Optimizing Without Derivatives: What Does the No Free Lunch Theorem Actually Say?

Loris Serafino

One of the most important stages in many areas of engineering and applied sciences is modeling and the use of optimization techniques to increase the quality and performance of products or processes. Financial market operations, radiation therapy technologies, protein folding determination, material sciences, and power system facilities design: developments in these disparate scientific and technological areas, as well as many others, rely also on the help of some quiet, hidden workers called optimization methods. Beyond what seems to be just a machine-like application of some algorithm, there exists a world of challenges and developments. Generally in literature the term *optimization* is related to (the output of) a mathematical technique or algorithm used to identify the extreme value of an arbitrary objective function through the manipulation of a known set of variables and subject to a set of constraints. More technically, a maximization problem with an explicit objective can in general be expressed in the following mathematical form: Finding the value

$$\arg\max_{x \in \mathcal{H}} f(x),$$

where x is a given vector in a generic multidimensional space $\mathcal{H}$ and $f : \mathcal{H} \to \mathbb{R}$ is a function of the vector x and $\mathcal{H} \subset \mathbb{R}^n$ is a (discrete or continuous, but here the focus will be on continuous) subset of the multidimensional real Euclidean space. From now on we will refer to $\mathcal{H}$ as the *search space.*[1] In most real-world engineering optimization problems, no analytical expression exists for accurately evaluating the response of a candidate solution. Sometimes the objective function consists just in the possibility to observe different sets of pairs of input and output from a computational simulation or an experiment: $\mathcal{D} = \{(x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$. This is the ***black-box scenario*** that will be considered here. In black-box optimization many issues are at stake. Some good reviews are [2, 3, 4]. Further, a number of engineering design tasks as well those in other contexts are modeled as multi-objective problems; this makes the optimization process even harder, but this case will not be considered here. Here the focus will be mainly on the following issues: the practical meaning of the ***No Free Lunch Theorem*** (NFLT) and on its natural connection with the Bayesian inference. This choice is due to the fact that they both represent a critical link between optimization theory and optimization practice.

Optimizing in the Black-Box Scenario

Black-box optimization is the reign of *metaheuristics*. A metaheuristic describes the way an optimization method decides which part of the search space to explore in the next step [1]. In general, every metaheuristic can be abstractly

Loris Serafino is mathematics lecturer at the Australian College of Kuwait. His email address is geoloris@gmail.com.

DOI: http://dx.doi.org/10.1090/noti1140

[1] *Even if there is no explicit mention of constraints here, the formulation is nonetheless general enough since they can be incorporated through an appropriate definition of the search space $\mathcal{H}$.*

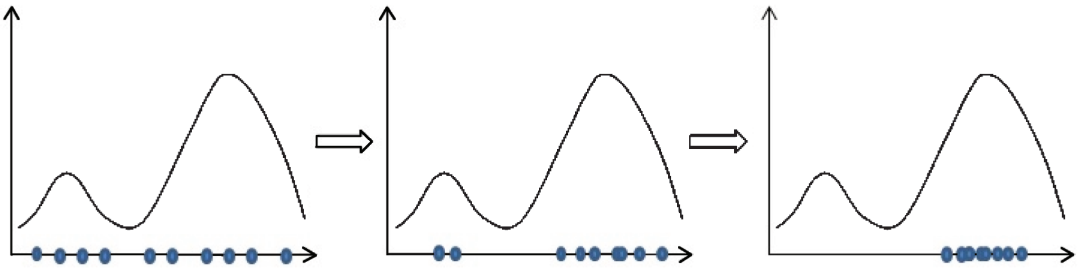

Figure 1. Genetic Algorithm: Generation after generation the population converges to a good solution.

considered as a sampling process acting on the search space. It starts with a set of values, and then step after step it generates new samples according to some specified mechanism based on current samples and the objective function values:

$$\left(x_1^{i+1},\ldots,x_m^{i+1}\right) = Alg\left(x_1^i,\ldots,x_n^i, f\left(x_1^i\right),\ldots,f\left(x_n^i\right),\Theta\right)$$

where Θ is a random variable and index i is the iteration counter.

So differences among different optimization methods come from the specific mechanism an algorithm uses to generate and accept new samples and in so doing alternating an exploration (global) phase with an exploitation (local) phase. For example, just to mention one class, *nature-inspired* algorithms, like the well-known genetic algorithm [5], are based on the idea of mimicking some natural phenomena that leads to the maximization of some defined quantity. Starting with a randomly generated population of candidate solutions (called chromosomes in usual terminology), a Genetic Algorithm (GA) carries out a process of fitness-based[2] selection and recombination to produce a successor population, the next generation. During recombination, *parent* solutions are selected and their genetic material is recombined to produce child chromosomes. After this step, in practical implementation, a mutation operator is applied. Mutation perturbs the recombined solutions slightly to explore their immediate neighborhood. These then pass into the successor population. As this process is iterated, a sequence of successive generations evolves and the average fitness of the chromosomes tends to increase until some stopping criterion is reached. In this way, a GA "evolves" a best solution to a given problem (see Figure 1). In the process of evolution, one population is replaced by another and so on.

GA represents just one tool in the hands of the practitioner. Other examples are: Particle Swarms, Ant Colonies, Simulated Annealing, and many other families [6]. In general, practice shows that any successful application depends on careful tuning of operators, parameters, and problem-dependent features.

Potentially there is an infinity of possible optimization problems, one for any possible function. At the same time there is an infinity of thinkable optimization methods, one for every possible exploration-exploitation trade-off combination. The choice of the correct metaheuristics for a given class of problem is a crucial theme that leads us to take into account the role played by the NFLT in terms of its theoretical and practical relevance.

The Practical Meaning of the No Free Lunch Theorem

We will start with the usual formal statement of the NFLT for optimization [9]. Wolpert and Macready's result considered a finite search space $\mathcal{X}$ and the space of all the possible objective functions $f : \mathcal{X} \mapsto \mathcal{Y}$ defined on it called $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$. They defined with $P(y_k|f, k, Alg)$ the conditional probability of finding a value $y_k \in \mathcal{Y}$ given a function f, after k iterations with algorithm Alg. This can be seen as a performance measure of the algorithm—its ability to locate a given function value after a given amount of iterations. Under some quite general conditions, the theorem states that, for any pair of algorithms Alg_1 and Alg_2:

$$\sum_f P(y_k|f, k, Alg_1) = \sum_f P(y_k|f, k, Alg_2).$$

Where the sum is carried out over the set of all the possible functions $\mathcal{F}$.

According to the most common understanding, the NFLT implies that there is no optimization method superior to others for all possible optimization problems. For some functions Alg_1 will be able to locate the maximum faster than Alg_2; for some other functions it will be the opposite. Averaging over the whole space $\mathcal{F}$, the performance will be the same. Equivalently, it is possible to say that, over $\mathcal{F}$, no algorithm will perform better

[2] *In GA literature the objective function is usually called* fitness.

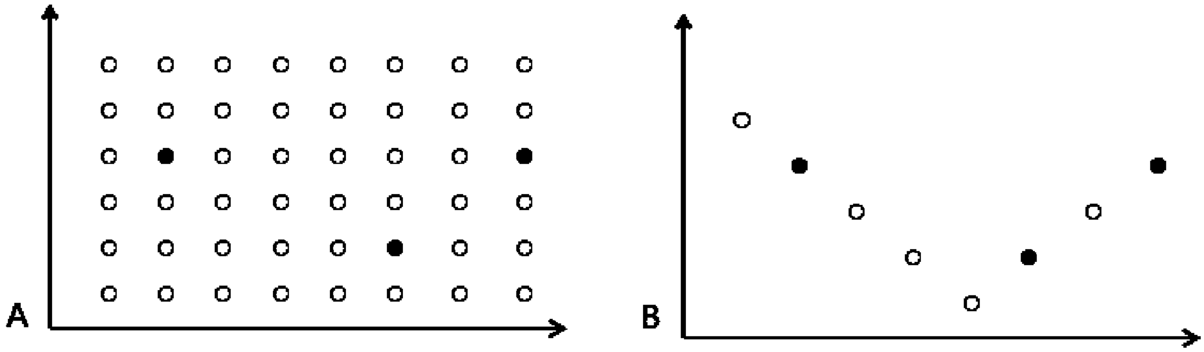

Figure 2. The No Free Lunch Theorem. The information collected so far will not say anything about the values of the function in other regions (case A). For a given subclass of functions, i.e., convex (case B), those algorithms that can take advantage of this structure will perform better than others.

than pure random search. Wolpert and Macready adopted a probabilistic framework. Their result holds if we assume a uniform distribution over $\mathcal{F}$, i.e., any functional form is uniformly admissible, and then they prove it by induction over the index k. To understand the practical implication of this theorem for black-box optimization problems, let's restate it while adopting a different perspective. Substantially the theorem states that:

With no prior knowledge about the function $f : \mathcal{X} \mapsto \mathcal{Y}$, in a situation where any functional form is uniformly admissible, the information provided by the value of the function in some points of the domain will not say anything about the values of the function in other regions of the domain.

This interpretation of the NFLT is pictured in Figure 2. So in this scenario the information collected with the data sample is not helpful in guiding the search as to which direction is better to explore next. In this sense, there is no optimization method superior to others for all possible optimization problems. Of course every function has its own structure; the problem is when prior knowledge about the functional form is not available because no rationale can guide an optimization strategy— i.e., to decide which optimization method to use, which set of parameters, and so on.

The lesson that has to be learned from NFLT is in the implications for a rational optimization strategy able to tackle black-box optimization problems. It is clear now that, for the practitioner the correct question is not *which algorithm I have to use* but first of all *what is the geometry of the objective function*: the optimization problem becomes an inferential problem as will be clear in the discussion below. Knowing the structure of the objective landscape makes it (theoretically) possible to properly tune an algorithm in terms of a trade-off between local search and global search. It is also true that, even if many studies are in progress, general results about which class of algorithms best suits which kind of problems are still far off. As we will see below, Bayesian probability theory will appear to be the natural foundational framework for metaheuristics.

In the NFLT scenario, where nothing is known in the literature about the structure of the problem at hand, practitioners tend to decide the optimization method according their background knowledge, practical availability of code, simulation software, and so on. In a different scenario where something is known about the function, like a lower bound on the function value or some information about the response landscape, this information must be used to tailor the algorithm and the optimization strategy accordingly. Knowledge of the objective function structure is a key to adjust effective algorithms in terms of a better trade-off between exploration and exploitation.

Summarizing, from the discussion of the NFLT, black-box optimization involves two main ingredients: input-output data and some prior knowledge. This leads naturally to the next section where the connection with the Bayesian framework will be explored.

NFLT and the Bayesian Philosophy: Two Sides of the Same Coin

Our interpretation of the NFLT in the context of black-box optimization shows a strong connection with the problem of *underdetermination* (see Figure 3). In the black-box context only a finite sample data set concerning the functional response is available: $\mathcal{D} = \{(x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$. In this case locating a function value (the optimum) becomes a purely inferential problem that leads naturally to a Bayesian framework. The rationale of

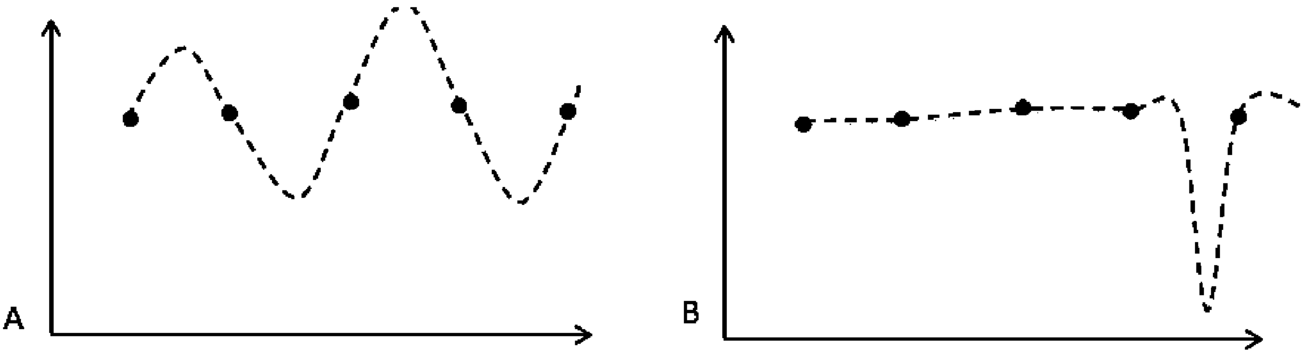

Figure 3. The problem of underdetermination: a sample of data can be described by quite different models (in this case function A and function B).

Given a sample data set $\mathcal{D} = \left\{\left(x_1, f\left(x_1\right)\right), \ldots, \left(x_n, f\left(x_n\right)\right)\right\}$			
NFLT terminology: optimum location		**Bayesian terminology: model generation**	
Uniform over $\mathcal{F}$	Nonuniform over $\mathcal{F}$	No prior	With prior
No *Alg* better than pure random search in locating $\arg\max f(x)$	Some *Alg* better than pure random search in locating $\arg\max f(x)$	No valid inference for the model realizing $\mathcal{D}$	Inference can be made about the model realizing $\mathcal{D}$

Table 1. NFLT vs. Bayesian theory

Bayesian probability theory is briefly summarized here in the context of optimization. The problem is to construct a model of the function that we need to optimize. In the Bayesian framework there are two ingredients: a data set and a prior. A prior distribution $P(f)$ over the space of functions is combined with the likelihood $P(\mathcal{D}|f)$ to generate a posterior

$$P(f|\mathcal{D}) \propto P(\mathcal{D}|f)P(f)$$

which takes into account the information given by the data set $\mathcal{D}$. In terms of modeling a function, if we assume as a prior every possible function (which is equivalent to saying that there is no prior) we are exactly in the NFLT case: no useful inferential information can be extracted from the data set. If the prior can be restricted to a proper class of functions, then the inference about the model realizing the data set will be more accurate. As we said about the implications of the NFLT, if it is possible to restrict the problem to a given subclass of functions, a proper optimization algorithm choice can produce good results. **For an optimization method, given $\mathcal{D}$, to have indications about where promising areas for the optimum can be located is equivalent to saying that a (posterior) refined model of the objective function has been inferred in Bayesian terms**. As is now clear, *there is a strong connection between the NFLT and the Bayesian framework* at the point where they can be considered two sides of the same coin. This situation is summarized in Table 1. The NFLT assumes a uniform distribution over the space of possible functions. In the Bayesian terminology this is equivalent to saying that there is no prior knowledge about the objective function: every functional form is admitted. According to the implication of the NFLT, it is essential to restrict the class of possible functions (in the NFLT terminology) to a proper subclass in order to tailor an effective optimization strategy. This restricted class of functions represents the prior that appears in the Bayesian formula. So the secret of a successful black-box optimization, assuming a *good* sample $\mathcal{D}$,[3] lies in the possibility of narrowing the prior.

Some Implications

The discussion above about the interpretation of NFLT in terms of the Bayesian approach (and vice versa) can be useful in two ways: to use the NFLT for understanding surrogate-based optimization techniques and, on the other side, to use a Bayesian framework for understanding the metaheuristics working logic. In the context of black-box optimization there is a large amount of literature about methods for objective function approximation (also called *meta-models* or *surrogates*) as a way to generate functional models that are computationally efficient and that approximate the true function. All of them *assume*, often implicitly, a prior in terms of a restricted class of modeling functions. Functional surrogate models can be algebraic representations of the true problem functions. The most popular ones are polynomials, in a method often known in the statistical literature as *response surface methodology*. Several related methods are now commonly used for function approximation: neural networks, Kriging model, radial-basis-function networks and support vector machines (for a general reference, see [11, 12]). After the model has been developed, some classical derivative-based or derivative-free methods can

[3] *For Design of Experiments techniques see* [11].

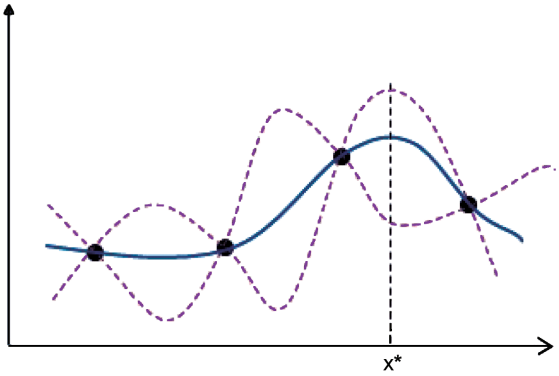

Figure 4. Bayesian optimization. Data points are shown as black dots. The GP modeled distribution has a mean shown by the blue line that fits the data, and a standard deviation represented here by the red dotted lines. Here x^* represents the next point to explore since maximum improvement is expected.

be applied [13]. *Bayesian optimization* is briefly discussed here as a tool that is getting relevance in optimization practice. This technique conceptually combines surrogate estimation with optimum localization (for a good introduction see [14, 15, 16]). It uses the Bayesian rationale to infer about a starting function model. The second step is to infill new points of the search space in those regions where a maximum improvement is expected according to the current best value and the predictive variance (Figure 4). Then the Bayesian procedure is updated. Usually in the literature, the prior is supposed to be a realization of a **Gaussian process**. This is a distribution over the space of functions modeled as a multivariate Gaussian distribution, so one can think of the Gaussian process as defining a distribution over the space of functions:

$$f(x) \sim \mathrm{GP}(m(x), k(x_i, x_j)),$$

meaning that the function f is distributed as a GP this means that function $m(x)$ and covariance function $k(x_i, x_j)$. A common choice for the covariance function is defined by:

$$k(x_i, x_j) = \exp\left(-\tfrac{1}{2}(x_i - x_j)^T \operatorname{diag}(\phi)^{-2}(x_i - x_j)\right),$$

where ϕ represents a family of hyper-parameters that have to be properly estimated.

The power of the Bayesian optimization philosophy briefly sketched out above lies in its ability to tackle optimization problems with a limited number of function calls, and this is very important for computationally expensive simulations. Again, the key point in the Bayesian rationale is given by the choice of the prior, i.e., the nonuniform distribution over $\mathcal{F}$ in the NFLT terminology. The choice of the prior as a *Gaussian process* implies that the function is supposed to be in the class of smooth well-behaved functions; this can be an arbitrary hypothesis. For objective functions with discontinuities or jumps this prior will not be able to provide very good results. Further, the construction of the surrogate model is essentially an exercise of design space exploration exactly in the same way any optimization algorithm result is. If the dimension of the search space is high (more than 100 variables), then the ability to sample promising regions becomes weaker and weaker. Bayesian optimization cannot avoid the *curse of dimensionality*: the number of candidate solutions grows exponentially with increasing dimensionality [7].

From the discussion above it is now clear how a Bayesian framework can be used to shed light on the main metaheuristics operating principles. All metaheuristics *assume* a model class (the Bayesian prior) of the objective function during the search and combine it with the sampled points at a given iteration to direct the search in the most promising directions. For example, genetic algorithms, as well as other optimizers, assume (and they work well if there is) *strong causality*: Small causes produce small effects. For functions with low causality they tend to suffer [2]. In metaheuristics the prior about the function is implicitly defined by the search operators used (*crossover*, *mutation*, etc.) and the set of internal working parameters. Many automated self-adapting strategies which are able to change the internal parameters values according to partial results obtained during the search process have been studied in recent years [8]. What these (so-called *smart*) optimization algorithms do is incorporate a more explicit Bayesian operating logic. They generate a posterior model using the sampled points collected at a given iteration and some assumptions about the function structure (prior). Iteration after iteration, they adjust the set of internal parameters to better tailor the function model and to foster the search in more promising regions (according to the *posterior* model).

Just to mention one, Covariance Matrix Adaptation Evolutionary Strategies (CMA-ES) is one of the most famous in the field of continuous global optimization. Again, the main idea of this algorithm is to use the information collected during the iterations in terms of sampled points to generate *on-the-fly* a model of the function to optimize by assuming a second-order functional structure [17]. This again shows the strong relationship between optimum location, model determination and the prior knowledge about the function geometry over which the optimization process is supposed to be carried out. The internal prior of a metaheuristic defines and constrains the exploration-exploitation trade-off and so the overall capabilities of the search process.

Summarizing, the ability of every optimization method is connected to the following points: a) how much the adopted internal prior model fits the geometry of the problem at hand and b) on the other side, the possibility of access to a *good* sample $\mathcal{D} = \{(x_1, f(x_1)), \ldots, (x_n, f(x_n))\}$ of $\mathcal{H} \times \mathrm{Im}(f)$. This can be a problem for computationally expensive functions and/or for high-dimensional search spaces. As we have said, the curse of dimensionality affects the ability to generate a valid posterior since, with increasing dimensions, the needed number of sampled points able to say something useful in the inferential process increases exponentially.

Conclusions

Summarizing the discussion above:

- Black-box optimization = prior function structure knowledge + sample data set + algorithm.
- In general, knowledge about the objective function landscape is the key for a better optimization in terms of the possibility to better tune the trade-off between local search (exploration) and global search (exploitation).
- The meaning of NFLT can be made clear if we reconsider it in terms of Bayesian logic. At an abstract level, the Bayesian approach is the natural framework on which to base an effective black-box optimization strategy.
- The choice of the ***prior*** is the critical point. The Gaussian process in Bayesian optimization is a valid choice whereas the fitness to be optimized is in the class of smooth functions.
- All metaheuristics more or less implicitly assume a functional model (prior). They work well if this model matches the geometry of the problem under optimization.
- The curse of dimensionality affects the ability to search every optimization method: In high-dimensional search spaces the number of sampled points needed for the inference process increases dramatically.

References

[1] C. Blum and A. Roli, Metaheuristics in combinatorial optimization: Overview and conceptual comparison **35** (3), *ACM Computing Surveys*, pp. 268–308, 2003.

[2] T. Weise, M. Zapf, R. Chiong, and A. J. Nebro Urbaneja, Why is optimization difficult?, *Nature-Inspired Algorithms for Optimisation*, Raymond Chiong (Ed.), pp. 1–50, April 30, 2009.

[3] S. Shan and G. Gary Wang, Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions, *Structural and Multidisciplinary Optimization*, **41** (2), pp. 219–241, March 2010.

[4] Y. Tenne, A computational intelligence algorithm for expensive engineering optimization problems, *Eng. Appl. Artif. Intell.* **25** (5), 1009–1021, August 2012.

[5] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, MA, USA, 1998.

[6] X. S. Yang, Review of metaheuristics and generalized evolutionary walk algorithm, *J. Bio-Inspired Computation* **3** (2), pp. 77–84, 2011.

[7] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, corrected edition, August 2003.

[8] A. E. Eiben, Zbigniew Michalewicz, Marc Schoenauer and James E. Smith, Parameter control in evolutionary algorithms, *Parameter Setting in Evolutionary Algorithms*, 19–46, 2007.

[9] D. H. Wolpert and W. G. Macready, No free lunch theorems for optimization, *IEEE Transactions on Evolutionary Computation* **1** (1), pp. 67–82, 1997.

[10] C. Garcia-Martinez, F. J. Rodriguez and M. Lozano, Arbitrary function optimisation with metaheuristics, *Soft Computing* **16** (12), pp. 2115–2133, December 2012.

[11] A. Sudjianto, K. T. Fang and R. Li, *Design and Modeling for Computer Experiments* (Computer Science & Data Analysis), Chapman & Hall/CRC, 2005.

[12] Y. Jin, A comprehensive survey of fitness approximation in evolutionary computation, Soft Computing—A Fusion of Foundations, *Methodologies and Applications* **9** (1):312, 2005.

[13] A. R. Conn, K. Scheinberg and L. N. Vicente, *Introduction to Derivative-Free Optimization*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2009.

[14] E. Brochu, V. M. Cora and N. de Freitas, A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning, CoRR abs/1012.2599, 2010.

[15] T. Alpcan, A framework for optimization under limited information, *J. Glob. Optim.* **55**, pp. 681–706, 2013.

[16] M. Hoffman, E. Brochu and N. de Freitas, Portfolio Allocation for Bayesian Optimization, UAI, 2011, arXiv:1009.5419v2.

[17] N. Hansen, The CMA evolution strategy: A comparing review, Towards a new evolutionary computation *Advances on Estimation of Distribution Algorithms*, Springer, pp. 1769–1776, 2006.