Evolution Strategies with an RBM-Based Meta-Model

Kirill Makukhin

The University of Queensland, Brisbane, Australia
mailto:k.makukhin@webage.net.au

Abstract. Evolution strategies have been demonstrated to offer a state-of-the-art performance on different optimisation problems. The efficiency of the algorithm largely depends on its ability to build an adequate meta-model of the function being optimised. This paper proposes a novel algorithm RBM-ES that utilises a computationally efficient restricted Boltzmann machine for maintaining the meta-model. We demonstrate that our algorithm is able to adapt its model to complex multidimensional landscapes. Furthermore, we compare the proposed algorithm to state-of the art algorithms such as CMA-ES on different tasks and demonstrate that the RBM-ES can achieve good performance.

Keywords: evolution strategies, restricted Boltzmann machine, meta-model, surrogate model, estimation of distribution algorithm.

1 Introduction

Evolution Strategies (ES) are a sub-set of genetic algorithms that use only mutation operator for generating offspring. The ES have been demonstrated to offer a state-of-the-art performance on different optimisation problems with discontinuous, non-differential, noisy and multimodal environments, often outperforming gradient-based techniques (e.g. [1]).

In general, a well acceptable solution to a problem can be found by an ES algorithm after a large number of fitness evaluations. However, such evaluations in many real-world applications are not trivial and/or resource-wise expensive. Various approaches has been proposed to reduce this cost by exploiting the history of previously evaluated exemplars. One of successive approaches is focused on building an approximate probabilistic meta-model of the target function (or surrogate function) that allows to sample offspring only in promising areas.

There is a large variety of meta-models used in conjunction with ES. One kind of algorithms are using Gaussian centring the peak to the global (hopefully) extremum [2, 3]. Other algorithms tries to estimate the target function landscape, or its fraction around the global extremum. They include statistical models with local regression [4], radial basis function interpolation [5], covariance matrix adaptation [6], Gaussian process model [7], Kriging interpolation [8], and adaptive Gaussian mixture models [9] to name a few; see also [10] for a comprehancive review.

In this paper, we propose a new algorithm that utilises a product of Gaussians in a form of a restricted Boltzmann machine that is well-suited for the approximation of

Y.S. Kim et al (Eds.): PKAW 2014, LNCS 8863, pp. 246–259, 2014.

complex *multimodal* target functions, thus improving the performance of (μ,λ) ES in such environments.

This paper first briefly describes the concept of restricted Boltzmann machines, then it introduces the RBM-ES algorithm. Afterwards, the algotithm is analised on an artificial multimodal landscape, compared with other ES algorithms on the BBOB platform [11], and it is finally evaluated on a 'real' cart-pole nalancing control problem in a simulated environment.

2 Product of Gaussians

Combining tractable models (such as Gaussian) to a mixture seems to be beneficial, because the mixture can approximate complex objective functions much better than individual models, especially in a multimodal case. However, mixtures of models seem to be inefficient in high-dimentional spaces [12].

Another way of combining models is by taking the product of individual models followed by renormalisation. If the models are Gaussians with one or more latent variables, then their product becomes a powerfull tool for modelling complex distributions with nearly arbitrary shape and sharpness, and the sharpness is not limited by the width of each individual kernel [13]. A well-know example of such a product is a restrictred Boltzmann machine (RBM). RBMs and their modifications have been excessively studied for the last decade, and they have been shown to be very efficient in many classification, recognition and dimensionality reduction tasks [14–22].

Generally speaking, a binary RBM is a two-layer network of visible and hidden units that are bi-directionally connected to each other with no lateral connections within layers, Fig. 1. An RBM layout. The units make stochastic decisions about whether to be "on" or "off" with probability defined by their net input [12, 23].

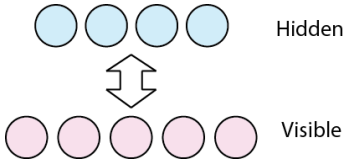

Fig. 1. An RBM layout

Specifically, the probability of turning a unit "on" is:

$$p(h_j|\mathbf{v}) = \frac{1}{1+e^{-z_j}}, \tag{1}$$

$$z_j = b_j + \sum_i v_i \cdot w_{ij},$$

where w_{ij} is the weight of the connection between the *i-th* visible and *j-th* hidden units, *b*, *c* are visual and hidden units' biases respectively.

The binary RBM could be extended to encode real-valued variables in few different ways [16]. In our implementation we have used an RBM with Gaussian visible and binary hidden units [24, 25], where Gaussian units denote linear units with added Gaussian noise.

Under this real-valued model, the conditional probabilities become:

$$p(v_i|h) = \mathcal{N}\left(c_i + \sum h_j w_{ij}\,, \sigma^2\right)\,, \tag{2}$$

$$p\left(h_j|v\right) = \frac{1}{1 + \exp\left(-\frac{z_j}{\sigma^2}\right)}\,, \tag{3}$$

where $\mathcal{N}(m, \sigma^2)$ denotes Gaussian noise with mean m and variance σ^2.

The model is trained to minimise the joint probability of visible and hidden units $P(\boldsymbol{v},h)$. For improving the efficiency of ES, we want to learn $P(\boldsymbol{v},h)$ in a way that makes it as close as possible to the target function.

In [12] has been proposed a fast method of trining RBMs, namely contrastive divergence (CD), where the weight are updated according to the following equation:

$$\Delta w_{ij} = \langle v_i h_j \rangle^{data} - \langle v_i h_j \rangle^{equilibrium}\,. \tag{4}$$

In this equation, $\langle\,\rangle^{data}$ denotes the expected value of the input multiplied with the inferred hidden states while the input is clamped on the data points, and $\langle\,\rangle^{equilibrium}$ is the expectation of $v_i h_j$ when the alternating Gibbs sampling of the hidden and visible units was (infinitely) iterated to get samples from the equilibrium distribution. Fortunately, it was shown that the learning could be acceptably efficient even if the Gibbs sampling chain has been stopped after the first update.

Importantly, an RBM is a generative model. That is, by alternating Gibbs sampling steps until equilibrium state [26], it is possible to obtain an unbiased exemplar from the model's probability density distribution [16] that allows efficiently produce offspring for ES. Similarly to the learning stage, the Gibbs sampling chain in practice could be stopped after a few iterations.

3 RBM-ES Algorithm

The proposed RBM-ES algorithm enhances the efficiency of evolution by generating offspring from a meta-model that mimics a target function.

The algorithm is conceptually derived from the simplest form of the real-valued ES, namely PBIL-C [3]. Unlike the PBIL-C, in our algorithm the meta-model is built upon a Gaussian-binary RBM that allows to perform efficiently both the adaptation of the model to the objective function, and the sampling from the learned probability distribution.

A naïve version of the algorithm pseudo-code is listed in Alg. 1. The first population of μ individuals is uniformly randomly generated keeping in mind that it should cover the whole search space. Afterwards, the best λ individuals are used as parent samples to train the RBM. Since then, every new offspring is generated by Gibbs

sampling from the meta-model. It usually helps to add some exponentially decaying noise to the step of sampling of visual units to enforce more exploration at the beginning, and also to improve initial search space coverage:

$$p(v_i|h) = \mathcal{N}\left(c_i + \sum h_j w_{ij}\,, \sigma^2\right) + \mathcal{N}(0, T)\ . \tag{5}$$

Alg. 1. The RBM-ES algorithm

```
1   initialise RBM(w, b, c, σ²), T
2   generate first population X⁰ with uniform distribution
3   repeat until converge or other stop criteria
4       select best λ individuals X^best from population X^t
5       train RBM with CD algorithm on X^best
6       for k = 1 to μ                  % generate new population X^t
7           v = N(0,1)
8           for n = 1 to g              % Gibbs sampling
9               calculate h, eq. (3)
10              calculate v, eq. (5)
11          end
12          x_k = v                     % update offspring with the new individual
13      end
14      T = εT                          % decay the added noise
15  end
16  return individual with the best fitness
```

The RBM is trained to minimise joint probability of data and hidden units with CD method described earlier. We have found that updating weighs with momentum $\alpha = 0.7\,..\,0.9$ improves stability and increases the speed of the learning of the meta-model [16]:

$$w^t = \alpha w^{t-1} + \Delta w\,. \tag{6}$$

The variance σ^2 in eq. (5) is set to 0.4 (see Discussion section for more information on the choice of this parameter). The noise decay value ε is set to 0.99, but it might require to be adjusted to account for the complexity and dimensionality of the target function.

The Gibbs sampling chain (lines 8-11 in Alg. 1) in theory should be repeated until the system reaches its equilibrium. However in practice, small number of iterations is usually sufficient. That is, we use $g = 6$ iterations in all our experiments.

The learning rate should be set to a small value; otherwise it might lead to the explosion of the RBM's weights. The value of 0.0005 seems to be a good starting point.

4 Experimental Study

This section describes three experiments. First, we evaluate the adaptation of the meta-model probability distribution on a non-linear multimodal function, illustrating that the RBM is indeed a powerful tool for modelling complex probability density functions (pdf). Next, we run a benchmark test in order to compare the RBM-ES performance to other algorithms. Specifically, we use the BBOB (black-box optimisation benchmarks) platform [11]. Finally, we evaluate the proposed algorithm on a practical task and compare it with other algorithms belonging to the ES family.

4.1 The Adaptation of the Meta-Model's Probability Density to a Target Function

This study illustrates the adaptation of an RBM-based meta-model to a target function. Specifically, a variation of the Gallagher 101 Gaussian peaks function from the BBOB platform was used.

The experiment uses the naïve version of the algorithm listed in Alg. 1 with the parameters described in the section above. The number of RBM's hidden units was 10. We run the algorithm to minimise the target function and plot the probability density function of the meta-model every few dozen generations.

The results are presented in Fig. 2. The fine line contour map (magenta in colour version) shows a 2D crossection of the target function with three deep optima, where the central minimum is the global one. Theoverlaping solid line contour map shows the RBMs pdf. Thus, the four consequent plots (A-D) show the evolution of the meta-model pdf with training.

The problem of finding the global minimum of functions like Gallagher 101 seems to be difficult for many black-box optimisation algorithms, because it has multiple local extremes and valleys. Intuitively, having an algorithm with a meta-model that is able to explore multiple peaks at the same time seems to be advantageous comparing to uni-modal meta-models. Indeed, the RBM-ES demonstrates such an ability. Plot A shows the initial pdf that has a bowl-like shape. With RBM learning, the probability distribution shape gradually changes (Plots B and C) to account for the target function landscape, covering all deepest minima. This allows to localise the sampling of the offspring to promising areas. Finally, the pdf become centred over the global minimum, Plot D.

Although this experiment does not explicitly evaluates the performance of the proposed algorithm, it does uncover the important process underlying the algorithms strength – its ability to handle multimodal functions. The following experiments will compare the performance in more challenging environments.

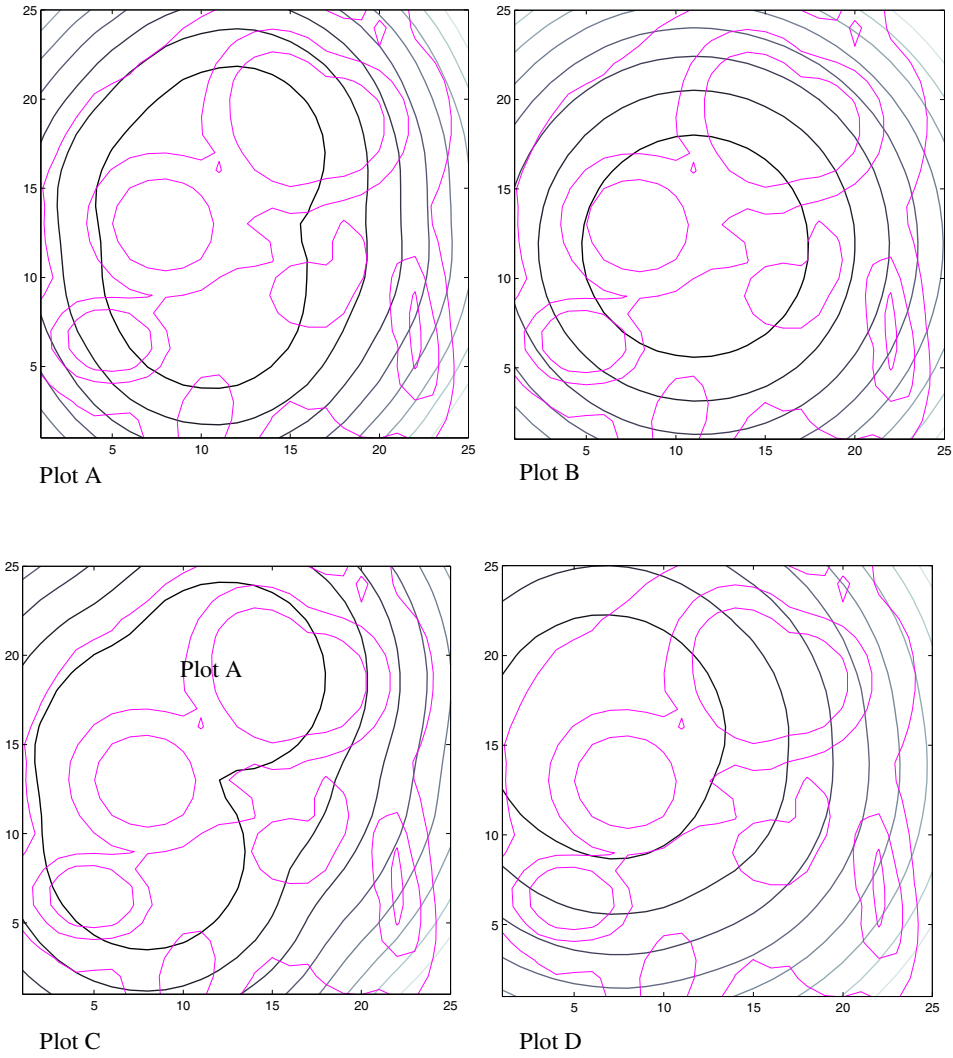

Fig. 2. The adaptation of the meta-model's probability density to the target function in 2D projection. Plot A shows the initial pdf of the meta-model. Plots B and C shows the adaptation of the meta-model in order to account the deepest minimum. Plot D shows the final shape of the pdf that has became centred over the global minimum. The fine-line contour map (magenta in colour version) is the target function, and the solid-line contour is the model pdf. The global minimum is located approximately at (7,15). Best viewed in colour.

4.2 Comparing the RBM-ES with other Algorithms on the BBOB Black-Box Optimisation Benchmark Platform

The following experiment evaluates the performance of the RBM-ES algorithm on the BBOB 2013 platform with noiseless functions. The platform allows comparing results with many algorithms presented on GECCO conferences.

The testbed contains 24 real-valued functions that have high complexity and are known to be difficult to minimise. The benchmarking procedure measures how many times problems were solved by an algorithm within limited budged (the number of function evaluations) for different dimensions and for different target precision values.

Fig. 3 presents results for 5-D and 20-D problems for all functions (right plots), as well as for a multimodal subgroups of functions (left plots) that appears to be more difficult to optimise [27]. The charts also show the performance of few other algorithms belonging to the ES family: BayEDAcG [9], Bipop-CMAES [28], Lmm-CMAES [1]. In addition, the top curve with a diamond-shaped marker (in gold on the colour version) shows the best performance achieved on the BBOB'2009 workshop.

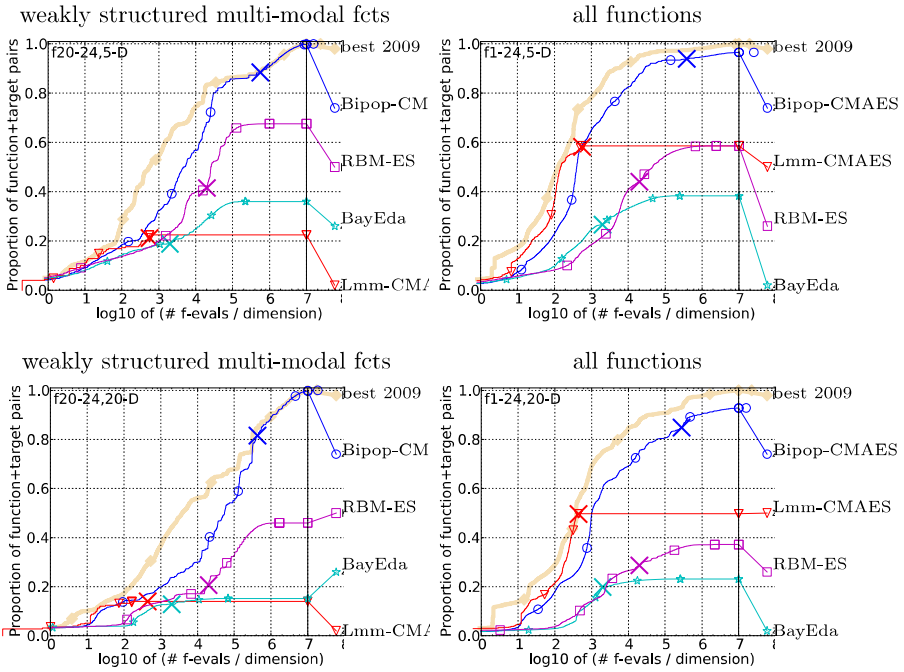

Fig. 3. Evaluation of the RBM-ES performance. The plots show the proportion of the function optimised w.r.t. the number of function evaluations divided by the dimension of the problem. Higher curves represent better performance. Best viewed in colour.

The plots show the number of solved problems with respect to the runtime measured as the number of function evaluations divided by dimension. The number of

problems include all functions (right plots) or only multi-model weakly structured functions (left plots). Every function is optimised several times with different target precisions from 10^{2} to 10^{-8}. For detailed testbed setup, please refer to [11].

The variant of the proposed RBM-ES algorithm demonstrates high performance on the BBOB testbed and is able to achieve comparable results with other ES algorithms, outperforming some of them, especially on the multimodal subclass of functions. In the next experiment we will evaluate the RBM-ES on a more practical multidimensional.

4.3 The Evaluation of Performance on a Practical Task

In this experiment we evaluate the proposed algorithm for more practical purposes and compare it with other algorithms belonged to the ES family. Specifically, in this experiment ES algorithms are compared on the task of evolving weights of a feed-forward neural network that is used as a controller for the classical cart-pole balancing problem.

The idea of using genetic algorithms to train artificial neural networks for reinforcement learning (RL) tasks is not new, and it has been around for few decades. Resent research has shown that often this approach is superior comparing to plain reinforcement learning, especially for complex tasks, because RL algorithms tend to suffer from the curse of dimensionality [29–32].

The purpose of the controller is to balance the pole, keeping the cart in the centre of the available area as long as possible by applying forces to the cart. The simulator math and sample code for the problem are readily available, i.e. as described in [33]. The task is episodic, and it ends either after 800 successive iterations or when the pole angle exceeds the critical value of ±15 degrees. In the case of losing balance, the controller is penalised with a negative reward -1. Every successive iterations is rewarded by +0.01, thus the total maximum reward is equal to 8. Keeping the cart position close to the centre is reinforced by adding s negative reward in proportion to the offset with a coefficient -0.005.

In this experiment we used a feed-forward network with one hidden layer that has four input units, ten hidden and two output logistic units, Fig. 4. Thus, the dimensionality of the problem, as the number of weights and biases, is equal to 72, which makes the problem relatively hard to solve.

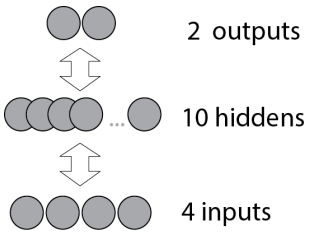

Fig. 4. The controller neural network layout

The input units receive the following pole and cart parameters:
x – cart position;
$\dot{x}$ – cart velocity;
φ – pole inclination;
$\dot{\varphi}$ – pole angular velocity.

The weight of the cart is 1kg, the weight and length of the pole are 0.1kg and 1m respectively. The simulation time step is 0.02 s. The force applied to the cart can be ±10N and the sign depends on weather the output of the first unit is larger than the second one.

We evaluated three different algorithms on the problem: PBIL-C [3], CMA-ES [34], and our RBM-ES. We have picked the CMA-ES algorithm for comparison, because it can be considered as a state-of-the-art algorithm and its Octave/Matlab code is available. Conversely, we included the PBIL-C, the simplest algorithm in the family of ES, as a unitary reference point. Every individual (as the set of controller network's weights) was tested on the cart-pole task five times with slightly different initial cart and pole position to get accurate fitness estimates.

First, we tested the basic ES algorithm PBIL-C. The PBIL algorithm is known to have a good performance, but it's application is limited to separable problems [35]. Thus not surprising, it was able to find a solution to the problem only occasionally, see Fig. 5. In contrast, the CMA-ES has demonstrated much better performance and was able to find a solution at almost every run, Fig.6. Finally, we evaluated the proposed RBM-ES algorithm, Fig. 7. Noticeably, this algorithm was able to find a solution every run.

Our results are inline with the previous work that evaluates different algorithms, including the CMA-ES, on the pole balancing problem [31, 32].

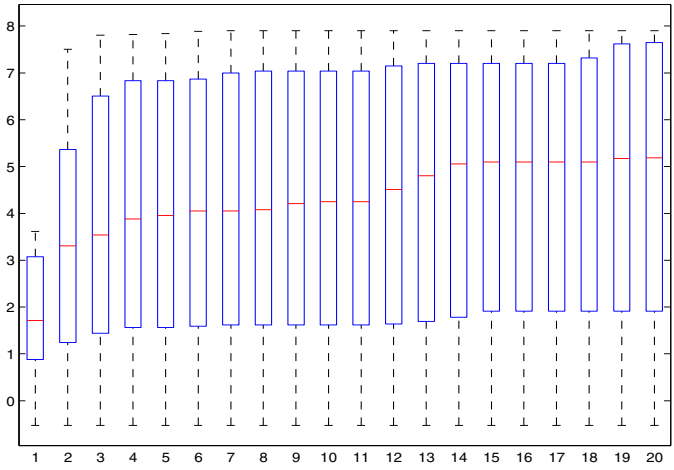

Fig. 5. The evaluation of the PBIL-C algorithm. The vertical axis represents the total average reward, the horizontal is the iteration number $^{x}10$. The maximum possible reward is 8. The central mark is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme values.

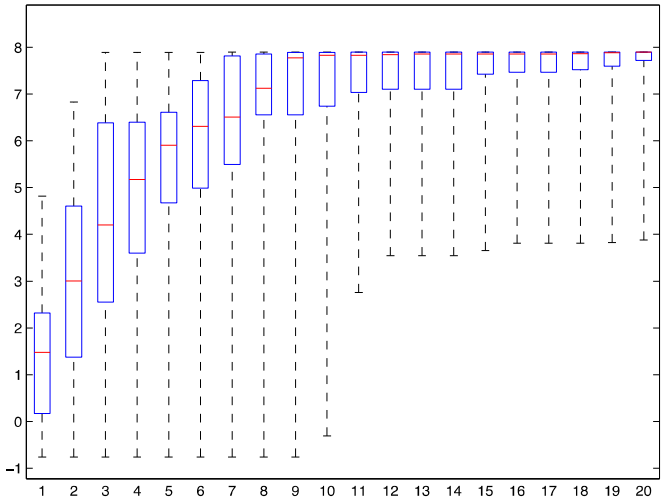

Fig. 6. The evaluation of the CMA-EC algorithm, a naïve version without restarts. See description of Fig. 5.

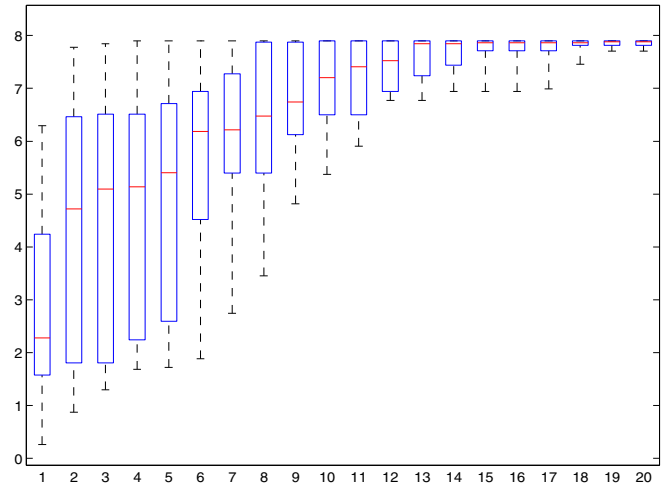

Fig. 7. The evaluation of the RBM-ES algorithm. See description of Fig. 5.

5 Discussion

Apparently, the efficacy of the adaptation of a meta-model to a target function is one of the most important properties of any ES algorithm. The first experiment shows that the proposed approach provides an effective tool for such adaptation. The complex meta-model allows to sample candidate solutions from the most promising areas.

Interestingly, the problem of being trapped in a local minimum seems to be less severe for the RBM-ES comparing to algorithms that maintain a uni-modal model of the target function. Fig. 2 provides an example of the adaptation of the pdf to a function with multiple local extrema. Noticeably, the RBM tries to cover every good candidate to the global solution, thus decreasing the chance of being attracted by one of the local minima instead of the global one.

The second experiment compares the performance of the propose algorithm with several state-of-the art algorithms from the same class of evolution strategies. The RBM-ES demonstrates comparable results and even outperform some algorithms, especially on the multimodal class of problems.

It is worth mentioning that we used a naïve variant of the RBM-ES in this experiment, while other algorithms were specifically tuned for the competition (e.g. multiple restarts, adaptive learning rate, etc.). In addition, the BBOB problems are typically designed with the goal of achieving a very low residual error up to 10^{-8}. This magnitude of error is hardly achievable with a stochastic model based on the RBM with Gaussian-binary units. Although such low error is not always desired in practical tasks, it seems that it is possible to further improve the algorithm performance by proper scaling of the data, and we leave this to the future work.

The final experiment with the pole balancing controller explores a relatively complex optimisation problem with large dimensionality, but presumably simple landscape. Nevertheless, the simplest algorithm from the ES family PBIL-C has not shown good performance, being able to find a solution only occasionally. In contrast, a naïve version of the RBM-ES algorithm achieved the performance slightly better than the CMA-ES, demonstrating relatively quick and robust convergence to a solution.

This cart-pole balancing experiment did not imply noisy environments. However, all the algorithms evaluated are stochastic in nature, and our preliminary tests do not show any significant difference with the addition of a moderate level of noise to the position and velocity "readings" from the cart-pole simulator. Thus, we believe that the RBM-ES algorithm would have shown good results on noisy problems as well.

Comparing to the CMA-ES, our algorithm has more parameters to tune. First, the RBM's hidden layer unit number must be chosen depending on the complexity of the target function. Although there is no a precise recipe for choosing the RBM layout, all common sense methods dealing with the trade-off between complexity and possible overfitting apply here. These methods may also include the addition of a regulariser such as weight decay or sparsity regulation [16].

Another RBM parameter, the variance of the Gaussian visual units σ^2, has a great effect on the shape and sharpness of the learned distribution. The influence of this parameter is discussed in the recent work [25], and a few practical advises are given in [16].

There are also two evolution-specific meta-parameters, λ and μ. Clearly, the parameters must satisfy $\lambda \leq \mu$. We were using $\lambda=3$ and $\mu=9$ in our experiments, an we found that the common approach of changing these values depending on the dimensionality of the problem (e.g. [1]) does not seem to noticeably improve the efficiency, measured as the number of target function evaluations.

While approaches for optimising both RBMs and evolution strategies are well explored separately and described in the literature, it seems also beneficial to implement self-adaptation mechanisms, i.e. as discussed in [36, 37]. This would allow to make

the algorithm more robust for accommodating a wider range of different tasks and we will be working in this direction in future.

6 Conclusion

The paper proposed a new algorithm RBM-ES from the evolution strategies family, where the offspring are generated from the probability distribution function of a meta-model, formed by a restricted Boltzmann machine.

The RBM-based meta-model seems to be highly efficient in learn functions with complex landscapes. Specifically, there are two appealing properties of the RBM that are exploited in the proposed RBM-ES algorithm. First, it is able to model *multimodal* distributions that allows doing parallel search in multiple modes. This property helps dealing with the problem of being trapped in a local minimum that is a typical issue for algorithms that maintain a uni-modal model of the target function.

Second, the RBM is able to model distributions that are narrower than its each expert, thus easing another important problem – choosing the initial sigmas. Unlike the RBM-ES, it is know that some algorithms can become unstable and diverge if the initial the deviation was chosen incorrectly [38].

Finally, the RBM-ES is computationally efficient, because RBMs offer a simple and efficient mechanism for sampling from the model's distribution and training.

Acknowledgments. The author thanks Marcus Gallagher and Gleb Yakubov for fruitful discussions, and anonymous reviewers for useful comments that helped to improve this paper. This work was funded by auPilot project.

References

1. Auger, A., Brockhoff, D., Hansen, N.: Benchmarking the Local Metamodel CMA-ES on the Noiseless BBOB'2013 Test Bed. In: GECCO 2013 (2013)
2. Baluja, S., Caruana, R.: Removing the genetics from the standard genetic algorithm. In: ICML, pp. 1–11 (1995)
3. Sebag, M., Ducoulombier, A.: Extending population-based incremental learning to continuous search spaces. Parallel Probl. Solving from Nat. 5, 418–427 (1998)
4. Branke, J., Schmidt, C., Schmec, H.: Efficient Fitness Estimation in Noisy Environments. In: Proceedings of Genetic and Evolutionary Computation (2001)
5. Gutmann, H.-M.: A Radial Basis Function Method for Global Optimization. J. Glob. Optim. 19, 201–227 (2001)
6. Olhofer, M., Sendhoff, B.: A framework for evolutionary optimization with approximate fitness functions. IEEE Trans. Evol. Comput. 6, 481–494 (2002)
7. Büche, D., Schraudolph, N.N., Koumoutsakos, P.: Accelerating Evolutionary Algorithms With Gaussian Process Fitness Function Models. IEEE Trans. Syst. Man, Cybern. Part C Appl. Rev. 35, 183–194 (2005)
8. Runarsson, T.P.: Constrained evolutionary optimization by approximate ranking and Surrogate models. In: Yao, X., et al. (eds.) PPSN VIII. LNCS, vol. 3242, pp. 401–410. Springer, Heidelberg (2004)

9. Emmerich, M., Giotis, A., Özdemir, M., Bäck, T., Giannakoglou, K.: Metamodel-Assisted Evolution Strategies. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN VII. LNCS, vol. 2439, pp. 361–370. Springer, Heidelberg (2002)
10. Gallagher, M.: Black-box optimization benchmarking: Results for the BayEDAcG algorithm on the noiseless function testbed. In: GECCO 2009, p. 2383. ACM Press, New York (2009)
11. Forrester, A., Sobester, A., Keane, A.: Engineering Design via Surrogate Modelling A Practical Guide. Wiley (2008)
12. Hansen, N., Auger, A., Finck, S., Ros, R.: Real-Parameter Black-Box Optimization Benchmarking 2009: Experimental Setup (2009)
13. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Comput. 14, 1771–1800 (2002)
14. Martens, J., Chattopadhyay, A., Pitassi, T., Zemel, R.: On the Representational Efficiency of Restricted Boltzmann Machines. In: NIPS, pp. 1–21 (2013)
15. Hinton, G.E.: To Recognize Shapes, First Learn to Generate Images. Prog. Brain Res. 165, 535–547 (2007)
16. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning. ACM Press (2010)
17. Hinton, G.E.: A Practical Guide to Training Restricted Boltzmann Machines (2010)
18. Desjardins, G., Courville, A.: Parallel tempering for training of restricted Boltzmann machines. In: AISTATS, pp. 145–152 (2010)
19. Larochelle, H., Mandel, M.: Learning Algorithms for the Classification Restricted Boltzmann Machine. J. Mach. Learn. Res. 13, 643–669 (2012)
20. Desjardins, G., Bengio, Y.: Empirical evaluation of convolutional RBMs for vision (2008)
21. Luo, H., Shen, R., Niu, C.: Sparse Group Restricted Boltzmann Machines. Arxiv Prepr. arXiv1008.4988, pp. 1–9 (2010)
22. Cho, K., Ilin, A., Raiko, T.: Improved learning of Gaussian-Bernoulli restricted Boltzmann machines. In: Honkela, T., Duch, W., Girolami, M., Kaski, S. (eds.) ICANN 2011, Part I. LNCS, vol. 6791, pp. 10–17. Springer, Heidelberg (2011)
23. Cho, K., Raiko, T., Ilin, A.: Enhanced gradient and adaptive learning rate for training restricted Boltzmann machines. Neural Computation 25(3) (2011)
24. Welling, M.: Product of Experts, http://www.scholarpedia.org/article/Product_of_experts
25. Welling, M., Rosen-Zvi, M., Hinton, G.E.: Exponential family harmoniums with an application to information retrieval. In: Adv. Neural Inf. Process. Syst. (2005)
26. Wang, N., Melchior, J., Wiskott, L.: An analysis of Gaussian-binary restricted Boltzmann machines for natural images. In: ESANN (2012)
27. Bengio, Y.: Learning Deep Architectures for AI. Found. Trends Mach. Learn. 2, 1–127 (2009)
28. Hansen, N., Auger, A., Ros, R., Finck, S., Pošík, P.: Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In: Proceedings of the 12th Annual Conference Comp. on Genetic and Evolutionary Computation, GECCO 2010, p. 1689. ACM Press, New York (2010)
29. Hansen, N.: Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference, GECCO 2009, p. 2389. ACM Press, New York (2009)

30. Taylor, M.E., Whiteson, S., Stone, P.: Comparing evolutionary and temporal difference methods in a reinforcement learning domain. In: GECCO 2006, p. 1321. ACM Press, New York (2006)
31. Heidrich-Meisner, V., Igel, C.: Similarities and differences between policy gradient methods and evolution strategies. In: ESANN, pp. 23–25 (2008)
32. Riedmiller, M., Peters, J., Schaal, S.: Evaluation of Policy Gradient Methods and Variants on the Cart-Pole Benchmark. In: 2007 IEEE ISADPRL, pp. 254–261 (2007)
33. Hansen, N.: The CMA Evolution Strategy: A Comparing Review. In: Lozano, J.A., Larrañaga, P., Inza, I., Bengoetxea, E. (eds.) Towards a New Evolutionary Computation. STUDFUZZ, vol. 192, pp. 75–102. Springer, Heidelberg (2006)
34. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press (2004)
35. Whitacre, J.M.: Adaptation and Self-Organization in Evolutionary Algorithms (2007)
36. Meyer-Nieberg, S., Beyer, H.: Self-adaptation in evolutionary algorithms. In: Lobo, F.G., Lima, C.F., Michalewicz, Z. (eds.) Parameter Setting in Evolutionary Algorithms. SCI, vol. 54, pp. 47–75. Springer, Heidelberg (2007)