Search methods for a vision system configurator

Olivier Hüsser, Heinz Hügli

Institut de Microtechnique

Université de Neuchâtel

Rue A.-L. Breguet 2

CH-2000 Neuchâtel

Phone: ++41 32 718 34 57, Fax: ++41 32 718 34 02

Olivier.huesser@unine.ch || Heinz.hugli@unine.ch

Summary

Flexible production sets new requirements in quality control. Vision systems must be highly versatile to cope with the ever changing production cycles and setup time must be reduced to a minimum so as to reduce downtime. This paper considers an automatic configurator tool that assists the operator during the setup of a visual inspection system and discusses heuristic search methods used to speed up the configuration optimisation process.

1. Introduction

Most vision-based inspection systems use image processing methods with inherent parameters. The quality of the inspection process will depend on a careful choice for these parameters values. This work is related to the automated configuration of vision-based inspection systems which means the search of the best set among the sets of possible parameters values. This search process is based on a score function reflecting the quality of the inspection system.

The present paper will discuss the search methods employed, their characteristics and the way they were improved to take advantage of the peculiarities of score functions of inspection systems. Next section briefly summarizes the way used to perform the system setup and how the search methods will be evaluated. Third section will present the *simulated annealing* search method, and discuss its theoretical efficiency in this context. Fourth section will describe the used *genetic search method*. Fifth section will present some results gathered with various versions of the search methods on an industrial inspection system for integrated circuits markings. Finally a conclusion ends this paper.

2. Setup of an inspection system

The inspection system must be setup when it is applied on a new task . This setup process involves several operations as depicted in figure 1. The operator defines the control rules: measurements and decision rules. He also selects some representative devices forming a learning set of samples. This learning set is used to configure some parameters of the control rules, mainly those of the image processing (thresholds, kernel size, …) and decision boundaries. This step may be complex and lengthy. Thus, it is desirable to facilitate this control rules configuration by mean of an automatic configurator.

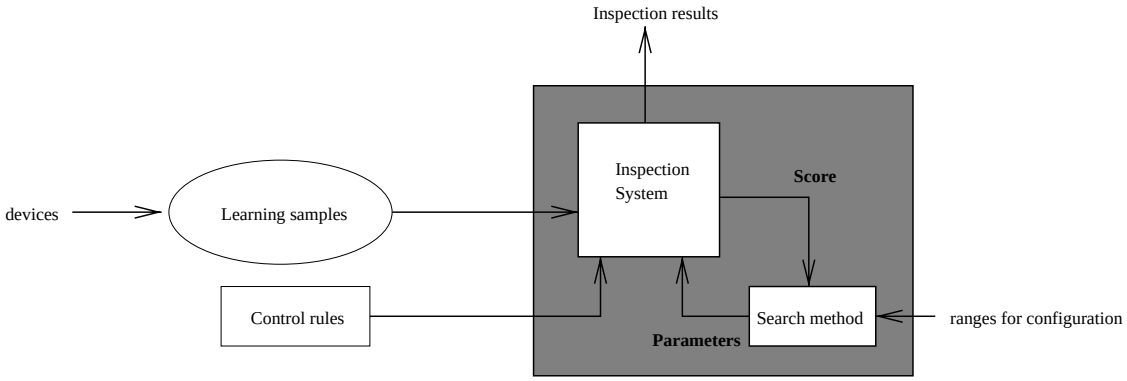

Figure 1:Setup of a quality control system

To achieve this, the developed configurator will do an iterative process, during which it will alternately propose a possible configuration (set of parameters) for the vision system and rate this configuration. The rating will be done with the help of a score function **S** whose definition will depend on the quality control rules used. An example of such a function has been discussed in a previous publication [1]. The configuration proposal will be done by a search method aimed at maximizing the score.

The search should be limited to meaningful values for the different parameters. Each parameter can be given a permitted range of values. On this basis, a high dimensional discrete score function **S** that rates the control system can be defined over a parametric space **P**. The goal of the configurator is to find the configuration within **P** that maximizes the score. Due to the size and high dimensionality of **P**, exhaustive search is not an option and better search methods have to be used.

The systems configuration should be done quickly and thus, the faster the better for the search method. We will consider two heuristic search methods and take a *random trial* method as basis to rate them. An **acceleration function** $A(\alpha)$ will be defined as

$$A(\alpha) = \frac{\tau_{rnd}(\alpha)}{\tau\ (\alpha)} \qquad (1)$$

where τ is the number of trials giving a 50% confidence to reach a fraction α of the maximum score.

A few peculiarities of the score function must be noted :

- The score is interdependent on all the parameters and so it is a complex and unpredictable function
- As the different parameters ranges are given independently, meaningful configurations are only a small subset of **P**
- A realistic quality control system should exhibit a minimum of stability and thus the "good" configurations are expected to form an extended cluster in **P**.

3. Simulated annealing

This search method is a mixture of systematic search by *gradient ascent* with a random process. It is also known as *stochastic relaxation* [2]. We use, here, a particular form of this method that can be summarized as an iterative process consisting of 2 operations:

- Select a random configuration of **P** - *random initialization* –
- Do the best transition within the 2^n closest neighbors until a maximum is reached

n is the dimensionality of parametric space and "best" means the one whose configuration gives the highest score.

3.1. Acceleration estimation

We shall now consider how this method theoretically compares to random search according to acceleration. To do this we make use of the concepts of *cumulative distribution function* and *convergence basin* of the score function.

For the random reference method, if successive trials are independent (no memory of past trials), each has a probability equal to 1/card{**P**} to be the best configuration (supposed to be unique), Consequently, a finite number of random trials n_{rnd} can guarantee a given probability Pr_{MAX} to find the maximum:

$$n_{rnd}(\Pr_{MAX}) = \frac{\log(1-\Pr_{MAX})}{\log\left(\frac{card\{P\}-1}{card\{P\}}\right)} \tag{2}$$

In a similar way, the simulated annealing method will have a minimum number of trials to guarantee a probability to reach the maximum:

$$n_{SA}(\Pr_{MAX}) = N\ (\Pr_{MAX}) < l > \tag{3}$$

where <l> is the average numbers of trials performed for one iteration of the algorithm, and N is the number of iterations that guarantee a given probability to find the maximum. N will depend on the set of points in **P** from which gradient ascent converges to the optimal score. This set will be named *convergence basin (CB)*.

Acceleration A(α) is simply the ratio between equations 2 and 3, To calculate it in a general case (α<1), we define an "annealing score function" S' where score of a given configuration is replaced by maximum score of the CB to which it belongs. Figure 2 shows a one-dimensional convergence space

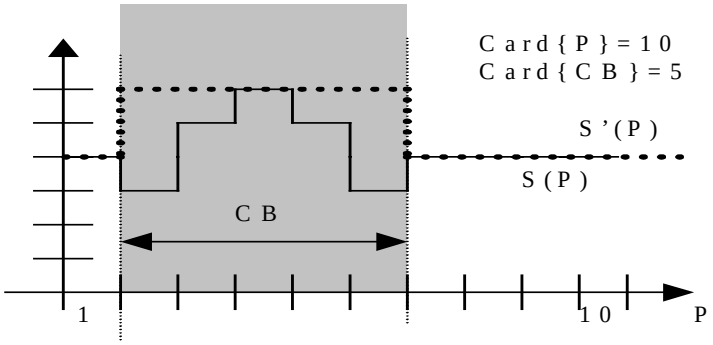

Figure 2: score and modified score function according to convergence space

With the help of cumulative distribution functions D_S and $D_{S'}$ of S resp. S', the acceleration was calculated:

$$A(\alpha) = \frac{\log(D_{S'}(\alpha))}{\log(D_S(\alpha))} \cdot \frac{1}{< l >} \tag{4}$$

To help understand this result we consider two example cases. In left half of figure 3 we have a 2-dimensional score function with a single meaningful configuration. Even in this case, simulated annealing perform as well as random search(A=1). On the right we calculate the acceleration for a score function being nil except for a circular space of radius 20. We observe that acceleration strongly depends on the α factor. This means that simulated annealing is faster than random search as long as the expected performance for the system is high. This corresponds to our requirements.

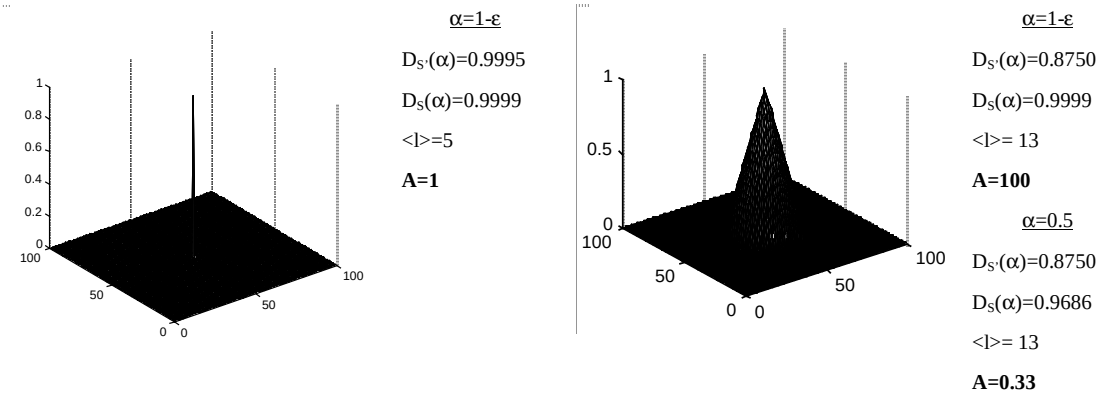

Figure 3: Score functions and corresponding acceleration

Now, we shall take into account the peculiarities of the score function S to increase the acceleration A.

3.2. Initialization enhancement

As stated above, lots of possible configuration are useless and the score function take 0-value over a large fraction of **P**. Furthermore, a quality control system should be stable and it is unlikely to have isolated good configurations. So, we will modify the search method, by iterating the *random initialization* step until a usable configuration (score>0) is found. This will spare 2n controls many times, resulting in a much lower <l> while only slightly degrading the first term of equation 4.

3.3. Addition of "memory"

A second improvement will be the addition of memory to the search: each time a score for a configuration has to be calculated, a check will verify that it has not yet be done; if it was previously calculated the score is just fetched from the memory, otherwise the score is calculated and stored into memory. This might improve the configuration speed for systems when the scores calculation takes a significant amount of time. This improvement will lower the average number of effective trials <l> performed at each iteration.

4. Genetic Algorithm (GA)

Genetic algorithms [3,4] that make use of a *survival of the fittest* law inspired from biology have been used to solve this configuration optimization problem. Each parameter of the quality control system will be considered as a chromosome of the configuration entity, and selection will be based upon the associated score of the entity.

Among the numerous variants of the method, we did some choices to reach a good performance (acceleration) for the configurator. Here is a short review of the operations that are performed, after the initialization, at each iteration of the GA:

1. Selection of pairs of entities (configurations) among the previously tested
2. **crossover**: creation of new entities by combination of the selected pairs
3. **mutation**: perturbation of these new entities
4. evaluation of the new entities

Next figure shows the cycle of this method.

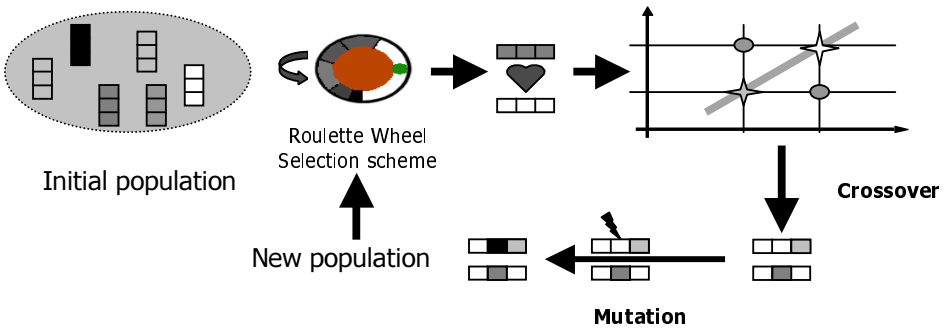

Figure 4: genetic algorithm summary

We want this method to be able to come out of suboptimal solutions by an extended exploration of the untested parts of **P**. Somehow in a similar way as our simulated annealing that starts over randomly after each completed gradient ascent. We do this with a **high rate of random mutations** – 10% - on all parameters of the newly generated configurations. The remaining unchanged 90% of parameters will exploit the promising configurations of the former iteration. An **extended linear crossover operator** will also help to come out of local traps of the score function. It will generate offsprings that may be extrapolation of the parents[1].

4.1 Algorithm enhancements

The peculiarities of the score function may also be used here to enhance the algorithm. The initialization step can be iterated at the start of the algorithm to guarantee that meaningful configurations (score>0) are within the initial set of entities. This will prevent a probable waste of search time in an uninteresting subspace of **P**. The introduction of the "memory" feature to avoid unnecessary evaluations (score calculations) will probably save some time when the algorithm converges towards a small subspace of **P** that has already been expoited.

5. Experiments

A configurator was implemented for a quality control system involved in the inspection of integrated circuits. The particular application of our test is the inspection of the markings printed on the top cover of such circuits. This involves the configuration of five parameters related to image processing, and two additional parameters for the decision rule. These parameters span a discrete 7-dimensional parametric space **P**. Given specific ranges for the parameters, the search space comprises over 10^6 configurations. So in this context, as each evaluation lasts about 100 ms, search acceleration is required. Next figure displays the inspected marking after preprocessing with 2 different configurations.

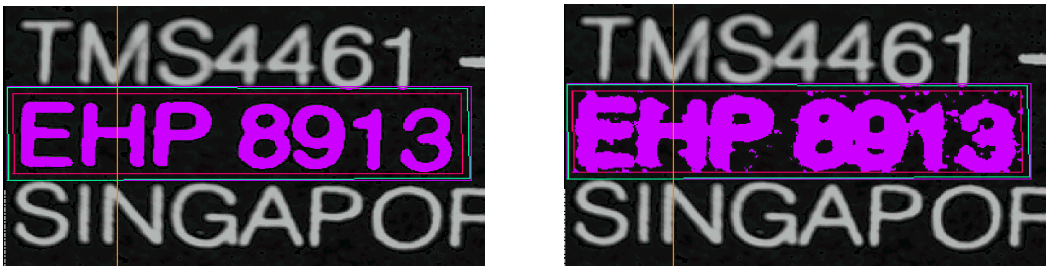

Figure 5: resulting images from a well configured system (left) and from a poorly configured(right)

We will now present the results of the two discussed search methods and see the effect of the enhancements in this application

Repeated automatic configurations of the system (with fixed duration) were done to gather statistical significant data. Plots of the score reached on average after a given time of search let us appreciate how quick the search algorithms are to find a good solution.

The curves of the simulated annealing method in figure 6 show that this method brings a real improvement versus a random search approach. This is remarquable as less than 10% of **P** corresponds to a meaningful configuration of the system. Thus, iterated initialization, is an effective improvement(factor 2 speedup of curves 1,2 versus 3,4). The *memory* feature doesn't show any real speedup of the search in this case as the largest part of **P** is still untested and thus it is not used.

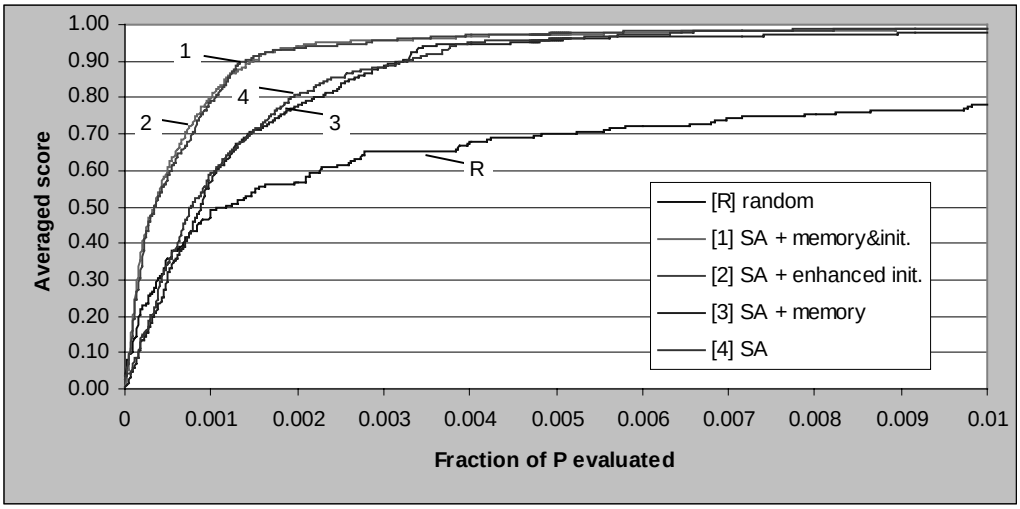

Figure 6 : Simulated annealing versus random method

The genetic algorithm approach exhibits similar behaviour as simulated annealing: a sensible speedup against random trials. The improvement due to a condition on the initial population slightly improves the performance of the method whereas the *memory* feature doesn't make much difference.

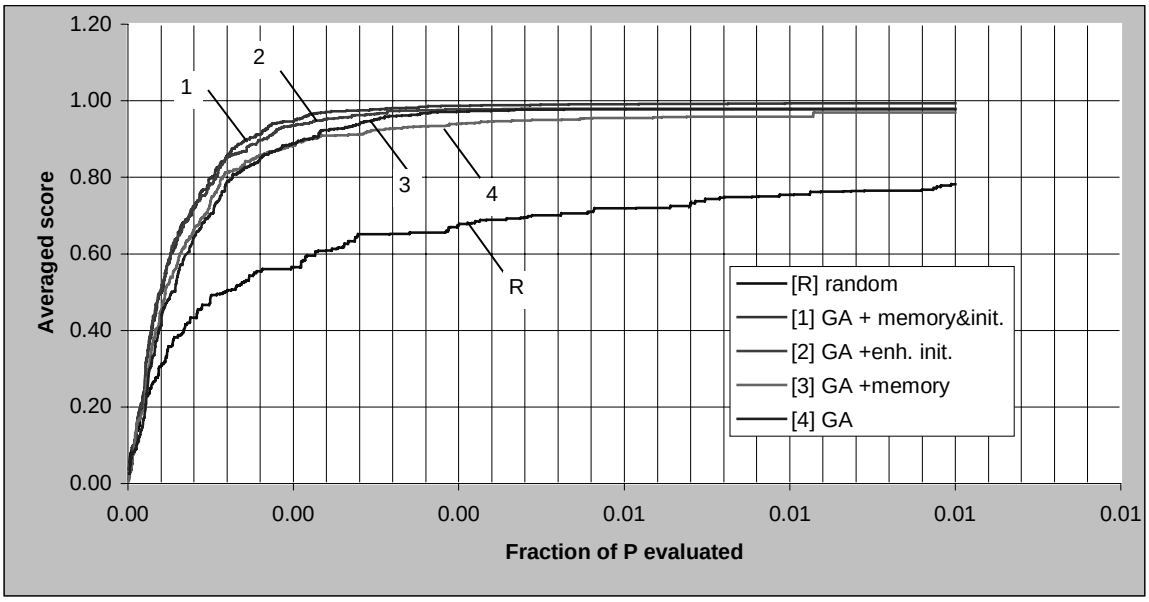

Figure 7 : genetic algorithm versus random method

On the basis of a data set resulting from an exhaustive search over **P** for the configuration of the inspection system, we calculated the acceleration function of the basic simulated annealing method. The curve of figure 8 is the logarithm of acceleration. *Simulated annealing* is efficient only for good solutions (α>0.5). This was expected as one example of figure 3 showed that not so good solutions tend to have a much higher convergence basin. Luckily we are not interested in poor solutions and the search quickly reaches the range where α tends to 1. The observed acceleration in this experiment is around a factor 10. So, the acceleration due to the search method presented here will effectively help to speed up the configuration of the quality control system.

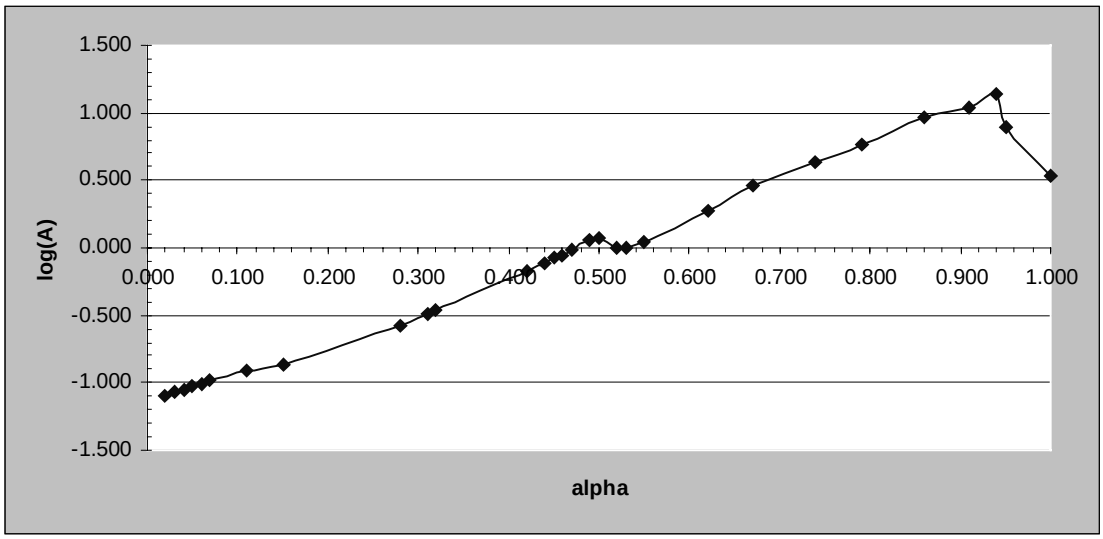

Figure 8: Acceleration logarithm as a function of expected score (SA search)

6. Conclusions

In the context of the setup of a versatile vision-based inspection system, two heuristic search algorithms were considered for automatizing its configuration: *simulated annealing* and *genetic search*. First the methods are described and specific choices are explained. An analysis relates the acceleration of the simulated annealing method to the convergence basin of the score function and shows that weak conditions are sufficient for this method to be efficient. Then the methods are tested in an integrated circuit marking inspection task. They achieve a 10 times speedup versus a random trial method. They permit the system's configuration to be done in a few minutes, significantly reducing the production downtime.

References

[1] O.Husser, H.Hugli, *A configuration assistant for versatile vision-based inspection systems*, Proceedings of SPIE vol.3966 pp259-269, 2000.

[2] S.Geman, D.Geman, *Stochastic Relaxation, Gibbs Distribution and the Bayesian Restoration of Images*, IEEE PAMI. November 1984

[3] D.E.Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading Massachussets, 1989.

[4] M.Srinivas, *Genetic Algorithms: a survey*, IEEE Computer, June 1994