A Benchmark Generator of Tree Decomposition Mk Landscapes

Dirk Thierens
Utrecht University
Utrecht, Netherlands
d.thierens@uu.nl

Tobias van Driessel
Utrecht University
Utrecht, Netherlands
tobiasvandriessel@startmail.com

ABSTRACT

We introduce the CliqueTreeMk algorithm to construct tree decomposition (TD) Mk Landscapes and to compute their global optimum efficiently. TD Mk Landscapes are well suited to serve as benchmark functions for blackbox genetic algorithms that are not given a priori the structural problem information as specified by the tree structure and their associated codomain fitness values. Specifically, for certain types of codomains the use of linkage learning might prove to be necessary in order to be able to solve these type of fitness functions.

CCS CONCEPTS

• **Computing methodologies → Heuristic function construction**;

KEYWORDS

Benchmarking, Decomposable Landscapes, Dynamic Programming

ACM Reference Format:
Dirk Thierens and Tobias van Driessel. 2021. A Benchmark Generator of Tree Decomposition Mk Landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference 2021 (GECCO '21).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3449726.3459427

1 INTRODUCTION

Suitable benchmark functions are vital to test the effectiveness and performance of evolutionary algorithms. Ideally, these benchmark functions should be completely understood in the sense that we know their structure and, importantly, their global optimum (or optima) so that we can check if a given EA has actually found the best possible solution. A problem with designing benchmark functions is that for many interesting problem classes it is not possible to compute the global optimum efficiently. Not knowing whether an EA has found the best solution limits the practical use of the benchmark and only allows relative comparisons between different algorithms - or different parameter settings of a given algorithm - but it does not allow to evaluate the overall performance and effectiveness. For example, [1] propose an interesting class of benchmark functions, but unfortunately there is no way to efficiently compute the global optimum. Similarly, the well known NK Landscapes does not allow to compute the global optimum. For this reason, EA researchers often use the Adjacent NK Landscapes where the interaction between the variables is limited to adjacent problem variables, allowing the use of dynamic programming to compute the global optimum.

GECCO '21, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07…$15.00
https://doi.org/10.1145/3449726.3459427

2 TREE DECOMPOSITION MK LANDSCAPES

Although (Adjacent) NK Landscapes are popular as a benchmark for optimization algorithms, its constraints ($M = N$, $k = K + 1$, and variable x_i must appear in subfunction f_i) are unnecessary for most benchmark purposes, as they turn out not to be important for most fundamental theoretical properties of NK Landscapes [2]. Whitley et al.[3] therefore introduced the term *Mk Landscapes* to refer to any k-bounded pseudo-Boolean optimization problem, thus a generalization of NK Landscapes without these constraints. M is the number of subfunctions and k is a constant that provides an upper bound on the interaction order size of the subfunctions, with M being polynomial in n.

Whitley introduced an important subclass of Mk Landscapes: the Tree Decomposition (TD) Mk Landscapes. The global optimum of Adjacent NK Landscapes can be computed efficiently because they control the tree-width of the variable interaction graph by considering only adjacent variables for the subfunctions. Similarly, Whitley et al. created Tree Decomposition Mk Landscapes that control the tree-width of the interaction graph in a more general way, thereby allowing the use of dynamic programming to do optimization in polynomial time. A Tree Decomposition Mk Landscape (TD Mk Landscape) is any Mk Landscape which has a known and bounded tree-width of k [3]. Tree Decomposition Mk Landscapes can be optimized in polynomial time if $k \in O(\log n)$ when one knows the tree decomposition and the associated fitness values of the subfunctions of the clique variables.

2.1 CliqueTreeMk

Whitley et al.[3] introduced a construction algorithm for TD Mk Landscapes, however, it limits the construction to TD Mk Landscapes which have a chain as tree decomposition, just like Adjacent NK Landscapes. Obviously, chains are restricted trees, they cannot represent the added complexity that arises from TD Mk landscape composed of a branching tree. Here, we introduce an algorithm to construct branching - or tree-based - TD Mk Landscapes, and apply dynamic programming to efficiently compute the global optimum (or multiple global optima).

The CliqueTreeMK algorithm constructs a TD Mk Landscape by progressively selecting subsets of the problem variables that are put in a random order, and each subset becomes a clique in the clique tree. The first k variables from the shuffled variable list are assigned to become the root of the clique tree C_0. Next, for every clique C_i, b children cliques $C_{j \in \text{children}_i}$ are selected until we have constructed M cliques. Each child C_j overlaps with its parent C_i for o variables, described by the separator S_j between C_i and C_j, and

the remaining $k - o$ variables are taken from the shuffled variable list to complete C_j.

A dynamic programming algorithm then uses this clique tree structure with its cliques and separators to calculate the global optimum. Starting at the leaves of the tree, for each separator S_j we store for each of the instances of the separator variables the maximizing variable values for its child clique C_j and the resulting score. Then, we can iterate in the reverse direction and assign values to the clique variables in C_j based on the maximizing values for its variables stored in its parent separator C_j.

We illustrate the CliqueTreeMK algorithm during these phases using an example instance with number of subfunctions $M = 7$, subfunction size $k = 3$, and overlap $o = 2$. Together, these define length $N = 9$. Furthermore, we choose a branching factor $b = 2$. The construction algorithm uses fixed values for k, o, and b, but the algorithm can be extended to allow for non-fixed values during construction. In the example the variables are randomly ordered: $(x_4, x_2, x_7, x_5, x_1, x_9, x_3, x_8, x_6)$.

2.1.1 *CliqueTreeMK Algorithm.*

(1) At the start of the algorithm, take next k variables as clique C_0. Otherwise take next already constructed clique C_i.
(2) Choose random o variables from parent clique C_i, assign to separator S_j
(3) Take next $(k - o)$ unchosen variables and add the variables from S_j to construct child clique C_j
(4) Go to step 2 until b branches have been built
(5) Go to 1 to build the whole tree

Following the algorithm with the given example instance could result in the following list of cliques: $(x_4, x_2, x_7), (x_4, x_7, x_5), (x_4, x_2, x_1), (x_7, x_5, x_9), (x_4, x_7, x_3), (x_4, x_1, x_8), (x_2, x_1, x_6)$
Figure 1 shows the constructed clique tree with its separators.

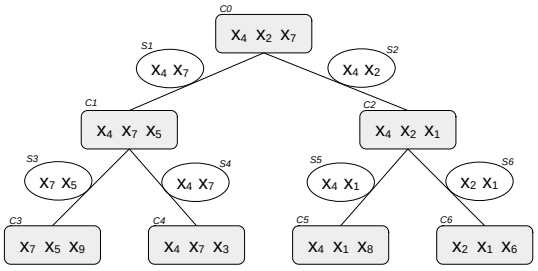

Figure 1: Clique tree with Cliques C_i and Separators S_i.

3 EXPERIMENTAL RESULTS

We have implemented the CliqueTreeMk algorithm and a dynamic program to generate and compute the global optimum (or global optimal) of TD Mk landscapes with a given number of subfunctions, length of the subfunctions, overlap size, and branching factor. These functions are very well suited to act as a benchmark[1] for blackbox genetic algorithms that are not given the structural information of the optimization problem - this is, the position of the subfunctions.

[1] Available at https://github.com/tobiasvandriessel/problem-generator

After specifying the tree structure of the TD Mk landscape a codomain for the subfunctions needs to be chosen. Possible values could be random real numbers between 0 and 1 as in the original NK landscape, quantized values that enforce the fitness contribution values to take one of Q possible discrete values as in the NK-Q landscape, deceptive trap functions, etc . . . .

As an example we generated TD Mk landscapes with random real values and with quantized values ($Q = 2$) for a tree with overlap $o = 1$, branching factor $b = 2$, clique sizes $k = 3, 4, 5$, and number of cliques $M = 4, 5, 6, 7, 8, 9, 10$. We applied a multi-start, single bit-flipping, local search algorithm (MLS) to explore these landscapes. The results shown are median values averaged over 25 runs of 100,000 fitness function evaluations with MLS. Clearly, the landscapes generated are very different for the two types of codomain, especially the actual number of global optima which can all be computed efficiently by the CliqueTreeMK algorithm.

k	M	
	4	5
3	60/60-28(28)	179/179-64(64)
4	773/756-587(587)	2548/1670-1638(1910)
5	5237/2007-4547(8254)	5685/267-4876(61960)

Table 1: NK-Q (o = 1, b = 2). #unique local optima found / #unique local optima found more than once - #unique global optima found (exact number of unique global optima).

k	M				
	6	7	8	9	10
3	16/16-1	22/22-1	28/28-1	46/46-1	64/64-1
4	128/122-1	271/217-1	423/260-1	678/238-1	689/173-1
5	853/284-1	930/138-0	858/59-0	790/17-0	730/4-0

Table 2: Random-NK (o = 1, b = 2). #unique local optima found / #unique local optima found more than once - #unique global optima found (there is 1 global optimum).

4 CONCLUSIONS

We have outlined the CliqueTreeMk algorithm to generate tree decomposable Mk landscapes, and to efficiently compute their global optimum (or optima). By varying the codomain of the landscape, multiple types of problems can be created. TD Mk Landscapes are well suited to serve as benchmark functions for blackbox Genetic Algorithms that are not given the structural problem information as specified by the clique tree. Specifically, for particular codomains - for instance (partially) deceptive functions - linkage learning techniques will be necessary to be able to find the global optimum (or optima) reliably and efficiently.

REFERENCES

[1] Kei Ohnishi, Shota Ikeda, and Tian-Li Yu. 2020. A Test Problem with Difficulty in Decomposing into Sub-Problems for Model-Based Genetic Algorithms. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion (GECCO '20).* Association for Computing Machinery, New York, NY, USA, 221?222. https://doi.org/10.1145/3377929.3389993
[2] Darrell Whitley. 2015. Mk landscapes, NK landscapes, MAX-kSAT: A proof that the only challenging problems are deceptive. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation.* 927–934.
[3] L Darrell Whitley, Francisco Chicano, and Brian W Goldman. 2016. Gray box optimization for Mk landscapes (NK landscapes and MAX-kSAT). *Evolutionary computation* 24, 3 (2016), 491–519.