Linkage Learning using the Maximum Spanning Tree of the Dependency Graph

B. Hoda Helmi
University of Missouri at St. Louis
St. Louis, MO 63121
Iran University of Science and Technology
Tehran, Iran 13114-16846
helmi@iust.ac.ir

Martin Pelikan
MEDAL Lab
Dept. of Mathematics and Computer Science
University of Missouri
St. Louis, MO 63121
martin@martinpelikan.net

Adel T. Rahmani
SCOMAS Lab
Dept. of Computer Science
Iran University of Science and Technology
Tehran, Iran 13114-16846
rahmani@iust.ac.ir

ABSTRACT

This paper presents a simple offline approach to linkage learning based on the graph theory. The proposed approach has the advantage of being able to learn the linkage without the need for the costly fit-to-data evaluations for model search. Based on the experimental results it can successfully find the linkage groups in a polynomial number of fitness evaluations.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization; I.2.6 [**Artificial Intelligence**]: Learning; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithm, Design, Experimentation, Performance

Keywords

Linkage learning, optimization problems, decomposable functions, nonlinearity detection

1. INTRODUCTION

In this paper, an offline linkage identification approach is introduced that is built upon the simple concept of maximum spanning tree. The proposed algorithm uses perturbation based pairwise dependencies to construct the underlying dependency graph of the problem. Maximum spanning tree of the graph is then found. There are two closely related linkage learning approaches that do the linkage learning separately from the optimization search. Both [1] [2] use nonlinearity detection to construct the dependency structure matrix (DSM) and then cluster the DSM using two different approaches. The offline utility of DSM genetic algorithm (DSMGA) [1] cluster the DSM by an evolutionary strategy with a fit-to-data objective function. In [2] a density based iterative clustering algorithm is introduced to cluster the DSM. Both of the above approaches need to set some parameters and thresholds which is not an easy task.

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

2. PROPOSED APPROACH

The pseudo-code of the proposed approach is depicted as Algorithm 1. The main steps are discussed below:

1. Constructing the dependency graph: The dependency graph is a weighted, undirected graph where each vertex corresponds to one decision variable and each edge of the graph has a real value weight e_{ij} representing the strength of dependency between vertex i and vertex j. The pairwise dependency metric defined in ref. [1] is used for constructing the dependency graph.

2. Finding the maximum spanning tree of the underlying graph of the problem: Once the graph is constructed, the maximum spanning tree (MST) of the graph is constructed using Prim's algorithm. Assuming that the problem can be decomposed into independent subproblems (linkage groups) and that the population is large enough, we would expect each linkage group to be a connected subgraph of the resulting MST, with several edges (false linkages) of smaller weight connecting the different linkage groups.

3. Cutting off the false linkages: The final task consists of cutting off the edges between the linkage groups. If a sufficiently population is used for constructing the graph, there would be a sensible gap between weights of the correct linkages within one subproblem and false linkages between subproblems in MST. To distinguish the two types of linkages, we used the simple k-means clustering algorithm with $k = 2$. For more details see [3].

Compexity Analysis: The number of fitness evaluations required is equal to the population size (N). Pairwise dependencies (f values) can be computed in $O(n^2 \times N)$, where n is the problem size. Finding MST can be done in $O(n^2)$ time. For cutting the false linkages a simple 2-means clustering algorithm is used. The clustering is done using a constant number of update iterations, l=20, so the computation complexity of cutting the false linkages is $O(n)$.

3. EXPERIMENTAL RESULTS

The results are shown for the concatenated trap functions

Algorithm 1 Algorithm of the proposed approach

Output: Linkage groups

1: Create random population, P.
2: Evaluate the population.
3: **for** each variable i and variable j, **do**
4: **for** each individual a in the population, **do**
5: Update $f_{a_i=0,a_j=0}||f_{a_i=0,a_j=1}||f_{a_i=1,a_j=0}||f_{a_i=1,a_j=1}$
6: **end for**
7: $e_{ij} = |f_{a_i=0,a_j=1} - f_{a_i=0,a_j=0} - f_{a_i=1,a_j=1} + f_{a_i=1,a_j=0}|$
8: **end for**
9: Find the maximum spanning tree T of the graph.
10: Find threshold ρ by 2-means clustering on the edges of T.
11: Cut the edges weighted less than ρ.

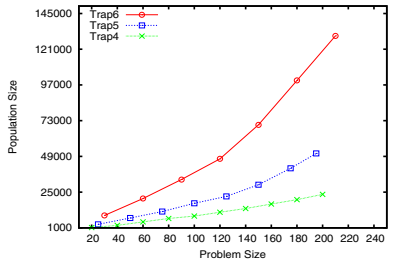

Figure 1: Population size needed for the proposed algorithm to successfully learn all the linkage groups for the concatenated trap function.

with linkage groups of sizes 4 (for problem sizes 20 to 160), 5 (for problem sizes 25 to 200) and 6 (for problem sizes 24 to 240). A concatenated trap function with m subproblems, has one global optimum and $2^m - 1$ local optima. Population size is determined by bisection with 50 successful independent runs in which all the linkage groups are identified correctly. We have fitted our results to $N \approx an^b$ (polynomial scaling) and $N \approx an^b \log n$. Based on the coefficient of determination (R^2) both models are acceptable. In figure 2, the number of fitness evaluations for trap4, trap5 and trap6 are plotted together with the fit according to the model $N \approx an^b$. b is observed to be in the range (1,1.7) for the proposed algorithm.

Comparison with reference algorithms: For offline utility of DSMGA [1], the results on the concatenated trap function with 10 subfunctions of order 5 is reported; 11712 function evaluations were needed to correctly identify 99.8% of linkage groups. For the same problem 7625 function evaluations are needed for the proposed algorithm to find all the linkage groups. The results of both DSMC and the proposed algorithm are depicted in figure 3. DSMC needs smaller number of fitness evaluations at the cost of setting and defining several metrics and parameters for the DSM clustering.

4. SUMMARY AND CONCLUSION

This paper introduced a simple offline linkage learning approach. The algorithm consists of three main steps. First, a dependency graph is created using a pairwise metric. Second, a maximum spanning tree is built for the dependency graph. Third, edges corresponding to weakest dependencies

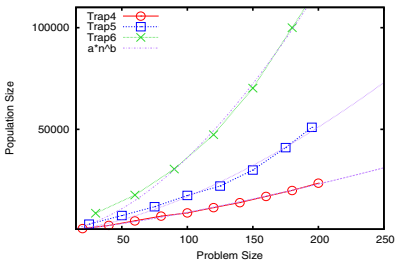

Figure 2: Number of fitness evaluations and the fit $N \approx an^b$ for the concatenated trap function.

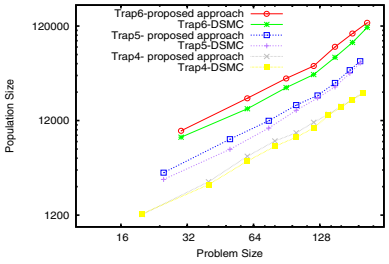

Figure 3: Number of fitness evaluations for the concatenated trap problem for DSMC and for the proposed approach.

in the MST are eliminated and the connected components are used as linkage groups. The proposed method does not need to do the costly fit-to-data check. It is shown that the proposed approach can find all the linkage information correctly (at least for the tested problems) with a polynomial number of fitness evaluations. The main advantage of the proposed approach compared to prior work in offline linkage learning and DSMC is that the algorithm contains no parameters that must be tuned. As the future work, performance of the proposed algorithm should be tested on other problems as well, like functions with overlapping linkage groups and exponentially scale problems. The algorithm should also be applied in the online setting.

Acknowledgments: M. Pelikan and H. Helmi were supported by NSF under grants ECS-0547013 and IIS-1115352. Any opinions, findings, and conclusions expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

5. REFERENCES

[1] T.-L. Yu and D. E. Goldberg. Dependency structure matrix analysis: Offline utility of the dependency structure matrix genetic algorithms. In *GECCO-2004*, pages 355–366. Springer, 2004.

[2] A. Nikanjam, H. Sharifi, B. H. Helmi, and A. Rahmani. A new DSM clustering algorithm for linkage groups identification. In *GECCO-2010*, pages 367–368. ACM, 2010.

[3] B. H. Helmi, M. Pelikan, A. T. Rahmani. Technical report, MEDAL Report No. 2012005, Missouri Estimation of Distribution Algorithms Laboratory (MEDAL), University of Missouri, St. Louis, MO, 2012.