Landscape properties of the 0-1 knapsack problem
Alyahya, Khulood; Rowe, Jonathan

DOI:
10.1145/2739482.2764688

License:
None: All rights reserved

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (Harvard):

Link to publication on Research at Birmingham portal

General rights
Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

• Users may freely distribute the URL that is used to identify this publication.
• Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
• User may use extracts from the document in line with the concept of ‘fair dealing’ under the Copyright, Designs and Patents Act 1988 (?)
• Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy
While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.
Landscape Properties of the 0-1 Knapsack Problem

Khulood Alyahya
School of Computer Science
University of Birmingham, B15 2TT, UK
kya020@cs.bham.ac.uk

Jonathan E. Rowe
School of Computer Science
University of Birmingham, B15 2TT, UK
J.E.Rowe@cs.bham.ac.uk

ABSTRACT
This paper studies two landscapes of different instances of the 0-1 knapsack problem. The instances are generated randomly from varied weight distributions. We show that the variation of the weights can be used to guide the selection of the most suitable local search operator for a given instance.

Categories and Subject Descriptors
F.2.0 [Theory of Computation]: Analysis of algorithms and problem complexity.

Keywords
Fitness Landscapes, Instance Difficulty, Constraint Handling

1. THE 0-1 KNAPSACK PROBLEM
The classical 0-1 knapsack problem (0-1 KP) is defined as follows: 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 $f(x) = \sum_{i=1}^{n} x_i p_i$, subject to $\sum_{i=1}^{n} x_i w_i \leq C$, where $x \in \{0,1\}^n$, $C = \lambda \sum_{i=1}^{n} w_i$, and $0 \leq \lambda \leq 1$. 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.

To avoid trivial instances, we only study instances where $w_i \leq C$ for all $i = 1, \ldots, n$, and $\lambda < 1$.

We generated instances randomly with different correlation between the fitness function coefficient $p_i$ and constraint coefficient $w_i$ as in [4]. Based on the value of the correlation coefficient, an instance is classified as uncorrelated, weakly correlated, strongly correlated, inverse strongly correlated, or subset sum. For each instance type we varied the tightness of the capacity constraint by setting $\lambda$ value between 0.1 to 0.9 with 0.1 step interval. Also, for each instance type, weights are drawn randomly from five different discrete probability distributions: uniform, normal, negatively skewed, positively skewed and bimodal distribution with peaks at both ends. The variation of the weights is then measured by their coefficient of variation ($CV = \sigma/\mu$).

An infeasible solution $x$ that violates the given constraint is penalised by a value $Pen(x) > 0$, while $Pen(x) = 0$ for a feasible solution $x$. The fitness functions after adding the penalty term is $f(x) = \sum_{i=1}^{n} x_i p_i - Pen(x)$. In this paper, we use the linear penalty function from [3] where the value of the penalty grows linearly with respect to the degree of constraint violation. We also add the term $\sum_{i=1}^{n} p_i$ to the penalty function as an offset term that ensures that all infeasible solutions achieve lower fitness values than all feasible solutions [2]. The penalty after adding the offset term:

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

where $\rho = \max \{ p_i/w_i \mid i = 1, \ldots, n \}$. We use this penalty function with all instance types except for 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. We therefore choose this simple penalty function to use with subset sum instances:

$$Pen(x) = \sum_{i=1}^{n} x_i w_i - C$$  (2)

2. LANDSCAPE OF THE 0-1 KP
Landscape analysis allows us to study an optimisation problem in connection with a neighbourhood structure defined over the search space. The aim of such analysis is to lead to a better understanding of the optimisation problem structure. The gained insights could then be used to guide the design or selection of the best search operators or parameter values to use. In this paper we study two landscapes of the 0-1 KP induced by the Hamming 1 ($H1$) and Hamming 1+2 ($H1+2$) neighbourhood operators. The $H1$ operator is the simple 1-bit flip mutation operator, the neighbourhood size is thus $|N(x)| = n$. The neighbourhood of the $H1+2$ operator includes the Hamming one neighbours plus the Hamming two neighbours of the current solution. The neighbourhood size for this operator is $|N(x)| = n + (n(n-1)/2)$. Only small problem sizes were considered to allow exhaustive enumeration of the entire search space ($n = 14, 18$).

2.1 Number of Optima
A very large difference in the number of local optima was found between the two landscape. The $H1+2$ landscape has
less number of local optima in all instance types and all values of $CV$ and $\lambda$, with the largest difference being found in instances with small $CV$ values. The number of optima in the $H1$ landscape was found to be highly and negatively correlated with the $CV$ of the weights in the middle region between very high and very low constraints ($0.3 \leq \lambda \leq 0.7$) as depicted in Figure 1. In the $H1+2$ landscape, only inverse strongly correlated instances were found to have a strong correlation between the optima number and the $CV$ with high and positive correlation across all constraint levels. The correlation in the subset sum and strongly correlated instances is strong and positive but only in the very constrained and very low constrained regions ($\lambda \leq 0.2, \lambda \geq 0.8$).

2.2 Basins of Attraction

We define the basin of attraction of an optimum as the set of points that leads to it after applying steepest ascent to them. The neighbours of a point are evaluated in an increasing order and the first best improving neighbour is always selected. The correlation between the optimum basin size and fitness has been conjectured to be related to problem difficulty. Previous studies have shown that landscapes, where fitter optima have larger basins, tend to be easier to search [5]. Figure 2 shows a very low positive and even negative correlation between the basin size and fitness in inverse strongly correlated and subset sum instances. This indicates that fitter optima do not necessarily have larger basins in these instances and even worse bad optima tend to have larger basins (when the correlation is negative). This could mean that these instances are difficult for local search.

2.3 Local Search

We compare here the cost of finding the optimal value using a local search algorithm based on the underlying neighbourhood of each landscape. Steepest ascent algorithm with random restart was run 30 times per instance and with each of the $H1$ and $H1+2$ neighbourhood operators. The cost is then calculated using the number of used fitness evaluations until the optimal value was found. Inverse strongly correlated and subset sum instances had the highest average cost of finding the optimal solution while uncorrelated instances had the lowest average cost. Results from section 2.2 suggests that the high cost in inverse strongly correlated and subset sum instances is due to steepest ascent being attracted to the large basins of local optima and having to perform many restarts until it could find the small basin of the optimal or one of the optimal solutions.

There is a trade-off between the number of fitness evaluations needed to explore the neighbourhood and the difference between the number of local optima between the two landscapes. When the difference is very large, it is better to use the $H1+2$ operator even though the size of the neighbourhood is much larger as Figure 3 shows ($CV < 0.3$). When the difference is moderate, it is better to use the $H1$ operator (in most instance types) even though the number of local optima in the $H1$ landscape is higher ($CV > 0.9$).

![Figure 1: Correlation between the $CV$ of the weights and the number of local optima versus $\lambda$ ($n = 18$).](image1)

![Figure 2: Correlation between basin size and fitness in the two landscapes ($n = 14$).](image2)

![Figure 3: Number of instances where each operator performed significantly better and instances where no significant difference was found (Draw). Significance determined using Wilcoxon rank-sum test.](image3)

3. CONCLUSIONS

The number of local optima in the $H1$ landscape was found to be strongly and negatively correlated with weights $CV$ in moderately constrained instances of the 0-1 KP. We showed that this correlation can be used to guide the choice of the best search operator. These results are similar to the results we obtained for the number partitioning problem [1].

4. REFERENCES