

Optimisation algorithms for ECG data compression

D.Haugland*, J.G.Heber*, J.H.Husøy*

Abstract

This paper demonstrates the use of exact optimisation algorithms for compressing digital ElectroCardioGrams (ECG). As opposed to traditional time-domain methods using heuristics for selecting a small subset of representative signal samples, we formulate the problem of selecting this subset in rigorous mathematical terms. This approach enables us to derive algorithms guaranteeing the smallest possible reconstruction error when interpolating a bounded selection of signal samples. The proposed model resembles well-known network models, and is solved by a cubic dynamic programming algorithm. When applied to standard test problems, the algorithm produces a compressed representation for which the distortion is about one half of that obtained by traditional time-domain compression techniques at reasonable compression ratios. This illustrates that in terms of accuracy of decoded signals, existing time-domain heuristics for ECG compression may be far from what is theoretically achievable. The current paper is an attempt to bridge this gap.

Keywords: Electrocardiography, Data compression, Time-domain algorithm, Optimisation, Graphs, Shortest path, Resource constraint, Dynamic programming.

1 Introduction

Owing to the large amount of data involved, storage or transmission of sampled ElectroCardioGraphy (ECG) signals is expensive. Compressing the data in such a way that accurate reproductions can be made is therefore of vital importance. Due to high redundancy in sampled ECG signals, savings in transmission time and storage requirements may be considerable.

The problem of compressing ECG data has traditionally been attacked by heuristic time-domain methods such as AZTEC (COX et al., 1968), CORTES (ABERNSTEIN and THOMPSON, 1982) and FAN (BARR, 1988). Several attempts have been made recently to improve the time-domain methods, and among these are techniques such as SLOPE and AZT-DIS (TAI, 1991; 1993). The basic idea of all these algorithms is to break the ECG signal into consecutive linear segments. This is done by selecting a set of *significant samples* in the encoder. The reconstruction, in the decoder, is done by performing linear interpolation between the retained samples. Selection of significant samples is based on some heuristic which is unique for each method. Instead of minimising the *total* reconstruction error, the heuristic typically attempts to limit the *maximum* reconstruction error.

In this paper we study how graph theory and optimisation can be used to solve the compression problem, and we formulate a mathematical model expressing the best possible

*Dept. of Electrical Engineering and Computer Science, Stavanger College, P.O. Box 2557, N-4004 Stavanger, Norway.

compression subject to given conditions. The notion of goodness may be defined in various ways, and the question of what conditions to impose may have multiple answers. We make a subjective choice of optimisation criterion and corresponding restrictions to constitute the model. Only time-domain methods are considered in this paper.

The advantage of optimisation models and algorithms is that given some realistic assumptions, they guarantee the best possible data compression. However, such approaches may suffer from lack of ability to satisfy the real-time requirements implied by the application in question. Although this is not true in every case, the processing time of exact optimisation algorithms may by far exceed the one of a simple heuristic. Hence the goal of the current work is not to replace existing methods, but rather the following:

- Provide a tool for analysing heuristic approaches to the problem.
- Serve as a source of ideas to alternative methods.

In the references above, heuristics are compared experimentally. If the answer to how well the methods theoretically can perform were available, such a comparison could be given another dimension. The gap between the quality of proposed methods and theoretically achievable quality also holds important information on the need for improvements. Such considerations justify the first goal above.

In cases where exact algorithms are too inefficient, these may still be a starting point to simpler but sufficiently accurate methods. If some conditions may be relaxed without serious loss of realism, or if shortcuts in the exact versions can be made with only modest performance degradations, useful techniques related to optimisation can be obtained. In later sec-

tions we illustrate how this idea can be applied to the problem under study.

A suggestion to how the optimisation problem can be formulated is given in the next section. In Section 3, we discuss how the model can be solved, and next the prospectives of the method are discussed. In Section 5 we demonstrate experimentally how optimisation can contribute to ECG data compression, and in the concluding section practical aspects of the method are considered.

2 Optimisation model

Let $y(1), y(2), \dots, y(N)$ be samples of an ECG-signal made with constant interval length, and define the *sample set* $S = \{y(1), y(2), \dots, y(N)\}$. We seek an appropriate *compression set* $C = \{n_1, n_2, \dots, n_M\} \subset \{1, 2, \dots, N\}$ and corresponding sampled values to represent S . Assume $n_1 = 1$ and $n_M = N$. Define the approximations $\hat{y}(n) = y(n)$ if $n \in C$, and $\hat{y}(n) = \frac{y(n_{m+1}) - y(n_m)}{n_{m+1} - n_m}(n - n_m)$ when $n_m < n < n_{m+1}$.

Hence we arrive at a continuous piecewise linear function which interpolates $\{(n, y(n)) \mid n \in C\}$. How well the function fits the other sampled values depends on the choice of C . One acceptable fitness criterion might be the *root mean square error* (RMS) defined as $\rho_C = \sqrt{\sum_{n=1}^N (\hat{y}(n) - y(n))^2}$. The contribution to ρ_C^2 from points along the straight line between $(n_m, y(n_m))$ and $(n_{m+1}, y(n_{m+1}))$, equals $\sigma_m = \sum_{n=n_m+1}^{n_{m+1}-1} (\hat{y}(n) - y(n))^2$ and hence $\rho_C^2 = \sum_{m=1}^{M-1} \sigma_m$.

Define the *digraph* $D = (V, A)$ whose *vertex set* $V = \{1, 2, \dots, N\}$. Furthermore, the *arc set* A contains the ordered pair (i, j) if and only if $i, j \in V$ and $i < j$. For any two vertices $n_1, n_M \in V$, the ordered set (n_1, n_2, \dots, n_M) is said to be a *path* from n_1 to n_M in D if

$n_1, \dots, n_M \in V$ are distinct vertices such that $(n_{m-1}, n_m) \in A$ for $m = 2, \dots, M$. In our case this is equivalent to $n_1 < n_2 < \dots < n_M$. In this paper P_n denotes a path from vertex 1 to vertex n . Whenever the origin of a path is unmentioned, it can be assumed to be vertex 1.

Any arc $(i, j) \in A$ represents the possibility of including i and j as *consecutive* members of C . Hence we actually explore D for a suitable path P_N to N . The set of vertices traversed by this path will constitute C . One effect of including i and j as consecutive vertices in P_N is a contribution to ρ_C^2 from all intermediate vertices $i+1, \dots, j-1$. This is expressed as $c_{ij} = \sum_{n=i+1}^{j-1} (\hat{y}(n) - y(n))^2$ where $\hat{y}(n)$ is found by linear interpolation from $(i, y(i))$ to $(j, y(j))$. It is important to note that c_{ij} hence only depends on $y(i), y(i+1), \dots, y(j)$.

The heuristics proposed in (BARR, 1988), (TAI, 1991) and (TAI, 1993), strive to find a preferably small collection C such that the total RMS also is small. Computational experiments show that the heuristics provide reasonable compromise solutions to these conflicting criteria. The idea in TAI(1993) is to start with a sufficiently small C and gradually extend it by samples for which the interpolation error currently is large. Termination occurs when this error at every sample is below a user-supplied upper bound.

In an optimisation approach, one of the criteria is defined the *ultimate goal*, whereas the other is represented in terms of a *constraint*. One possible model is hence to minimise RMS subject to the constraint that no more than a prescribed number (M) of samples may enter C .

Referring to the digraph defined above, the problem amounts to minimising the *length* of P_N , such that P_N contains no more than M vertices. The length of a path is defined as the sum of the lengths of the arcs connecting the vertices on the path, and the length of each

arc is defined as c_{ij} .

The problem is an instance of the *resource-constrained shortest path problem* (ANEJA et al., 1983; BEASLEY and CHRISTOFIDES, 1989). The resource in question is the number of vertices on the path. Unlike the general version of the problem, our model contains only one resource constraint (JOKSCH, 1966; ANEJA and NAIR, 1978; HANDLER and ZANG, 1980; RIBEIRO and MINOUX, 1985). Omitting the resource constraint, we simply face the frequently studied *shortest path problem* (DIJKSTRA, 1958).

Because of the particular choice of resource in our case, we term our problem the *cardinality constrained shortest path problem* (CCSP). In the next section, we derive a recursive formulation of CCSP, and based on this we give an algorithm providing the exact solution.

3 Optimisation algorithm

In order to establish an efficient solution scheme, we propose the following precise problem formulation. Define P_{jm} as the shortest path to j visiting *exactly* m vertices, and let $f(j, m)$ denote the length of P_{jm} . We actually search for a $1 < m^* \leq M$ and the corresponding P_{Nm^*} for which $f(N, m^*) = \min_{1 < m \leq M} f(N, m)$. In our search, we compute all such paths P_{Nm} in the order given by increasing values of m . The CCSP problem is hence solved when these quantities become available.

Consider the path $P_{j,m+1}$, and denote the second last vertex in $P_{j,m+1}$ by i . Obviously, $i < j$ and $i \geq m$. Hence $P_{j,m+1}$ contains a subpath through m vertices to i . But this subpath has to be $P_{i,m}$, because otherwise we could find a shorter path through m vertices to i . Augmenting this with vertex j yields a path shorter than $P_{j,m+1}$ (through $m+1$ vertices) ending at j , contradicting the fact that $P_{j,m+1}$

```

Algorithm CCSP
for  $j = 2, \dots, N$ 
   $f(j, 2) = c_{1j}$  // Length of two-vertex path
   $p(j, 2) = 1$  // from 1 to  $j$ 
   $m^* = 2$  // Assume the two-vertex path to  $N$  is optimal
for  $m = 2, \dots, M - 1$  // Find  $m + 1$ -vertex paths
  for  $j = m + 1, \dots, N$  // Find the path to  $j$ 
     $p(j, m + 1) = m$  // Assume  $P_{j, m+1} = \{1, 2, \dots, m, j\}$ 
     $f(j, m + 1) = f(m, m) + c_{m, j}$  // The length of this path
    for  $i = m + 1, m + 2, \dots, j - 1$  //  $P_{j, m+1}$  may equal  $P_{im} \cup \{j\}$ 
      if  $f(i, m) + c_{ij} < f(j, m + 1)$  // Shorter!
         $f(j, m + 1) = f(i, m) + c_{ij}$  // Update the shortest length
         $p(j, m + 1) = i$  // Record the last step
    if  $f(N, m + 1) < f(N, m^*)$  // Shortest path to  $N$  so far
       $m^* = m + 1$  // Optimal number of vertices in path to  $N$ 
end

```

Figure 1: Algorithm for the CCSP problem

is the shortest of all such paths.

Hence we have that $f(j, m + 1) = f(i, m) + c_{ij}$ for some $i = m, m + 1, \dots, j - 1$. Furthermore, it is clear that i must be the vertex *minimising* the right hand side in this equation. Supplying the obvious condition that $f(j, 2) = c_{1j}$, we arrive at the recursive equations

$$f(j, 2) = c_{1j} \quad (1)$$

$$f(j, m + 1) = \min \{f(i, m) + c_{ij} \mid i = m, \dots, j - 1\} \quad (2)$$

When $j = 2, \dots, n$ and $m = 2, \dots, M - 1$ are inserted in (1)-(2), $f(N, M)$ is uniquely defined.

Equations(1)-(2) constitute a *dynamic programming* formulation of CCSP. Similar formulations have been proposed for various constrained shortest path problems (SAIGAL, 1968; ROSSEEL, 1968; BEASLEY and CHRISTOFIDES, 1989). The formulation above resembles the one given in (SAIGAL,

1968), but in (2) we exploit the fact that $(i, j) \in A$ only if $j > i$. Furthermore, we disregard $f(i, m)$ for all $i < m$ since all paths terminating at i have at most i vertices.

From the above formulation, the algorithm in Figure 1 suggests itself ($p(j, m)$ signifies the predecessor of j in P_{jm}).

The compression set can thus be recorded by the algorithm:

$$\begin{aligned}
n_{m^*} &= N \\
\text{for } m &= m^*, m^* - 1, \dots, 2 \\
n_{m-1} &= p(n_m, m)
\end{aligned}$$

This produces the interpolation points $(n_1, y(n_1)), (n_2, y(n_2)), \dots, (n_{m^*}, y(n_{m^*}))$.

It is easily seen that when all arc lengths are available, the computations above involve $\mathcal{O}(MN^2)$ arithmetic operations. Computation of all c_{ij} -values can be shown to require no more than a total of $\mathcal{O}(N^2)$ operations.

4 Benefits of CCSP

Given the compression rate, there exists no better (in terms of RMS) encoded representation than the one generated by the algorithm in Figure 1. Although the truth of this statement is restricted to time-domain methods involving linear interpolation between selected sample points, the algorithm guarantees to produce more accurate representations than several popular methods. Experiments reported in the next section show that compared to conventional methods, the RMS reduction achieved by the CCSP approach is considerable.

Usually, significant error reductions are computationally expensive, and the suggested approach is definitely more time consuming than traditional methods. But even in cases where the algorithm might violate the time constraint induced by the real time application, it serves as a powerful tool when analysing possible heuristic methods.

When considering what compression method to use in a given project, various heuristics may be compared in terms of e.g. computational efficiency and total error in the decoded signal. In order to assess the potential in further refinements of the heuristics, the theoretical minimum signal error holds important information. If results close to this bound are observed when applying a suggested method to realistic data, there is not much point in further enhancements. Otherwise, it is indicated that attempts to improve the sample selection may be worthwhile. Since our method produces the minimum interpolation error, the CCSP solution hence is a *reference* for any other time-domain method in question.

In addition to purposes of the kind mentioned above, considering data compression problems in terms of graph theory may be beneficial when developing heuristic methods. Once the problem is recognised as a cardinality

constrained shortest path problem, we know precisely what the heuristic should look for. Possible ways of locating short paths with a limited number of vertices include

- Divide the samples into d periods, and perform CCSP on each of these. Execution time is reduced by about d^2 .
- Start with the path involving vertices 1 and N only, and augment it gradually by one vertex until an M -vertex path is achieved.
- Start with an arbitrary path with M vertices, and change vertices on the path successively such that the path length is gradually reduced.

This paper will be followed up by research on heuristics of the above kind.

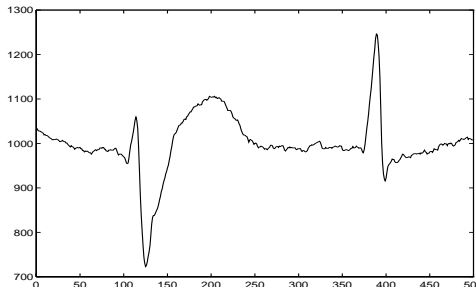


Figure 2: Recording represented by 500 samples from the mit200 record

5 Numerical experiments

To investigate the performance of the CCSP approach, and hence to evaluate the performance of existing time-domain ECG coders, two arbitrary ECG recordings (Figure 2 and Figure 3) from the MIT (MOODY, 1992)

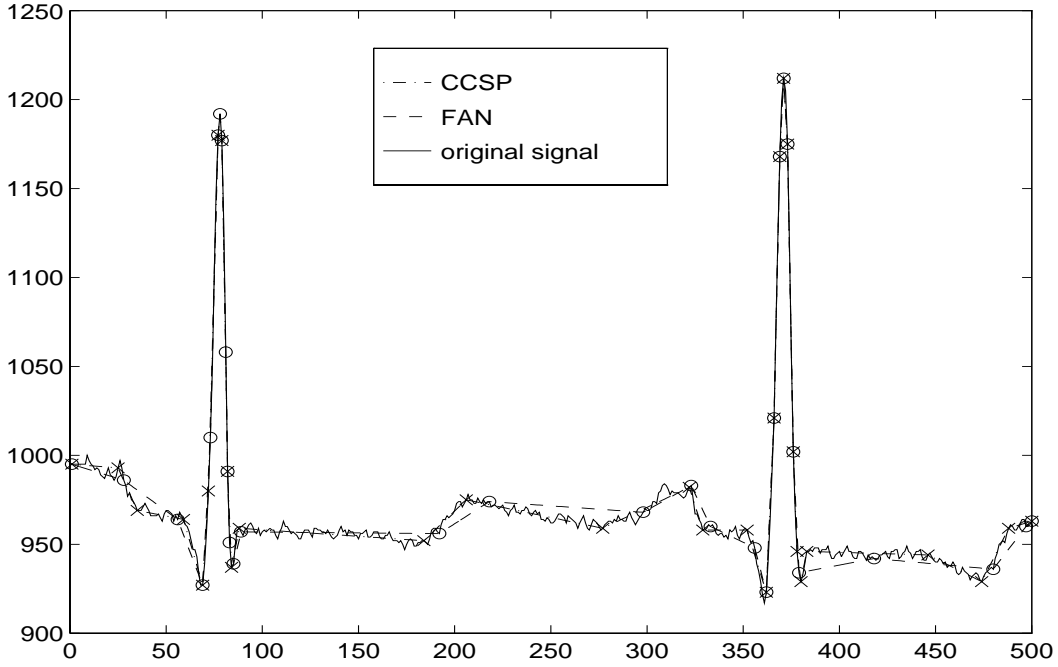


Figure 3: The reconstructed and the original signal of a 500 samples long recording from the mit100 record. The 30 samples selected by the FAN algorithm and the CCSP algorithm are indicated with an 'o' and a 'x', respectively.

databases were selected. In comparison with other time-domain coders, the FAN algorithm has been reported to give high compression ratio, in addition to producing a reconstructed signal with high fidelity (JALALEDDINE et al., 1990). Thus, an appropriate choice will be to represent the traditional time-domain techniques by the FAN method, and compare it with the CCSP algorithm in the following numerical experiments.

In practice the selected samples are usually entropy coded. However, in our case we shall restrict ourselves to present compression ratio in terms of the fraction between the number of samples in the original signal and the number

of retained samples, so that

Sample reduction rate =

$$\frac{\text{Number of samples in original signal}}{\text{Number of retained samples}}. \quad (3)$$

Assuming that the probability distributions of the coded sequences produced by each technique are approximately equal, the compression ratio in terms of bit reduction will be proportional to the sample reduction ratio.

An error estimate which appears frequently in the ECG coding literature is the Percent

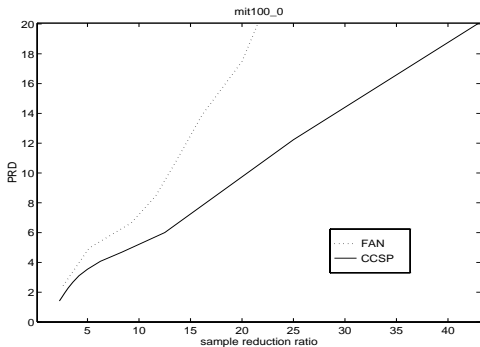


Figure 4: Sample reduction ratio versus PRD for record mit100

Root-mean-square Difference (PRD) defined as

$$PRD = \sqrt{\frac{\sum_{n=1}^N [y(n) - \hat{y}(n)]^2}{\sum_{n=1}^N [y(n) - \bar{y}]^2}} \times 100\%, \quad (4)$$

where \bar{y} is the mean value of the original signal. This estimate is proportional to the error criterion RMS used in the CCSP algorithm, and the results shown in Figure 4 and 5 clearly illustrate the impressive performance of this algorithm in terms of compression efficiency.

For a low sample reduction ratio, the difference in PRD is minor. However, as we increase the sample reduction ratio, the CCSP algorithm becomes superior. This trend has been manifested by numerous experiments with a larger selection of ECG records from the MIT database.

6 Conclusions

This paper demonstrates that a graph theoretic approach to the problem of compressing

ECG data can be very useful. We consider the digraph where each vertex represents a sample point in the original ECG signal, and each arc signifies inclusion of the samples corresponding to its connecting vertices as consecutive retained signal samples. Hence we seek a selection of arcs and incident vertices in the digraph, and this selection is to constitute a path from the first to the last sample point.

The algorithm proposed for the problem proves to converge in cubic time, and has been successfully applied to standard test problems. Compared to conventional time-domain heuristics, our approach contributes to a drastic reduction in the total error of the decoded signal, depending on the compression rate. From this we learn the lesson that frequently applied time-domain methods suffer from significant deviations from the theoretical optimum. Furthermore, the lower error bound can actually be reached by applying the suggested method.

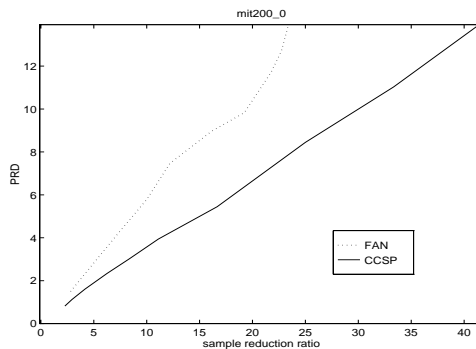


Figure 5: Sample reduction ratio versus PRD for record mit200

Numerical experiments reported in this paper show that exact solutions to data compression problems, where the signal length is 500 and the sample reduction rate is 10, can

be found in about 8.8 CPU-seconds on an HP/9000 work station. This is fairly close to the real time requirement of about 1 second. The need for faster methods is however present, and it is our belief that graph theoretic considerations may contribute to efficient heuristics as well.

Shorter execution time can be achieved by using a higher sample reduction rate, as the number of operations required by the CCSP algorithm are proportional to the number of samples retained. In Section 5 we observed that the difference in reconstruction error between CCSP and the FAN algorithm increased, as the sample reduction ratio increased. Thus, in applications where high compression is desirable, we can conclude that the CCSP algorithm at this stage can be a realistic alternative to the traditional time-domain coders.

References

- ABERNSTEIN, J.P. and THOMPSON, W.J. (1982) A new data reduction algorithm for real time ECG analysis. *IEEE Trans.*, **BME-29**, 43-48.
- ANEJA, Y.P., AGGARWAL, V. and NAIR, K.P.K. (1983) Shortest chain subject to side conditions. *Networks*, **13**, 295-302.
- ANEJA, Y.P. and NAIR, K.P.K. (1978) The constrained shortest path problem. *Nav. Res. Log. Q.*, **25**, 549-553.
- BARR, R.C. (1988) Adaptive sampling of cardiac waveforms. *J. Electrocard.*, **21**, 57-60.
- BEASLEY, J.E. and CHRISTOFIDES, N. (1989) An algorithm for the Resource Constrained Shortest Path Problem. *Networks*, **19**, 379-394.
- COX, J.R., NOILE, F.M., FOZZARD, H.A. and OLIVER, G.C. (1968) AZTEC: preprocessing program for real-time ECG rhythm analysis. *IEEE Trans.*, **BME-15**, 128-129.
- DIJKSTRA, E.W. (1959) A note on two problems in connection with graphs. *Numerische Math.*, **1**, 269-271.
- HANDLER, G.Y. and ZANG, I. (1980) A dual algorithm for the constrained shortest path problem. *Networks*, **10**, 293-310.
- JALALEDDINE, S.M.S., HUTCHENS, C.G., STRATTAN, R.D. and COBERLY, W.A. (1990) ECG data compression techniques - a unified approach. *IEEE Trans.*, **BME-37(4)**, 329-343.
- JOKSCH, H.C. (1966) The shortest route problem with constraints. *J. Math. Anal. Appl.*, **14**, 191-197.
- MOODY, G. (1992) MIT-BIH Arrhythmia Database CD-ROM (second edition), overview. *Massachusetts Institute of technology*.
- RIBEIRO, C. and MINOUX, M. (1985) A heuristic approach to hard constrained shortest path problems. *Discrete Appl. Math.*, **10**, 125-137.
- ROSSEEL, M. (1968) Comments on a paper by R.Saigal: A constrained shortest route problem. *Opns. Res.*, **16**, 1232-1234.
- SAIGAL, R. (1968) A constrained shortest route problem. *Opns. Res.*, **16**, 205-209.
- TAI, S.C. (1991) SLOPE - a real time ECG data compressor. *Med. & Biol. Eng. & Comput.*, **29**, 175-109.
- TAI, S.C. (1993) AZTDIS - a two-phase real-time ECG data compressor. *J. Biomed. Eng.*, **15**, 510-515.