Improved Runtime Bounds for the Univariate Marginal Distribution Algorithm via Anti-Concentration

Phan Trung Hai Nguyen

June 26, 2017

School of Computer Science
University of Birmingham
Birmingham B15 2TT
United Kingdom

Outlines

Background

- Estimation of Distribution Algorithms (EDAs)
- Univariate Marginal Distribution Algorithm (UMDA)
- Ones-counting problem (OneMax)

Useful tools

- Level-based theorem
- Anti-concentration bound
- Feige's inequality

Our result

- Improved upper bound on runtime of UMDA on OneMax

Conclusion

Evolutionary Algorithms (EAs)

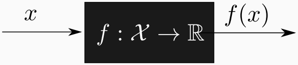


Figure 1: Black-box optimisation

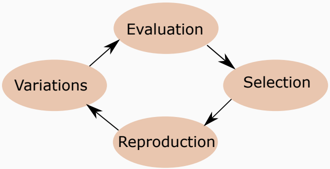


Figure 2: Flowchart of EAs

- **Black-box** optimisation: optimise an objective function $f : \mathcal{X} \to \mathbb{R}$.
- Inspiration from **natural evolution** and **molecular genetics**.
- Bio-inspired mechanisms: mutation, crossover.
- **Runtime analysis** of EAs
  - Generally provide insights into its behaviour,
  - Number of fitness evaluations,
  - Interest in the expected runtime.

Estimation of Distribution Algorithms (EDAs)

Belonging to class of Evolutionary Algorithms.

Do not use mutation or crossover.

Building and sampling using a probabilistic model.

Many practical applications:

- Bio-informatics: Gene structure/expression analysis, protein structure prediction, protein design (Armañanzas et al. 2008).
- Engineering: military antenna design (Yu et al. 2006), forest management (Ducheyne et al. 2004), analog circuit design (Zinchenko et al. 2002), environment monitoring network design (Kollat et al. 2008).

Algorithm 1: EDA pseudo-code

begin
- Initialise the probabilistic model.
- **repeat**
  - Sample λ individuals using the current model.
  - Select $\mu \leq \lambda$ individuals using some selection mechanism.
  - Update the model using selected individuals.
- **until** *termination condition fulfilled*

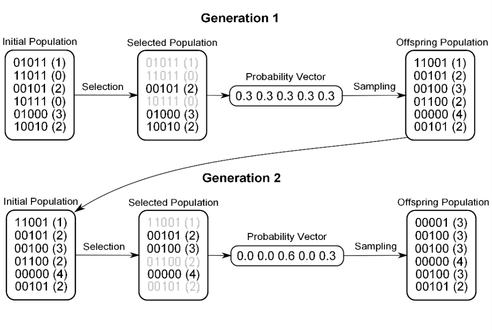


Figure 3: Sample runs of a simple EDA (Hauschild & Pelikan 2011)

EDAs can be categorised into

- Univariate: Decision variables are treated **independently**.
  E.g. UMDA (Mühlenbein & Paaß 1996), cGA (Harik et al. 1997), PBIL (Baluja 1994).
- Bivariate: **Pairwise** interactions between variables can be captured The model structure is often a chain/tree.
  E.g. BMDA (Pelikan & Muehlenbein 1999), MIMIC (Bonet et al. 1996).
- Multivariate: More **complex** interactions can be captured using Markov network or Bayesian network.
  E.g. BOA (Pelikan et al. 2000), EBNA (Etxeberria & Larrañaga 1999), LTGA (Thierens 2010), ECGA (Harik et al. 1999).

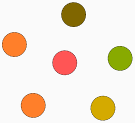


Figure 4: Independence

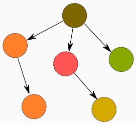


Figure 5: Tree-based model

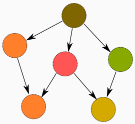


Figure 6: Bayesian Network

Univariate Marginal Distribution Algorithm (UMDA)

Solution representation: $(X_1, \ldots, X_n) \in \{0,1\}^n$.

Probabilistic model at generation t: $\mathcal{M}_t := (p_t^1, p_t^2, \ldots, p_t^n)$.

Marginal probabilities: $p_t^i := \Pr(X_i = 1)$.

Algorithm 2: UMDA

begin
  initial model $\mathcal{M}_0 = (1/2, 1/2, \ldots, 1/2)$
  repeat
    Sample P_t of λ individuals using joint probability

$$\Pr(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} \Pr(X_i).$$

    Select the μ best individuals according to fitness.
    Update the probabilistic model

$$p_{t+1}^i := \frac{1}{\mu} \sum_{j=1}^{\mu} X_i^{[j]} \in \left[\frac{1}{n}, 1 - \frac{1}{n}\right]$$

    where $X_i^{[j]}$ is the i-th bit of the j-th individual.
  until *termination condition fulfilled*

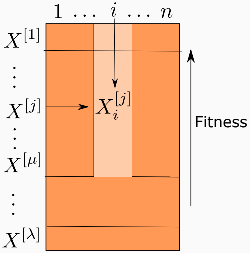


Figure 7: UMDA illustration

Previous Work

	(1+1) EA	cGA[1]		
OneMax	$\Theta(n \log n)$	$\Theta(K\sqrt{n})$	(Droste 2006)	EDA worse
Linear	$\Theta(n \log n)$	$\Omega(Kn)$	(Droste 2006)	EDA worse

	(1+1) EA	UMDA	
LeadingOnes	$\Theta(n^2)$	$\mathcal{O}(n\lambda \log \lambda + n^2)$	(Dang & Lehre 2015)[2]
BVLeadingOnes	$\Theta(n \log n)$	∞ w.o.p.	(Chen et al. 2010)
		$\mathcal{O}(n\lambda)$	(Chen et al. 2010)[3]
SubString	$2^{\Omega(n)}$ w.o.p.	$\mathcal{O}(n\lambda)$	(Chen et al. 2009)[4]
OneMax	$\Theta(n \log n)$	$\mathcal{O}(n\lambda \log \lambda)$	(Dang & Lehre 2015)[3]
		$\Omega(\mu\sqrt{n} + n \log n)$	(Krejca & Witt 2017)[4]

[1] $K = n^{1/2+\epsilon}$
[2] $\lambda = \Omega(\log n)$
[3] $\lambda = \omega(n^2)$
[4] $\lambda = (1 + \Theta(1))\mu$

Figure 8: Mastermind game

- Play as **sub-problems** for many real-world separable problems. The optimum is to optimise each individuals.
- **Mastermind** game with two colours and only black answer-pegs used.
- Formally defined as

$$\textsc{OneMax}(x_1, \ldots, x_n) := \sum_{i=1}^{n} x_i.$$

- **Objective**: counting the number of ones in the bitstring.
- **Optimal solution**: all-ones bitstring.

Open Question

Runtime analysis of UMDA on OneMax (latest results):

- Upper bound: $\mathcal{O}(n \log n \log \log n)$ by Dang & Lehre (Dang & Lehre 2015).
- Lower bound: $\Omega(\mu\sqrt{n} + n \log n)$ by Krejca & Witt (Krejca & Witt 2017).

The upper and lower bounds are still different by $\Theta(\log \log n)$.

Open question: Could this gap be closed?

Why Is Closing The Gap Important?

- Understanding how parameter settings could affect its performance.
- Aid in the selection of EDAs for a particular project.
- Also Relevant to Population Genetics (i.e. linkage equilibrium).
- Compare performance with other EAs like (1+1) EA.

Theorem

The expected optimisation time of UMDA with

- $c \log n \leq \mu \leq c' \sqrt{n}$ for some constants $c, c' > 0$,
- $\lambda \geq a\mu$ for some constant $a \geq 13e$

on OneMax is

$$\mathcal{O}(n\lambda).$$

Intuition: UMDA can optimise OneMax within $\mathcal{O}(n)$ generations.

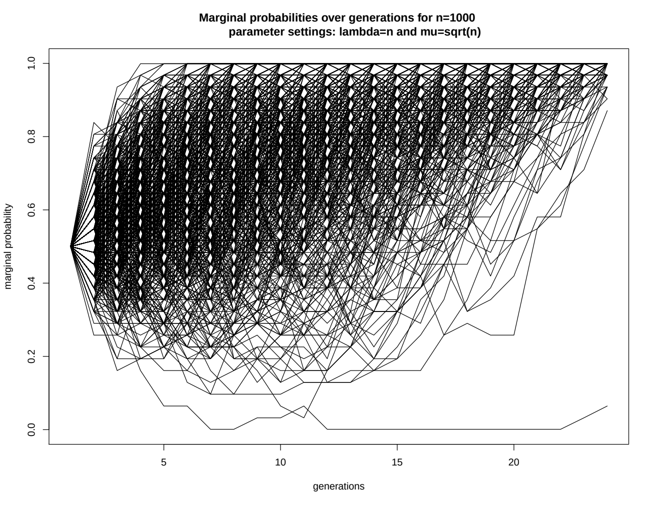
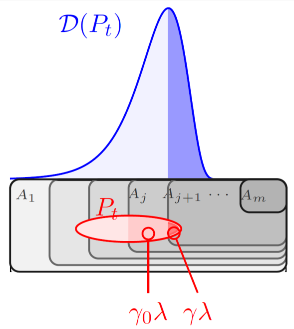


Figure 9: Marginal probabilities of UMDA on ONEMAX for $n = 1000$

Level-Based Theorem[5]

Provide **upper bounds** on the expected runtime of some population-based algorithms on a wide range of optimisation problems.



$y \sim \mathcal{D}(P_t)$

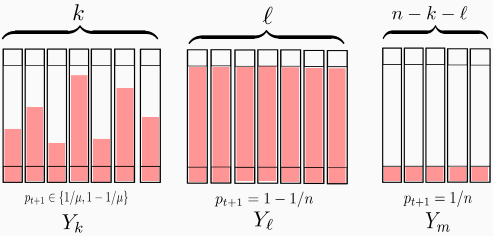
- **(G1)** $j \in [m-1]$, if $|P_t \cap A_{\geq j}| \geq \gamma_0 \lambda$ then
$$\Pr\left(y \in A_{\geq j+1}\right) \geq z_j.$$
- **(G2)** similar to (G1) and $|P_t \cap A_{\geq j+1}| \geq \gamma\lambda$ then
$$\Pr\left(y \in A_{\geq j+1}\right) \geq (1+\delta)\,\gamma.$$
- **(G3)** and the population size $\lambda \in \mathbb{N}$ satisfies
$$\lambda \geq \left(\frac{4}{\gamma_0 \delta^2}\right) \ln\left(\frac{128m}{z_* \delta^2}\right)$$
where $z_* := \min_{j\in[m-1]}\{z_j\}$, then expected runtime
$$\mathbb{E}\left[T\right] \leq \left(\frac{8}{\delta^2}\right) \sum_{j=1}^{m-1} \left[\lambda \ln\left(\frac{6\delta\lambda}{4+z_j\delta\lambda}\right) + \frac{1}{z_j}\right].$$

[5] See Corus et al. (2016) for more details on the theorem.

Proof idea (UMDA with margins)

- **Level definition:** $A_j := \{x \in \{0,1\}^n \mid \textsc{OneMax}(x) = j - 1\}$.]
- Let A_j denote the current level.
- We choose $\gamma_0 := \mu/\lambda \Rightarrow$ all μ best individuals have $\geq j - 1$ ones.
- Upgrade $\Rightarrow$ Sample an offspring with $\geq j$ ones, i.e. $\Pr(Y \geq j)$.
- Verify condition (G1): $\Pr(Y \geq j) \geq z_j$.
- Verify condition (G2): $\Pr(Y \geq j) \geq (1 + \delta)\gamma$.

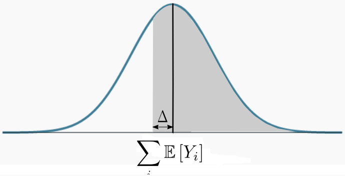
Verify condition (G2)



The probability of sampling j one-bits:

$$
\begin{aligned}
\Pr(Y \ge j) &\ge \Pr(Y_k + Y_m \ge j - \ell) \cdot \Pr(Y_\ell = \ell) \\
&\ge \Pr(Z > j - \ell - 1) \cdot \left(1 - \frac{1}{n}\right)^{\ell} && (Z := Y_k + Y_m) \\
&\ge \Pr\left(Z > \mathbb{E}[Z] - \frac{\gamma}{\gamma_0}\right) \cdot \frac{1}{e} && \left(\mathbb{E}[Z] \ge j - \ell - 1 + \frac{\gamma}{\gamma_0}\right) \\
&\ge \quad \dots \\
&\ge \quad \dots \\
&\ge (1+\delta)\gamma.
\end{aligned}
$$

Feige's Inequality



Theorem (Feige (2004))

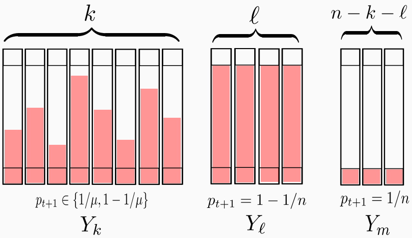
Given n independent r.v. $Y_1, \ldots, Y_n \in [0,1]$, then for all $\Delta > 0$

$$\Pr\left(\sum_{i=1}^{n} Y_i > \sum_{i=1}^{n} \mathbb{E}[Y_i] - \Delta\right) \geq \min\left\{\frac{1}{13}, \frac{\Delta}{1+\Delta}\right\}.$$

The probability of sampling j one-bits now becomes

$$\begin{aligned}
\Pr(Y \geq j) &\geq \Pr\left(Z > \mathbb{E}[Z] - \frac{\gamma}{\gamma_0}\right) \cdot \frac{1}{e} \\
&\geq \min\left\{\frac{1}{13}, \frac{\gamma/\gamma_0}{1+\gamma/\gamma_0}\right\} \cdot \frac{1}{e} \\
&\geq \frac{\gamma}{13\gamma_0} \cdot \frac{1}{e} \\
&\geq (1+\delta)\gamma.
\end{aligned}$$

Verify (G1): k is sufficiently large



The upgrade probability in this case:

$$
\begin{aligned}
\Pr(Y_k \geq j-\ell) &= \Pr(Y_k \geq j-\ell-1) - \Pr(Y_k = j-\ell-1) \\
&= {\color{red}\Pr(Y_k \geq \mathbb{E}[Y_k])} - {\color{blue}\Pr(Y_k = j-\ell-1)} \\
&\geq \quad \ldots \\
&\geq \quad \ldots
\end{aligned}
$$

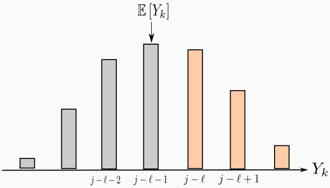
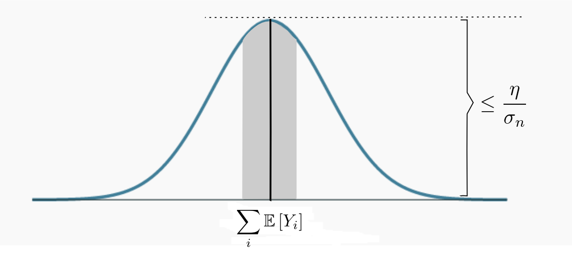


Figure 10: Probability distribution of Y_k

Anti-Concentration Bound



Theorem (Baillon et al. (2016))

Given n independent Bernoulli r.v. $Y_1, \ldots, Y_n$ with success probabilities $p_1, \ldots, p_n$. For all n, y and p_i.

$$\Pr\left(\sum_{i=1}^{n} Y_i = y\right) \leq \frac{\eta}{\sigma_n}$$

where $\sigma_n^2 := \sum_{i=1}^{n} p_i(1-p_i)$ and $\eta \sim 0.4688$ is an absolute constant.

Verify (G1): k is sufficiently large

Theorem (Theorem 3.2, Jogdeo & Samuels (1968))

Let $Y_1, Y_2, \ldots, Y_n$ be n independent Bernoulli random variables. Let $Y := \sum_{i=1}^{n} Y_i$ be the sum of these random variables. If the expectation of Y is an integer, then

$$\Pr(Y \geq \mathbb{E}[Y]) \geq \frac{1}{2}.$$

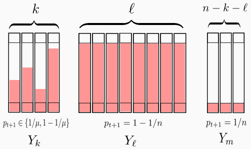
Now the upgrade probability now becomes

$$\begin{aligned} \Pr(Y_k \geq j - \ell) &= \Pr(Y_k \geq j - \ell - 1) - \Pr(Y_k = j - \ell - 1) \\ &= \Pr(Y_k \geq \mathbb{E}[Y_k]) - \Pr(Y_k = j - \ell - 1) \\ &\geq \frac{1}{2} - \frac{\eta}{\sqrt{\mathrm{Var}[Y_k]}} \\ &\geq \Omega(1). \end{aligned}$$

CASE 2: small k and large ℓ (so is j)

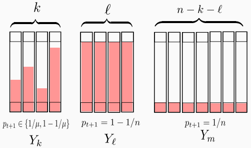
- Being very close to optimal level.
- If $j \geq n + 1 - \frac{n}{\mu}$, then according to Dang & Lehre (2015)

$$\Pr(Y \geq j) = \Omega\left(\frac{1}{\mu}\right) = \Omega\left(\frac{n-j+1}{n}\right).$$



CASE 3: Small k and not close to optimal level (i.e. not too large j).

$$\begin{aligned}\Pr(Y \geq j) &\geq \Pr(Y_k \geq j - \ell - 1) \cdot \Pr(Y_\ell = \ell) \cdot \Pr(Y_m \geq 1)\\ &\geq \frac{1}{2} \cdot \frac{1}{e} \cdot \frac{n-k-\ell}{n}\\ &\geq \Omega\left(\frac{n-j+1}{n}\right).\end{aligned}$$



Condition (G3) and Expected runtime

Combining all three cases

$$z_j := \min\left\{\Omega(1), \Omega\left(\frac{n-j+1}{n}\right)\right\} = \Omega\left(\frac{n-j+1}{n}\right).$$

(G3) satisfied iff

$$\lambda \geq \left(\frac{4}{\gamma_0\delta^2}\right)\ln\left(\frac{128m}{z_*\delta^2}\right) = \Omega(\log n).$$

The expected optimisation time is

$$\begin{aligned}
\mathcal{O}\left(\lambda\sum_{j=1}^{n}\ln\left(\frac{1}{z_j}\right) + \sum_{j=1}^{n}\frac{1}{z_j}\right) &= \mathcal{O}\left(\lambda\sum_{j=1}^{n}\ln\left(\frac{n}{n-j+1}\right) + \sum_{j=1}^{n}\frac{n}{n-j+1}\right) \\
&= \mathcal{O}\left(\lambda\ln\prod_{j=1}^{n}\frac{n}{n-j+1} + n\sum_{k=1}^{n}\frac{1}{k}\right) \\
&= \mathcal{O}\left(\lambda\ln\frac{n^n}{n!} + n\sum_{k=1}^{n}\frac{1}{k}\right) \qquad \text{(Stirling's approximation}^5\text{)} \\
&= \mathcal{O}(n\lambda) + \mathcal{O}(n\log n) \\
&= \mathcal{O}(n\lambda).
\end{aligned}$$

[5] Stirling's approximation: $n! \propto n^{n+0.5}e^{-n}$

Summary

- The upper bound $\mathcal{O}(n\lambda)$ **holds** for $a \log(n) \leq \mu \leq a' \sqrt{n}$ where a, a' are some positive constants.
- The bound is **tight** when $\lambda \leq c \log(n)$, yielding a tight bound $\Theta(n \log n)$ matching the bound of $(1+1)$EA on OneMax.
- The result finally **closes the gap** $\Theta(\log \log n)$ between the first upper bound $\mathcal{O}(n \log n \log \log n)$ (Dang & Lehre 2015) and recently discovered lower bound $\Omega(\mu\sqrt{n} + n \log n)$ (Krejca & Witt 2017).
- Anti-concentration may be applied to analyse runtime of other algorithms.

Future Work

Possible future work:

- Runtime of UMDA on OneMax for larger offspring population size, i.e. $\mu = \omega(\sqrt{n})$[6].
- Runtime of UMDA on more complex fitness landscapes.
- Generalise the result to linear functions

$$\textsc{Linear}(x_1, \ldots, x_n) := \sum_{i=1}^{n} \omega_i x_i.$$

For OneMax, all weights $\omega_i = 1$.

[6] $g(n) = \omega(f(n))$ means $\lim_{n\to\infty} g(n)/f(n) \to \infty$

Any question?

Phan Trung Hai Nguyen

pxn683@cs.bham.ac.uk

References

Armañanzas, R., Inza, I., Santana, R., Saeys, Y., Flores, J. L., Lozano, J. A., Peer, Y. V. d., Blanco, R., Robles, V., Bielza, C. & Larrañaga, P. (2008), 'A review of estimation of distribution algorithms in bioinformatics', *BioData Mining* **1**, 6.

Baillon, J.-B., Cominetti, R. & Vaisman, J. (2016), 'A sharp uniform bound for the distribution of sums of bernoulli trials', *Combinatorics, Probability and Computing* **25**(3), 352–361.

Baluja, S. (1994), 'Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning', *Technical report, Carnegie Mellon University, Pittsburgh, PA* .

Bonet, J. S. D., Isbell, C. L. & Viola, P. (1996), Mimic: finding optima by estimating probability densities, *in* 'Proceedings of the 9th International Conference on Neural Information Processing Systems', pp. 424–430.

Chen, T., Lehre, P. K., Tang, K. & Yao, X. (2009), When is an estimation of distribution algorithm better than an evolutionary algorithm?, *in* 'Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2009', pp. 1470–1477.

References II

Chen, T., Tang, K., Chen, G. & Yao, X. (2010), 'Analysis of computational time of simple estimation of distribution algorithms', *IEEE Transactions on Evolutionary Computation* **14**(1), 1–22.

Corus, D., Dang, D., Eremeev, A. V. & Lehre, P. K. (2016), 'Level-based analysis of genetic algorithms and other search processes', *CoRR* abs/1407.7663.
URL: *http://arxiv.org/abs/1407.7663*

Dang, D. & Lehre, P. K. (2015), Simplified runtime analysis of estimation of distribution algorithms, *in* 'Proceedings of Genetic and Evolutionary Computation Conference, GECCO'15'.

Droste, S. (2006), 'A rigorous analysis of the compact genetic algorithm for linear functions', *Natural Computing* **5**(3), 257–283.

Ducheyne, E., De Baets, B. & De Wulf, R. (2004), Probabilistic models for linkage learning in forest management, *in* Y. Jin, ed., 'Knowledge Incorporation in Evolutionary Computation', Vol. 167 of *Studies in Fuzziness and Soft Computing*, Springer, pp. 177–214.

Etxeberria, R. & Larrañaga, P. (1999), Global optimization using bayesian networks, *in* 'Second Symposium on Artificial Intelligence, CIMAF-99', pp. 332–339.

Feige, U. (2004), On sums of independent random variables with unbounded variance, and estimating the average degree in a graph, *in* 'Proceedings of the 36th STOC', pp. 594–603.

Harik, G. R., Lobo, F. G. & Goldberg, D. E. (1997), 'The compact genetic algorithm', *IlliGAL report No. 97006, University of Illinois at Urbana-Champaign, Urbana* .

Harik, G. R., Lobo, F. G. & Goldberg, D. E. (1999), 'The compact genetic algorithm', *IEEE Transactions on Evolutionary Computation* **3**(4), 287–297.

Hauschild, M. & Pelikan, M. (2011), 'An introduction and survey of estimation of distribution algorithms', *Swarm and Evolutionary Computation* **1**(3), 111–128.

Jogdeo, K. & Samuels, S. M. (1968), 'Monotone convergence of binomial probabilities and a generalization of ramanujan's equation', *The Annals of Mathematical Statistics* **39**(4), 1191–1195.

Kollat, J., Reed, P. & Kasprzyk, J. (2008), 'A new epsilon-dominance hierarchical bayesian optimization algorithm for large multiobjective monitoring network design problems', *Advances in Water Resources* **31**(5), 828 – 845.
URL: *http://www.sciencedirect.com/science/article /pii/S0309170808000298*

References IV

Krejca, M. S. & Witt, C. (2017), Lower bounds on the run time of the univariate marginal distribution algorithm on onemax, *in* ‘Proceedings of Foundation of Genetic Algorithms, FOGA’17’.

Mühlenbein, H. & Paaß, G. (1996), From recombination of genes to the estimation of distributions i. binary parameters, *in* H.-M. Voigt, W. Ebeling, I. Rechenberg & H.-P. Schwefel, eds, ‘Parallel Problem Solving from Nature - PPSN IV’, Vol. 1141 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 178–187.

Pelikan, M., Goldberg, D. E. & Cantú-Paz, E. (2000), ‘Linkage problem, distribution estimation, and bayesian networks’, *Evolutionary Computation* **8**, 311–340.

Pelikan, M. & Muehlenbein, H. (1999), ‘The bivariate marginal distribution algorithm’, *Advances in Soft Computing* pp. 521–535.

Thierens, D. (2010), The linkage tree genetic algorithm, *in* ‘Proceedings of the 11th International Conference on Parallel Problem Solving from Nature: Part I’, PPSN’10, Springer-Verlag, Berlin, Heidelberg, pp. 264–273.
URL: *http://dl.acm.org/citation.cfm?id=1885031.1885060*

Yu, T.-L., Santarelli, S. & Goldberg, D. E. (2006), *Military Antenna Design Using a Simple Genetic Algorithm and hBOA*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 275–289.
URL: *http://dx.doi.org/10.1007/978-3-540-34954-9_12*

Zinchenko, L., Mühlenbein, H., Kureichik, V. & Mahnig, T. (2002), Application of the univariate marginal distribution algorithm to analog circuit design, *in* 'Proceedings of the 2002 NASA/DOD Conference on Evolvable hardware'.