A Surrogate-Assisted and Informed Linkage Aware GA

Tomasz Oliwa
Department of Computer Science
The University of Georgia
Athens, Georgia 30602–7404, USA
oliwa@uga.edu

Khaled Rasheed
Department of Computer Science
The University of Georgia
Athens, Georgia 30602–7404, USA
khaled@cs.uga.edu

ABSTRACT

We present a genetic algorithm that combines linkage learning, surrogate models and informed operators. Linkage learning aims at measuring and exploiting interdependence of groups of genes. Surrogate models are fitness approximators to ease the task of calculating true fitness values. Informed operators generate, evaluate and rank a set of solutions according to their fitness model to return their most fit solution. Our described approach provides on-line perturbation based linkage learning and informed linkage exploitation with novel, specialized operators. Results of experimental runs on several synthetic fitness function compositions are provided to demonstrate significant improvement of the final result quality compared to a conventional GA setup.

Categories and Subject Descriptors

I.2.6 [**ARTIFICIAL INTELLIGENCE**]: Learning— *Parameter learning*; I.2.8 [**ARTIFICIAL INTELLIGENCE**]: Problem Solving, Control Methods, and Search— *Heuristic methods*

Keywords

Genetic Algorithms, Surrogate Models, Linkage, Operators

1. INTRODUCTION

The type and strength of interactions of problem variables in the context of global optimization is referred to as linkage or epistasis, and its character and possible exploitation has been subject to continuous studies in the area of evolutionary computation (EC) [1]. Another approach to save evaluation time and enhance the quality of final individuals is surrogate-assisted evolution (SAE). Here, surrogate models (for instance neural networks [3]) work as function approximators. Informed operators, introduced in [8], work by applying a surrogate model during offspring creation to rank an offspring pool. Surrogate models have been previously employed in linkage learning [5], but only in an offline linkage learning fashion without any operators to take advantage of it. We present a novel combination of these three approaches, the surrogate-assisted and informed linkage exploiting genetic algorithm (SAILEGA), which allows for on-line perturbation based linkage learning and informed exploitation of it.

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

2. LINKAGE LEARNING

One of the major goals of linkage learning is the detection of the underlying fitness function's decompositability. Formally, the fitness function $F(x)$ in Equation 1

$$F(x) = \sum_{i=0}^{k} f_i(x_{s_i}) \tag{1}$$

can be additively decomposed into k sub-functions, where the whole chromosome x is split into subchromosomes x_{s_i}, and where s_i defines which genes belong to the corresponding subchromosome (with permitted overlapping). Such a set of genes s_i is called a linkage group. SAILEGA[1] implements surrogate-assisted and regular versions of LINC-R [9] and LIMD [4]. To mend linkage estimation errors, we define existing linkage for two genes only if both surrogate-assisted LINC-R and LIMD return a successful linkage observation.

3. SURROGATE MODELS

All encountered (unique) individuals are stored in a population storage, which is clustered with X-means [6] every generation. A SVM for regression using Weka [2] with the Pearson VII function-based universal kernel (PUK) is created for each cluster that is represented by at least one member of the current population. Every generation, one member of each current cluster is subject to linkage analysis.

4. CROSSOVER OPERATORS

SAILEGA incorporates four novel crossover operators combining the merits of linkage exploitation, SAE and informed operators with the aforementioned SVMs and a crossover selector following GADO's [7] guideline. The informed linkage group line (ILGL) crossover is based on the double line crossover [7], it is especially helpful in regions of ellipsoidal space, the algorithm is given in Figure 1. The informed guided linkage group (IGLG) crossover is based on the guided crossover [7]. Similarly as with the ILGL crossover, each linkage group receives its own ratio variable r to let linked genes exploit the search space in the same direction. The informed linkage inter group (ILIG) crossover takes the idea of the uniform crossover to the level of linkage groups to keep linked genes from being disrupted while enabling mixing of genetic material. The informed linkage inter-intra group (ILIIG) crossover performs ILIG's recombination and then shuffles the offspring alleles of each linkage group.

[1]Our GA is built on top of the Watchmaker Framework 0.7.1: http://watchmaker.uncommons.org/

```
p1, p2 ← parentSelection(currentPopulation)
offspring ← empty offspring
allGroups ← set of linkage groups ∪ nonlinked group
for all group ∈ allGroups do
    r ← getRandom(−2, 2)
    for all gene ∈ group do
        newGene = r ∗ p1_gene + (1 − r) ∗ p2_gene
        offspring_gene ← newGene
    end for
end for
```

Figure 1: The ILGL crossover algorithm.

Table 1: Fitness Functions

Function	Definition
F_1: Sphere$_4$	$\sum_{i=1}^{D}(x_i+4)^2$
F_2: Sphere$_{-4}$	$\sum_{i=1}^{D}(x_i-4)^2$
F_3: Rosenbrock	$\sum_{i=2}^{D}(100(x_1-x_i^2)^2+(x_i-1)^2)$
F_4: Schwefel 1.2	$\sum_{i=1}^{D}(\sum_{j=1}^{i}x_j)^2$
F_5: Rastrigin	$10D+\sum_{i=1}^{D}(x_i^2-10cos(2\pi x_i))$

5. EXPERIMENTAL SETUP

Three configurations C are run 50 times on each composite function as given in Tables 1 and 2. $C_{default}$ uses surrogate models, C_{noSu} only calls the true fitness function for any evaluations and C_{1point} only applies the 1-point crossover. Their remaining setup is identical to the GA in [5].

6. EXPERIMENTAL RESULTS

Table 2 shows the mean of the final individuals and the corresponding standard deviation as well as one-tailed t-test results to compare $C_{default}$ and C_{noSu} against C_{1point}. On all runs, both $C_{default}$ and C_{noSu} clearly outperform C_{1point}, showing that their configuration leads the GA to vastly superior solutions. Furthermore, with expensive fitness functions, C_{noSu} may take much more time than $C_{default}$ due to C_{noSu}'s solely calls to the actual fitness function.

Table 2: Final Minimization Results

Config	Function^{Dim}	Mean	Std. Dev.	p-value
$C_{default}$	$F_1^7+F_2^7$	0.0027	7.68e-4	1.168e-10
C_{noSu}	$F_1^7+F_2^7$	0.002	0.00036	1.15e-10
C_{1point}	$F_1^7+F_2^7$	1.32	1.17	-
$C_{default}$	$F_3^6+F_2^8$	4.06	3.75	2.911e-10
C_{noSu}	$F_3^6+F_2^8$	1.94	2.27	7.383e-12
C_{1point}	$F_3^6+F_2^8$	20.44	14.99	-
$C_{default}$	$F_4^4+F_5^4+F_2^6$	2.03	1.43	2.567e-10
C_{noSu}	$F_4^4+F_5^4+F_2^6$	0.08	0.2	$<$ 2.2e-16
C_{1point}	$F_4^4+F_5^4+F_2^6$	4.86	2.42	-

7. CONCLUSION

In this paper, we provide a combination of surrogate models, linkage learning and informed linkage exploiting operators. The presented method SAILEGA is composed of an on-line perturbation based linkage learning module, an application of X-means clustering to dynamically adjust its surrogate model set of SVMs for regression, and an arsenal of novel informed linkage exploiting recombination operators. These operators are designed in a modular way such that it is possible to use them in a non-surrogate setting or without a pool of created offspring, making them potentially useful for an implementation by other optimizers. We empirically demonstrated that SAILEGA can successfully exploit linkage of genes from fitness functions to significantly improve final result quality.

8. REFERENCES

[1] Y.-P. Chen and M.-H. Lim, editors. *Linkage in Evolutionary Computation*, volume 157 of *Studies in Computational Intelligence*. Springer, 2008.

[2] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11:10–18, November 2009.

[3] Y. Jin, M. Hüsken, M. Olhofer, and B. Sendhoff. Neural networks for fitness approximation in evolutionary optimization. In Y. Jin, editor, *Knowledge Incorporation in Evolutionary Computation*, pages 281–305. Springer, Berlin, 2004.

[4] M. Munetomo and D. E. Goldberg. Linkage identification by non-monotonicity detection for overlapping functions. *Evolutionary Computation*, 7(4):377–398, 1999.

[5] T. Oliwa and K. Rasheed. A surrogate-assisted linkage inference approach in genetic algorithms. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 997–1004, New York, NY, USA, 2011. ACM.

[6] D. Pelleg and A. W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 727–734, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[7] K. Rasheed. *GADO: A genetic algorithm for continuous design optimization*. Ph.d. thesis, Rutgers University, 1998.

[8] K. Rasheed and H. Hirsh. Informed operators: Speeding up genetic-algorithm-based design optimization using reduced models. In L. D. Whitley, D. E. Goldberg, E. Cantú-Paz, L. Spector, I. C. Parmee, and H.-G. Beyer, editors, *GECCO*, pages 628–635. Morgan Kaufmann, 2000.

[9] M. Tezuka, M. Munetomo, and K. Akama. Linkage identification by nonlinearity check for real-coded genetic algorithms. In K. Deb, R. Poli, W. Banzhaf, H.-G. Beyer, E. K. Burke, P. J. Darwen, D. Dasgupta, D. Floreano, J. A. Foster, M. Harman, O. Holland, P. L. Lanzi, L. Spector, A. Tettamanzi, D. Thierens, and A. M. Tyrrell, editors, *GECCO (2)*, volume 3103 of *Lecture Notes in Computer Science*, pages 222–233. Springer, 2004.