On Selecting the Best Individual in Noisy Environments

Wojciech Jaśkowski
wjaskowski@cs.put.poznan.pl

Wojciech Kotłowski
wkotlowski@cs.put.poznan.pl

Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60965 Poznań, Poland

ABSTRACT

In evolutionary algorithms, the typical post-processing phase involves selection of the best-of-run individual, which becomes the final outcome of the evolutionary run. Trivial for deterministic problems, this task can get computationally demanding in noisy environments. A typical naive procedure used in practice is to repeat the evaluation of each individual for the fixed number of times and select the one with the highest average. In this paper, we consider several algorithms that can adaptively choose individuals to evaluate basing on the results evaluations which have already been performed. The procedures are designed without any specific assumption about noise distribution. In the experimental part, we compare our algorithms with the naive and optimal procedures, and find out that the performance of typically used naive algorithm is poor even for relatively moderate noise. We also show that one of our algorithms is nearly optimal for most of the examined situations.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: [Probabilistic algorithms]; G.4 [**Math. Software**]: [Algorithm design and analysis]

General Terms

Algorithms, Theory, Experimentation

Keywords

Evolutionary Algorithms, Evolutionary Computation, Noise, Robustness, Uncertainty, Approximation models

1. INTRODUCTION

In evolutionary algorithms, the typical post-processing phase involves selection of the best-of-run individual, which becomes the final outcome of the evolutionary run. In case of deterministic problems, it is straightforward: one simply chooses the individual with the highest fitness (either from the last generation of the entire run). In noisy environments, however, selecting the best individual is not such a trivial task. Evaluating each individual once and selecting the one with the highest fitness value is not enough, because of the disturbing noise. A selection procedure typical used in practice is resampling: each individual from a population is evaluated k times, the individual's fitness is estimated as the average of evaluations and the one with the highest average is the result of the procedure. This procedure would be optimal if the task was to estimate fitness of a set of individuals, but our task is to select the best one and estimating the fitness of others is not of our concern. Intuitively, a better procedure should pay more attention to individuals that give more hope being better than others. In this paper, we consider several procedures and experimentally show that they are significantly better than the naive resampling.

The problem of selecting the best individual from a population is of practical importance. When the noise is low and there are only a few individuals in the population, there is no need to go beyond simple resampling. However, when the population is large, the noise is heavy or the cost of the evaluation is high, the time required to select the best or close-to-best individual could be significant. Populations of 1,000,000 individuals or more are not unusual in, e.g., genetic programming and in some cases we would like to choose the best-of-run individual not only from the last generations of evolution, but from a bigger set of candidates. Finally, a researcher would sometimes like to observe the evolution dynamics in a form of the best-of-generation individual for every generation. This process could be more computationally demanding than the evolution itself, thus a better method than naive resampling is needed.

Noisy environments appear often in many kinds of simulations, e.g., in evolution of artificial life creatures, which is also an example of problem where the evaluation of an individual is computationally demanding, thus the cost of the measurement is high. The problem of noisy environment and costly evaluation function take place also in probabilistic games, for instance, in poker. The random distribution of cards requires the evaluation of a certain poker strategy to be repeated many times. Since the strategy itself can be computationally demanding, also the cost of the measurement is high.

2. PROBLEM DESCRIPTION

The problem is to select the individual with the highest fitness from a set of individuals. Usually, the set of our interest is the last generation of evolutionary algorithm in which the best-of-run individual is searched for. However, in noisy environments, we do not know if the best-of-run individual is in the last generation. In such a case an experimenter would like to select the best-of-run from a set consisting of, e.g., several last generations.

Generally, this set of individuals will be in the following referred as *population* and denoted by $A = \{a_i\}$. We denote the size of A as n. We assume that the selection procedure can use at most k evaluations; in other words, k expresses the computational effort the experimenter wants (is ready to) to devote to the task of best-of-run selection. The algorithm can distribute these k evaluations among individuals on-line. It means that the individuals are evaluated sequentially and, after getting the evaluation result, the algorithm can make decision about which individual to evaluate next.

Each individual a_i has a (hidden) fitness denoted by f_i. We do not assume any distribution of fitness in a population. Estimation of individual's fitness is subject to noise.

The noise can come from many different sources such as sensory measurements errors or randomized simulations and can be modeled in many ways. In this work, we assume additive noise. Evaluation of individual a_i returns $f_i + \varepsilon$ (observed fitness), where ε is a random error. We do not explicitly assume any distribution of ε apart from the obvious condition that the noise is unbiased, $E(\varepsilon) = 0$.

As the selection procedures are probabilistic by nature, in order to compare different selection procedures a performance measure is required. One possible measure is the success ratio, i.e., the probability that the individual selected by a procedure is the best-of-population individual. However, this measure does not take into account that two or more individuals have a very similar fitness and in practice it does not matter if we select the best or the nearly best one. Thus, in this paper, we will evaluate selection procedures using the expected fitness of the selected individual.

Several methods are proposed in this paper and to compare their performance we must assume some model for the experimental setup. Thus, for the purpose of experiment, we will assume that the fitness of individuals in A is normally distributed, $f_i \sim N(\nu, \tau)$. In practice, this is true or approximately true for many domains, since mutation and cross-over operators have a normalizing effect on the population fitness distribution [**?**]. We assume the noise distribution to be $N(f_i, \sigma)$. The noise can come from many different sources such as sensory measurements errors or randomized simulations. A similar noise model (normal distribution for the measurements and the population) was used by Miller and Goldberg in [**?**].

3. RELATED WORK

Evolutionary optimization in noisy environments is a popular topic, since many practical problems are noisy by nature. A comprehensive survey about uncertain environments can be found in [**?**]. One of the most common problem considered in literature is to determine the optimal sample size during the evolution, e.g., [**?**, **?**]. These and similar papers concentrate on the presence of noise during evolution, neglecting the problem of selecting the best-of-run individual from a population (see for example [**?**]). Nevertheless, in practice, such a procedure is desired.

The only work we are aware of that considers a similar problem in terms of evolutionary algorithms was done by Branke and Schmidt [**?**]. However, the authors designed only a simple statistical method in order to select the better of two individuals in the tournament selection.

The problem presented in this paper belongs to the domain called *experimental design*; in particular, it has the form of a *selection problem*, in which the objective is to construct a procedure for selecting the best of a finite set of alternatives, when stochastic simulation is used to infer the value of each alternative [**?**, **?**]. There are many selection procedures and we refer to [**?**] for a good review of related works in this subject. Most of them is designed to sample (evaluate) until some fixed accuracy of the result is reached. Two different utility/loss functions are considered: *the probability of correct selection* (PCS) and *the expected opportunity cost* (EOC). The latter aims at choosing the highest expected fitness value, which corresponds exactly to the formulation of the problem presented here.

The selection procedures are often based on some assumptions about the data. Most of them assume that output of evaluation has normal distribution with mean and variance specific for each alternative. The so called *Indifference Zone* procedure [**?**] assumes that the best system is better than other systems by at least some known constant value on which the efficiency of the algorithm depends. *Value of Information Procedures* (VIP) [**?**] and *Optimal Computing Budget Allocation* (OCBA) [**?**] work in Bayesian framework and therefore assume some form of prior distributions, which leads to employing specific function inside the procedures, such as density of t distribution. Moreover, each procedure has a large number of parameters, which influences its accuracy [**?**].

All the algorithms proposed here (except the optimal procedure derived for comparison only) were designed without any specific distribution, data or framework assumptions in mind. This led us to simpler parameter-free procedures that are ready to implement and use also for practitioners with no expert statistical knowledge. Despite their simplicity, as the experimental results show, one of the proposed heuristics is as good as the one-stage optimal procedure for most of the cases. Thus, our aproach is rather in the spirit of so called *racing algorithms* [**?**, **?**], which were propose to solve model selection problems in machine learning and were also used in metaheuristic optimization.

Finally, we notice that the model used for experimental setup is an example of *Bayesian experimental design* [**?**] (in particular *Bayesian selection problem*), the framework for optimal design of experiments based on Bayesian decision theory [**?**]. It dates back to Lindley [**?**], who presented a two-part decision-theoretic approach, providing a unifying theory for most work on Bayesian experimental design today. The one-stage optimal procedure presented in the Appendix of this paper also falls under this category.

4. THE ALGORITHMS

We designed four algorithms for selecting the best-of-population individual: *naive*, *tournament*, *candidate selection* and *optimal*. Input arguments for all of them are: the number of available evaluations k, the size of the population n and the population A itself. The *optimal* method knows also the hidden parameters of the distribution used in the experiment i.e., σ, υ, τ that would not be available in practice. The *optimal* algorithm can be seen as a benchmark for other (practical) algorithms in our experiment.

Algorithm 1 Naive procedure

```
1: procedure NAIVE($A, n, k$)
2:     $t \leftarrow k/n$
3:     for $a \in A$ do
4:         $t$ times evaluate $a$;
5:         update avg($a$)
6:     end for
7:     return element $a \in A$ with maximal $avg(a)$
8: end procedure
```

Algorithm 2 Tournament procedure

```
1: procedure TOURNAMENT($A, n, k$)
2:     $t \leftarrow k/n$
3:     while n > 1 do
4:         for $a \in A$ do
5:             $t/2$ times evaluate $a$;
6:             update avg($a$)
7:         end for
8:         $A \leftarrow n/2$ elements from $A$ with maximal avg()
9:         $n \leftarrow n/2$
10:    end while
11:    return the only element from $A$
12: end procedure
```

Algorithm 3 Candidate selection

```
1: procedure CANDIDATE SELECTION($A, n, k$)
2:     for $a \in A$ do
3:         evaluate $a$;
4:         update $avg(a)$ and $cnt(a)$
5:         $k \leftarrow k - 1$
6:     end for
7:     while $k > 0$ do
8:         $a_{\max} \leftarrow$ element from $A$ with maximal $avg()$
9:         $a_{\text{2nd}} \leftarrow$ element from $A \backslash \{a_{\max}\}$ maximizing $avg()$ (second from the top)
10:        $a_{\min} \leftarrow$ element from $A \backslash \{a_{\max}\}$ maximizing the confidence $conf(a, a_{\max})$
11:        if $conf(a_{\min}, a_{\max}) < conf(a_{\max}, a_{\text{2nd}})$ then
12:            evaluate $a_{\min}$;
13:            update $avg(a_{\min})$ and $cnt(a_{\min})$
14:        else
15:            evaluate $a_{\max}$;
16:            update $avg(a_{\max})$ and $cnt(a_{\max})$
17:        end if
18:        $k \leftarrow k - 1$
19:    end while
20:    return element $a \in A$ with maximal $avg(a)$.
21: end procedure
```

4.1 Naive

Naive algorithm distributes the k available evaluations fairly among the n elements[1]. Each element is evaluated $t = k/n$ times. The best-on-average element is returned as a result (see Algorithm 1). This procedure is also called *sampling* [?] and it is usually used to estimate the fitness of all individuals during the evolution.

4.2 Tournament

The *tournament* procedure is based on the observation that it pays off to evaluate more accurately individuals that give more hope to be better than others. The algorithm resembles the so-called single elimination tournament. First, each of the n individuals is evaluated t times; then half of the individuals with the the lowest estimated average fitness are discarded and $n/2$ individuals remain. Next, the remaining $n/2$ are again evaluated t times, their observed fitness is updated (their fitness is based on $2t$ evaluations by now), and the $n/4$ worst individuals are thrown out. This process is repeated until one individual remains, which is the result of the algorithm (see Algorithm 2). Notice that, the worst individuals are evaluated only t times, whereas the tournament finalists (best two) are evaluated $t \times \log_2 n$ times. Setting t to $k/(2n)$ ensures that approximately all k available evaluations are used:

$$nt + \frac{n}{2}t + \ldots + \frac{n}{n/2}t = n\frac{k}{2n}\left(1 + \frac{1}{2} + \ldots + \frac{1}{n/2}\right) = k - \frac{k}{n}.$$

In practice, the remaining $\frac{k}{n}$ evaluations can be distributed among the tournament finalists to better estimate their fitness, however its impact was found to be negligible.

The *tournament* procedure is intuitive, elegant and easy to implement. In the experimental part of this paper, we show that it is significantly better than the commonly used naive procedure. Nevertheless, using more statistical knowledge a better algorithm can be designed.

[1]If n does not divide k, the evaluations are distributed as fair as possible.

4.3 Candidate selection

The *candidate selection* procedure (termed simply *candidate* in the following) is based on iteratively choosing one individual (a "candidate" for the best-of-population individual) for a single evaluation and updating the fitness estimate of this individual. Two values are kept for each individual during the procedure: the average value of all evaluations $avg(a)$, and the number of evaluations $cnt(a)$.

The procedure is initialized by evaluating every individual $a \in A$ once. Next, in each iteration, for each $a \in A$ we calculate the *confidence* defined as:

$$conf(a, \bar{a}) = (avg(a) - avg(\bar{a}))^2 \cdot cnt(a) \qquad (1)$$

where $\bar{a}$ is defined as $\bar{a} = \arg\min_{a' \in A \backslash a} avg(a')$. In other words, if a is the individual with the highest $avg(a)$, then $\bar{a}$ is the second individual from the top. Else, $\bar{a}$ is the individual with the highest $avg(a)$. Next, the individual with the smallest confidence is chosen for evaluation. This process is repeated until all the evaluations are done. The individual the highest $avg(a)$ is the result of the procedure.

The idea behind the confidence term is following. Suppose $\bar{x}$ is the average value of m independent random variables x_i with mean μ and variance σ^2. Then for large m, the term $\frac{\bar{x}-\mu}{\sigma}\sqrt{m}$ (sometimes called *confidence* term or $Z - score$) becomes distributed approximately as $N(0,1)$. It is used for tests (such as t-test) whether to reject the hypothesis that x_i have mean value smaller than μ (if confidence exceeds some threshold value). In our case, since σ is common across all the individuals, we use the square of the confidence (to avoid calculating square roots) and we put $avg(\bar{a})$ in place of the mean value. Thus, we test the hypothesis whether a given individual has its mean value greater than the maximal average value among the individuals (or smaller than the

Algorithm 4 One-Step Optimal Bayesian Selection

```
1: procedure OPTIMAL BAYESIAN($A, n, k, \sigma, \tau, \nu$)
2:     for $a \in A$ do
3:         $avg_B(a) \leftarrow \nu$
4:         $cnt(a) \leftarrow 0$
5:     end for
6:     while $k > 0$ do
7:         $a_{\max} \leftarrow$ element from $A$ with maximal $avg_B()$
8:         $a_{2\text{nd}} \leftarrow$ element from $A \backslash \{a_{\max}\}$ maximizing $avg_B()$ (second from the top)
9:         $a_{\min} \leftarrow$ element from $A \backslash \{a_{\max}\}$ minimizing the utility $U(a, a_{\max})$
10:        if $U(a_{\min}, a_{\max}) > U(a_{\max}, a_{2\text{nd}})$ then
11:            evaluate $a_{\min}$;
12:            update $avg_B(a_{\min})$ and $cnt(a_{\min})$
13:        else
14:            evaluate $a_{\max}$;
15:            update $avg_B(a_{\max})$ and $cnt(a_{\max})$
16:        end if
17:        $k \leftarrow k - 1$
18:    end while
19:    return element $a \in A$ with maximal $avg_B(a)$.
20: end procedure
```

second from the top individual, if the maximal individual is tested). The individual with the smallest confidence of rejection (for which we keep the null hypothesis with highest confidence) is evaluated.

4.4 Bayesian One-Stage Ahead Procedure

It is worth considering the *optimal* procedure for the experimental setup, mentioned in Section 2 and described in more details in Section 5. If we assume that all hidden parameters (ν, τ, σ) are known, there exists an optimal procedure for the best-of-population selection problem, since this is a Bayesian framework. However, in real problems values of ν, σ and τ are rarely known, thus such a Bayesian procedure is impractical. Here, we present it only for comparison with heuristic algorithms that do not know the hidden parameters.

The procedure is presented as Algorithm 4. Comparing to the previous heuristics, a *Bayesian average* is used instead of normal average value, defined as:

$$avg_B(a) = \frac{\nu \cdot \sigma^2/\tau^2 + cnt(a) \cdot avg(a)}{\sigma^2/\tau^2 + cnt(a)}$$

Notice that ν and "signal to noise" ratio τ/σ is required to calculate $avg_B(a)$. Moreover, individuals are being selected for evaluation by calculating their *utility*:

$$U(a, \bar{a}) = \frac{\sigma}{\delta(a)} \left(\frac{1}{\sqrt{2\pi}} e^{-\frac{(\Delta(a,\bar{a}))^2}{2}} + \Delta(a, \bar{a}) \Phi(\Delta(a, \bar{a})) \right)$$

where we used $\delta(a) = \sqrt{(\sigma^2/\tau^2 + cnt(a) + 1)(\sigma^2/\tau^2 + cnt(a))}$ and $\Delta(a, \bar{a}) = |avg(a) - avg(\bar{a})| \cdot \delta(a)/\sigma$. The derivation of this expression is presented in the Appendix. For each $a \in A$, the individual $\bar{a}$ is obtained in the same way as in the *candidate* procedure.

This procedure is a one-stage optimal Bayesian procedure, which means that it is optimal procedure among all the procedures looking one step ahead. We do not know if this procedure is also a g*lobally* optimal procedure, but if not, our one-step optimal procedure should be a very good approximation of the globally optimal one.

4.5 Time complexity

The *naive* and *tournament* algorithms have time complexity $O(k)$; *candidate* and *optimal* algorithms have time complexity $O(kn)$. However, the impact of higher time complexity in case of *candidate* and *optimal* algorithms is not critical, if we assume that the cost of evaluation is much higher than the cost of arithmetic operations. In this case, all of the algorithms have time complexity $O(k)$, since all perform k evaluations.

5. EXPERIMENTS AND RESULTS

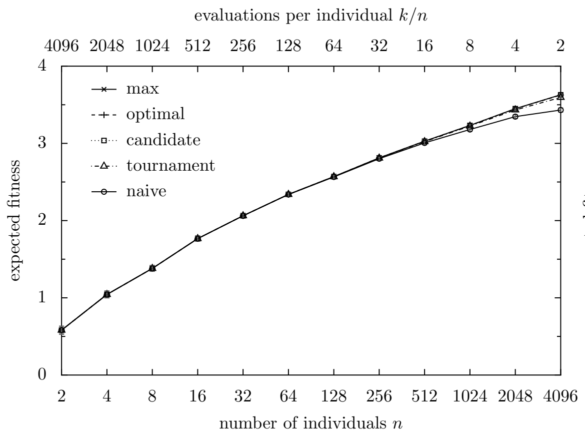
To compare the algorithms presented in the previous section, we carried out a computational experiment. In a simulation run, we first generated a set of n individuals with fitness drawn at random with normal distribution $N(\nu, \tau)$. Each algorithm under examination was then executed with the same input parameters A, n, k. During each run, the algorithms could k times evaluate a chosen individual a_i. Each such evaluation produced a value drawn at random with normal distribution, i.e., $\varepsilon \sim N(f_i, \sigma)$. The estimated expected fitness of individual returned by each algorithm was determined as the average of 1000 simulation runs.

The problem we have posed has five variables: n, k, ν, τ, σ. It would be infeasible to test all points of the five-dimensional space. Fortunately, only two combinations of those parameters are important. Firstly, notice that the standard deviation τ is barely the scale of the fitness values axis and is meaningful only in comparison with the noise σ, so that only the 'signal to noise' ratio τ/σ matters; thus we set $\tau = 1$ in the experiment without loss of generality. Secondly, ν (a priori mean) is only a fitness scale translation and does not influence any of the methods, so it is set to 0. Finally, we noticed experimentally that as we vary σ with varying k proportionally to σ^2, the results of all the procedures do not change (see Figure 1). This is due to the fact that accuracy of estimated mean values grows with the square root of evaluations and decreases proportional to σ. Therefore setting σ^2/k constant (with n being fixed) does not change the accuracy of estimates and leads to similar accuracy of the methods. Thus, we can fix k and change σ and n, so we end up with only two parameters.

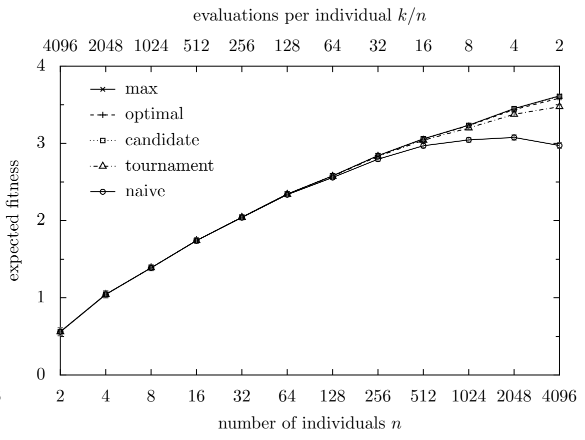
Therefore we set $\nu = 0$, $\tau = 1$, $k = 8192$. We examined different noise values $\sigma = (0.5, 1, 2, 4, 8, 16, 32, 64)$. For each noise value, we checked how the algorithms cope with different populations sizes $n = (2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096)$. As the number of evaluations is fixed ($k = 8192$), in the following analysis, we will sometimes use the term *the number of evaluations per individual k/n*.

The results are shown in Figure 2. Each graph (a)–(h) corresponds to one noise value. Notice that ticks at the top axis correspond to the number of evaluations per individual, whereas ticks at the bottom axis correspond to the population size. The confidence intervals (95%) were marked as gray bars.

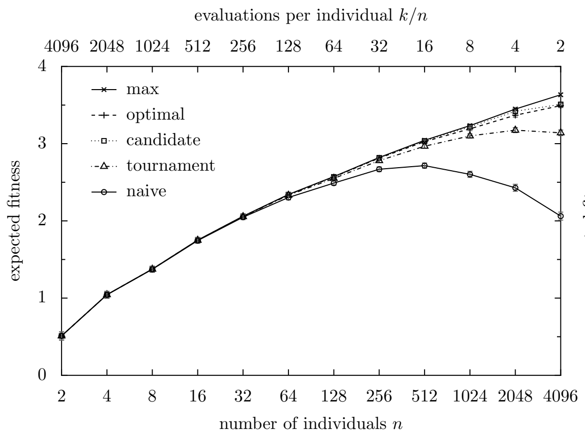
The curve labeled as *max* on the graph is not a real algorithm, but a maximal possible expected fitness that could be obtained if returning always the individual with maximal fitness. That is why *max* performance does not deteriorate when the noise increases. The maximum possible expected fitness increases with increasing number of individuals n, be-



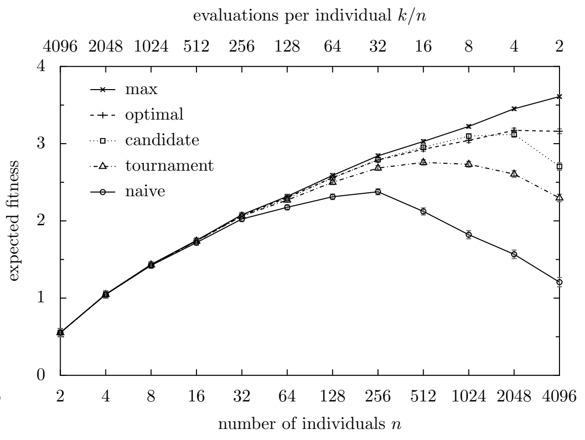
(a) $\sigma = 0.5$



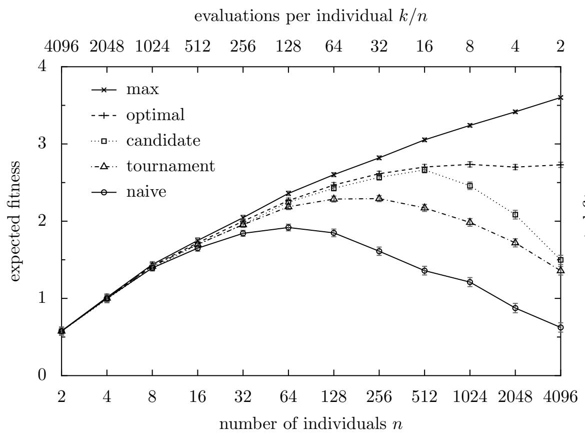
(b) $\sigma = 1.0$



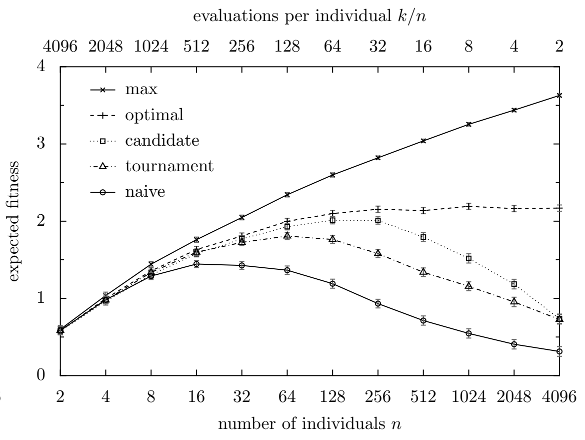
(c) $\sigma = 2.0$



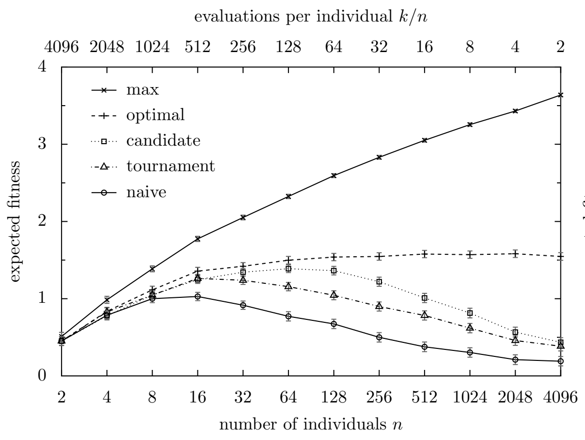
(d) $\sigma = 4.0$



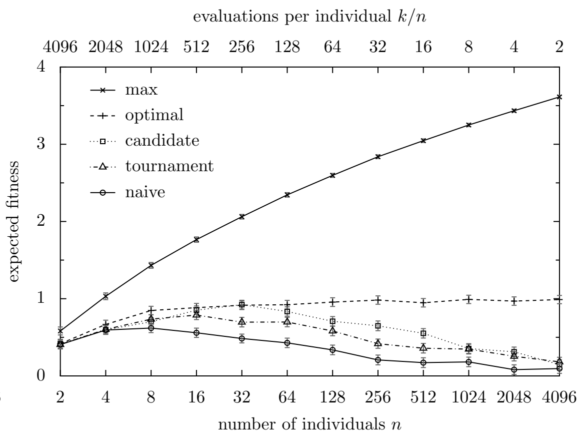
(e) $\sigma = 8.0$



(f) $\sigma = 16.0$



(g) $\sigma = 32.0$



(h) $\sigma = 64.0$

Figure 2: Each graph presents a comparison of four algorithms for different noises σ and population sizes n. The rest of the parameters were fixed constant: $\nu = 0.0$, $\tau = 1.0$, $k = 8192$.

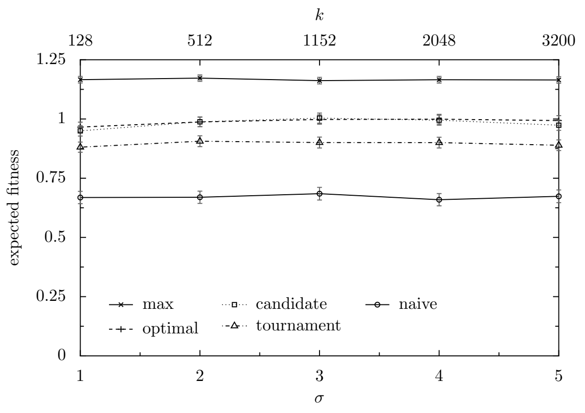


Figure 1: When k is proportional to σ^2 the expected fitness remains constant for all procedures. This graph was created using the following parameters: $n = 64$, $k/\sigma^2 = 128$, $\tau = 0.5$, $\nu = 0$. The simulation was repeated 1000 times.

cause it has the form of the maximum order statistics which expected value grows with the population size.

Let us first comment on how the noise affects the examined algorithms. Low noise (e.g. $\sigma = 0.5$) does not influence much on any algorithm, but when the noise increases, it is much harder to find the best-of-population individual; finally, in the presence of high noise ($\sigma = 64$), no algorithm can confidently select the best among two individuals even when the number of evaluations per individual is as high as 4096 (see Figure 2h). Moreover, when the noise is moderate ($\sigma = 8.0$) and the number of evaluations per individual medium ($k/n = 32$) even the *optimal* algorithm produces expected fitness significantly lower than the maximum possible. At the same time, the expected fitness of *naive* algorithm is nearly two times lower than *max*.

When the number of evaluations per individual is high and noise is low, it does not matter which algorithm we use, because all of them perform, approximately, as good as possible. When the noise is higher or the number of evaluations per individual lower, we can clearly see that *naive* algorithm is much worse than other algorithms. Already with a relatively small noise $\sigma = 1.0$, we can notice that having less than 32 evaluations per individual makes the naive algorithm perform worse than others. The most striking difference between *naive,* and *tournament* or *candidate* algorithm is when $k/n = 2$ and $\sigma = 4$.

We can also see, that *candidate* algorithm is, in most of the cases, equal or better than *tournament.* The only exception of this rule is when $\sigma = 32$ and $n \in [4, 16]$, but the difference is small and the confidence intervals overlap heavily. Among the considered methods, *candidate* is the most robust procedure.

Candidate procedure is for most examined cases nearly as good as the *optimal* one. Surprisingly, however when the number of individuals increases, the expected fitness of *candidate* procedure deteriorates. This effect can be observed, for example, for $\sigma = 8$. Expected fitness peak is achieved for $n = 512$, whereas for $n = 1024$ the method behaves significantly worse. This effect is, counter-intuitive to some extent, since we would expect that when the maximum possible fitness *max* increases the expected fitness of *candidate* algorithm should also increase, because there are better individuals in bigger populations. This effect is also noticeable for other algorithms (*naive* and *tournament*); only *optimal* holds the level. This problem appears, because when the number evaluations per individual drops, the algorithms loose to much time (evaluations) to assess if an individual gives a hope for being the best-of-population. On the other hand, the *optimal* algorithm has complete information about the distribution, i.e, ν and τ. In result, it quickly decide what is the probability that evaluated individual is better than others. That is why *optimal* algorithm does not evaluate most of the individuals in some cases—it selects a sample from A and concentrates its efforts to determine which of them is the best.

As we have noticed, for $\sigma = 8$ and $n = 1024$ the result of *candidate* algorithm is worse than for $k = 512$. Thus, for $k = 1024$, it would be wise to discard half of the individuals and run the *candidate* algorithm on the remaining 512 individuals receiving the same good result as for $k = 512$. Generally, if we could know the optimal $n_{\sigma,k}$ for all σ^2 and k value, we could design a better than *candidate* algorithm, by calling the procedure with argument $n_{\sigma,k}$ instead of n always when $n > n_{\sigma,k}$. Unfortunately, σ (nor τ) is not known to the algorithm in most practical cases. We could probably try to estimate σ/τ on-line, but this idea needs additional research.

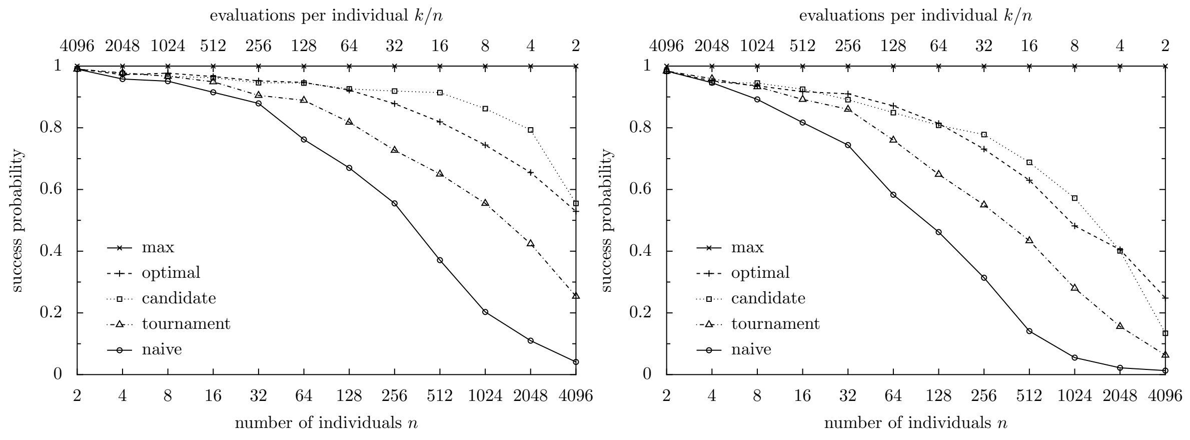
To further compare the profit of application *candidate* algorithm instead of *naive* one, we checked how many times more evaluations are required by *naive* algorithm to get at least the same expected fitness as *candidate* algorithm. The results for different number of individuals n and noise values σ are shown in Table 1. As we can see, for some parameters the *naive* algorithm requires two orders of magnitude times more evaluations than the *candidate.*

It is interesting whether our algorithms work well with different utility function, i.e., the probability of correctly choosing the optimal element (success probability). To investigate this, we conducted additional experiment with two different noise values in which we test the algorithms measuring the success probability. The results are shown on Figure 3 and look surprising on the first sight. It follows that in many cases the *candidate* procedure outperforms *optimal.* However, the *optimal* procedure is not optimal anymore for such utility function. Those results show that our best *candidate* procedure is robust to the changes of utility measure. On the other hand, performance of the *optimal* procedure deteriorated, which suggests that procedures optimized to specific conditions might work poorly when conditions are changed.

6. CONCLUSIONS

We presented an interesting problem of selecting the best individual from a population in a noisy environment and designed several procedures solving it. All presened procedures are straighforward to implement and use. In the experimental part, we compared our algorithms and found out that the commonly used resampling procedure is weak even if the noise is low. For a situation when both the noise and the fitness in population are normally distributed, we designed an optimal algorithm, which need complete information about the distributions. We found out, however that,

[2]To be exact, σ/τ.



(a) $\sigma = 2$ (b) $\sigma = 4$

Figure 3: Success probability.

Table 1: How many times more evaluations *naive* algorithm should use to get the same result as *candidate* algorithm in function of noise σ and number of individuals n.

$\sigma \backslash n$	32	64	128	256	512	1024	2048	4096
0.5	7.2	6.9	9.5	14.4	36.4	99.0	137.0	171.0
1	2.9	3.6	8.6	14.9	23.8	51.0	78.3	122.5
2	1.5	2.9	7.8	10.3	22.6	32.0	48.0	34.0
4	1.5	3.7	6.2	10.2	17.6	22.1	20.0	11.5
8	1.7	3.9	5.7	8.9	12.6	12.0	11.0	7.5
16	2.4	3.9	5.8	7.2	8.6	9.0	8.5	6.5
32	3.6	4.0	6.9	7.8	7.3	7.6	6.5	5.0
64	3.0	4.6	5.4	5.7	8.2	7.0	5.5	6.0

our *candidate* selection procedure, that have no information about the distribution, is nearly optimal in the most of parameters combinations.

We also showed that it does not always pay off to have a big set of candidate solutions. When the noise is high, to maximize the expected fitness, it is better to consider only part of the population. However, to answer the question how many individuals should be discarded and under what conditions requires further research.

Neither *tournament* nor *candidate* methods does not rely on any assumption concerning noise and population distributions. In the future, we would like to test our algorithms on other than normal distributions (t-Student or Cauchy-Lorentz) and also verify their utility on real-world data.

7. ACKNOWLEDGMENTS

This research has been supported by the Ministry of Science and Higher Education grant # N N519 3505 33.

8. REFERENCES

[1] R. E. Bechhofer, T. J. Santner, and D. M. Goldsman. *Design and Analysis for Statistical Selection, Screening, and Multiple Comparisons.* John Wiley & Sons, Inc., New York, 1995.

[2] James O. Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer, 2nd edition, 1993.

[3] Hans-Georg Beyer and Bernhard Sendhoff. Evolutionary algorithms in the presence of noise: To sample or not to sample. In *FOCI*, pages 17–24. IEEE, 2007.

[4] Jürgen Branke, Stephen E. Chick, and Christian Schmidt. Selecting a selection procedure. *Management Science*, 53(12):1916–1932, 2007.

[5] Jürgen Branke and Christian Schmidt. Sequential sampling in noisy environments. In Xin Yao et. al., editor, *Parallel Problem Solving from Nature - PPSN VIII*, 2004.

[6] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

[7] C.-H. Chen. A lower bound for the correct subset-selection probability and its application to discrete event simulations. *IEEE Transactions on Automatic Control*, 41(8):1227–1231, 1996.

[8] Stephen E. Chick and Koichiro Inoue. New two-stage and sequential procedures for selecting the best simulated system. *Operations Research*, 59(5):732–743, 2001.

[9] Shanti S. Gupta and Klaus J. Miescke. Bayesian look ahead one-stage sampling allocations for selection of the best population. *Journal of Statistical Planning and Inference*, 54:229–244, 1996.

[10] Koichiro Inoue, Stephen E. Chick, and Chun-Hung Chen. An empirical evaluation of several methods to select the best system. *ACM Transactions on Modeling and Computer Simulation*, 9(4):381–407, 1999.

[11] Yaochu Jin and J. Branke. Evolutionary optimization in uncertain environments-a survey. *Evolutionary Computation, IEEE Transactions on*, 9(3):303–317, June 2005. Survey of noisy environments.

[12] Dennis V. Lindley. *Bayesian Statistics – a Review.* SIAM, Philadelphia, 1972.

[13] O. Maron and A.W. Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. *Advances in Neural*

Information Processing Systems, 6:59–66, 1994.

[14] B.L. Miller. *Noise, Sampling, and Efficient Genetic Algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, 1997.

[15] B.L. Miller and D.E. Goldberg. Genetic algorithms, tournament selection, and the effects of noise. Technical Report 95006, Dept. of General Engineering, University of Illinois at Urbana-Champaign, July 1995.

[16] B.L. Miller and D.E. Goldberg. Optimal sampling for genetic algorithms. Technical Report 96005, Dept. of General Engineering, University of Illinois at Urbana-Champaign, 1996.

[17] B. Yuan and M. Gallagher. Statistical racing techniques for improved empirical evaluation of evolutionary algorithms. In *PPSN VIII*, volume 3242 of *LNCS*, pages 172–181, 2004.

APPENDIX

A. DERIVATION OF THE OPTIMAL PROCEDURE

In this Section we derive the Algorithm 4. It is based on the Bayesian decision theory [?] and is the one stage ahead optimal procedure, i.e. in each iteration $t = 1, \dots, k$ we choose one individual for evaluation such that the choice is optimal between all single-evaluation procedures. The algorithm is a special case of more general optimal procedure described in [?] but the derivation and the form of the algorithm presented here is simpler.

Let $\mathbf{m} = (m_1, \dots, m_n)$ be a vector of numbers of evaluations for each individual in iteration $t-1$. Assume that for ith individual, values $\{x_{i,1}, \dots, x_{i,m_i}\}$ has already been obtained in m_i evaluations. Let $\bar{\mathbf{x}} = (\bar{x}_1, \dots, \bar{x}_n)$ be a vector of average values for each individual in iteration $t-1$, i.e. $\bar{x}_i = \frac{\sum_{j=1}^{m_i} x_{ij}}{m_i}$ and $\bar{x}_i := 0$ if $m_i = 0$. Let $\tilde{\mathbf{x}} = (\tilde{x}_1, \dots, \tilde{x}_n)$ be a vector of so called *Bayesian* averages in iteration $t-1$ defined as:

$$\tilde{x}_i = \frac{\nu \cdot \sigma^2/\tau^2 + m_k \bar{x}_k}{\sigma^2/\tau^2 + m_k} \tag{2}$$

Let $\mathbf{m}', \bar{\mathbf{x}}', \tilde{\mathbf{x}}'$ be the above described values in iteration t, after updating them with new evaluation x of one of the individuals. Let d_1 be the index of the individual chosen for evaluation in iteration t. Let d_2 be the index of individual chosen to be the best, after iteration t. From Bayesian decision theory it follows that d_1 is of the form $d_1(\bar{\mathbf{x}})$ (depends only on $\bar{\mathbf{x}}$), since $\bar{\mathbf{x}}$ is so called *sufficient* statistics for normal distribution [?]. Similarly, and d_2 is of the form $d_2(\bar{\mathbf{x}})'$. We called the pair (d_1, d_2) the decision function.

We define the *utility* $u(\mathbf{f}, i)$ of choosing the ith individual, when the fitness vector is $\mathbf{f} = \{f_1, \dots, f_n\}$. Since we aim at maximizing the response of the procedure, we define utility simply as $u(\mathbf{f}, i) = f_i$. We define *expected utility* of decision function (d_1, d_2) as the expected value of u according to the random choice of data $\bar{\mathbf{x}}'$ and fitness vector $\mathbf{f}$.

$$U(d_1, d_2) = \int u(\mathbf{f}, d_2) dP(\bar{\mathbf{x}}', \mathbf{f}) \tag{3}$$

(notice that the dependence on d_1 follows from the fact that $\bar{\mathbf{x}}'$ depends on d_1). We aim at finding d_1 and d_2 which maximizes (3). But expression (3) can be transformed to:

$$U(d_1, d_2) = \int \left(\int \left(\int u(\mathbf{f}, d_2) dP(\mathbf{f}|\bar{\mathbf{x}}') \right) dP(x|\bar{\mathbf{x}}) \right) dP(\bar{\mathbf{x}})$$

which shows that to maximize (3), it is enough to find d_1 and d_2 maximizing

$$\int \left(\int u(\mathbf{f}, d_2) dP(\mathbf{f}|\bar{\mathbf{x}}') \right) dP(x|\bar{\mathbf{x}}) \tag{4}$$

for each $\bar{\mathbf{x}}$. From Bayesian theory it follows that $\mathbf{f}|\bar{\mathbf{x}}'$ has posterior distribution of the form $f_i|\bar{x}_i' \sim N(\tilde{x}_i', \rho_i)$ where $\tilde{x}_i'$ was defined above and $\rho_i^2 = \frac{\tau^2\sigma^2}{m_k'\tau^2 + \sigma^2}$. Then, one can easily show that

$$\int u(\mathbf{f}, d_2) dP(\mathbf{f}|\bar{\mathbf{x}}') = \tilde{x}_{d_2}' \tag{5}$$

But this expression is maximized for $d_2 = \arg\max_i \tilde{x}_i'$. Thus, from (4) and (5) it follows that the optimal decision d_1 is the one which maximizes:

$$U(d_1) = \int \max_i \tilde{x}_i' P(x|\bar{\mathbf{x}}) dx \tag{6}$$

To simplify expression (6), first notice that $P(x|\bar{\mathbf{x}}) = P(x|\bar{x}_{d_1})$. We can obtain this probability density by decomposition

$$P(x|\bar{x}_{d_1}) = \int P(x|f_{d_1}, \bar{x}_{d_1}) P(f_{d_1}|\bar{x}_{d_1}) df_{d_1}$$

Since $x|f_{d_1}, \bar{x}_{d_1} = x|f_{d_1} \sim N(f_{d_1}, \sigma)$ and $f_{d_1}|\bar{x}_{d_1} \sim N(\tilde{x}_{d_1}, \rho_{d_1})$, by solving typical Gaussian integral we obtain $x|\bar{x}_{d_1} \sim N(\bar{x}_{d_1}, \sigma_k)$ where $\sigma_k = \sqrt{\rho_{d_1}^2 + \sigma^2}$. To calculate the integral (6), first notice that $d_1 \neq d_2$ as long as $\tilde{x}_{d_1}' < \max_i \tilde{x}_i'$ which is equivalent to:

$$x < \tilde{x}_{d_2}'(\sigma^2/\tau^2 + m_k + 1) - m_{d_1}\bar{x}_{d_1} - \nu\sigma^2/\tau^2 =: \beta_{d_1}$$

But then, the integrand $\max_i \tilde{x}_i'$ is constant and equal to $\tilde{x}_{d_2}'$ or $\tilde{x}_{d_2}$, since $\tilde{x}_{d_2}$ was not changed and d_2 can be expressed as:

$$d_2 = \arg\max_{i \neq d_1} \tilde{x}_i \tag{7}$$

(since $d_2 \neq d_1$ and d_2 corresponds to the largest $\tilde{x}_i$). Moreover, if $x \geq \beta_{d_1}$ than $d_1 = d_2$. Therefore we decompose (6) as:

$$U(d_1) = \tilde{x}_{d_2} \int_{-\infty}^{\beta_{d_1}} P(x|\bar{\mathbf{x}}) dx + \int_{\beta_{d_1}}^{\infty} \tilde{x}_{d_1}' P(x|\bar{\mathbf{x}}) dx$$

where d_2 is defined in (7). The first integral equals to:

$$\tilde{x}_{d_2} \Phi\left((\beta_{d_1} - \tilde{x}_{d_1})/\sigma_k\right)$$

where Φ is the standard normal cumulative distribution function. The second integral equals to:

$$\tilde{x}_{d_1} \Phi\left(\frac{\tilde{x}_{d_1} - \beta_{d_1}}{\sigma_k}\right) + \frac{\frac{\sigma_k}{\sqrt{2\pi}} \exp\left(-\frac{(\beta_{d_1} - \tilde{x}_{d_1})^2}{2\sigma_k^2}\right)}{\sigma^2/\tau^2 + m_{d_1} + 1}$$

If we denote $\delta_{d_1} = \sqrt{(\sigma^2/\tau^2 + m_{d_1} + 1)(\sigma^2/\tau^2 + m_{d_1})}$, then it follows that:

$$\sigma_k = \sqrt{\rho_{d_1}^2 + \sigma^2} = \frac{\sigma}{\sigma^2/\tau^2 + m_{d_1}} \delta_{d_1}$$

and:

$$\frac{\tilde{x}_{d_1} - \beta_{d_1}}{\sigma_k} = \frac{\tilde{x}_{d_1} - \tilde{x}_{d_2}}{\sigma} \delta_{d_1}$$

So that we obtain:

$$U(d_1) = \tilde{x}_{d_2} + (\tilde{x}_{d_1} - \tilde{x}_{d_2})\Phi\left(\frac{\tilde{x}_{d_1} - \tilde{x}_{d_2}}{\sigma}\delta_{d_1}\right) - \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{(\tilde{x}_{d_1} - \tilde{x}_{d_2})^2 \delta_{d_1}^2}{2\sigma^2}} \tag{8}$$

This expression can be already used, however it is numerically unstable. Therefore, we transform it using the equivalences $\Phi(z) = 1 - \Phi(-z)$ and $\tilde{x}_{d_1} - \tilde{x}_{d_2} = \max\{\tilde{x}_{d_1}, \tilde{x}_{d_2}\} - |\tilde{x}_{d_1} - \tilde{x}_{d_2}|$, and noticing that $\max\{\tilde{x}_{d_1}, \tilde{x}_{d_2}\} = \max_i \tilde{x}_i$ is constant and can be dropped. Finally, we obtain:

$$U(d_1) = \frac{\sigma}{\delta_{d_1}}\left(\frac{1}{\sqrt{2\pi}} e^{-\frac{\Delta_{d_1}^2}{2}} + \Delta_{d_1}\Phi(\Delta_{d_1})\right) \tag{9}$$

where $\Delta_{d_1} = \frac{|\tilde{x}_{d_1} - \tilde{x}_{d_2}|}{\sigma}\delta_{d_1}$. Thus, in each iteration we choose to evaluate the individual with index d_1, which maximizes the above expression.