A Distributed Genetic Algorithm for Parameters Optimization to Detect Microcalcifications in Digital Mammograms

Alessandro Bevilacqua[1,2], Renato Campanini[2,3], and Nico Lanconelli[2,3]

[1] Department of Electronics, Computer Science and Systems,
University of Bologna, viale Risorgimento, 2,
40136 Bologna, Italy
abevilacqua@deis.unibo.it
[2] INFN (National Institute for Nuclear Physics), viale Berti Pichat, 6/2,
40127 Bologna, Italy
{bevila, lanconel, campanini}@bo.infn.it
[3] Department of Physics, University of Bologna, viale Berti Pichat, 6/2,
40127 Bologna, Italy

Abstract. In this paper, we investigate the improvement obtained by applying a distributed genetic algorithm to a problem of parameter optimization in medical images analysis. We setup a method for the detection of clustered microcalcifications in digital mammograms, based on statistical techniques and multiresolution analysis by means of wavelet transform. The optimization of this scheme requires multiple runs on a set of 40 images, in order to obtain relevant statistics. We aim to evaluate how fluctuations of some parameters values of the detection method influence the performance of our system. A distributed genetic algorithm supervising this process allowed to improve of some percents previous results obtained after having "hand tuned" these parameters for a long time. At last, we have been able to find out parameters not influencing performance at all.

1 Introduction

The presence of microcalcifications in breast tissue is one of the most important signs considered by radiologist for an early diagnosis of breast cancer, which is one of the most common form of cancer among women. Statistical analyses show how errors in microcalcifications detection are very high in population screening programmes. A feasible solution, in order to reduce these kind of errors, consists in providing doctors with a computer aided system, which could act as a "second radiologist". Experiments showed that these systems can significantly improve the accuracy in the detection task. Our Computer Aided Diagnosis (CAD) scheme described in [1] is quite complex and its effectiveness depends on the values of different parameters. Therefore it is necessary to optimize the choice of these parameters, in order to achieve good performance. Unfortunately, their number is very high (about thirty) and they are correlated with each other.

E.J.W. Boers et al. (Eds.) EvoWorkshop 2001, LNCS 2037, pp. 278–287, 2001.

Consequently, it is difficult to get an optimal choice of them. In the earlier study the selection were performed manually; we refer to this procedure as the "hand tuned" one. In this paper, we present an automated method for the selection of the parameters values by means of a genetic algorithm. Genetic Algorithms (GAs) search the solution space to maximize (minimize) a fitness (or cost) function by using simulated evolutionary operators such as mutation and sexual recombination. In this study the fitness function to be maximized reflects the goal of maximizing the number of true-positive detections while minimizing the number of false-positive detections.

GAs are currently applied to many diverse and difficult optimization problems (see [2] and [3]). In a number of applications where the search space was too large for other heuristic methods or too complex for analytic treatment GAs produced favorable results. Other researchers in [4] and [5] have shown that GAs could improve the performance of a CAD scheme.

In the present study, we will evaluate how the parameter values fluctuations influence the performance of the CAD scheme and which parameters more affect the cost function; our goal is as well to select, by using a GA, the most significant parameters. The GA needs to evaluate several generations, in order to obtain a good optimization. Due to the very long time required for one run, it would be almost impracticable to execute the GA on a sequential architecture. We therefore implement a distributed GA on a small Network Of Workstations (NOWs), by realizing a global parallelized GA. In this type of parallel GAs, there is only one population, as in the serial GA, and even if the evaluation of individuals is parallelized explicitly, the algorithm remains unchanged. In this way, we could easily apply the existing principles for sequential GAs.

We accomplish the optimization of our CAD scheme by using the 40 digitized mammograms of the Nijmegen database. Performances of the detection scheme are shown by means of Free Response Operating Characteristic (FROC) curves: they display the number of true positive clusters of microcalcifications detected versus the average number of false positives per image.

2 The Detection Method

Microcalcifications are very small spots which appear brighter than the surrounding normal tissue. Typically they are between 0.1 mm and 1 mm in size and are of particular clinical significance when found in clusters of five or more in a 1 cm^2 area. Most of the clusters consist of at least one evident microcalcification and other more hidden signals.

Our approach to the detection task includes two different methods: the first one (coarse) is able to detect the most obvious signals, while the second one (fine), based on multiresolution analyses, discovers more subtle microcalcifications (see [1]). Signals coming out from these two methods are combined through a logical OR operation and then clusterized to give the final result. There are some steps, common to both coarse and fine methods:

– *pre-processing*, which isolates breast tissue;

- *filtering*, in which structured background is removed;
- *signal extraction*, to find out microcalcifications candidates signals;
- *false positive reduction*, where microcalcifications are separated, by calculating a set of features, as described in [1] and [6] from false signals extracted.

In all these tasks there are several parameters to be tuned; we used twenty-nine of them (listed in Table 1) for the optimization process with the GA. A very critical phase of every CAD system is the false positive reduction step: here a detected signal is considered either microcalcification or false signal, according to the value of a set of its features.

Table 1. Parameters used in the optimization process

Parameters of the coarse method	Parameters of the fine method
Size of the local thresholding window	
Threshold for Gaussianity test h_t	
Values for local thresholding k	
Minimum edge gradient (EG)	
Maximum EG	
Minimum average local gradient (ALG)	
Maximum ALG	
Minimum area of signal	Minimum area of signal
Maximum area of signal	Maximum area of signal
Minimum gray level (GL)	Minimum gray level (GL)
Maximum GL	Maximum GL
Minimum degree of linearity (DL)	Minimum degree of linearity (DL)
Maximum DL	Maximum DL
ct_{11}, in: GL $> ct_{11}$ * EG + ct_{12}	p_1, in: EG $< p_1$ * tanh(p_2 * GL)
ct_{12}, in: GL $> ct_{11}$ * EG + ct_{12}	p_2, in: EG $< p_1$ * tanh(p_2 * GL)
ct_{21}, in: DL $< ct_{21}$ * ALG + ct_{22}	p_3, in: EG $> p_3$ * GL + p_4
ct_{22}, in: DL $< ct_{21}$ * ALG + ct_{22}	p_4, in: EG $> p_3$ * GL + p_4
ct_{31}, in: DL $> ct_{31}$ * ALG + ct_{32}	
ct_{32}, in: DL $> ct_{31}$ * ALG + ct_{32}	

Most of the parameters are thresholds to choose the range of features values in this false positive reduction phase; others are used for selecting regions of interest or extracting signals (see [1]). Any individuals of the population considered in GA optimization is therefore described by a string (gene) of twenty-nine values. Each one of them represents a parameter value and can be a real or an integer number, according to the domain of the parameter itself. The purpose of the optimization of a CAD scheme is to find out the set of parameters which gets the highest number of true positive clusters of microcalcifications with the lowest rate of false positive clusters, i.e. the best tradeoff between sensitivity and specificity. This tradeoff is described by the design of the fitness function.

3 The Genetic Algorithm

3.1 Design

The advantages in using GAs are that they require no knowledge or gradient information about the response surface, they are resistant to becoming trapped in local optima and they can be employed for a wide variety of optimization problems. On the other hand GAs could have trouble in finding the exact global optimum and they require a large number of fitness functions evaluations. It is very difficult to achieve analytic relationship between the sensitivity of the CAD and the parameters values to be optimized. Since a GA does not need this kind of information, it should be suitable in our optimization task.

If there is an explicit knowledge about the system being optimized, that information can be included in the initial population. In this study we initialize the population to the best "hand tuned" results.

An evolutionary strategy needs to be adopted in order to generate individuals for the next generation. We chose an *elitist generation* as replacement operator. Namely the individuals are ranked by their fitness and only the best of them (10% of the population) are taken unchanged into the next generation. In this way, we guarantee that good individuals are not lost during a run. Other children come from crossover and mutation, (their associated probabilities are p_{CO} and p_{MUT} respectively).

The aim of the fitness function is to numerically represent the performance of an individual. In our case a couple (true positives, false positives) is mapped by this function to a real number normalized between 0 and 1. That number encodes the excellence of a couple obtained by a particular individual (i.e. a set of parameters in the CAD scheme). We designed the cost function as a 2D gaussian, with the maximum in the most desired point (100% of true positives and 0 false positives) and variance equal to 15% true positives and 2.2 false positives.

In order to end the evolution of the population we choose the following termination criterion: we stop the evolution when the average of the fitness of the best six individuals has reached a plateau. The final result of the GA optimization is the best individual of the last iteration.

3.2 Implementation

The GA supervises the executions of an existent "basic program", which solves the domain problem and provide fitness evaluation. In the sequential program, the fitness evaluation step is computed independently for each individual of the population, by means of a *for* loop. We have one more loop that is innate in the "basic program" (the detection algorithm), which performs the detection scheme described in Sect. 2 over a whole database of mammographic images: the evaluation of one individual so requires the independent execution of the detection program *for each* image in the database. These loops are exploited in the parallel development of the algorithm.

Global parallelization is one of the most common way to realize a parallel GA. In global parallelization (see [7]), any individual can mate with any other because the operators and the evaluation of the individuals are explicitly parallelized, often by a "master processor" that sends individuals to other processors for the evaluation and applies genetic operator itself. The master program stores the entire population and performs an iterative decomposition: on each generation it sends a fraction of the population (one or more individuals) to each slave processors and waits results from the slaves to come back.

Slaves are self-scheduled, they ask the master for more work as their task ends. This algorithm behaves in a synchronous manner, since the master waits to receive the fitness values for *all* the individuals, before proceeding into the next generation. Once all individuals have been computed, the master performs replacement, crossover and mutation operations to create a new generation. In this way, the GA operations keep global and the existing design guidelines for simple (sequential) GAs are directly applicable.

In our GA, individuals are short strings of bytes and they are not time consuming, from a communication point of view. For this reason, the data parallelism of the detection algorithm can be exploited too. We recall that in the sequential algorithm an individual processes a whole images database. The easiest distributed implementation is to let an individual to sequentially process the entire set of images too. In this way, for each generation the program waits for the slowest slave to end its computation: in the worst case this time is the one necessary to process a database. Thus we create a new item, named *chunk*, which is constituted by an image identifier and by an individual. In this way, it is not necessary that one slave compute a whole database, but this task may be assigned to different slaves. Therefore, the master program sends a chunk, instead of one individual, every generation to each slave process, which sends back results and the individual identifier. In this way, the maximum idle time of the program shrinks from the time needed for computing a whole database to the time to compute one image. Here we use a modified version of the manager-workers paradigm, the "*working-manager*" model defined in [8], in which the manager uses its idle time to process data itself, by so increasing the overall performance.

4 Results

4.1 The Analysis of Performance

NOWs are cluster of workstations with quite slow communication links, that anyway can be suitable for a large number of applications. In addition, small Symmetric MultiProcessor (SMP) systems can be found within many of the modern computers which may lead to a powerful distributed computer if connected to each other.

The cluster we used is a heterogeneous computers network consisting of 6 workstations (10 computing nodes), connected to a LAN by a 100 Mbit

Ethernet. Workstations are listed below, according to their performance on the sequential algorithm:

W_1) 1 PentiumIII 450Mhz, 512MB RAM;
W_2) 1 SMP: 2 PentiumIII 450Mhz, 256MB RAM;
W_3) 1 SMP: 2 PentiumII 450Mhz, 512MB RAM;
$W_{4,5}$) 2 SMPs: 2 PentiumII 400Mhz, 512MB RAM;
W_6) 1 Mobile PentiumII 366Mhz, 128MB RAM;

We adopt static mapping and do not introduce any virtual parallelism degree, by assigning one task to each processor, for a total amount of 10 processes.

All the code is written in C, *Linux* is the operating system, *PVM* libraries supply the communication routines and *gcc* is the C compiler.

Each population is constituted by 30 individuals, and each generation required 27 individuals to be computed. Each generation takes about 1 h of elapsed time to be calculated on this cluster.

We get both *CPU time* and *wall clock time* measures necessary to obtain final results. This data parallel application has a very coarse grain structure and in addition small amount of data are transferred when communication takes place. For this reason, time due to communication between master and slaves is irrelevant and it has not been considered.

We then focus our attention on the "weighted" efficiency (Fig. 1) of each workstation:

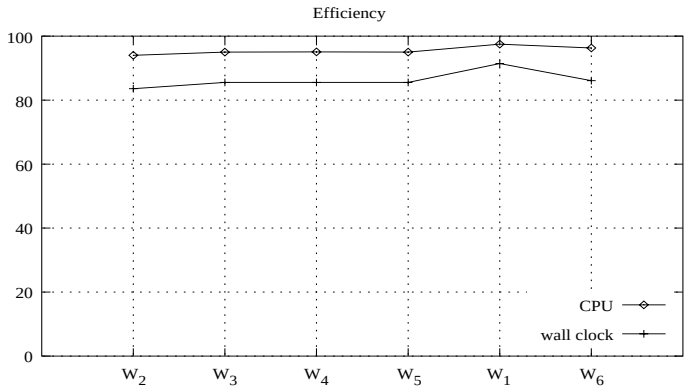

Fig. 1. The "Weighted" efficiency of each workstation

$$Eff_i = \frac{S_i}{I_i} \tag{1}$$

where S_i and I_i are the relative speedup and the "ideal" (theoretical) speedup, respectively. S_i is defined as:

$$S_i = \frac{T_s(W_i)}{T_p} \tag{2}$$

where $T_s(W_i)$ is the time it takes to the sequential algorithm on the workstation W_i, and T_p is the time of the parallel algorithm. Finally, I_i:

$$I_i = \frac{P_T}{P_i}. \tag{3}$$

Here, P_i=$T_s(W_1)/T_s(W_i)$ is the *power weight* of W_i compared with that of the fastest workstation W_1. P_T is the power weight of the *whole* cluster and its value is obtained by summing over all P_i.

We observe how the intrinsically parallel nature of this problem has been successfully exploited, by dividing original data in chunks. We obtain excellent results in term of efficiency, if we consider CPU time. Experiments showed a mean idle time value of about 10 sec.

Regarding with the time needed to read images, it would be quite significant, since images are only local to W_3. Each image (8 MB) takes about 1.5 sec to be read, and for each generation 27×40 images should be read. This sequential step causes a loss in terms of efficiency, even if we tried to keep in memory a few images, and it becomes more visible if we consider wall clock time.

4.2 Experimental Results

The main goal of the present study is to show that the performance of our CAD detection scheme improves due to the optimization based on the GA. To this

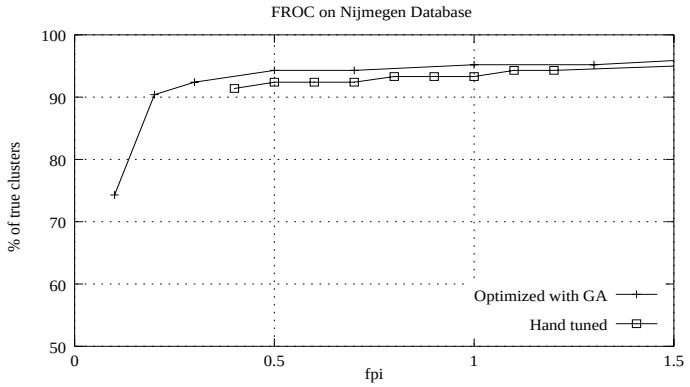

Fig. 2. FROC of the detection methods on the 40 images (Nijmegen) database

end we depict in Fig. 2 two FROC curves: one related to the "hand tuned" method and one for the optimized one. The optimized curve is derived from the best individual, by varying the value of the parameter p_1. Both the curves are obtained on the entire database of 40 images: here we do not divide the data into training and test groups, rather we use all the images for training and testing. We can see that the sensitivity of the optimized scheme is always greater than in the

"hand tuned" case. The GA allows to outperform the previous method of some percents. Even if at first look it could seem a slight improvement, nevertheless this is extremely important because here it is necessary to minimize the losses of clusters of microcalcifications, maintaining at the same time a low rate of false alarms. Indeed, to avoid any losses of suspect cases is a vital point in issues regarding the detection of lesions for early breast cancer diagnosis; therefore, any little step towards a sensitivity of 100% is crucial. The best solution obtained by the GA is an individual with a fitness value of 0.827, which corresponds to 94.3% of true clusters with 0.47 false positives per image. To obtain these results we utilize uniform crossover p_{CO}=0.8 and p_{MUT}=0.1. The convergence of the GA evolution has required the computation of 1974 individuals, corresponding to 73 generations: it took roughly 3 days (76 h) on our NOW. Let's take a look at the peculiarity of the best individual: focusing our attention on its genes values, we can find out the differences between them and the parameters values of the "hand tuned" results. We can notice that these changes reduce the range of the values which identify the true signals. In particular, regarding the coarse method, the ranges of area, gray level, edge gradient and degree of linearity are narrower than those achieved in the "hand tuned" study (e.g. the minimum area of signal increases from 3 to 5 pixels, whereas the maximum area decreases from 30 to 22). In Fig. 3 it is possible to observe a similar effect about the fine method. There are plotted the curves, described by parameters p_1 and p_2, which separate true signals from false detections. Signals above the curve are kept,

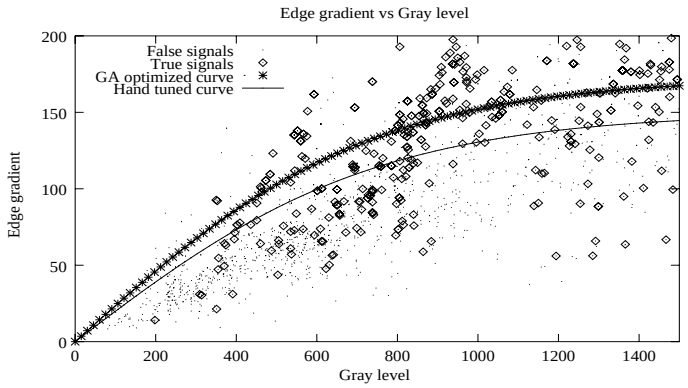

Fig. 3. Scatter plot of edge gradient versus gray level. The two curves, described by p_1 and p_2, separate false signals from true ones

while signals below it are eliminated. Also in this case the GA is more selective in maintaining true signals: if a signal has a given gray level, the GA keep it only if the edge gradient is higher, with respect to the "hand tuned" case. We can therefore summarize that the GA optimization tends to restrict the range of

the features, which characterize the true signals. That allows to cut the number of false positive signals, without losing too many true ones.

Another issue investigated is how the parameters values fluctuations influence the performance of the CAD system. Starting from the best individual we vary the value of the first gene around its best value (the one discovered by the GA), keeping the other genes fixed. The variation of the parameter ranges around ±50% its best value. We repeat this process for all the genes, each time maintaining the other values fixed at their best solution. In this way, we can see which parameter more affect the fitness value.

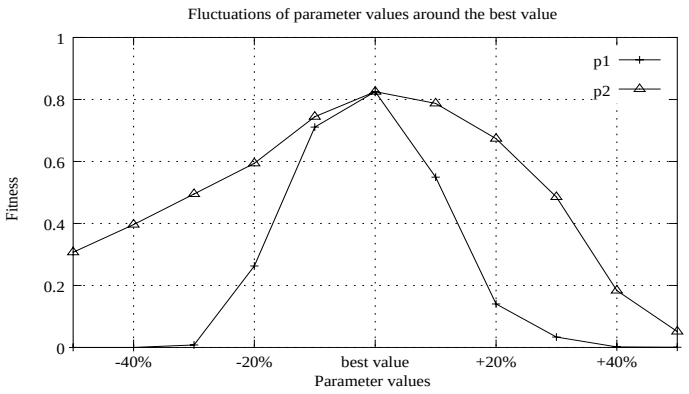

Fig. 4. Variation of fitness due to the change of parameters p_1 and p_2, with the other parameters fixed at their best values

In Fig. 4 we can see an example of how the fitness changes, due to the variation of p_1 and p_2 separately. We notice that p_1 is a very significant parameter, because the fitness goes to zero, with a modification of p_1 of only ±30% around its best value. It turns out that the k value for local thresholding (coarse method), p_1, p_2 and the maximum of the degree of linearity (fine method) are the most significant parameters. A little fluctuation of their values indeed implies a rapid fall of the fitness. On the other hand, there are some genes which do not affect the value of the fitness. They are, in particular: the maximum of the edge gradient, the minimum and the maximum of the average local gradient, ct_{31} and ct_{32} for the coarse method and p_4 for the fine method. A variation up to 50% of them around their best value does not imply any change in the fitness value. This fact has allowed us to perform an optimization without these parameters (keeping them fixed at their best value). With this reduced set of parameters (23 genes instead of 29) we obtain the same results of those cited above (fitness value of 0.827), by analyzing only 1623 individuals instead of 1974 (60 generations instead of 73).

5 Conclusion

We have been developing a Computer Aided Diagnosis system for the detection of clustered microcalcifications in digital mammograms. The performance of the system strongly depends on the settings of a set of several parameters. The goal of the present research is to optimize those parameters, in order to restrict the range of the features, which characterize the true signals, allowing the cut of the number of false alarms, without losing too many true signals detected. In an early version, this task was performed manually, and it took to us a very long time. In this paper, we presented an heuristic approach to this optimization problem by means of a distributed genetic algorithm. We have obtained better results in few generations, which correspond to a really short interval time. Finally, we have been able to discover the most significant parameters, being those that more affect the cost function.

Notes and Comments. Images were provided by courtesy of the National Expert and Training Center for Breast Cancer Screening and the Department of Radiology at the University of Nijmegen, the Netherlands.

This work is supported by the Italian National Institute for Nuclear Physics (CALMA project).

References

1. Bazzani, A., Bevilacqua, A., Bollini, D., Brancaccio, R., Campanini, R., Lanconelli, N., Romani, D.: System for automatic detection of clustered microcalcifications in digital mammograms. Int. J. Mod. Phys. C **11** (2000) 901–912
2. Dhawan, A. P., Chitre, Y., Kaiser-Bonasso, C., Moskowitz, M.: Analysis of mammographic microcalcifications using gray-level image structure features. IEEE Trans. Med. Imag. **15** (1996) 246–259
3. Dokur, Z., Olmez, T., Yazgan, E.: Classification of MR and CT images using genetic algorithms. Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society **20** (1998)
4. Anastasio, M., Yoshida, H., Nagel, R., Nishikawa, R. M., Doi, K.: A Genetic Algorithm-Based Method for Optimizing the Performance of a Computer-Aided Diagnosis Scheme for Detection of Clustered Microcalcifications in Mammograms. Med. Phys. **25** (1998) 1559–1566
5. Yoshida, H., Anastasio, M., Nagel, R., Nishikawa, R. M., Doi, K.: Computer-Aided Diagnosis for Detection of Clustered Microcalcifications in Mammograms: Automated Optimization of Performance Based on Genetic Algorithm. Proceedings of IWCAD 1997, (Elsevier Science B.V., The Netherlands) (1997)
6. Ema, T., Doi, K., Nishikawa, R. M., Jiang, Y., Papaioannou, J.: Image feature analysis and Computer Aided Diagnosis in digital radiography: reduction of false-positive clustered microcalcifications using local edge-gradient analysis. Med. Phys. **22** (1995) 161–169
7. Cantú-Paz, E.: A survey of Parallel Genetic Algorithms. Report No. 97003, (Univ. of Illinois, Urbana, 1997) (1997)
8. Bevilacqua, A.: A dynamic load balancing method on a heterogeneous cluster of workstations. Informat. **23** (1999) 49–56