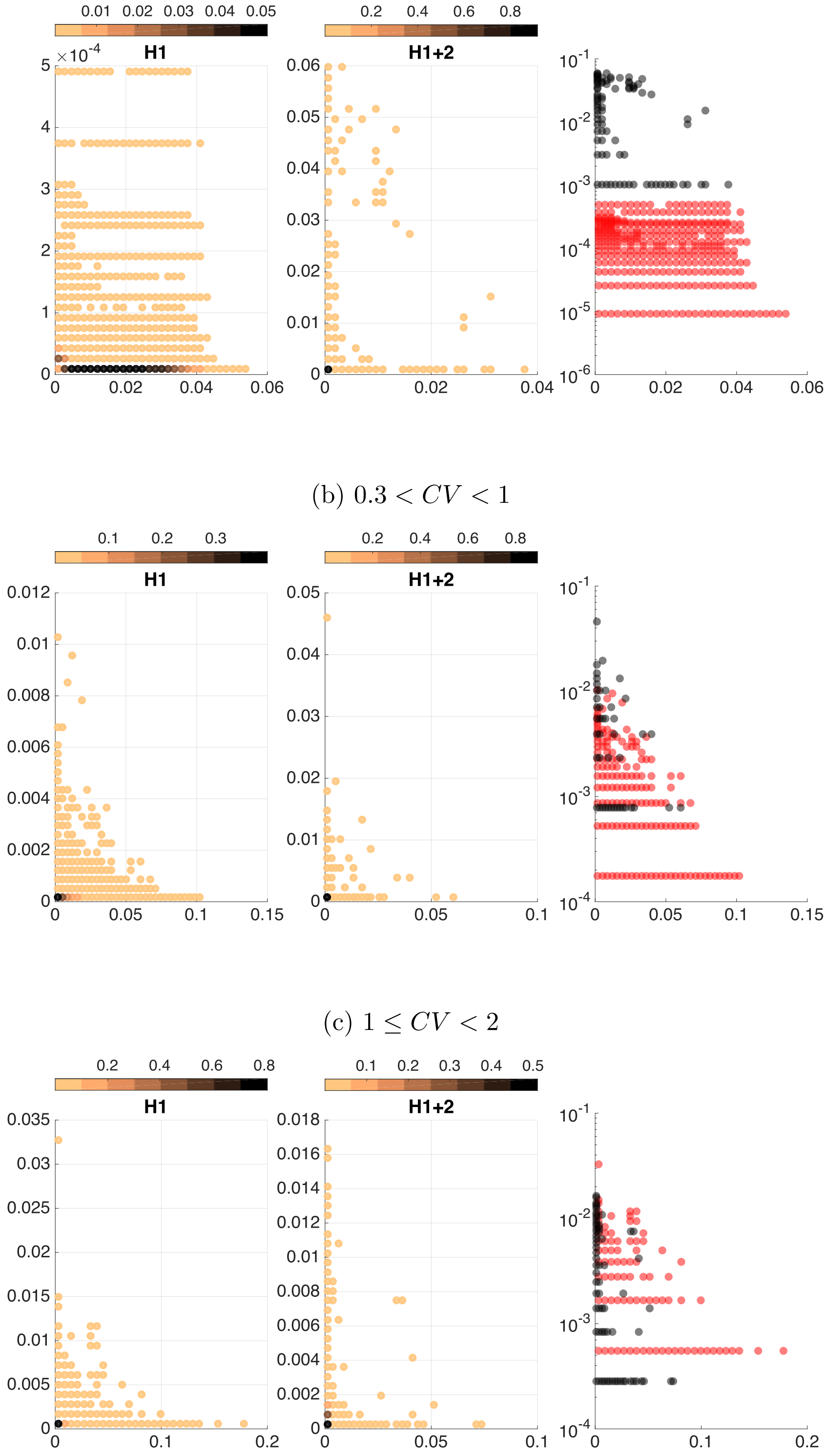


(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

Figure A.7: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n} w_i$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of $n = 20$ and $k = 1$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.



(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

Figure A.8: Basin size/2^n (y-axis) against $f(x)/\sum_{i=1}^{n} w_i$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the strict optima and plateaux found in 600 instance of $n = 20$ and $k = 1.2$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.

APPENDIX B

0-1 KNAPSACK PROBLEM SUPPLEMENTARY RESULTS

Tables of the search position types found in the feasible region of randomly generated instances of the 0-1 Knapsack problem with linear penalty function. These tables are used to support the observations/conclusions made in section 5.2.

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.54e-07	0.00E+00	0.00E+00	3.84E-01	0.00E+00	0.00E+00	1.16E-01
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	3.84E-01	0.00E+00	0.00E+00	1.16E-01
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	3.85E-01	0.00E+00	0.00E+00	1.15E-01
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	3.85E-01	0.00E+00	0.00E+00	1.15E-01
Subset Sum	9.54e-07	0.00E+00	0.00E+00	3.86E-01	0.00E+00	0.00E+00	1.15E-01
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	3.84E-01	0.00E+00	0.00E+00	1.16E-01
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	3.85E-01	0.00E+00	0.00E+00	1.15E-01
Circle	9.54e-07	0.00E+00	0.00E+00	3.86E-01	0.00E+00	0.00E+00	1.14E-01

(a) *H1* Landscape

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	3.64E-06
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	3.96E-01	1.04E-01	1.53E-07	3.54E-06
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	1.86E-01	3.12E-01	1.26E-03	6.67E-04
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	2.04E-01	2.95E-01	1.14E-03	7.52E-04
Subset Sum	9.54e-07	0.00E+00	0.00E+00	1.87E-01	3.12E-01	1.26E-03	6.93E-04
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	1.96E-01	3.04E-01	3.30E-04	1.82E-04
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	3.37E-02	4.64E-01	2.56E-03	2.85E-04
Circle	9.54e-07	0.00E+00	0.00E+00	1.99E-01	3.00E-01	1.19E-03	7.33E-04

(b) *H1+2* Landscape

Table B.1: Proportion of the search position types in the feasible region, averaged over 600 instances of KP with $k = 0.4$, $n = 20$, and $0 < CV \leq 0.3$.

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.54e-07	0.00E+00	0.00E+00	4.81E-01	0.00E+00	0.00E+00	1.95E-02
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	4.82E-01	0.00E+00	0.00E+00	1.82E-02
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.82E-01	0.00E+00	0.00E+00	1.87E-02
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.80E-01	0.00E+00	0.00E+00	2.04E-02
Subset Sum	9.54e-07	0.00E+00	0.00E+00	4.81E-01	0.00E+00	0.00E+00	1.89E-02
Uncorrelated Spanner	9.54e-07	0.00E+00	0.00E+00	4.71E-01	0.00E+00	0.00E+00	2.96E-02
Weakly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	4.71E-01	0.00E+00	0.00E+00	2.93E-02
Strongly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	4.72E-01	0.00E+00	0.00E+00	2.89E-02
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.81E-01	0.00E+00	0.00E+00	1.92E-02
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	4.80E-01	0.00E+00	0.00E+00	2.00E-02
Circle	9.54e-07	0.00E+00	0.00E+00	4.81E-01	0.00E+00	0.00E+00	1.91E-02

(a) *H1* Landscape

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	5.31E-06
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	3.02E-01	1.98E-01	4.02E-07	4.29E-06
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	2.69E-01	2.29E-01	1.33E-03	1.53E-03
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	3.01E-01	1.97E-01	1.17E-03	1.65E-03
Subset Sum	9.54e-07	0.00E+00	0.00E+00	2.93E-01	2.04E-01	1.15E-03	1.65E-03
Uncorrelated Spanner	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	2.45E-05
Weakly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	3.69E-01	1.31E-01	1.43E-05	1.39E-04
Strongly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	2.42E-01	2.57E-01	3.06E-04	2.76E-04
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	2.89E-01	2.10E-01	2.57E-04	4.01E-04
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	1.11E-01	3.85E-01	2.55E-03	9.82E-04
Circle	9.54e-07	0.00E+00	0.00E+00	2.93E-01	2.04E-01	1.13E-03	1.68E-03

(b) *H1+2* Landscape

Table B.2: Proportion of the search position types in the feasible region, averaged over 600 instances per each instance type, with $k = 0.4$, $n = 20$, and $0.3 < CV < 1$.

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.54e-07	0.00E+00	0.00E+00	4.92E-01	0.00E+00	0.00E+00	8.82E-03
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	4.92E-01	0.00E+00	0.00E+00	8.95E-03
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.92E-01	0.00E+00	0.00E+00	8.71E-03
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.92E-01	0.00E+00	0.00E+00	8.29E-03
Subset Sum	9.54e-07	0.00E+00	0.00E+00	4.92E-01	0.00E+00	0.00E+00	7.99E-03
Uncorrelated Spanner	9.54e-07	0.00E+00	0.00E+00	4.92E-01	0.00E+00	0.00E+00	8.99E-03
Weakly Correlated Spanner	9.50e-07	0.00E+00	0.00E+00	4.91E-01	0.00E+00	0.00E+00	9.05E-03
Strongly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	4.91E-01	0.00E+00	0.00E+00	9.39E-03
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.92E-01	0.00E+00	0.00E+00	8.68E-03
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	4.92E-01	0.00E+00	0.00E+00	8.52E-03
Circle	9.54e-07	0.00E+00	0.00E+00	4.91E-01	0.00E+00	0.00E+00	9.44E-03

(a) *H1* Landscape

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.25e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	4.81E-06
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	1.57E-01	3.44E-01	1.09E-06	5.06E-06
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	1.92E-01	3.06E-01	1.59E-03	9.67E-04
Inverse Strongly Correlated	9.31e-07	0.00E+00	0.00E+00	1.75E-01	3.24E-01	1.54E-03	8.52E-04
Subset Sum	9.42e-07	0.00E+00	0.00E+00	1.86E-01	3.12E-01	1.47E-03	9.48E-04
Uncorrelated Spanner	8.86e-07	0.00E+00	0.00E+00	5.01E-01	0.00E+00	0.00E+00	7.73E-06
Weakly Correlated Spanner	8.96e-07	0.00E+00	0.00E+00	2.10E-01	2.91E-01	1.99E-06	1.14E-05
Strongly Correlated Spanner	8.83e-07	0.00E+00	0.00E+00	2.31E-01	2.70E-01	8.02E-06	1.87E-05
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	1.91E-01	3.09E-01	3.48E-04	2.51E-04
Profit Ceiling	9.19e-07	0.00E+00	0.00E+00	3.18E-02	4.65E-01	2.95E-03	3.95E-04
Circle	9.31e-07	0.00E+00	0.00E+00	2.02E-01	2.96E-01	1.56E-03	1.03E-03

(b) *H1+2* Landscape

Table B.3: Proportion of the search position types in the feasible region, averaged over 600 instances of KP with $k = 0.4$, $n = 20$, and $1 \leq CV < 2$.

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.54e-07	0.00E+00	0.00E+00	3.83E-01	0.00E+00	0.00E+00	1.17E-01
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	3.83E-01	0.00E+00	0.00E+00	1.17E-01
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	3.83E-01	0.00E+00	0.00E+00	1.17E-01
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	3.82E-01	0.00E+00	0.00E+00	1.18E-01
Subset Sum	9.54e-07	0.00E+00	0.00E+00	3.82E-01	0.00E+00	0.00E+00	1.18E-01
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	3.83E-01	0.00E+00	0.00E+00	1.17E-01
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	3.83E-01	0.00E+00	0.00E+00	1.17E-01
Circle	9.54e-07	0.00E+00	0.00E+00	3.82E-01	0.00E+00	0.00E+00	1.18E-01

(a) *H1* Landscape

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	3.50E-06
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	3.71E-05
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.99E-01	0.00E+00	0.00E+00	1.20E-03
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.99E-01	0.00E+00	0.00E+00	1.23E-03
Subset Sum	9.54e-07	0.00E+00	0.00E+00	4.99E-01	0.00E+00	0.00E+00	1.19E-03
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	3.05E-04
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	4.98E-01	1.25E-03	3.56E-06	1.17E-03
Circle	9.54e-07	0.00E+00	0.00E+00	4.98E-01	1.25E-03	2.84E-06	1.19E-03

(b) *H1+2* Landscape

Table B.4: Proportion of the search position types in the feasible region, averaged over 600 instances of KP with $k = 1$, $n = 20$, and $0 < CV \leq 0.3$.

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.54e-07	0.00E+00	0.00E+00	4.82E-01	0.00E+00	0.00E+00	1.75E-02
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	4.82E-01	0.00E+00	0.00E+00	1.81E-02
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.83E-01	0.00E+00	0.00E+00	1.70E-02
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.82E-01	0.00E+00	0.00E+00	1.75E-02
Subset Sum	9.54e-07	0.00E+00	0.00E+00	4.82E-01	0.00E+00	0.00E+00	1.76E-02
Uncorrelated Spanner	9.54e-07	0.00E+00	0.00E+00	4.71E-01	0.00E+00	0.00E+00	2.94E-02
Weakly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	4.71E-01	0.00E+00	0.00E+00	2.93E-02
Strongly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	4.72E-01	0.00E+00	0.00E+00	2.78E-02
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.84E-01	0.00E+00	0.00E+00	1.63E-02
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	4.83E-01	0.00E+00	0.00E+00	1.72E-02
Circle	9.54e-07	0.00E+00	0.00E+00	4.83E-01	0.00E+00	0.00E+00	1.74E-02

(a) *H1* Landscape

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	5.04E-06
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	2.61E-04
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.98E-01	0.00E+00	0.00E+00	2.09E-03
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.98E-01	0.00E+00	0.00E+00	2.09E-03
Subset Sum	9.54e-07	0.00E+00	0.00E+00	4.98E-01	0.00E+00	0.00E+00	2.03E-03
Uncorrelated Spanner	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	2.27E-05
Weakly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	4.99E-01	0.00E+00	0.00E+00	7.40E-04
Strongly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	4.98E-01	0.00E+00	0.00E+00	1.68E-03
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.99E-01	0.00E+00	0.00E+00	9.21E-04
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	4.98E-01	0.00E+00	0.00E+00	2.08E-03
Circle	9.54e-07	0.00E+00	0.00E+00	4.98E-01	0.00E+00	0.00E+00	2.10E-03

(b) *H1+2* Landscape

Table B.5: Proportion of the search position types in the feasible region, averaged over 600 instances per each instance type, with $k = 1$, $n = 20$, and $0.3 < CV < 1$.

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.54e-07	0.00E+00	0.00E+00	4.93E-01	0.00E+00	0.00E+00	7.15E-03
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	4.92E-01	0.00E+00	0.00E+00	7.89E-03
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.93E-01	0.00E+00	0.00E+00	7.22E-03
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.92E-01	0.00E+00	0.00E+00	7.83E-03
Subset Sum	9.54e-07	0.00E+00	0.00E+00	4.92E-01	0.00E+00	0.00E+00	8.23E-03
Uncorrelated Spanner	9.46e-07	0.00E+00	0.00E+00	4.95E-01	0.00E+00	0.00E+00	5.03E-03
Weakly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	4.95E-01	0.00E+00	0.00E+00	4.81E-03
Strongly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	4.95E-01	0.00E+00	0.00E+00	4.83E-03
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	4.92E-01	0.00E+00	0.00E+00	7.83E-03
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	4.92E-01	0.00E+00	0.00E+00	7.63E-03
Circle	9.54e-07	0.00E+00	0.00E+00	4.92E-01	0.00E+00	0.00E+00	8.23E-03

(a) *H1* Landscape

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.42e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	4.41E-06
Weakly Correlated	9.48e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	7.39E-05
Strongly Correlated	9.31e-07	0.00E+00	0.00E+00	4.99E-01	0.00E+00	0.00E+00	1.45E-03
Inverse Strongly Correlated	9.34e-07	0.00E+00	0.00E+00	4.99E-01	0.00E+00	0.00E+00	1.37E-03
Subset Sum	9.36e-07	0.00E+00	0.00E+00	4.99E-01	0.00E+00	0.00E+00	1.40E-03
Uncorrelated Spanner	8.43e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	6.70E-06
Weakly Correlated Spanner	8.49e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	1.09E-04
Strongly Correlated Spanner	8.69e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	2.24E-04
Multiple Strongly Correlated	9.23e-07	0.00E+00	0.00E+00	4.99E-01	0.00E+00	0.00E+00	5.53E-04
Profit Ceiling	9.37e-07	0.00E+00	0.00E+00	4.99E-01	0.00E+00	0.00E+00	1.44E-03
Circle	9.48e-07	0.00E+00	0.00E+00	4.98E-01	0.00E+00	0.00E+00	1.51E-03

(b) *H1+2* Landscape

Table B.6: Proportion of the search position types in the feasible region, averaged over 600 instances of KP with $k = 1$, $n = 20$, and $1 \leq CV < 2$.

Tables of the search position types found in the infeasible region of randomly generated instances of the 0-1 Knapsack problem with linear penalty function. These tables are used to support the observations/conclusions made in section 5.2.

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.00e-07	1.08E-07	0.00E+00	4.74E-01	2.53E-02	0.00E+00	0.00E+00
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Subset Sum	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Circle	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

(a) *H1* Landscape

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.00e-07	1.08E-07	0.00E+00	4.74E-01	2.53E-02	0.00E+00	0.00E+00
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	4.98E-01	1.25E-03	0.00E+00	0.00E+00
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	1.87E-01	3.13E-01	0.00E+00	0.00E+00
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	2.03E-01	2.96E-01	0.00E+00	0.00E+00
Subset Sum	9.54e-07	0.00E+00	0.00E+00	1.87E-01	3.12E-01	0.00E+00	0.00E+00
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	1.96E-01	3.04E-01	0.00E+00	0.00E+00
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	1.36E-01	3.64E-01	0.00E+00	0.00E+00
Circle	9.54e-07	0.00E+00	0.00E+00	1.99E-01	3.00E-01	0.00E+00	0.00E+00

(b) *H1+2* Landscape

Table B.7: Proportion of the search position types in the infeasible region, averaged over 600 instances of KP with $k = 0.4$, $n = 20$, and $0 < CV \leq 0.3$.

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.02e-07	1.02E-07	0.00E+00	4.73E-01	2.66E-02	0.00E+00	0.00E+00
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Subset Sum	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Uncorrelated Spanner	9.46e-07	1.59E-08	0.00E+00	4.95E-01	4.07E-03	0.00E+00	0.00E+00
Weakly Correlated Spanner	9.51e-07	1.03E-08	0.00E+00	4.98E-01	1.33E-03	0.00E+00	0.00E+00
Strongly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Multiple Strongly Correlated	9.50e-07	7.75E-09	0.00E+00	4.98E-01	1.98E-03	0.00E+00	0.00E+00
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Circle	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

(a) *H1* Landscape

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.02e-07	1.02E-07	0.00E+00	4.73E-01	2.66E-02	0.00E+00	0.00E+00
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	4.88E-01	1.21E-02	0.00E+00	0.00E+00
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	2.70E-01	2.30E-01	0.00E+00	0.00E+00
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	3.02E-01	1.98E-01	0.00E+00	0.00E+00
Subset Sum	9.54e-07	0.00E+00	0.00E+00	2.95E-01	2.05E-01	0.00E+00	0.00E+00
Uncorrelated Spanner	9.46e-07	1.59E-08	0.00E+00	4.95E-01	4.07E-03	0.00E+00	0.00E+00
Weakly Correlated Spanner	9.51e-07	1.03E-08	0.00E+00	4.74E-01	2.52E-02	0.00E+00	0.00E+00
Strongly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	3.32E-01	1.68E-01	0.00E+00	0.00E+00
Multiple Strongly Correlated	9.50e-07	7.75E-09	0.00E+00	2.88E-01	2.12E-01	0.00E+00	0.00E+00
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	2.91E-01	2.09E-01	0.00E+00	0.00E+00
Circle	9.54e-07	0.00E+00	0.00E+00	2.94E-01	2.05E-01	0.00E+00	0.00E+00

(b) *H1+2* Landscape

Table B.8: Proportion of the search position types in the infeasible region, averaged over 600 instances of KP with $k = 0.4$, $n = 20$, and $0.3 < CV < 1$.

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.07e-07	9.30E-08	0.00E+00	4.75E-01	2.43E-02	0.00E+00	0.00E+00
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Subset Sum	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Uncorrelated Spanner	9.50e-07	7.95E-09	0.00E+00	4.97E-01	2.06E-03	0.00E+00	0.00E+00
Weakly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	4.99E-01	0.00E+00	0.00E+00	0.00E+00
Strongly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	4.99E-01	0.00E+00	0.00E+00	0.00E+00
Multiple Strongly Correlated	9.48e-07	9.78E-08	0.00E+00	4.96E-01	3.20E-03	0.00E+00	0.00E+00
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Circle	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

(a) *H1* Landscape

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.07e-07	9.30E-08	0.00E+00	4.75E-01	2.43E-02	0.00E+00	0.00E+00
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	4.55E-01	4.49E-02	0.00E+00	0.00E+00
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	1.92E-01	3.07E-01	0.00E+00	0.00E+00
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	1.75E-01	3.25E-01	0.00E+00	0.00E+00
Subset Sum	9.54e-07	0.00E+00	0.00E+00	1.87E-01	3.13E-01	0.00E+00	0.00E+00
Uncorrelated Spanner	9.50e-07	7.95E-09	0.00E+00	4.97E-01	2.06E-03	0.00E+00	0.00E+00
Weakly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	3.85E-01	1.15E-01	0.00E+00	0.00E+00
Strongly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	3.09E-01	1.90E-01	0.00E+00	0.00E+00
Multiple Strongly Correlated	9.48e-07	9.78E-08	0.00E+00	1.91E-01	3.09E-01	0.00E+00	0.00E+00
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	1.92E-01	3.08E-01	0.00E+00	0.00E+00
Circle	9.54e-07	0.00E+00	0.00E+00	2.02E-01	2.97E-01	0.00E+00	0.00E+00

(b) *H1+2* Landscape

Table B.9: Proportion of the search position types in the infeasible region, averaged over 600 instances of KP with $k = 0.4$, $n = 20$, and $1 \leq CV < 2$.

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.01e-07	1.05E-07	0.00E+00	4.75E-01	2.46E-02	0.00E+00	0.00E+00
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Subset Sum	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Circle	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

(a) *H1* Landscape

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.01e-07	1.05E-07	0.00E+00	4.75E-01	2.46E-02	0.00E+00	0.00E+00
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	4.99E-01	1.16E-03	0.00E+00	0.00E+00
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Subset Sum	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	4.99E-01	1.25E-03	0.00E+00	0.00E+00
Circle	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

(b) *H1+2* Landscape

Table B.10: Proportion of the search position types in the infeasible region, averaged over 600 instances of KP with $k = 1$, $n = 20$, and $0 < CV \leq 0.3$.

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	8.93e-07	1.22E-07	0.00E+00	4.68E-01	3.17E-02	0.00E+00	0.00E+00
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Subset Sum	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Uncorrelated Spanner	9.45e-07	1.68E-08	0.00E+00	4.96E-01	4.31E-03	0.00E+00	0.00E+00
Weakly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Strongly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Circle	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

(a) *H1* Landscape

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	8.93e-07	1.22E-07	0.00E+00	4.68E-01	3.17E-02	0.00E+00	0.00E+00
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Subset Sum	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Uncorrelated Spanner	9.45e-07	1.68E-08	0.00E+00	4.96E-01	4.31E-03	0.00E+00	0.00E+00
Weakly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Strongly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Circle	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

(b) *H1+2* Landscape

Table B.11: Proportion of the search position types in the infeasible region, averaged over 600 instances of KP with $k = 1$, $n = 20$, and $0.3 < CV < 1$.

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.02e-07	1.03E-07	0.00E+00	4.73E-01	2.69E-02	0.00E+00	0.00E+00
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Subset Sum	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Uncorrelated Spanner	9.32e-07	4.42E-08	0.00E+00	4.88E-01	1.16E-02	0.00E+00	0.00E+00
Weakly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Strongly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Circle	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

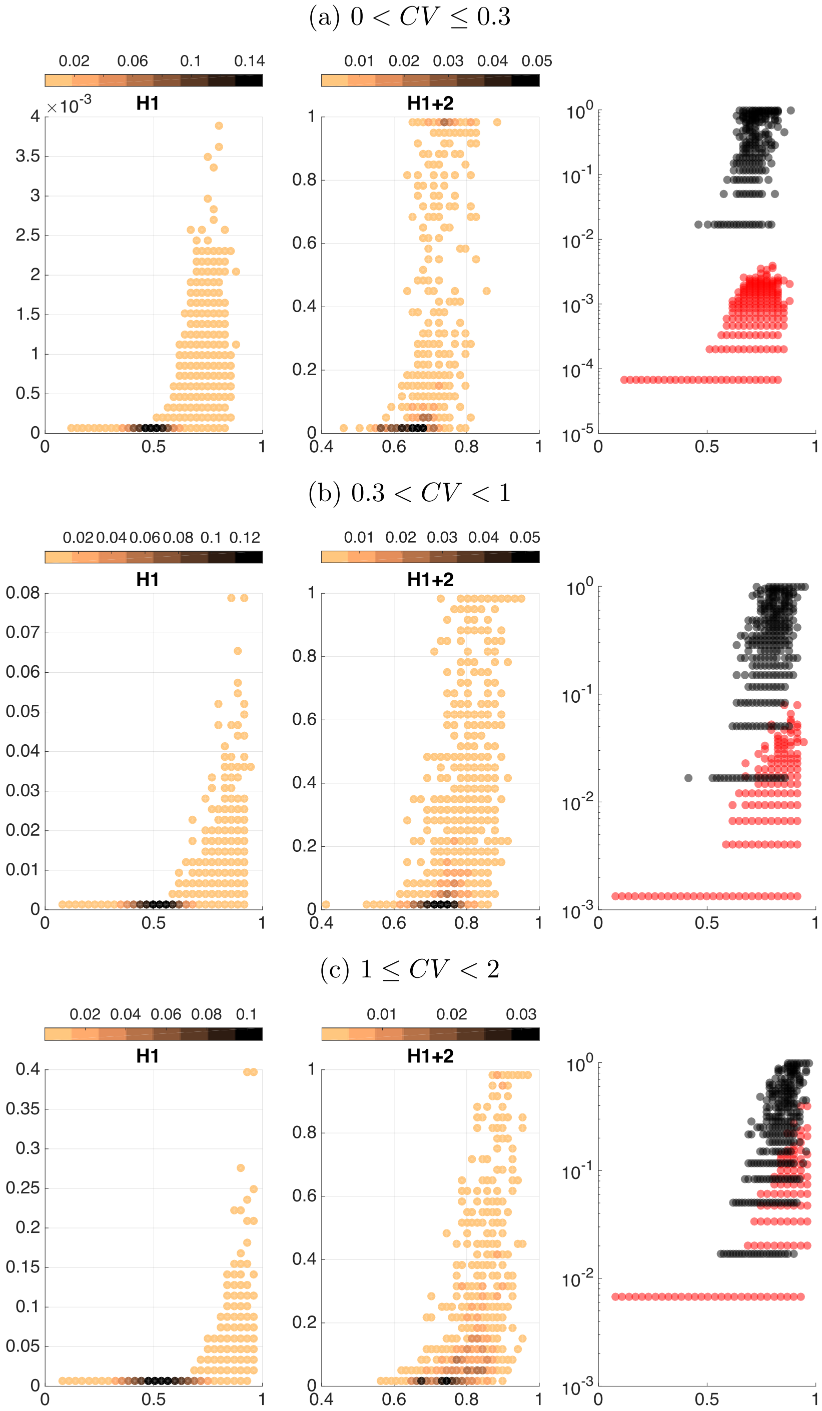
(a) *H1* Landscape

Instance Type	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
Uncorrelated	9.02e-07	1.03E-07	0.00E+00	4.73E-01	2.69E-02	0.00E+00	0.00E+00
Weakly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Inverse Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Subset Sum	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Uncorrelated Spanner	9.32e-07	4.42E-08	0.00E+00	4.88E-01	1.16E-02	0.00E+00	0.00E+00
Weakly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Strongly Correlated Spanner	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Multiple Strongly Correlated	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Profit Ceiling	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
Circle	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

(b) *H1+2* Landscape

Table B.12: Proportion of the search position types in the infeasible region, averaged over 600 instances of KP with $k = 1$, $n = 20$, and $1 \leq CV < 2$.

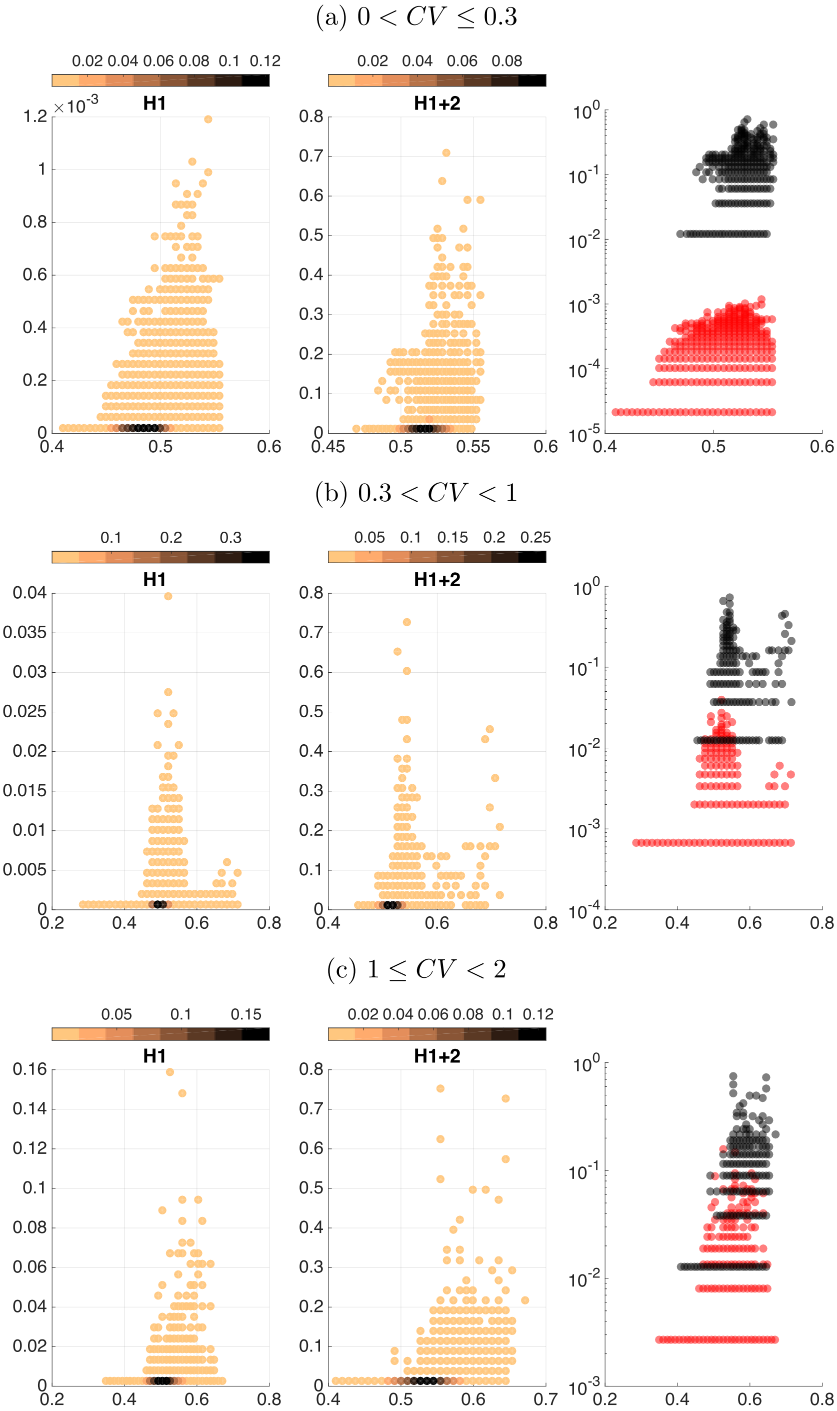
The following figures are used to support the observations/conclusions made in subsection 5.4.1.


(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

Figure B.1: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n} p_i$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of uncorrelated knapsack of size $n = 20$ and $k = 1$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.

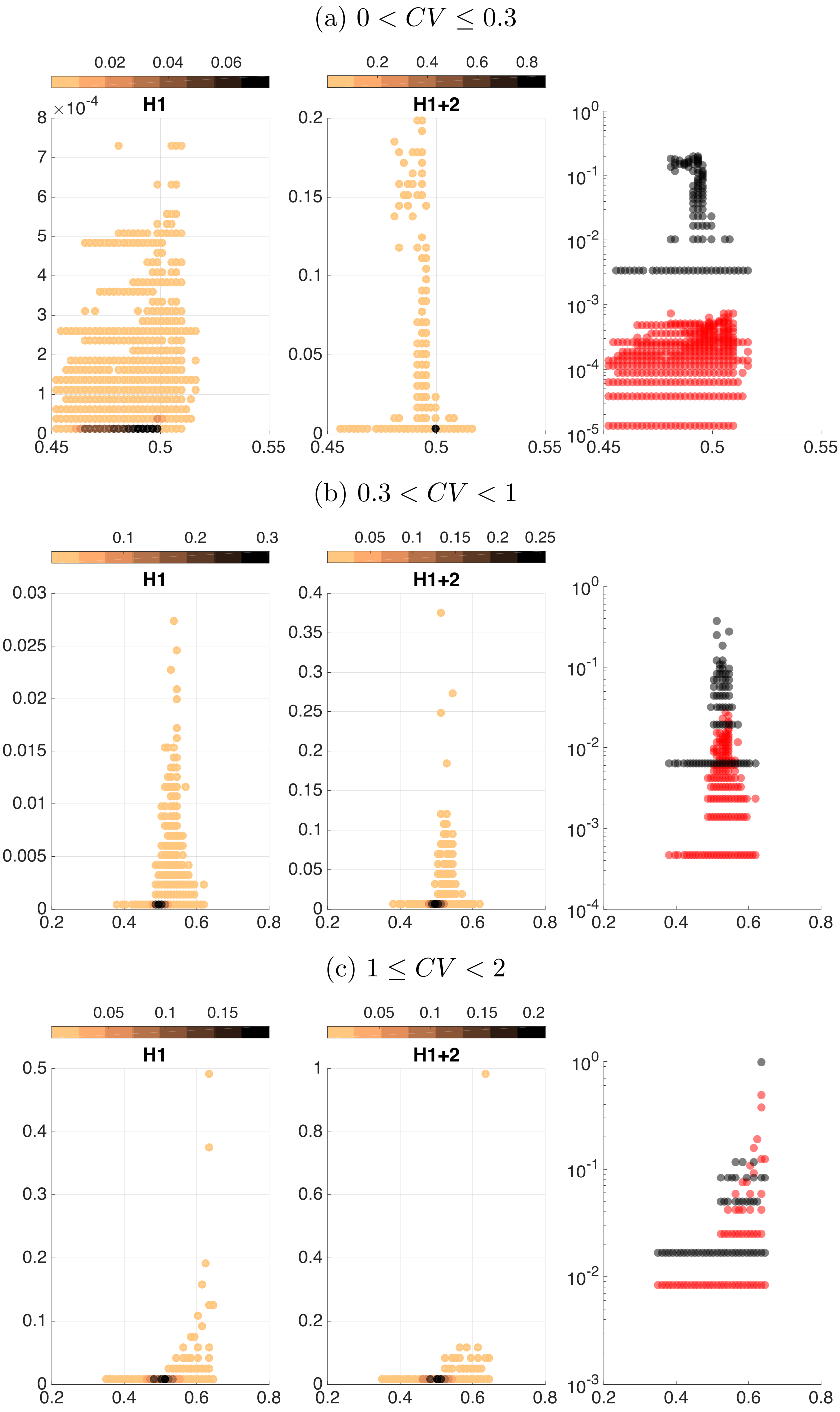


(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

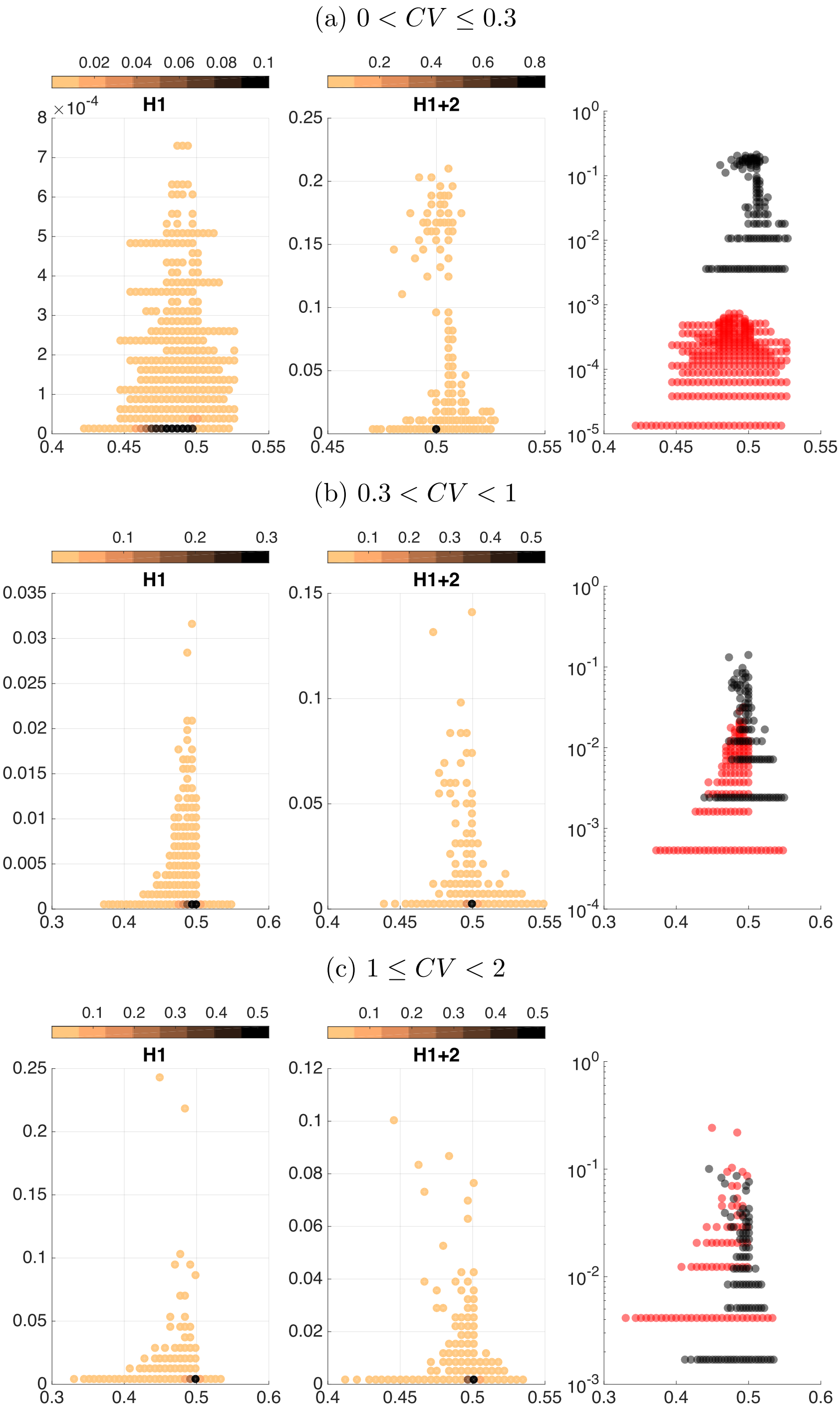
Figure B.2: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n} p_i$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of weakly correlated knapsack of size $n = 20$ and $k = 1$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.


(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

Figure B.3: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n} p_i$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of strongly correlated knapsack of size $n = 20$ and $k = 1$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.

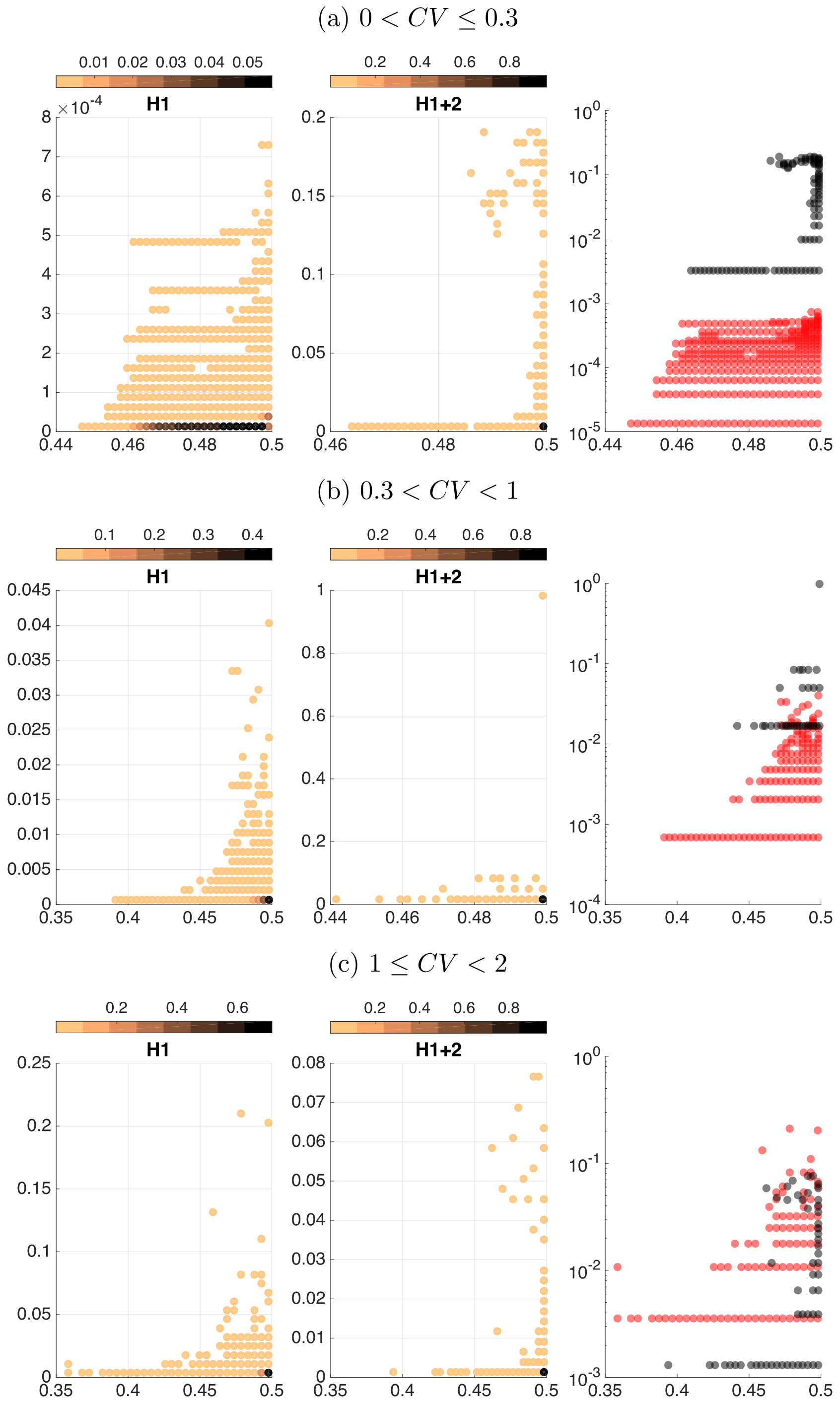


(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

Figure B.4: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n} p_i$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of inverse strongly correlated knapsack of size $n = 20$ and $k = 1$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.

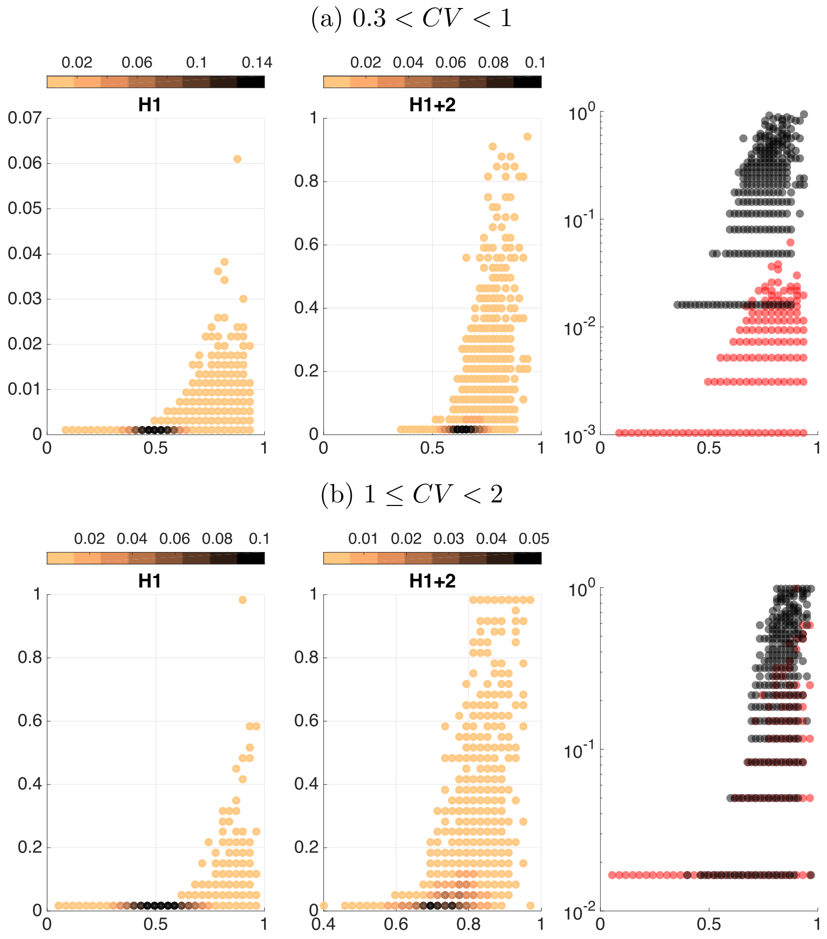


(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

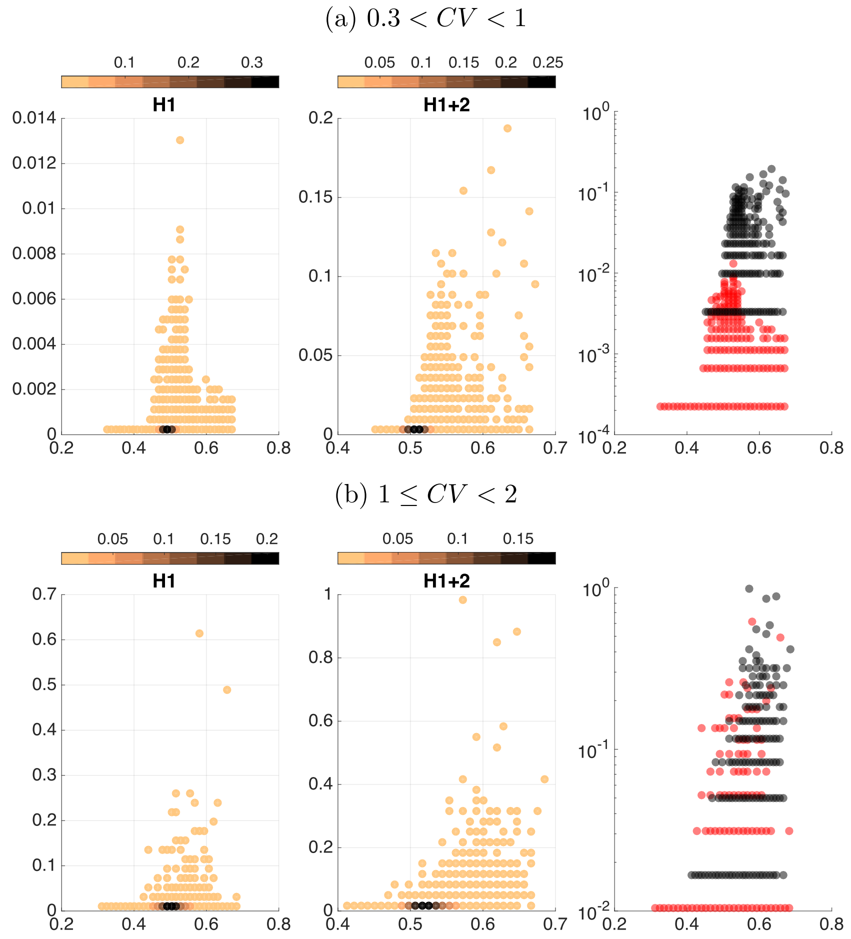
Figure B.5: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n} p_i$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of subset sum knapsack of size $n = 20$ and $k = 1$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.



(a) $0.3 < CV < 1$

(b) $1 \leq CV < 2$

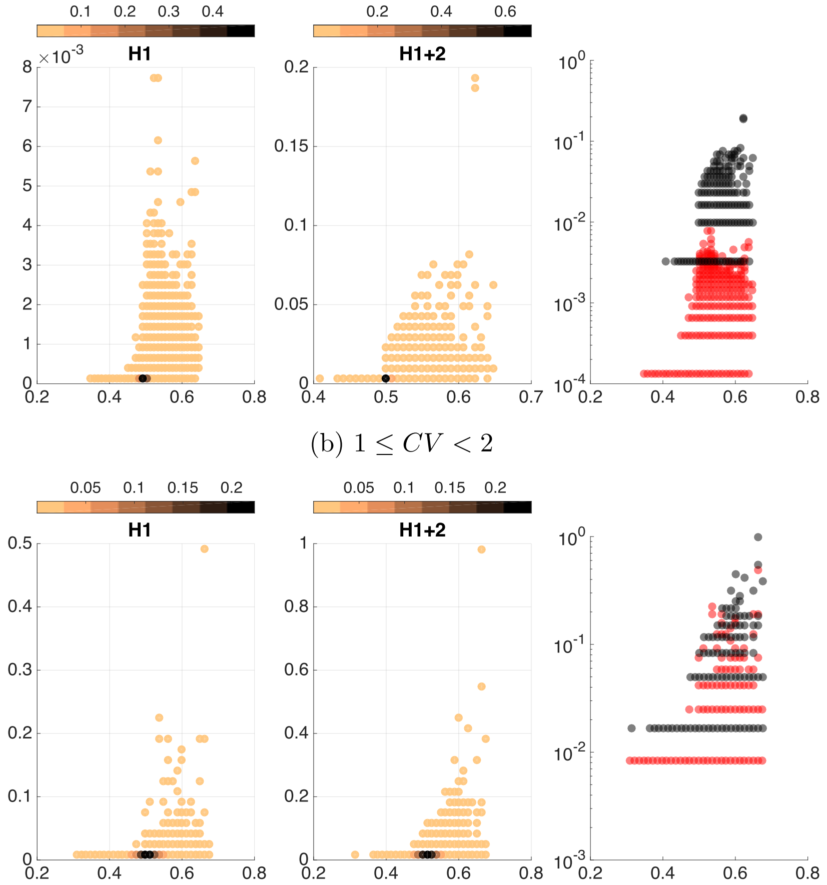
Figure B.6: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n} p_i$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of uncorrelated spanner knapsack of size $n = 20$ and $k = 1$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.



(a) $0.3 < CV < 1$

(b) $1 \leq CV < 2$

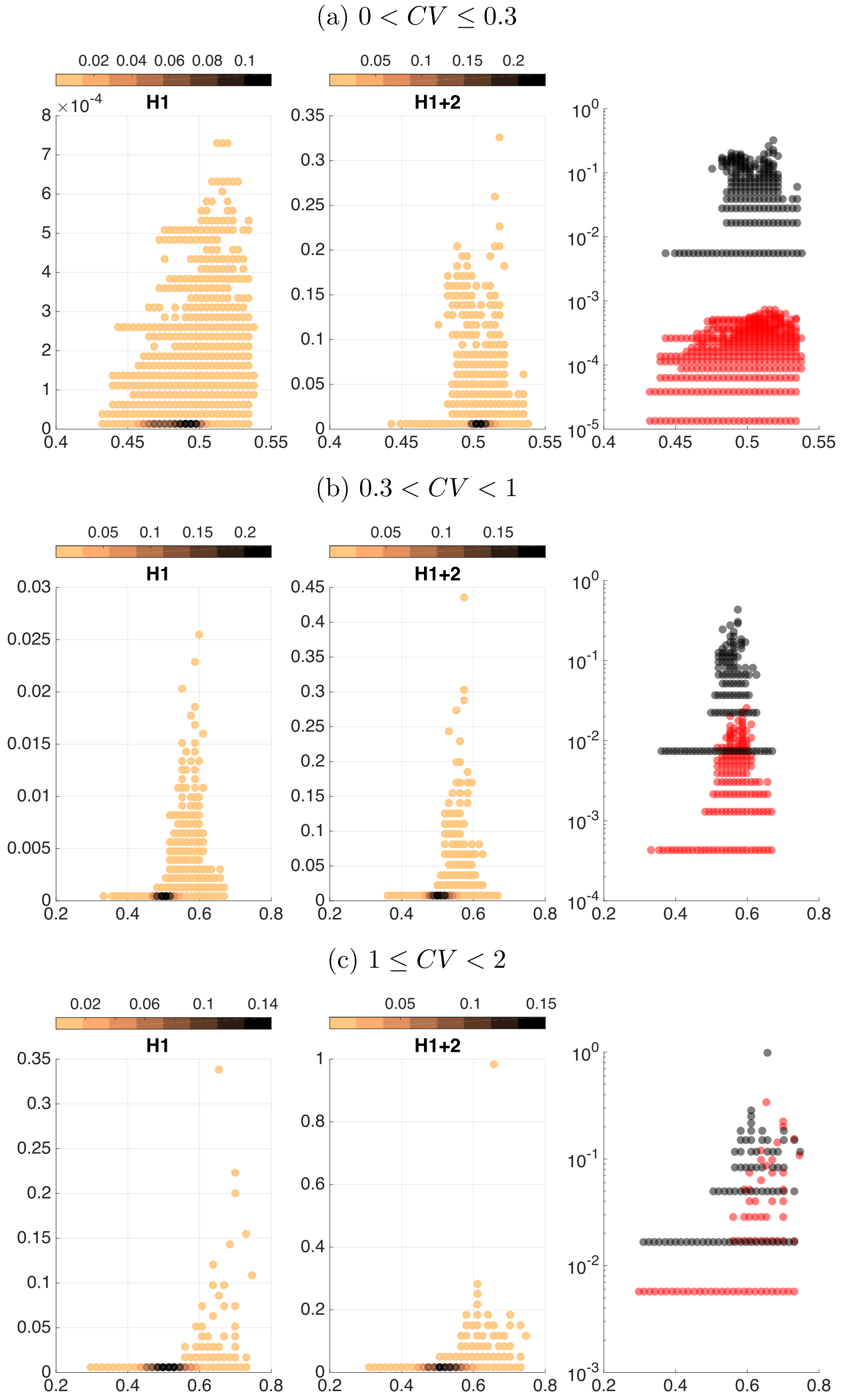
Figure B.7: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n} p_i$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of weakly correlated spanner knapsack of size $n = 20$ and $k = 1$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.



(a) $0.3 < CV < 1$

(b) $1 \leq CV < 2$

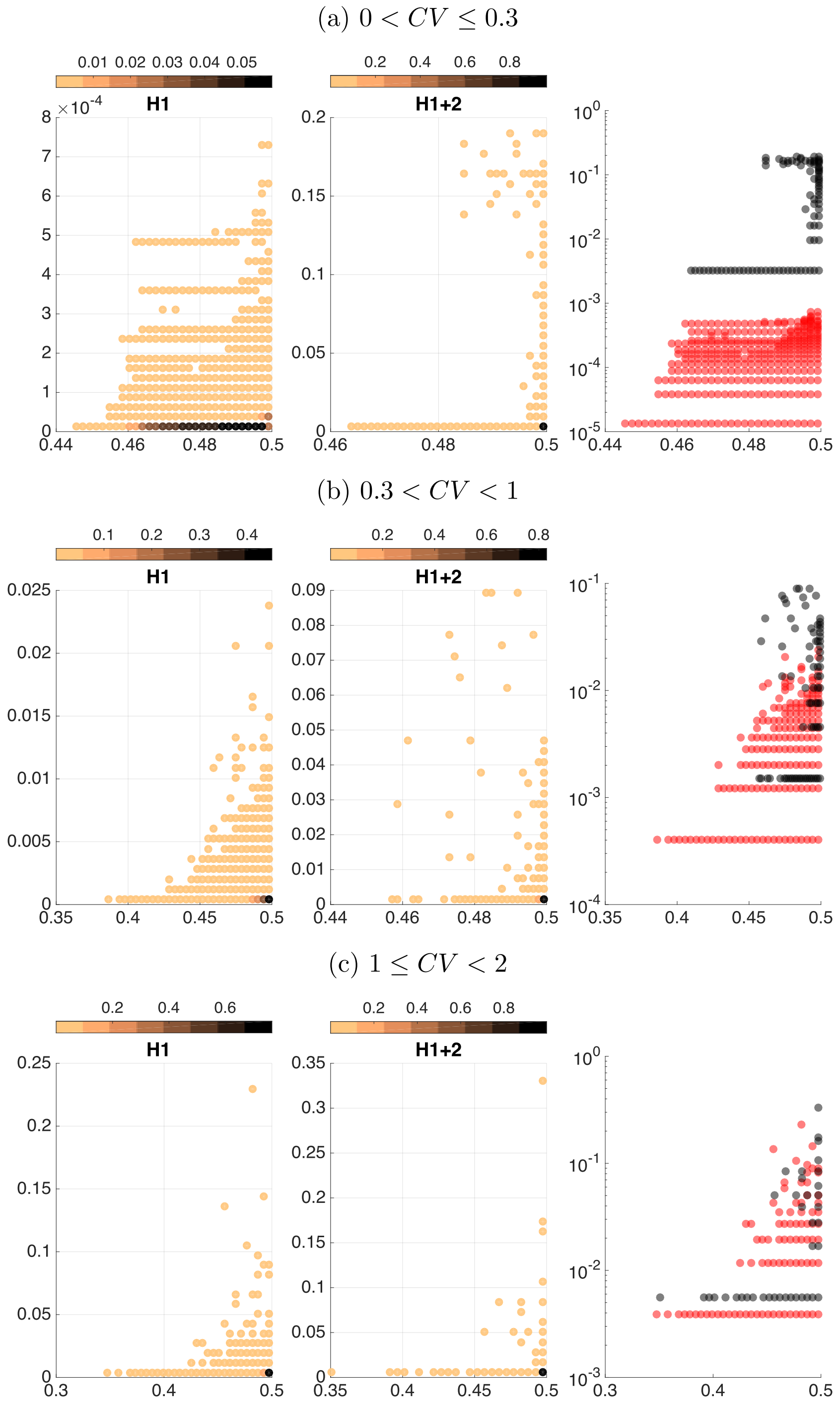
Figure B.8: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n} p_i$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of strongly correlated spanner knapsack of size $n = 20$ and $k = 1$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.


(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

Figure B.9: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n} p_i$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of multiple strongly correlated knapsack of size $n = 20$ and $k = 1$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.

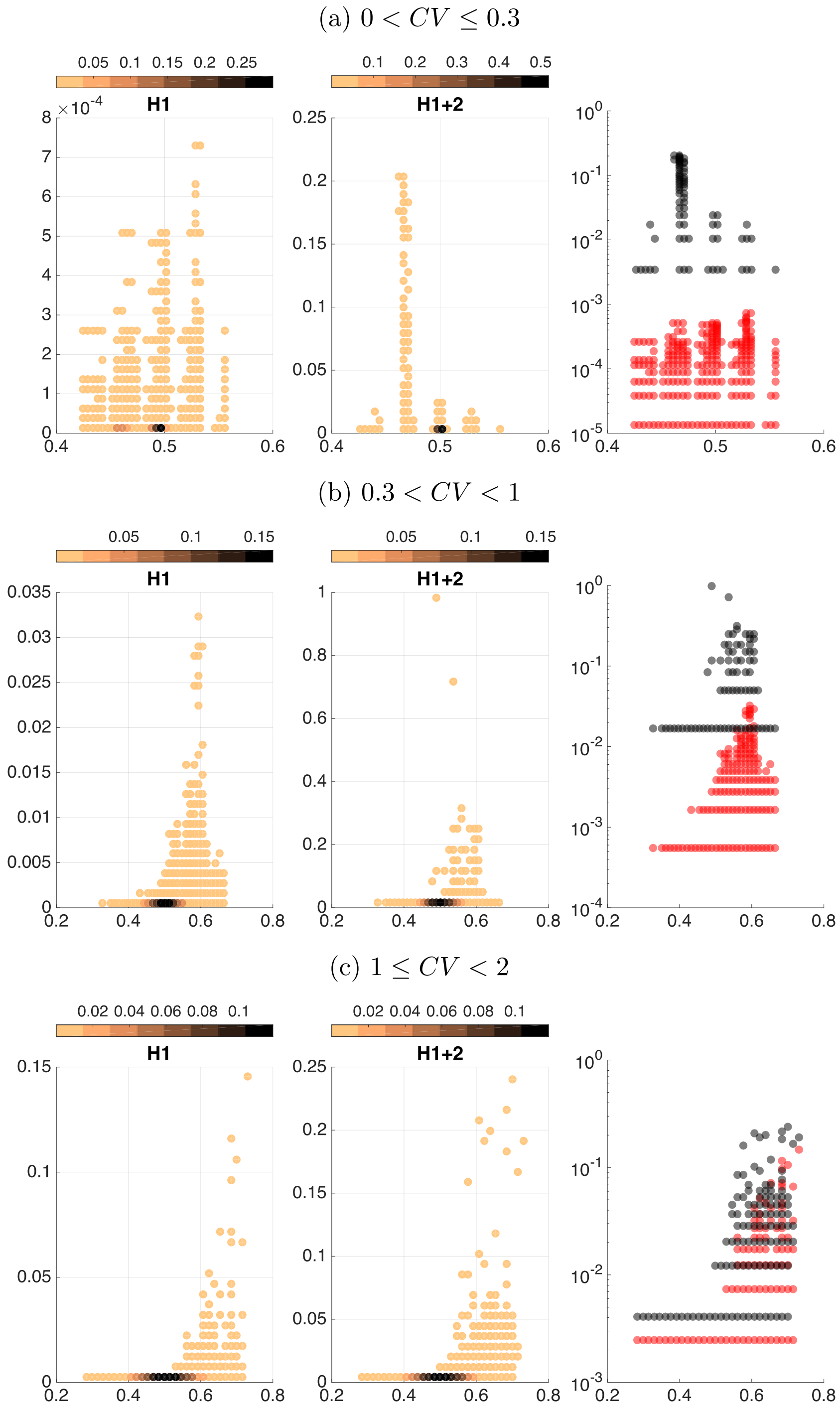


(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

Figure B.10: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n} p_i$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of profit ceiling knapsack of size $n = 20$ and $k = 1$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.



(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

Figure B.11: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n} p_i$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of circle knapsack of size $n = 20$ and $k = 1$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.

APPENDIX C

QUADRATIC 0-1 KNAPSACK PROBLEM SUPPLEMENTARY RESULTS

The following tables are used to support the observations/conclusions made in section 6.2.

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	0.00e+00	1.70E-02	1.30E-06	7.59E-04	3.60E-01	1.20E-01	3.40E-03
0.25	0.00e+00	8.27E-04	4.84E-09	8.37E-02	3.02E-01	4.85E-02	6.52E-02
0.5	0.00e+00	5.37E-05	0.00E+00	3.55E-01	3.05E-02	1.34E-03	1.13E-01
0.75	4.87e-09	8.79E-06	0.00E+00	3.85E-01	6.92E-04	4.81E-06	1.15E-01
0.95	2.99e-07	1.69E-06	0.00E+00	3.84E-01	2.85E-06	0.00E+00	1.16E-01
1	9.54e-07	0.00E+00	0.00E+00	3.86E-01	0.00E+00	0.00E+00	1.15E-01

(a) *H1* Landscape

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	0.00e+00	1.38E-02	0.00E+00	3.71E-03	4.83E-01	1.28E-05	6.60E-06
0.25	0.00e+00	8.24E-04	0.00E+00	1.33E-01	3.66E-01	5.71E-07	8.42E-06
0.5	0.00e+00	5.37E-05	0.00E+00	4.36E-01	6.40E-02	6.88E-08	8.50E-06
0.75	4.87e-09	8.79E-06	0.00E+00	4.71E-01	2.95E-02	9.24E-08	8.85E-06
0.95	2.99e-07	1.69E-06	0.00E+00	4.71E-01	2.93E-02	1.20E-07	8.71E-06
1	9.54e-07	0.00E+00	0.00E+00	4.70E-01	3.05E-02	9.24E-08	8.21E-06

(b) *H1+2* Landscape

Table C.1: Proportion of the search position types in the feasible region, averaged over 600 instances for each Δ value, with $k = 0.4$, $n = 20$, and $0 < CV \leq 0.3$.

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	0.00e+00	1.30E-02	2.85E-06	1.90E-03	4.56E-01	2.85E-02	6.25E-04
0.25	0.00e+00	8.09E-04	0.00E+00	1.43E-01	3.36E-01	8.31E-03	1.18E-02
0.5	0.00e+00	5.68E-05	0.00E+00	4.51E-01	2.97E-02	1.94E-04	1.95E-02
0.75	8.05e-09	9.18E-06	0.00E+00	4.80E-01	5.92E-04	3.14E-07	1.99E-02
0.95	3.24e-07	1.59E-06	0.00E+00	4.82E-01	2.56E-06	0.00E+00	1.87E-02
1	9.54e-07	0.00E+00	0.00E+00	4.81E-01	0.00E+00	0.00E+00	1.94E-02

(a) *H1* Landscape

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	0.00e+00	7.70E-03	0.00E+00	2.29E-03	4.90E-01	2.18E-05	3.64E-06
0.25	0.00e+00	6.94E-04	0.00E+00	1.41E-01	3.59E-01	7.93E-07	8.19E-06
0.5	0.00e+00	5.57E-05	0.00E+00	4.41E-01	5.94E-02	6.11E-08	8.91E-06
0.75	8.05e-09	9.14E-06	0.00E+00	4.73E-01	2.69E-02	4.83E-08	8.13E-06
0.95	3.24e-07	1.59E-06	0.00E+00	4.73E-01	2.68E-02	1.11E-07	8.29E-06
1	9.54e-07	0.00E+00	0.00E+00	4.72E-01	2.83E-02	8.93E-08	9.66E-06

(b) *H1+2* Landscape

Table C.2: Proportion of the search position types in the feasible region, averaged over 600 instances for each Δ value, with $k = 0.4$, $n = 20$, and $0.3 < CV < 1$.

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	0.00e+00	1.02E-02	1.86E-06	5.78E-04	4.75E-01	1.39E-02	2.42E-04
0.25	0.00e+00	6.85E-04	0.00E+00	1.48E-01	3.42E-01	4.06E-03	5.33E-03
0.5	0.00e+00	5.13E-05	0.00E+00	4.64E-01	2.74E-02	1.01E-04	8.63E-03
0.75	5.71e-09	9.31E-06	0.00E+00	4.91E-01	6.27E-04	6.51E-07	8.46E-03
0.95	3.08e-07	1.60E-06	0.00E+00	4.92E-01	2.71E-06	0.00E+00	8.86E-03
1	9.54e-07	0.00E+00	0.00E+00	4.92E-01	0.00E+00	0.00E+00	8.58E-03

(a) *H1* Landscape

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	0.00e+00	2.62E-03	0.00E+00	7.61E-04	4.97E-01	3.99E-05	2.23E-06
0.25	0.00e+00	3.16E-04	0.00E+00	1.40E-01	3.60E-01	8.50E-07	5.90E-06
0.5	0.00e+00	3.66E-05	0.00E+00	4.45E-01	5.57E-02	6.10E-08	5.79E-06
0.75	5.71e-09	7.92E-06	0.00E+00	4.75E-01	2.57E-02	5.14E-08	6.20E-06
0.95	3.02e-07	1.49E-06	0.00E+00	4.74E-01	2.59E-02	5.33E-08	6.25E-06
1	9.48e-07	0.00E+00	0.00E+00	4.73E-01	2.71E-02	2.82E-08	6.12E-06

(b) *H1+2* Landscape

Table C.3: Proportion of the search position types in the feasible region, averaged over 600 instances for each Δ value, with $k = 0.4$, $n = 20$, and $1 \leq CV < 2$.

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	0.00e+00	1.58E-02	8.20E-07	6.40E-04	3.60E-01	1.20E-01	3.98E-03
0.25	0.00e+00	8.53E-04	4.77E-09	8.03E-02	3.00E-01	5.11E-02	6.76E-02
0.5	0.00e+00	5.22E-05	0.00E+00	3.52E-01	3.01E-02	1.25E-03	1.16E-01
0.75	0.00e+00	8.85E-06	0.00E+00	3.81E-01	6.03E-04	1.11E-06	1.18E-01
0.95	3.72e-07	1.52E-06	0.00E+00	3.82E-01	2.84E-06	0.00E+00	1.18E-01
1	9.54e-07	0.00E+00	0.00E+00	3.83E-01	0.00E+00	0.00E+00	1.17E-01

(a) *H1* Landscape

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	0.00e+00	1.30E-02	0.00E+00	4.62E-03	4.82E-01	9.18E-06	6.18E-06
0.25	0.00e+00	8.49E-04	0.00E+00	1.48E-01	3.51E-01	3.96E-07	8.27E-06
0.5	0.00e+00	5.22E-05	0.00E+00	4.69E-01	3.14E-02	4.77E-09	8.71E-06
0.75	0.00e+00	8.85E-06	0.00E+00	4.99E-01	6.11E-04	0.00E+00	8.89E-06
0.95	3.72e-07	1.52E-06	0.00E+00	5.00E-01	1.03E-05	0.00E+00	8.31E-06
1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	7.68E-06	0.00E+00	8.56E-06

(b) *H1+2* Landscape

Table C.4: Proportion of the search position types in the feasible region, averaged over 600 instances for each Δ value, with $k = 1$, $n = 20$, and $0 < CV \leq 0.3$.

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	0.00e+00	1.35E-02	1.65E-06	1.43E-03	4.55E-01	2.93E-02	5.62E-04
0.25	0.00e+00	8.39E-04	0.00E+00	1.40E-01	3.41E-01	7.45E-03	1.07E-02
0.5	0.00e+00	5.37E-05	0.00E+00	4.50E-01	3.15E-02	1.88E-04	1.78E-02
0.75	0.00e+00	8.67E-06	0.00E+00	4.80E-01	5.96E-04	4.57E-07	1.90E-02
0.95	3.79e-07	1.56E-06	0.00E+00	4.82E-01	2.68E-06	0.00E+00	1.77E-02
1	9.54e-07	0.00E+00	0.00E+00	4.82E-01	0.00E+00	0.00E+00	1.80E-02

(a) *H1* Landscape

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	0.00e+00	8.15E-03	0.00E+00	1.98E-03	4.90E-01	2.11E-05	3.79E-06
0.25	0.00e+00	7.37E-04	0.00E+00	1.51E-01	3.48E-01	8.83E-07	8.04E-06
0.5	0.00e+00	5.30E-05	0.00E+00	4.68E-01	3.17E-02	8.44E-09	9.00E-06
0.75	0.00e+00	8.65E-06	0.00E+00	4.99E-01	6.04E-04	0.00E+00	8.91E-06
0.95	3.79e-07	1.56E-06	0.00E+00	5.00E-01	1.00E-05	0.00E+00	8.51E-06
1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	7.01E-06	0.00E+00	8.67E-06

(b) *H1+2* Landscape

Table C.5: Proportion of the search position types in the feasible region, averaged over 600 instances for each Δ value, with $k = 1$, $n = 20$, and $0.3 < CV < 1$.

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	0.00e+00	8.99E-03	9.71E-07	1.17E-03	4.78E-01	1.16E-02	2.77E-04
0.25	0.00e+00	6.63E-04	0.00E+00	1.49E-01	3.42E-01	3.42E-03	5.02E-03
0.5	0.00e+00	5.13E-05	0.00E+00	4.62E-01	3.01E-02	1.08E-04	7.62E-03
0.75	0.00e+00	8.78E-06	0.00E+00	4.91E-01	6.35E-04	7.18E-07	7.94E-03
0.95	3.16e-07	1.69E-06	0.00E+00	4.93E-01	3.02E-06	0.00E+00	7.38E-03
1	9.54e-07	0.00E+00	0.00E+00	4.92E-01	0.00E+00	0.00E+00	8.12E-03

(a) *H1* Landscape

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	0.00e+00	2.54E-03	0.00E+00	1.45E-03	4.96E-01	2.46E-05	2.43E-06
0.25	0.00e+00	3.25E-04	0.00E+00	1.54E-01	3.46E-01	8.69E-07	6.47E-06
0.5	0.00e+00	3.57E-05	0.00E+00	4.70E-01	3.03E-02	0.00E+00	5.97E-06
0.75	0.00e+00	7.82E-06	0.00E+00	4.99E-01	6.43E-04	0.00E+00	6.45E-06
0.95	3.16e-07	1.58E-06	0.00E+00	5.00E-01	9.44E-06	0.00E+00	6.41E-06
1	9.26e-07	0.00E+00	0.00E+00	5.00E-01	7.12E-06	0.00E+00	6.28E-06

(b) *H1+2* Landscape

Table C.6: Proportion of the search position types in the feasible region, averaged over 600 instances for each Δ value, with $k = 1$, $n = 20$, and $1 \leq CV < 2$.

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.25	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.5	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.75	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.95	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

(a) *H1* Landscape

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	9.54e-07	0.00E+00	0.00E+00	4.57E-01	4.32E-02	0.00E+00	0.00E+00
0.25	9.54e-07	0.00E+00	0.00E+00	4.98E-01	1.44E-03	0.00E+00	0.00E+00
0.5	9.54e-07	0.00E+00	0.00E+00	4.99E-01	3.40E-04	0.00E+00	0.00E+00
0.75	9.54e-07	0.00E+00	0.00E+00	4.99E-01	2.98E-04	0.00E+00	0.00E+00
0.95	9.54e-07	0.00E+00	0.00E+00	4.99E-01	2.95E-04	0.00E+00	0.00E+00
1	9.54e-07	0.00E+00	0.00E+00	4.99E-01	2.78E-04	0.00E+00	0.00E+00

(b) *H1+2* Landscape

Table C.7: Proportion of the search position types in the infeasible region, averaged over 600 instances for each Δ value, with $k = 0.4$, $n = 20$, and $0 < CV \leq 0.3$.

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.25	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.5	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.75	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.95	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

(a) *H1* Landscape

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	9.54e-07	0.00E+00	0.00E+00	4.72E-01	2.81E-02	0.00E+00	0.00E+00
0.25	9.54e-07	0.00E+00	0.00E+00	4.99E-01	1.09E-03	0.00E+00	0.00E+00
0.5	9.54e-07	0.00E+00	0.00E+00	4.99E-01	1.89E-04	0.00E+00	0.00E+00
0.75	9.54e-07	0.00E+00	0.00E+00	5.00E-01	1.70E-04	0.00E+00	0.00E+00
0.95	9.54e-07	0.00E+00	0.00E+00	5.00E-01	1.22E-04	0.00E+00	0.00E+00
1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	1.75E-04	0.00E+00	0.00E+00

(b) *H1+2* Landscape

Table C.8: Proportion of the search position types in the infeasible region, averaged over 600 instances for each Δ value, with $k = 0.4$, $n = 20$, and $0.3 < CV < 1$.

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.25	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.5	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.75	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.95	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

(a) *H1* Landscape

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	9.54e-07	0.00E+00	0.00E+00	4.30E-01	6.95E-02	0.00E+00	0.00E+00
0.25	9.54e-07	0.00E+00	0.00E+00	4.97E-01	2.28E-03	0.00E+00	0.00E+00
0.5	9.54e-07	0.00E+00	0.00E+00	4.99E-01	2.92E-04	0.00E+00	0.00E+00
0.75	9.54e-07	0.00E+00	0.00E+00	4.99E-01	2.96E-04	0.00E+00	0.00E+00
0.95	9.54e-07	0.00E+00	0.00E+00	4.99E-01	2.99E-04	0.00E+00	0.00E+00
1	9.54e-07	0.00E+00	0.00E+00	4.99E-01	3.44E-04	0.00E+00	0.00E+00

(b) *H1+2* Landscape

Table C.9: Proportion of the search position types in the infeasible region, averaged over 600 instances for each Δ value, with $k = 0.4$, $n = 20$, and $1 \leq CV < 2$.

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.25	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.5	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.75	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.95	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

(a) *H1* Landscape

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.25	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.5	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.75	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.95	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

(b) *H1+2* Landscape

Table C.10: Proportion of the search position types in the infeasible region, averaged over 600 instances for each Δ value, with $k = 1$, $n = 20$, and $0 < CV \leq 0.3$.

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.25	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.5	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.75	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.95	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

(a) *H1* Landscape

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.25	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.5	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.75	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.95	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

(b) *H1+2* Landscape

Table C.11: Proportion of the search position types in the infeasible region, averaged over 600 instances for each Δ value, with $k = 1$, $n = 20$, and $0.3 < CV < 1$.

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.25	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.5	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.75	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.95	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

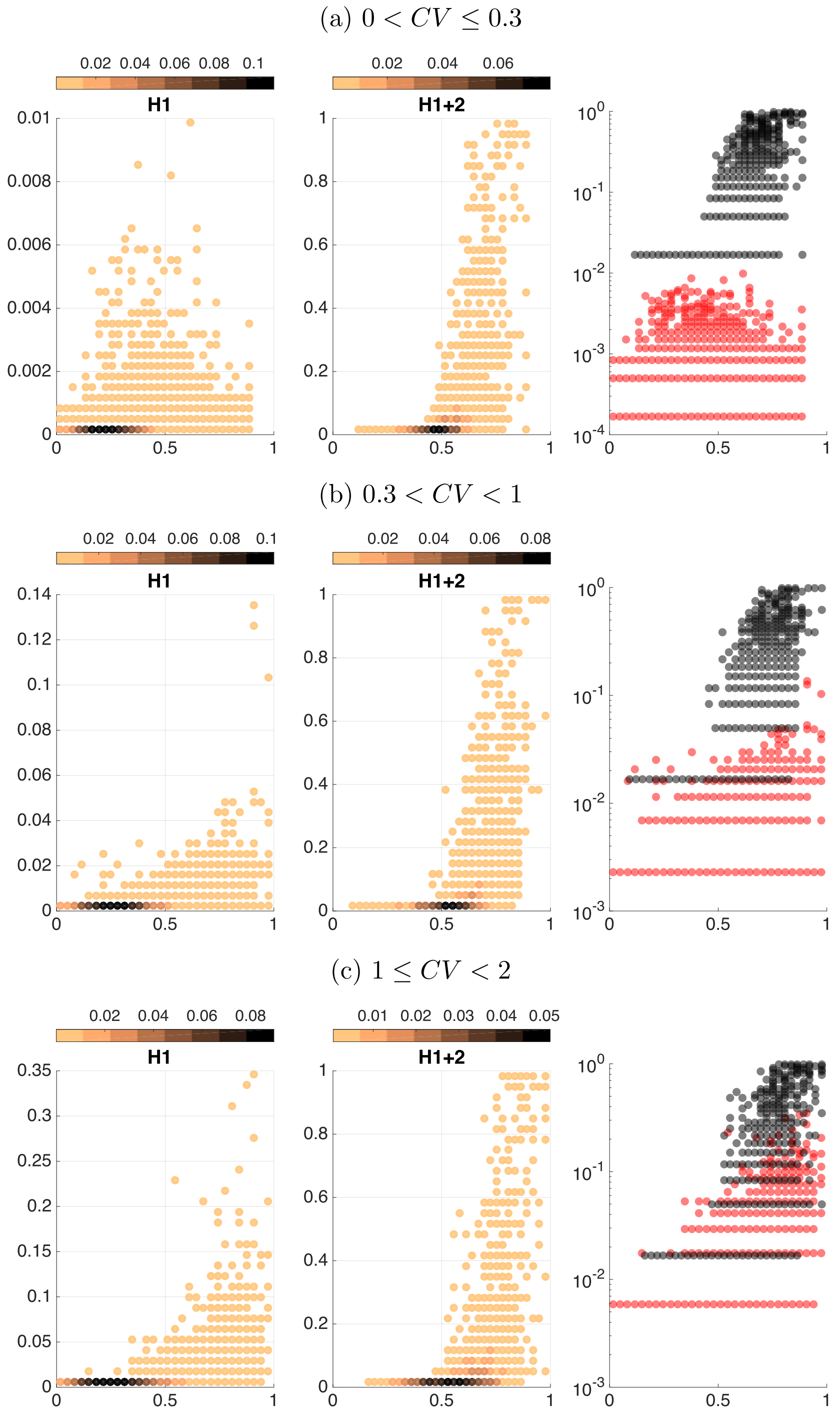
(a) *H1* Landscape

Δ	SLMIN	NSLMIN	IPLAT	LEDGE	SLOPE	NSLMAX	SLMAX
0.1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.25	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.5	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.75	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
0.95	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00
1	9.54e-07	0.00E+00	0.00E+00	5.00E-01	0.00E+00	0.00E+00	0.00E+00

(b) *H1+2* Landscape

Table C.12: Proportion of the search position types in the infeasible region, averaged over 600 instances for each Δ value, with $k = 1$, $n = 20$, and $1 \leq CV < 2$.

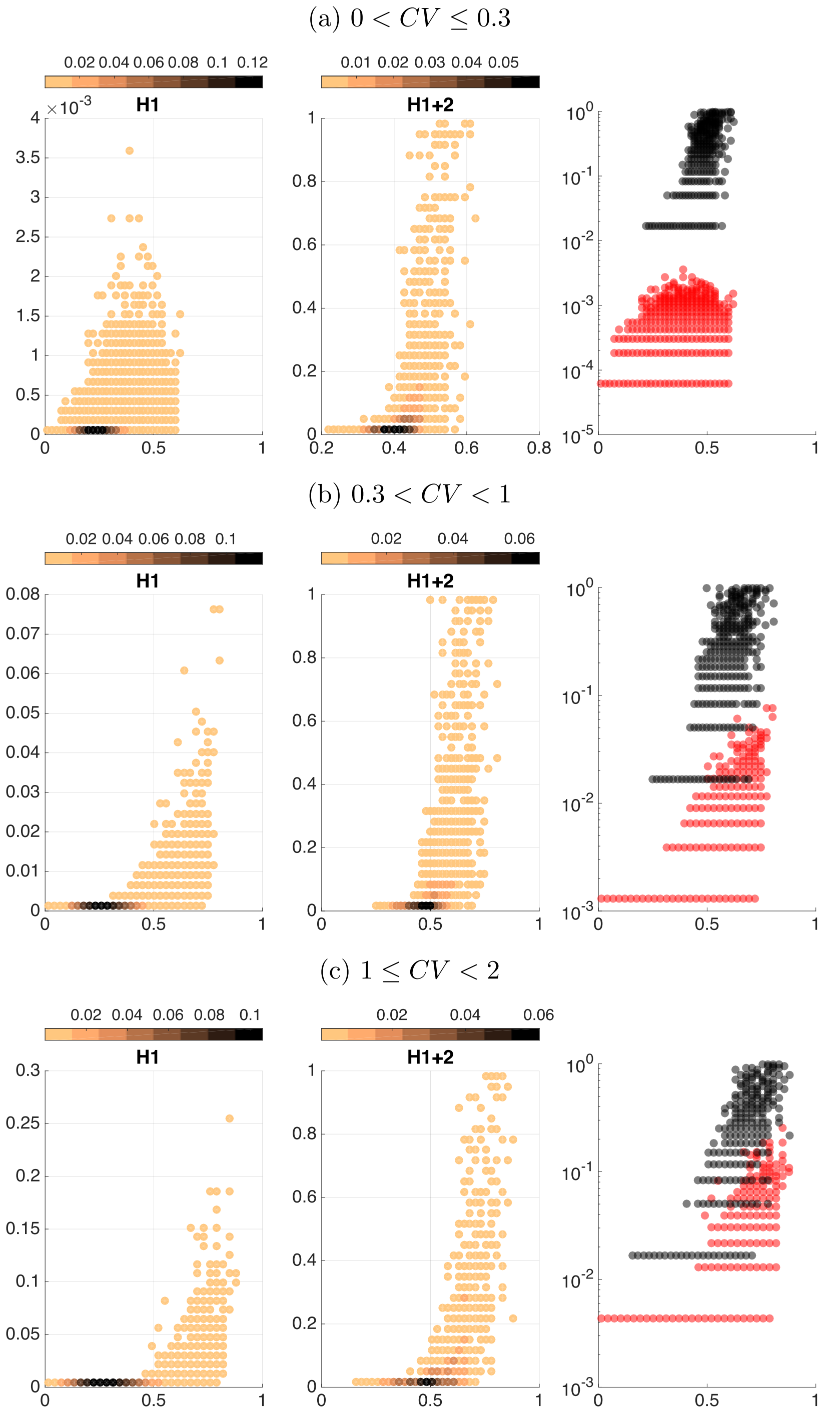
The following figures are used to support the observations/conclusions made in subsection 6.4.1.


(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

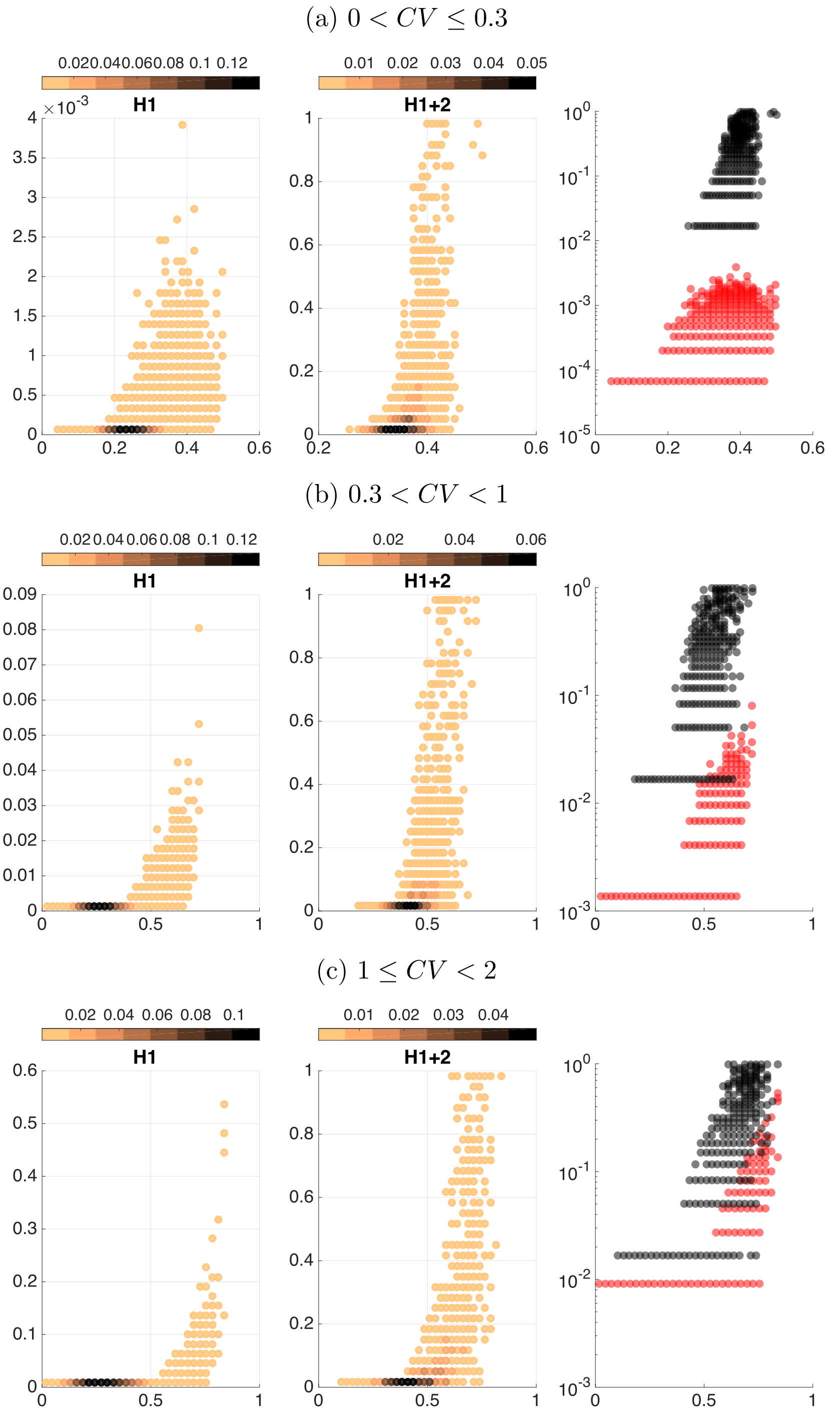
Figure C.1: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of Δ =0.1 knapsack of size n = 20 and k = 1. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.


(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

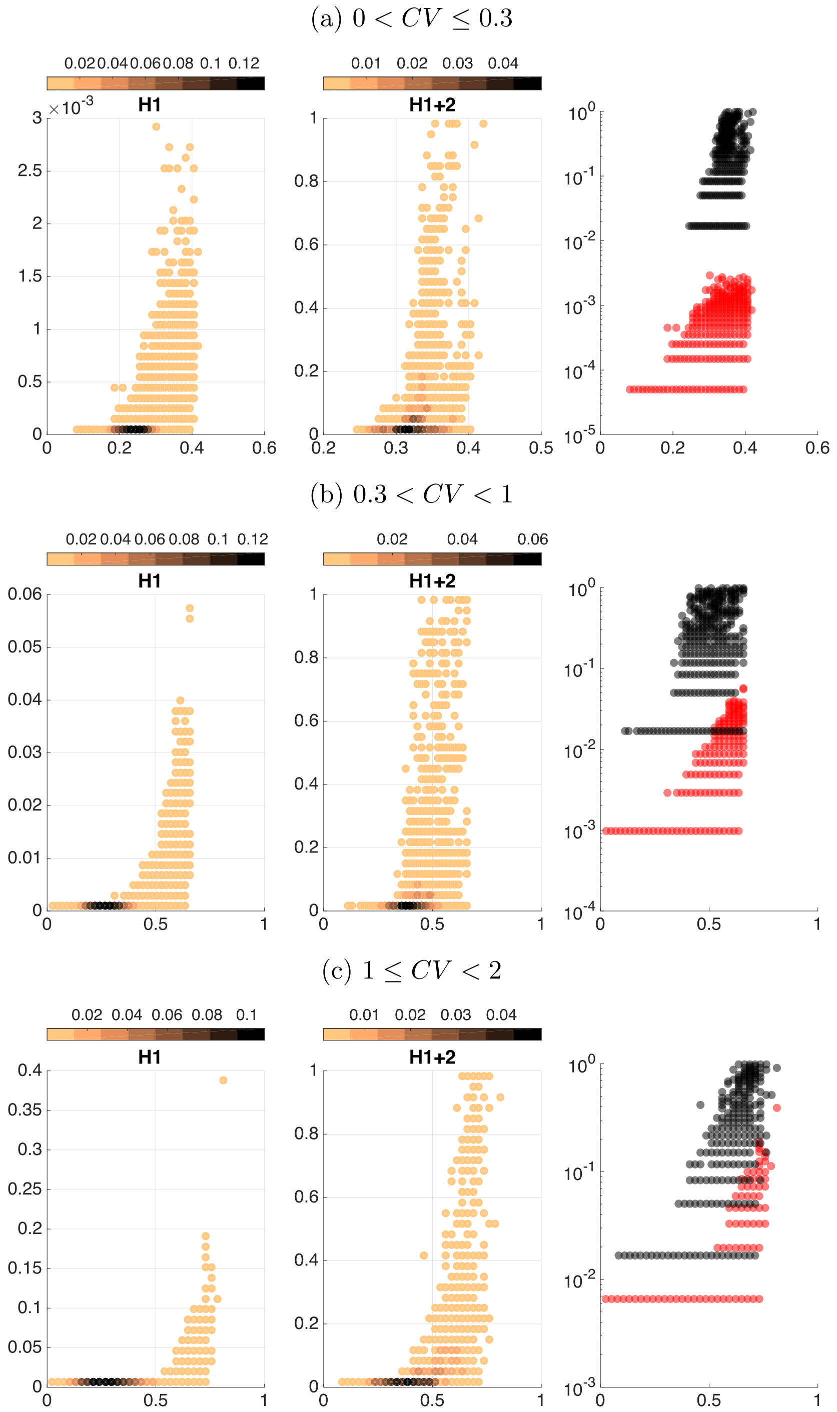
Figure C.2: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of Δ =0.25 knapsack of size $n = 20$ and $k = 1$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.


(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

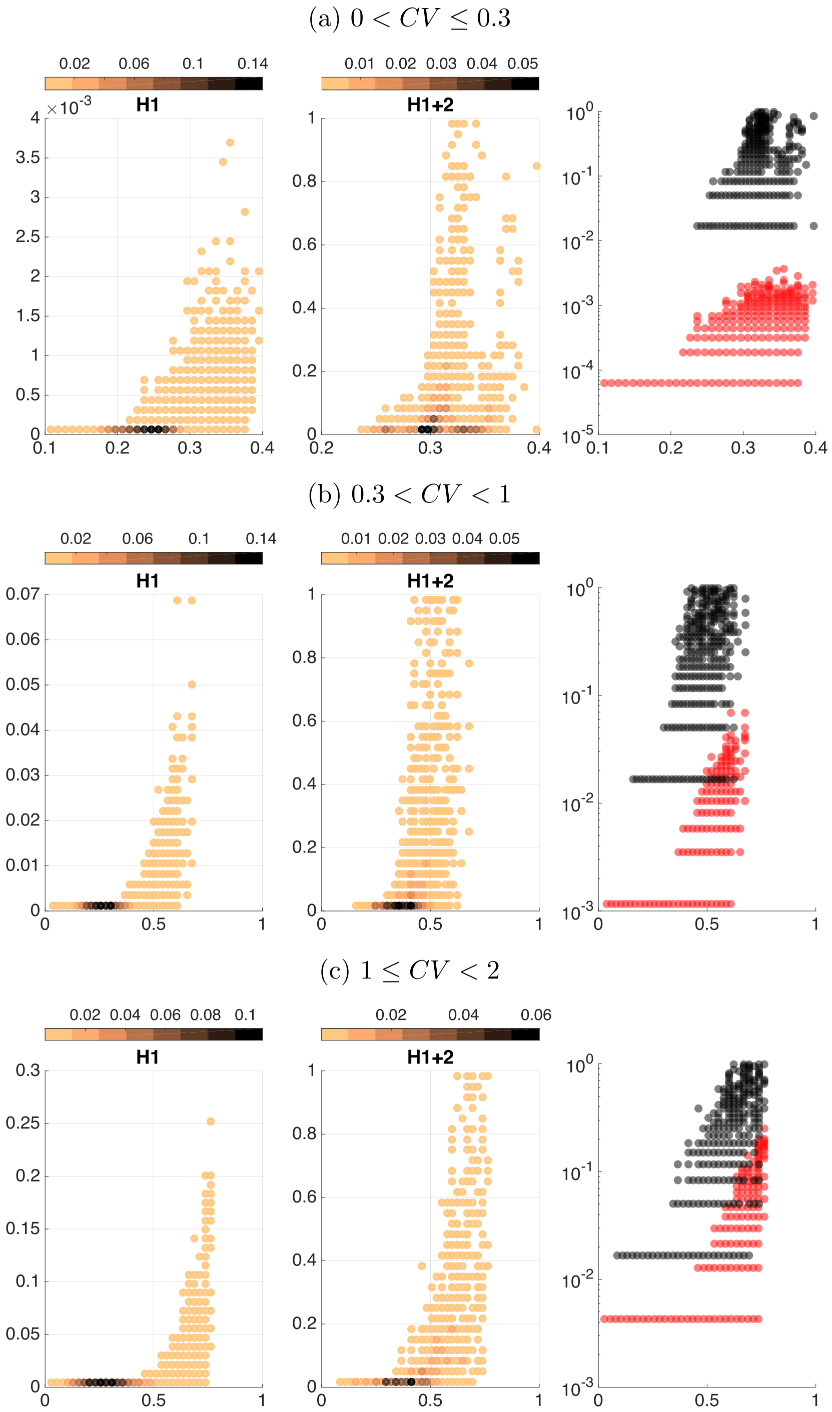
Figure C.3: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of Δ =0.5 knapsack of size $n = 20$ and $k = 1$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.


(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

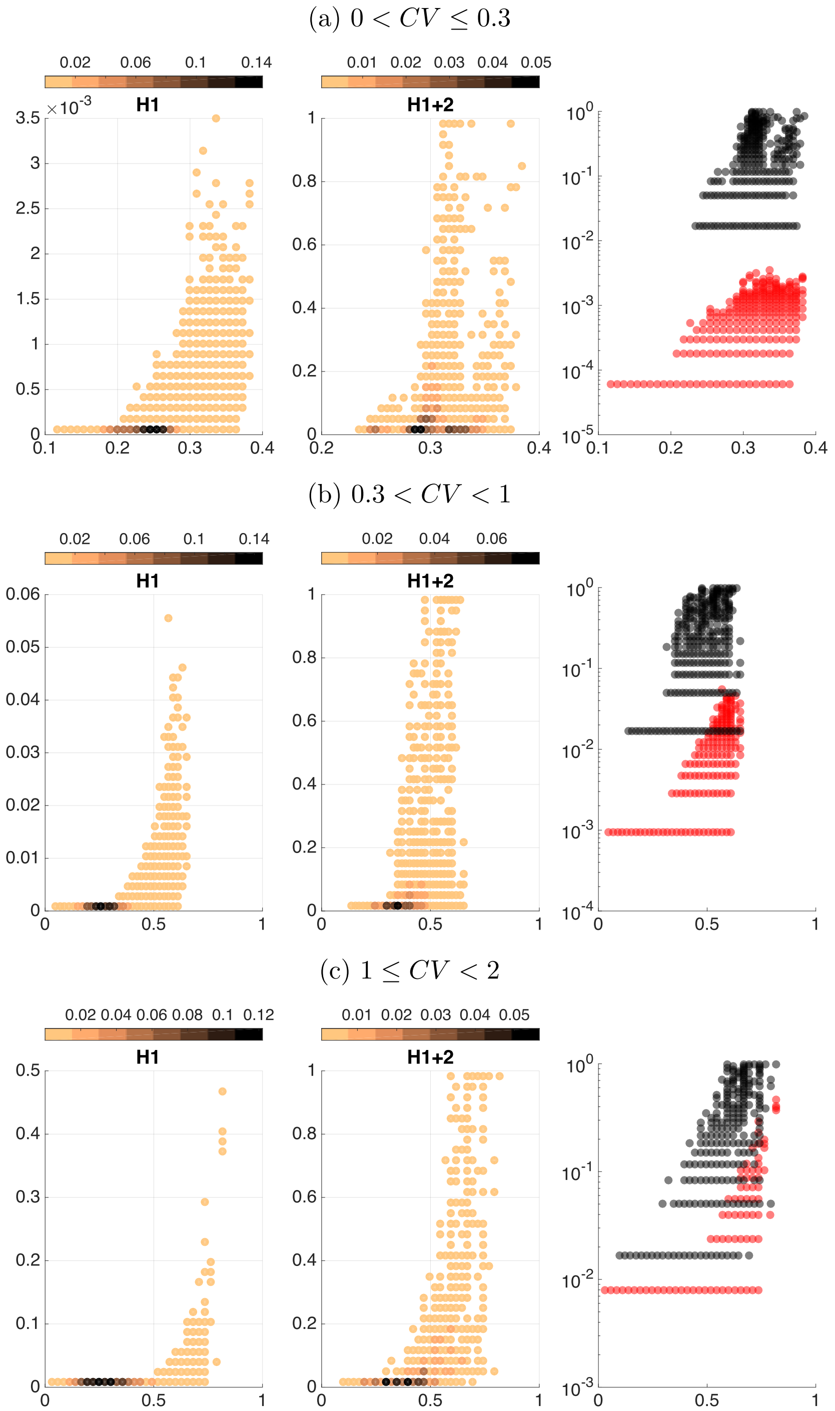
Figure C.4: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of Δ =0.75 knapsack of size n = 20 and k = 1. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.


(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

Figure C.5: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of Δ =0.95 knapsack of size n = 20 and k = 1. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.



(a) $0 < CV \leq 0.3$

(b) $0.3 < CV < 1$

(c) $1 \leq CV < 2$

Figure C.6: $|B(x^*)|/2^n$ (y-axis) against $f(x^*)/\sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}$ (x-axis), the colour bars show the frequency of each data point. The results show the fitness and basin size of all the optima found in 600 instances of Δ =1 knapsack of size $n = 20$ and $k = 1$. The rightmost plot is just the *H1* (red) and the *H1+2* (black) results (without the frequency) overlaid on the same axes (y-axis in log scale) to facilitate comparison of values.

BIBLIOGRAPHY

[1] *Journal of Heuristic Policies on Heuristic Search Research*, (June 1, 2016). http://www.springer.com/journal/10732.

[2] A. Agresti and B.A. Coull. Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician*, 52(2):119–126, 1998.

[3] L. Altenberg. Fitness distance correlation analysis: An instructive counterexample. In *proceedings of the Seventh International Conference on Genetic Algorithms*, pages 57–64, 1997.

[4] K. Alyahya and J.E. Rowe. Local optima and weight distribution in the number partitioning problem. In *Parallel Problem Solving from Nature - PPSN XIII*, volume 8672 of *Lecture Notes in Computer Science*, pages 862–871. 2014.

[5] K. Alyahya and J.E. Rowe. Phase transition and landscape properties of the number partitioning. In *14th European Conference on Evolutionary Computation in Combinatorial Optimisation, EvoCOP*, volume 8600 of *Lecture Notes in Computer Science*, pages 206–217. 2014.

[6] K. Alyahya and J.E. Rowe. Landscape properties of the 0-1 knapsack problem. In *proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 1343–1344, 2015.

[7] K. Alyahya and J.E. Rowe. Simple random sampling estimation of the number of local optima. In *Parallel Problem Solving from Nature - PPSN XIV*, volume 9921 of *Lecture Notes in Computer Science*, pages 932–941. 2016.

[8] R.S. Barr, B.L. Golden, J.P. Kelly, M.G.C. Resende, and W.R. Stewart. Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics*, 1(1):9–32, 1995.

[9] Y. Borenstein and R. Poli. Information landscapes. In *proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 1515–1522, 2005.

[10] Y. Borenstein and R. Poli. Information landscapes and problem hardness. In *proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 1425–1431, 2005.

[11] C. Borgs, J. Chayes, and B. Pittel. Phase transition and finite-size scaling for the integer partitioning problem. *Random Structures & Algorithms*, 19(3-4):247–288, 2001.

[12] L.D. Brown, T.T. Cai, and A. DasGupta. Interval estimation for a binomial proportion. *Statistical Science*, 16(2):101–117, 2001.

[13] E.K. Burke, T. Curtois, G. Kendall, M. Hyde, G. Ochoa, and J.A. Vazquez-Rodriguez. Towards the decathlon challenge of search heuristics. In *proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, pages 2205–2208, 2009.

[14] L. Caccetta and A. Kulanoot. Computational aspects of hard knapsack problems. *Nonlinear Analysis: Theory, Methods & Applications*, 47(8):5547–5558, 2001.

[15] M.C. Cario, J.J. Clifford, R.R. Hill, I. Yang, K. Yang, and C.H. Reilly. An investigation of the relationship between problem characteristics and algorithm performance: a case study of the GAP. *IIE Transactions*, 34(3):297–312, 2002.

[16] R. Caruana and M. Mullin. Estimating the number of local minima in big, nasty search spaces. In *proceedings of IJCAI-99 Workshop on Statistical Machine Learning for Large-Scale Optimization*, 1999.

[17] F. Chicano, F. Daolio, G. Ochoa, S. Vérel, M. Tomassini, and E. Alba. Local optima networks, landscape autocorrelation and heuristic search performance. *Parallel Problem Solving from Nature - PPSN XII*, pages 337–347, 2012.

[18] F. Chicano, L.D. Whitley, and E. Alba. A methodology to find the elementary landscape decomposition of combinatorial optimization problems. *Evol. Comput.*, 19(4):597–637, 2011.

[19] A. Clauset, C.R. Shalizi, and M.E.J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

[20] P. Collard, S. Vérel, and M. Clergue. Local search heuristics: Fitness cloud versus fitness landscape. In *ECAI*, pages 973–974, 2004.

[21] Z.J. Czech. Statistical measures of a fitness landscape for the vehicle routing problem. In *IEEE International Symposium on Parallel and Distributed Processing*, pages 1–8, 2008.

[22] J. Czogalla and A. Fink. Fitness landscape analysis for the no-wait flow-shop scheduling problem. *Journal of Heuristics*, 18(1):25–51, 2012.

[23] F. Daolio, A. Liefooghe, S. Verel, H. Aguirre, and K. Tanaka. Global vs local search on multi-objective NK-landscapes: Contrasting the impact of problem features. In *proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 369–376, 2015.

[24] F. Daolio, S. Verel, G. Ochoa, and M. Tomassini. Local optima networks of the quadratic assignment problem. In *IEEE Congress on Evolutionary Computation*, pages 1–8, 2010.

[25] Y. Davidor. Epistasis variance: A viewpoint on ga-hardness. In *FOGA*, pages 23–35, 1990.

[26] M. Englert, H. Röglin, and B. Vöcking. Worst case and probabilistic analysis of the 2-opt algorithm for the TSP. *Algorithmica*, 68(1):190–264, 2013.

[27] A. Eremeev and C.R. Reeves. Non-parametric estimation of properties of combinatorial landscapes. In *Applications of Evolutionary Computing*, volume 2279 of *Lecture Notes in Computer Science*, pages 31–40. 2002.

[28] R.M. Feldman and C. Valdez-Flores. *Applied probability and stochastic processes.* Springer Science & Business Media, 2009.

[29] F.F. Ferreira and J.F. Fontanari. Probabilistic analysis of the number partitioning problem. *Journal of Physics A: Mathematical and General*, 31(15):3417, 1998.

[30] C. Fonlupt, D. Robilliard, and P. Preux. A bit-wise epistasis measure for binary search spaces. In *proceedings of the 5th International Conference on Parallel Problem Solving from Nature - PPSN V*, pages 47–56, 1998.

[31] J.F. Fontanari. A statistical analysis of the knapsack problem. *Journal of Physics A: Mathematical and General*, 28(17):4751, 1995.

[32] J. Frank, P. Cheeseman, and J. Stutz. When gravity fails: Local search topology. *Journal of Artificial Intelligence Research*, 7:249–281, 1997.

[33] Y. Fu. The use and abuse of statistical mechanics in computational complexity. In *Lectures in the Sciences of Complexity*, volume 1, pages 815–826, 1989.

[34] G. Gallo, P.L. Hammer, and B. Simeone. Quadratic knapsack problems. In *Combinatorial Optimization*, volume 12 of *Mathematical Programming Studies*, pages 132–149. 1980.

[35] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Series of books in the mathematical sciences. W. H. Freeman, 1979.

[36] J. Garnier and L. Kallel. How to detect all maxima of a function. *Theoretical aspects of evolutionary computing*, pages 343–370, 2001.

[37] J. Garnier and L. Kallel. Efficiency of local search with multiple local optima. *SIAM J. Discret. Math.*, 15(1):122–141, January 2002.

[38] I.P. Gent and T. Walsh. Analysis of heuristics for number partitioning. *Computational Intelligence*, 14(3):430–451, 1998.

[39] D.E. Goldberg. *Genetic algorithms in search, optimization, and machine learning.* Artificial Intelligence. Addison-Wesley, 1989.

[40] J. Gottlieb. On the feasibility problem of penalty-based evolutionary algorithms for knapsack problems. In *Applications of Evolutionary Computing*, volume 2037 of *Lecture Notes in Computer Science*, pages 50–59. 2001.

[41] L.K. Grover. Local search and the local structure of NP-complete problems. *Operations Research Letters*, 12(4):235–243, October 1992.

[42] L.K. Grover. Local search and the local structure of NP-complete problems. *Operations Research Letters*, 12(4):235–243, 1992.

[43] D.A. Grundel, P.A. Krokhmal, C.A.S. Oliveira, and P.M. Pardalos. On the number of local minima for the multidimensional assignment problem. *Journal of Combinatorial Optimization*, 13(1):1–18, 2007.

[44] J. Hallam and A. Prügel-Bennett. Large barrier trees for studying search. *IEEE Transactions on Evolutionary Computation*, 9(4):385–397, Aug 2005.

[45] A.K. Hartmann and M. Weigt. *Phase Transitions in Combinatorial Optimization Problems*. John Wiley & Sons, 2006.

[46] J. He, C. Reeves, C. Witt, and X. Yao. A note on problem difficulty measures in black-box optimization: Classification, realizations and predictability. *Evolutionary Computation*, 15(4):435–443, 2007.

[47] L. Hernando, A. Mendiburu, and J.A. Lozano. An evaluation of methods for estimating the number of local optima in combinatorial optimization problems. *Evolutionary computation*, 21(4):625–658, 2013.

[48] J.L. Hintze and R.D. Nelson. Violin plots: A box plot-density trace synergism. *The American Statistician*, 52(2):181–184, 1998.

[49] J.N. Hooker. Testing heuristics: We have it all wrong. *Journal of Heuristics*, 1(1):33–42, 1995.

[50] H.H. Hoos. *Stochastic local search-methods, models, applications*. PhD thesis, TU Darmstadt, FB Informatik, Darmstadt, Germany, 1998.

[51] H.H. Hoos and T. Stützle. *Stochastic Local Search: Foundations & Applications*. The Morgan Kaufmann Series in Artificial Intelligence. Elsevier Science, 2005.

[52] W. Hordijk. A measure of landscapes. *Evolutionary Computation*, 4(4):335–360, 1996.

[53] J. Horn and D.E. Goldberg. Genetic algorithm difficulty and the modality of fitness landscapes. In *Foundations of Genetic Algorithms 3*, pages 243–269. 1995.

[54] J. Inoue. Statistical mechanics of the multi-constraint continuous knapsack problem. *Journal of Physics A: Mathematical and General*, 30(4):1047, 1997.

[55] T. Jansen. On classifications of fitness functions. *Theoretical aspects of evolutionary computing*, pages 371–386, 2001.

[56] T. Jones. *Evolutionary Algorithms, Fitness Landscapes and Search.* PhD thesis, The University of New Mexico, 1995.

[57] T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *proceedings of the 6th International Conference on Genetic Algorithms*, pages 184–192, 1995.

[58] L. Kallel, B. Naudts, and C.R. Reeves. Properties of fitness functions and search landscapes. *Theoretical aspects of evolutionary computing*, pages 175–206, 2001.

[59] S. Khuri, T. Bäck, and J. Heitkötter. The zero/one multiple knapsack problem and genetic algorithms. In *proceedings of the 1994 ACM Symposium on Applied Computing*, pages 188–193, 1994.

[60] R. E. Korf. A complete anytime algorithm for number partitioning. *Artificial Intelligence*, 106(2):181–203, 1998.

[61] E. Korutcheva, M. Opper, and B. Lopez. Statistical mechanics of the knapsack problem. *Journal of Physics A: Mathematical and General*, 27(18):L645, 1994.

[62] J. Liu, H.A. Abbass, D.G. Green, and W. Zhong. Motif difficulty (md): a predictive measure of problem difficulty for evolutionary algorithms using network motifs. *Evolutionary Computation*, 20(3):321–347, 2012.

[63] G. Lu, J. Li, and X. Yao. *the 11th European Conference on Evolutionary Computation in Combinatorial Optimization, EvoCOP*, chapter Fitness-Probability Cloud and a Measure of Problem Hardness for Evolutionary Algorithms, pages 108–117. 2011.

[64] S. Martello, D. Pisinger, and P. Toth. Dynamic programming and strong bounds for the 0-1 knapsack problem. *Manage. Sci.*, 45(3):414–424, March 1999.

[65] K.E. Mathias and L.D. Whitley. Transforming the search space with gray coding. In *proceedings of the First IEEE Conference on Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence*, pages 513–518, 1994.

[66] D.C. Mattfeld, C. Bierwirth, and H. Kopfer. A search space analysis of the job shop scheduling problem. *Annals of Operations Research*, 86:441–453, 1999.

[67] S. Mertens. Phase transition in the number partitioning problem. *Physical review letters*, 81(20):4281–4284, 1998.

[68] S. Mertens. A physicist's approach to number partitioning. *Theoretical Computer Science*, 265(1-2):79–108, 2001.

[69] P. Merz. Advanced fitness landscape analysis and the performance of memetic algorithms. *Evolutionary Computation*, 12(3):303–325, 2004.

[70] P. Merz and B. Freisleben. New ideas in optimization. chapter Fitness Landscapes and Memetic Algorithm Design, pages 245–260. 1999.

[71] P. Merz and B. Freisleben. Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Transactions on Evolutionary Computation*, 4(4):337–352, 2000.

[72] Z. Michalewicz and J. Arabas. Genetic algorithms for the 0/1 knapsack problem. In *Methodologies for Intelligent Systems*, volume 869 of *Lecture Notes in Computer Science*, pages 134–143. 1994.

[73] C. Moore and S. Mertens. *The Nature of Computation*. OUP Oxford, 2011.

[74] C.L. Müller and I.F. Sbalzarini. Global characterization of the CEC 2005 fitness landscapes using fitness-distance analysis. In *Applications of Evolutionary Computation: EvoApplications*, pages 294–303, 2011.

[75] B. Naudts. *Measuring GA-hardness*. PhD thesis, University Of Antwerp, Belgium, 1998.

[76] B. Naudts and L. Kallel. A comparison of predictive measures of problem difficulty in evolutionary algorithms. *Evolutionary Computation, IEEE Transactions on*, 4(1):1 –15, apr 2000.

[77] B. Naudts and A. Verschoren. Epistasis and deceptivity. In *Bulletin of the Belgian Mathematical Society - Simon StevinS*, 1999.

[78] G. Ochoa, M. Tomassini, and C. Vérel, S.and Darabos. A study of NK landscapes' basins and local optima networks. In *proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 555–562, 2008.

[79] A.L. Olsen. Penalty functions and the knapsack problem. In *proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, pages 554–558, 1994.

[80] A.J. Parkes. Clustering at the phase transition. In *proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence*, pages 340–345, 1997.

[81] A.M. Pires and C. Amado. Interval estimators for a binomial proportion: Comparison of twenty methods. *REVSTAT–Statistical Journal*, 6(2):165–197, 2008.

[82] D. Pisinger. Core problems in knapsack algorithms. *Operations Research*, 47(4):570–575, 1999.

[83] D. Pisinger. Where are the hard knapsack problems? *Comput. Oper. Res.*, 32(9):2271–2284, 2005.

[84] D. Pisinger. The quadratic knapsack problema survey. *Discrete Applied Mathematics*, 155(5):623–648, 2007.

[85] E. Pitzer and M. Affenzeller. A comprehensive survey on fitness landscape analysis. *Recent Advances in Intelligent Engineering Systems*, pages 161–191, 2012.

[86] E. Pitzer, M. Affenzeller, and A. Beham. A closer look down the basins of attraction. In *UK Workshop on Computational Intelligence*, pages 1–6, 2010.

[87] R. Poli and L. Vanneschi. Fitness-proportional negative slope coefficient as a hardness measure for genetic algorithms. In *proceedings of the 9th annual conference on Genetic and evolutionary computation*, GECCO '07, pages 1335–1342, 2007.

[88] A. Prügel-Bennett and M.-H. Tayarani-N. Maximum satisfiability: Anatomy of the fitness landscape for a hard combinatorial optimization problem. *IEEE Transactions on Evolutionary Computation*, 16(3):319–338, 2012.

[89] M. Qasem and A. Prügel-Bennett. Learning the large-scale structure of the max-sat landscape using populations. *IEEE Transactions on Evolutionary Computation*, 14(4):518 –529, aug. 2010.

[90] C.R. Reeves. Direct statistical estimation of GA landscape properties. *Foundations of Genetic Algorithms*, 6:91–107, 2001.

[91] C.R. Reeves and A.V. Eremeev. Statistical analysis of local search landscapes. *Journal of the Operational Research Society*, 55(7):687–693, 2004.

[92] C.R. Reeves and J.E. Rowe. *Genetic Algorithms - Principles and Perspectives: A Guide to GA Theory.* Operations Research/Computer Science Interfaces Series. Springer, 2002.

[93] C.M. Reidys and P.F. Stadler. Neutrality in fitness landscapes. *Applied Mathematics and Computation*, 117(2):321–350, 2001.

[94] C.M. Reidys and P.F. Stadler. Combinatorial landscapes. *SIAM review*, 44(1):3–54, 2002.

[95] J.M.W. Rhys. A selection problem of shared fixed costs and network flows. *Management Science*, 17(3):200–207, 1970.

[96] H Richter and A. Engelbrecht, editors. *Recent advances in the theory and application of fitness landscapes.* Springer, 2014.

[97] T. Sasamoto. Phase transitions of subset sum and shannon's limit in source coding. *Physica A: Statistical Mechanics and its Applications*, 321(12):369–374, 2003.

[98] T. Sasamoto, T. Toyoizumi, and H. Nishimori. Statistical mechanics of an NP-complete problem: subset sum. *Journal of Physics A: Mathematical and General*, 34(44):9555, 2001.

[99] J. Slaney and T. Walsh. Backbones in optimization and approximation. In *proceedings of the 17th international joint conference on Artificial intelligence*, IJCAI'01, pages 254–259, 2001.

[100] J. Slaney and T. Walsh. Backbones in optimization and approximation. In *International Joint Conference on Artificial Intelligence*, pages 254–259, 2001.

[101] T. Smith, P. Husbands, and M. O'Shea. Fitness landscapes and evolvability. *Evolutionary computation*, 10(1):1–34, 2002.

[102] K. Smith-Miles and L. Lopes. Measuring instance difficulty for combinatorial optimization problems. *Computers Operations Research*, 39(5):875–889, 2012.

[103] P.F. Stadler. Towards a theory of landscapes. In *Complex Systems and Binary Networks*, volume 461-461 of *Lecture Notes in Physics*, pages 78–163. 1995.

[104] P.F. Stadler. Fitness landscapes. In *Biological Evolution and Statistical Physics*, volume 585 of *Lecture Notes in Physics*, pages 183–204. 2002.

[105] P.F. Stadler, W. Hordijk, and J.F. Fontanari. Phase transition and landscape statistics of the number partitioning problem. *Physical Review E*, 67(5):056701, 2003.

[106] P.F. Stadler and W. Schnabl. The landscape of the traveling salesman problem. *Physics Letters A*, 161(4):337–344, 1992.

[107] P.F. Stadler and C.R. Stephens. Landscapes and effective fitness. *Comments on Theoretical Biology*, 8(4-5):389–431, 2002.

[108] J. Swan, S. Adriaensen, M. Bishr, E.K. Burke, J.A. Clark, P. De Causmaecker, J. Durillo, K. Hammond, E. Hart, C.G. Johnson, Z.A. Kocsis, B. Kovitz, K. Krawiec, S. Martin, J.J. Merelo, L.L. Minku, E. Ozcan, G.L. Pappa, E. Pesch, P. Garcia-Sanchez, A. Schaerf, K. Sim, J. Smith, T. Stutzle, S. VoB, S. Wagner, and X. Yao. A research agenda for metaheuristic standardization. In *MIC 2015: The 11th Metaheuristics International Conference*, 2015.

[109] K. Srensen. Metaheuristicsthe metaphor exposed. *International Transactions in Operational Research*, 22(1):3–18, 2015.

[110] E.G. Talbi. *Metaheuristics: From Design to Implementation*. Wiley Series on Parallel and Distributed Computing. Wiley, 2009.

[111] J. Tavares, F.B. Pereira, and E. Costa. Multidimensional knapsack problem: A fitness landscape analysis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(3):604–616, 2008.

[112] M.-H. Tayarani-N. and A. Prügel-Bennett. On the landscape of combinatorial optimization problems. *IEEE Transactions on Evolutionary Computation*, 18(3):420–434, 2014.

[113] M.-H. Tayarani-N. and A. Prügel-Bennett. An analysis of the fitness landscape of travelling salesman problem. *Evolutionary Computation*, pages 1–38, 2015.

[114] M.-H. Tayarani-N. and A. Prügel-Bennett. Anatomy of the fitness landscape for dense graph-colouring problem. *Swarm and Evolutionary Computation*, 22:47–65, 2015.

[115] M.-H. Tayarani-N. and A. Prügel-Bennett. Quadratic assignment problem: a landscape analysis. *Evolutionary Intelligence*, 8(4):165–184, 2015.

[116] M.-H. Tayarani-Najaran. Analysis of the fitness landscape for the class of combinatorial optimisation problems. May 2013.

[117] C.A. Tovey. Hill climbing with multiple local optima. *SIAM Journal on Algebraic Discrete Methods*, 6(3):384–393, 1985.

[118] M.F. Triola. *Elementary Statistics*. Pearson, 12th edition, 2012.

[119] L. Vanneschi, M. Clergue, P. Collard, M. Tomassini, and S. Vérel. Fitness clouds and problem hardness in genetic programming. In *proceedings of the Genetic and Evolutionary Computation- GECCO 2004*, pages 690–701, 2004.

[120] V.K. Vassilev, T.C. Fogarty, and J.F. Miller. Information characteristics and the structure of landscapes. *Evolutionary Computation*, 8(1):31–60, 2000.

[121] S. Vérel, F. Daolio, G. Ochoa, M. Tomassini, et al. Local optima networks with escape edges. In *procedings of International Conference on Artificial Evolution (EA-2011)*, pages 10–23, 2011.

[122] S. Verel, G. Ochoa, and M. Tomassini. The connectivity of NK landscapes basins: A network analysis. *Artificial Life*, 11:648, 2008.

[123] M.D. Vose and A.H. Wright. Stability of vertex fixed points and applications. *Foundations of genetic algorithms 3*, 3:103, 1995.

[124] J.-P. Watson. An introduction to fitness landscape analysis and cost models for local search. In *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, pages 599–623. 2010.

[125] J.-P. Watson, J.C. Beck, A.E. Howe, and L.D. Whitley. Problem difficulty for tabu search in job-shop scheduling. *Artificial Intelligence*, 143(2):189–217, 2003.

[126] E. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63(5):325–336, 1990.

[127] T. Weise, M. Zapf, R. Chiong, and A. Nebro. Why is optimization difficult? In *Nature-Inspired Algorithms for Optimisation*, volume 193 of *Studies in Computational Intelligence*, pages 1–50. 2009.

[128] D. Whitley and F. Chicano. *Learning and Intelligent Optimization: 6th International Conference, LION 6*, chapter Quasi-elementary Landscapes and Superpositions of Elementary Landscapes, pages 277–291. 2012.

[129] D. Whitley and F. Chicano. *the 14th European Conference on Evolutionary Computation in Combinatorial Optimisation, EvoCOP*, chapter Elementary Landscape Decomposition of the Hamiltonian Path Optimization Problem, pages 121–132. 2014.

[130] D. Whitley, A.M. Sutton, and A.E. Howe. Understanding elementary landscapes. In *proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 585–592, 2008.

[131] D. Whitley, A.M. Sutton, and A.E. Howe. Understanding elementary landscapes. In *proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, pages 585–592, 2008.

[132] S.W. Wilson. GA-easy does not imply steepest-ascent optimizable. In *proceedings of the Fourth International Conference on Genetic Algorithms*, pages 85–91, 1991.

[133] C. Witt. *the 22nd Annual Symposium on Theoretical Aspects of Computer Science*, chapter Worst-Case and Average-Case Approximations by Simple Randomized Search Heuristics, pages 44–56. Springer, 2005.

[134] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[135] S. Wright. The roles of mutation, inbreeding, crossbreeding and selection in evolution. In *proceedings of the sixth international congress on genetics*, volume 1, pages 356–366, 1932.

[136] X. Yao, Y. Liu, and G. Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3(2):82–102, 1999.

[137] M. Yokoo. Why adding more constraints makes a problem easier for hill-climbing algorithms: Analyzing landscapes of CSPs. In *Principles and Practice of Constraint Programming-CP97*, pages 356–370. 1997.

[138] H. Yoshizawa and S. Hashimoto. Landscape analyses and global search of knapsack problems. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 3, pages 2311–2315, 2000.

[139] B. Yossi and R. Poli. Information landscapes and the analysis of search algorithms. In *proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 1287–1294, 2005.

[140] W. Zhang. Phase transitions and backbones of 3-SAT and maximum 3-SAT. In *Principles and Practice of Constraint Programming - CP 2001*, volume 2239 of *Lecture Notes in Computer Science*, pages 153–167. 2001.

[141] D. Zwillinger and S. Kokoska. *CRC Standard Probability and Statistics Tables and Formulae.* CRC Press, 1999.