An Hybrid Neural/Genetic Approach to Continuous Multi-objective Optimization Problems

Mario Costa[1], Edmondo Minisci[2], and Eros Pasero[1]

[1] Politecnico di Torino – Dept. of Electronics
Corso Duca degli Abruzzi 24, 10129 Turin, Italy
{mario.costa,eros.pasero}@polito.it
[2] Politecnico di Torino – Dept. of Aerospace Engineering
Corso Duca degli Abruzzi 24, 10129 Turin, Italy
edmondo.minisci@polito.it

Abstract. Evolutionary algorithms perform optimization using the information derived from a population of sample solution points. Recent developments in this field regard optimization as the evolutionary process of an explicit, probabilistic model of the search space. The algorithms derived on the basis of this new philosophy maintain every feature of the classic evolutionary algorithms, but are able to overcome some drawbacks. In this paper an evolutionary multi-objective optimization tool based on an estimation of distribution algorithm is proposed. It uses the ranking method of non-dominated sorting genetic algorithm-II and the Parzen estimator to approximate the probability density of solutions lying on the Pareto front. The proposed algorithm has been applied to different types of test case problems and results show good performance of the overall optimization procedure in terms of the number of function evaluations.

1 Introduction

The extensive use of evolutionary algorithms in the last decade demonstrated that an optimization process can be obtained by combining effects of interactive operators such as selection - whose task is mainly to identify the best individuals in the current population - and crossover and mutation, which try to generate new and better solutions starting from the selected ones. But, if the mimicking of natural evolution in living species has been a source of inspiration of new strategies, the attempt to copy natural techniques as they are sometimes introduces a great complexity without a corresponding improvement of algorithms performance. Moreover standard evolutionary algorithms can be ineffective when problems exhibit a high level of interaction among variables. This is mainly due to the fact that recombination operators are likely to disrupt promising sub-structures of optimal solutions.

Alternatively, in order to make a rational use of the evolutionary metaphor and/or to create optimization tools that are able to handle very hard problems (with several parameters, with difficulties in linkage learning, deceptive), some algorithms have been proposed that automatically learn the structure of the search space. Following this way, several works, based on explicit probabilistic-statistic tools, have been carried out.

B. Apolloni et al. (Eds.): WIRN VIETRI 2003, LNCS 2859, pp. 61–69, 2003.

Generally, these methods, starting from results of current populations, try to identify a probabilistic model of the search space, and crossover and mutation operators are replaced with sampling. Those methods have been named Estimation of Distribution Algorithms (EDAs).

Most EDAs have been developed to manage optimization processes for mono-objective, combinatorial problems, but several works regarding problems in continuous domains have been proposed.

We can distinguish three types of EDAs depending on the way the probabilistic model is built: a) without dependences among variables [1]; with bivariate dependences among variables [2]; c) with multivariate dependences ([3], [4]).

Recently, EDAs handling multi-objective optimizations have been proposed. References [5] and [6] respectively extend the mono-objective version in [4] and [3]. They describe the algorithms and present some results when applied to well known test problems.

In this paper we propose a multi-objective optimization algorithm for continuous problems that uses the Parzen method to build a probabilistic representation of Pareto solutions, with multivariate dependences among variables.

Similarly to what was done in [6] for multi-objective Bayesian Optimization Algorithm (BOA), the already known and implemented techniques of Non Dominated Sorting Genetic Algorithm II (NSGA-II) [7] are used to classify promising solutions, while new individuals are obtained by sampling from the Parzen model.

The Parzen method, as introduced in the next section, can appear analogous to the normal kernel method described and used in [4]. Actually, the two methods are different and, even if both put kernels on each sampled point, our method uses classical Parzen dictates to set terms of the covariance matrix (non-diagonal) of kernels in order to directly approximate the joint Probability Density Function (PDF).

A brief introduction on the general problem of building probabilistic models is followed by a description of the main characteristics of the Parzen method. In section 3 the structure of the algorithm and the practical implementation are discussed; results of application to test cases are detailed. A final section of concluding remarks summarizes the present work and indicates future developments.

2 Parzen Method

When dealing with continuous-valued random variables, most statistical inferences rely on the estimation of PDFs and/or associated functionals from a finite-sized sample. Whenever something is known in advance about the PDF to be estimated, it is worth exploiting that knowledge as much as we can in order to shape a special-purpose estimator. In fact any additional information we are able to implement in the estimator as a built-in feature is equivalent to some effective increase in the sample size. Otherwise stated, in so doing we improve the estimator's efficiency.

In the statistician's wildest dream some prime principles emerge and dictate that the true PDF must belong to a certain parametric family of model PDFs. This restricts the set of admissible solutions to a finite-dimensional space, and cuts the problem down to the identification of the parameters thereby introduced. In fact parametric estimation is so appealing that few popular families of model PDFs are applied almost everywhere even in lack of any guiding principle, and often little effort is made to check

their actual faithfulness. On the other hand, a serious check has to rely on composite hypothesis tests that are like to be computationally very expensive.

While designing an EDA for general-purpose multi-objective optimization there is really no hint on how the true PDF should look like. For instance, that PDF could well have several modes, whereas most popular models are uni-modal. The possible occurrence of multiple modes is usually handled through *mixtures* of uni-modal kernel PDFs. Since the "correct" number of kernels is not known in advance, the size of the mixture is optimized (e.g. by data clustering) just like any other parameter: that is, the weight and the inner parameters of each kernel.

The usage of mixtures does however not alleviate us from worrying about faithfulness. Otherwise stated, the choice of the parametric family the kernels belong to still matters. In fact the overall number of parameters (and therefore the number of kernels) must grow sub-linearly with the sample size n, or else the variance of the resulting estimator would not vanish everywhere as $n \to \infty$, thus precluding ubiquitous converge to the true PDF in the mean square sense. But if that condition is met, then even a single "wrong" kernel can spoil convergence wherever it injects some bias. This is nothing but another form of the well-known bias-variance dilemma.

The Parzen method [8] pursues a non-parametric approach to kernel density estimation. It gives rise to an estimator that converges everywhere to the true PDF in the mean square sense. Should the true PDF be uniformly continuous, the Parzen estimator can also be made uniformly consistent. In short, the method allocates exactly n identical kernels, each one "centered" on a different element of the sample. In contrast with parametric mixtures, here no experimental evidence is spent to identify parameters. This is the reason why the presence of so many kernels does not inflate the asymptotic variance of the estimator. As a consequence, the detailed shape of the kernels is irrelevant, and the faithfulness problem is successfully circumvented. Of course some restrictions are in order: here is a brief explanation.

Let $\boldsymbol{z}$ be a real-valued random variable. Let $p^{\boldsymbol{z}}(\cdot):\Re \to \Re_+ \cup \{0\}$ be the associated PDF. Let $\boldsymbol{D}_n=\{\boldsymbol{z}_1,..,\boldsymbol{z}_n\}$ be a collection of n independent replicas of $\boldsymbol{z}$. The empirical estimator $\hat{\boldsymbol{p}}_n^E(\cdot)$ of $p^{\boldsymbol{z}}(\cdot)$ based on $\boldsymbol{D}_n$ is defined as follows:

$$\forall z \in \Re \quad \hat{\boldsymbol{p}}_n^E(z) = \frac{1}{n}\sum_{i=1}^{n} \delta(z - \boldsymbol{z}_i) \cdot \tag{1}$$

The estimator just defined is unbiased everywhere but it converges nowhere to $p^{\boldsymbol{z}}(\cdot)$ in the mean square sense because $Var\left[\hat{\boldsymbol{p}}_n^E(z)\right] = \infty$ irrespective of both n and z. This last result is not surprising, since the Dirac's delta is not squared integrable.

The Parzen estimator $\hat{\boldsymbol{p}}_n^S(\cdot)$ of $p^{\boldsymbol{z}}(\cdot)$ based on $\boldsymbol{D}_n$ is obtained by convolving the empirical estimator with some squared integrable kernel PDF $g_S(\cdot)$:

$$\forall z \in \Re \quad \hat{\boldsymbol{p}}_n^S(z) = \int_{-\infty}^{\infty} \hat{\boldsymbol{p}}_n^E(x) \frac{1}{h_n} g_S\left(\frac{z-x}{h_n}\right) dx = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h_n} g_S\left(\frac{z-\boldsymbol{z}_i}{h_n}\right) \cdot \tag{2}$$

The kernel acts as a low-pass filter whose "bandwidth" is regulated by the scale factor $h_n \in \Re_+$. It exerts a "smoothing" action that lowers the sensitivity of $\hat{\boldsymbol{p}}_n^S(z)$ w.r.t. $\boldsymbol{D}_n$ so as to make $Var\left[\hat{\boldsymbol{p}}_n^S(z)\right] < \infty \ \ \forall z \in \Re$. Thus for any given sample size the larger is

the scale factor, the smaller is the variance of the estimator. But the converse is also true: since $\hat{\boldsymbol{p}}_n^S(z)$ is nothing but a mean, then for any given scale factor the larger is the sample size, the smaller is the variance of the estimator (indeed it is inversely proportional to the sample size). Both statements are in fact special cases of the following property:

$$\forall z \in \Re \quad \lim_{n\to\infty} nh_n = \infty \Rightarrow \lim_{n\to\infty} Var\left[\hat{\boldsymbol{p}}_n^S(z)\right] = 0 \ . \tag{3}$$

On the other hand, the same smoothing action produces an unwanted "blurring" effect that limits the resolution of the approximation. Intuitively the scale factor should therefore vanish as $n\to\infty$ in order to let the estimator closely follow finer and finer details of the true PDF. Also this last remark finds a precise mathematical rendering in the following property:

$$\forall z \in \Re \quad \lim_{n\to\infty} h_n = 0 \Rightarrow \lim_{n\to\infty} E\left[\hat{\boldsymbol{p}}_n^S(z)\right] = p^z(z) \ . \tag{4}$$

To summarize, the conflicting constraints dictated by the bias-variance dilemma can still be jointly satisfied by letting the scale factor decrease slowly enough as the sample size grows. The resulting estimator converges everywhere to the true PDF in the mean square sense irrespective of the kernel employed, provided that it is squared integrable.

The above results were later extended to the multi-variate case by Cacoullos [9].

3 Parzen EDA

The main idea of the work is the use of the Parzen method to build a probabilistic model and to sample from the estimated PDF in order to obtain new promising solutions. A detailed description of the Multi-Objective Parzen EDa (MOPED algorithm) follows, and some results are presented in order to show capabilities and potentialities of the algorithm.

Moreover, an extensive use of the Parzen method could lead to simplify the overall optimization procedure towards a parameter-less tool. As a first step in this direction, at the end of section we introduce a different spreading technique for solutions in the Pareto front.

3.1 General Algorithm

As summarized in figure 1, the general optimization procedure can be described as follows:

1. Starting: N_{ind} individuals are sampled from a uniform *m*-dimensional PDF.
2. Classification & Fitness evaluation: by using NSGA-II techniques [7], individuals of current population are ranked and ordered in terms of dominance criterion and crowding distance in the objective function. A fitness value, linearly varying from 2-α (best individual) to α (worst individual), with $0 < \alpha < 1$, is assigned to each individual.

3. Building model & sampling: on the basis of information given by N_{ind} individuals, by means of the Parzen method a probabilistic model of promising search space portion is built. For generic processes can be useful adopting different kernels alternatively from a generation to the other in order to obtain an effective exploration. In this work Gauss and Cauchy distributions are used. Actually, these types of kernel, for their intrinsic characteristics, are complementary and results will show that the use of only one of them could be inefficient for some problems.

   From the probabilistic model so determined, τN_{ind} new individuals are sampled. Fitness values are used to calculate variance of kernels (the fitness values are related to the scale factors introduced in section 2) and to favor sampling from most important kernels.
4. Evaluation: New τN_{ind} individuals are evaluated in terms of objective functions.
5. Classification & Fitness evaluation: following NSGA-II criteria, individuals of intermediate population, of which dimension is $(1+\tau) N_{ind}$, are ordered. A fitness value, linearly varying from $2-\alpha$ (best individual) to α (worst individual), with $0 < \alpha < 1$, is assigned to each individual.
6. New population: best N_{ind} individuals are selected to be next generation.
7. EXIT or NEXT ITER: if convergence criteria are achieved the algorithm stops, otherwise it restarts from point 3.

The algorithm presented above demonstrated satisfactory performance in solving several test cases, when performance is measured in terms of objective function evaluations to obtain a good approximation of the Pareto front. Some results will be shown in the next paragraph.

The still open question is finding an efficient convergence criterion that could be adopted for a generic optimization. That is finding a convergence criterion that guaranties an optimal approximation of Pareto front (efficacy) and requires a number of objective function evaluations as low as possible (efficiency).

Following results show that neither the maximum generation number nor all of the individuals in first class are without gaps. The former because of an extremely low efficiency if a too high maximum number of generations is used, the latter because of premature convergence on a local, non-optimal, front.

Consequently, the maximum generation number is always used. An upper limit for iteration is imposed as suggested from literature results, even if this kind of stopping criterion makes the algorithm inefficient.

3.2 Test Cases Results

In order to have some ideas regarding effectiveness and efficiency of the method, the proposed algorithm has been applied to some well-known test problems taken from literature [10].

For all of test cases 10 independent runs have been performed and results in terms of number of function evaluations are given as average values.

As said in the previous description of the algorithm, in absence of an effective and efficient criterion a maximum number generation criterion has been adopted. In order to allow comparison with obtained results in literature, our results are presented in terms of effective number of iteration, or better, in terms of number of functions evaluations required to obtain the approximation of the optimal front as well.

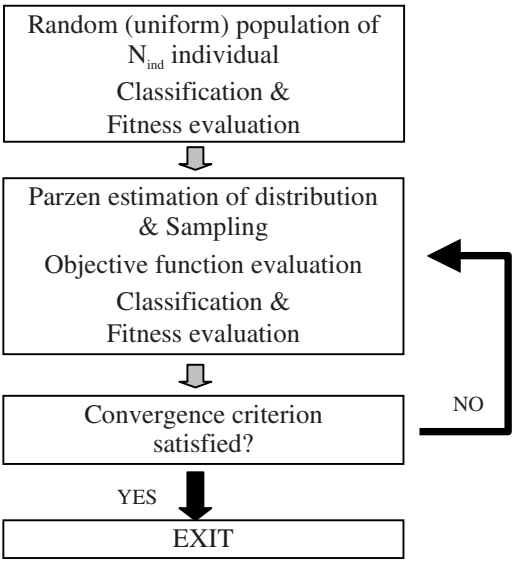


Fig. 1. General structure of the algorithm

All tests have been run with the same values of the following parameters: a) the number of individuals (N_{ind} = 100), the sampling parameter ($\tau = 2$), and the fitness parameter ($\alpha = 0.2$).

In figure 2 one of the fronts obtained for the MOP4 problem is shown in the upper left corner. The other three parts of the figure represent the marginal bivariate PDFs of variables when normal kernels are used. The triple structure of the approximated front can be identified from every marginal PDF, even if for this run it is more evident in the x_1 - x_2 PDF.

For this problem a maximum number of iterations is set equal to 55 (11,100 function evaluations), but still in this case in order to have a stable configuration of the solutions a less number of iterations is needed, which is 44.9 (9,090 function evaluations).

Problems EC4 and EC6 are more complex and a presentation of relative results allows a deeper discussion of advantages and gaps of the proposed algorithm.

EC6 is presented as a problem that tests the ability of algorithms to spread solutions on the whole front. MOPED demonstrates to be able to cover the entire optimal range, even if most of runs produce one or two sub-optimal solutions on the left part of the Pareto front (figure 3.a shows one of the fronts). What happens is similar to the results of the Strength Pareto Evolutionary Algorithm (SPEA) when applied to the same problem as reported in [7].

For both EC4 and EC6 we know that achievement of optimal front corresponds to $g(x)$=1. Therefore, for these problems we adopted the following exit criterion: when the $g(x)$ value averaged on the whole population is ≤ 1.01, this allows to have an error less that 1%.

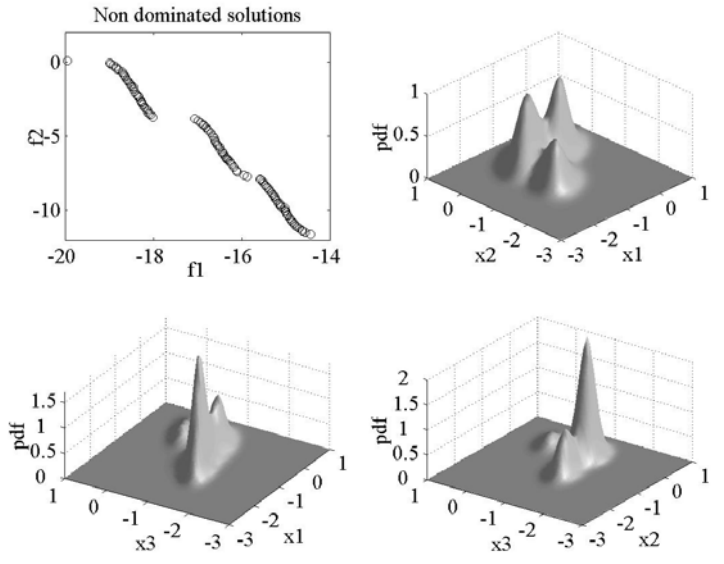
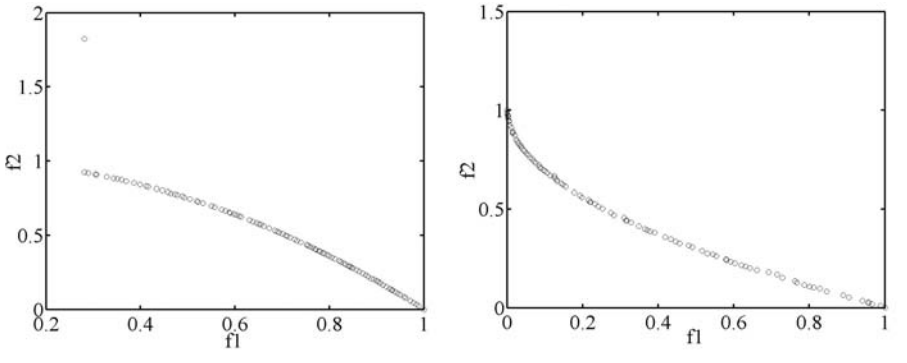


Fig. 2. MOP4 problem. In the left upper corner non-dominated solutions in the objective plain are shown. The other three parts of the figure show the marginal bivariate PDFs of problem's variables when normal kernels are used



a) b)

Fig. 3. a) Obtained non-dominated solutions on EC6 problem. Most of obtained fronts display some sub-optimal solutions. - b) Obtained non-dominated solutions on EC4 problem

For EC6 problem we imposed 15,000 maximum function evaluations, but the convergence criterion related to $g(x)$ function has been reached after approximately 8,300 evaluations.

Problem EC4 is the hardest in terms of number of function evaluations needed to reach the true optimal front. Because of the form of the functions to optimize, process tends to get stuck on sub-optimal fronts.

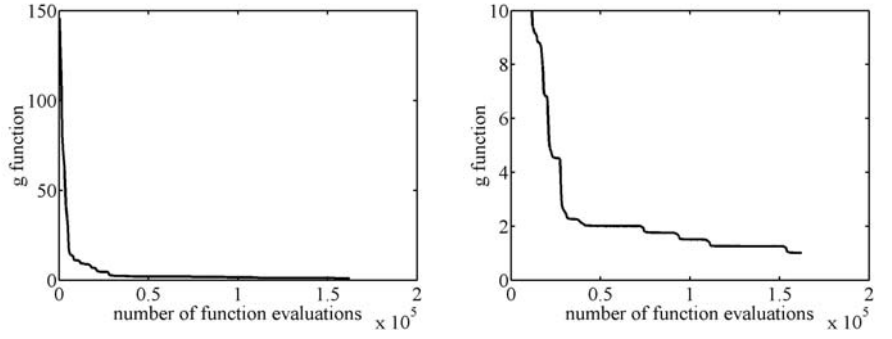


Fig. 4. General trend of *g(x)* function for EC4 problem. On the left the trend during the whole process is shown. On the right side, last part of the process allows a deeper comprehension of difficulties in terms of function evaluation to jump from a local front to a better one

Results demonstrate that the optimal solution (figure 3.b shows one of the fronts) can be obtained after 153,710 function evaluations, with a minimum value of 66,100 in one of the ten runs, and a maximum of 244,100 , when the upper limit of function evaluations is set to 300,000.

From figure 4, which shows a general trend of *g(x)* function, it is possible to see how the process goes on. Ranges with null slope mean a transitorily convergence on a local Pareto-optimal front. In order to allow some comparison we monitored the *g(x)* function and it reaches the value of 3 after 40,862 objective function evaluations.

4 Conclusions

Here we have presented a new estimation of distribution algorithm that is able to manage multi-objective problems following Pareto criterion. The algorithm uses the Parzen method in order to create a non-parametric, model-independent probabilistic representation of promising solutions in the search space. Results obtained when the algorithm is applied to well-known test cases show good performance of the optimization process in terms of the number of objective function evaluations and in the spreading of solutions on the whole front.

Contrary to previous works, in this paper we do not attempt to identify a conditionally independent structure in the genome. We know that this may increase the efficiency of the Parzen estimator. It is our intention to address this important point in the future along a frequentist approach with minor changes in the underlying philosophy. In fact the hypothesis testing inherent in the frequentist approach allows the user to impose a known error of the first kind.

Acknowledgment

This work was partly funded by the MIUR National Program “Sistemi elettronici dedicati basati su algoritmi di apprendimento statistico per l'analisi di dati sperimentali in applicazioni scientifiche ed industriali”.

References

1. Mühlenbein, H., The equation for the response to selection and its use for prediction, Evolutionary Computation 5(3), pp. 303-346, 1998.
2. Pelikan, M., Mühlenbein, H., The bivariate marginal distribution algorithm. In Roy, R., Furuhashi, T., & Chawdhry, P. K. (Eds.), Advances in Soft Computing Engineering Design and Manufacturing, pp. 521-535, London: Springer-Verlag, 1999.
3. Pelikan, M., Goldberg, D. E., and Cant'u-Paz, E. (1999). BOA: The Bayesian optimization algorithm. In Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M., & Smith, R. E. (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99, Vol. I, pp. 525-532. Orlando, FL, Morgan Kaufmann Publishers, San Fransisco, CA, 1999.
4. Bosman, P.A.N., Thierens, D., Expanding from discrete to continuous estimation of distribution algorithms: The IDEA, in M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo, and H.-P. Schwefel, eds., Parallel Problem Solving from Nature, pp 767-776, Springer, 2000.
5. Thierens, D., Bosman, P.A.N., Multi-Objective Mixture-based Iterated Density Estimation Evolutionary Algorithms L. Spector, E.D. Goodman, A. Wu, W.B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M.H. Garzon and E. Burke, editors, Proceedings of the Genetic and Evolutionary Computation Conference - GECCO-2001, pages 663-670, Morgan Kaufmann Publishers, 2001.
6. Khan, N., Goldberg, D.E., and Pelikan, M., Multi-Objective Bayesian Optimization Algorithm, IlliGAL Report No. 2002009, March 2002.
7. Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., A fast and elitist Multiobjective Genetic Algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, April 2002.
8. Parzen, E., On Estimation of a Probability Density Function and Mode, Ann. Math. Stat., Vol. 33, pp. 1065-1076, 1962.
9. Cacoullos, T., Estimation of a Multivariate Density, Ann. Inst. Stat. Math., Vol. 18, pp. 179-189, 1966.
10. Deb, K., Multi-Objective Genetic Algorithm: Problem Difficulties and Construction of Test Problems, Evolutionary Computation, Vol. 7, No. 3, pp. 205-230, The MIT Press, 1999.