Optim Eng (2006) 7: 81–96
DOI 10.1007/s11081-006-6592-3

Local search methods for ℓ_1-minimization in frame based signal compression

Dag Haugland · Sverre Storøy

Received: 13 February 2003 / Revised: 18 August 2004

Abstract We consider the problem of finding a linear combination of at most t out of K column vectors in a matrix, such that a target vector is approximated as closely as possible. The motivation of the model is to find a lower-dimensional representation of a given signal vector (target) while minimizing loss of accuracy. We point out the computational intractability of this problem, and suggest some local search heuristics for the unit norm case. The heuristics, all of which are based on pivoting schemes in a related linear program, are compared experimentally with respect to speed and accuracy.

Keywords Signal compression · Linear approximation · Sparsity constraint · Local search · Pivoting

1. Introduction

Data compression is an important operation both in storage of large amounts of data and in transmission of digital signals through channels with limited capacity. There are two distinct classes of compression methods: *lossless* compression and *lossy* compression. By lossless compression it is understood that the original data can be reconstructed exactly from the compressed version. Lossy compression, on the other hand, gives an approximation of the original data, and consequently has a better compression potential.

One way to deal with lossy compression of a signal vector (target), is through dimension reduction, i.e. find a target approximation in some lower dimensional space. The subspace in question is not necessarily given, and the accuracy of the approximation can largely depend on how the subspace is selected. Dimension reduction techniques based on *frames* fall into this category. A frame is a matrix with columns representing potential basis vectors of the subspace. Consequently, the *subspace selection problem* attains a bi-level nature where the first level consists in selecting columns from the frame, and the second level problem is to

D. Haugland (✉) · S. Storøy
Department of Informatics, University of Bergen, N-5020 Bergen, Norway
e-mail: dag@ii.uib.no
e-mail: sverre@ii.uib.no

find the best target approximation in the subspace spanned by this selection. The subspace selection problem has previously been referred to as the *subset* selection problem (Golub and van Loan, 1989; Couvreur and Bresler, 2000), but it is rephrased here in order to underline that the set that is searched for is a vector space.

Frames have recently been applied quite extensively to compute alternative representations of given signals. The purpose of this can be a sparse exact representation, and thereby a lossless compression. The goal can also be an exact representation minimizing a given norm, allowing for more dense binary representation of the signal. Techniques in these directions include the popular *matching pursuit* method (Mallat and Zhang, 1993; Gharavi-Alkhansari and Huang, 1998), the *method of frames* (Daubechies, 1998), *best orthogonal basis* (Coifman and Wickerhauser, 1992), and *basis pursuit* (Chen et al., 1998).

Dimension reduction is achieved when the number of selected frame columns is smaller than the target length, and the compression becomes lossy if the target does not fall within the spanned subspace. Frame based compression approaches of this kind include interrupted versions of some of the representation techniques above (see Engan (2000) for a general study). More specialized methods for finding sparse solutions to linear systems are given in Natarajan (1995), Couvreur and Bresler (2000), Reeves (1999) and Haugland (2001).

Most of the work cited above assumes a Euclidean distance measure. In many practical applications this is very reasonable, but when less emphasis is to be put on the larger components of the residual vector, the unit norm may be preferable. Some attempts in this direction have been suggested by Ryen (2001). Through discretization of the weights, his binary programming model does not solely do dimension reduction, but is also able to suggest a compression that can be coded by a specified number of bits. The computational burden of such a model is however considerable.

The main focus of the current paper is on how to solve the subspace selection problem with unit norm minimization. In the next section we give a hardness result that is general with respect to vector norm, before we turn the attention to search methods aiming for locally optimal solutions in the unit norm case. Section 3 provides the foundations for our local search method, and variants of the heuristic are given in Section 4. Their computational properties are discussed in Section 5, and finally in Section 6 we give some computational results.

2. Notation and preliminaries

In the following we will to a large extent use the concepts and notation defined in Engan (2000).

Let t, N and K be positive integers where $t \leq N$, let the *target* $x \in \Re^N$ and the *frame* $F \in \Re^{N \times K}$ be given, and suppose F satisfies the Haar-condition (all sets of no more than N columns from F are linearly independent). We let $\Gamma_1 = \{1, \ldots, K\}$ and $\Gamma_2 = \{1, \ldots, N\}$ denote the column and row index set of F, respectively. We let $e_i \in \Re^N$ denote the ith unit vector in $\Re^N$, and if v is a real vector, we let v_i denote its ith component. Define $\varphi : 2^{\Gamma_1} \times \Re^K \mapsto \Re$ such that, for any $\Omega_1 \subseteq \Gamma_1$ and $w \in \Re^K$, $\varphi(\Omega_1, w) = \|x - \sum_{j \in \Omega_1} w_j f_j\|_p$, where $1 \leq p \leq +\infty$ and $f_j = (f_{1j}, \ldots, f_{Nj})^T$ is the jth column of F. The subspace selection problem with minimum ℓ_p-norm can be formulated as

$$\varphi^* = \min_{\Omega_1 \subseteq \Gamma_1} \min_{w \in \Re^K} \{\varphi(\Omega_1, w) : |\Omega_1| \leq t\} \tag{1}$$

Proposition 1. *Problem* (1) *is NP-hard.*

Proof: Consider the problem
$\mathcal{P}_\varepsilon : \min_{\Omega_1 \subseteq \Gamma_1} \min_{w \in \Re^K} \{|\Omega_1| : \varphi(\Omega_1, w) \leq \varepsilon\}$. By a reduction from *exact cover by 3-sets*, as demonstrated for $p = 2$ by Natarajan (1995), it follows that $\mathcal{P}_\varepsilon$ is NP-hard. Since the decision versions of $\mathcal{P}_\varepsilon$ and (1) are identical, also the latter problem is NP-hard. □

From now and throughout the paper we assume that $p = 1$. Problem (1) can be written $\min_{\Omega_1 \subseteq \Gamma_1} \{\phi(\Omega_1) : |\Omega_1| \leq t\}$ where $\phi : 2^{\Gamma_1} \mapsto \Re$ is defined such that $\phi(\Omega_1) = \min_{w \in \Re^K} \{\varphi(\Omega_1, w)\}$. Hence the goal is to determine a set $\Omega_1 \subseteq \Gamma_1$ and the solution to the associated ℓ_1-approximation problem. When Ω_1 is given, this is nothing but the LP

$$\phi(\Omega_1) = \min \sum_{i=1}^{N} (r_i^+ + r_i^-) \tag{2}$$

$$\text{s.t. } Fw + r^+ - r^- = x \tag{3}$$

$$r^+, r^- \in \Re_+^N \tag{4}$$

$$w \in \Re^K \tag{5}$$

$$w_j = 0 \qquad j \in \Gamma_1 \setminus \Omega_1 \tag{6}$$

We have the following result:

Proposition 2. *If* $|\Omega_1| \leq N$, *there exist a minimizer* (w, r^+, r^-) *of* $\phi(\Omega_1)$ *and a set* $\Omega_2 \subseteq \Gamma_2$ *such that* $|\Omega_2| = N - |\Omega_1|$ *and* $r_i^+ = r_i^- = 0 \quad \forall i \in \Gamma_2 \setminus \Omega_2$.

Proof: Since F satisfies the Haar-condition, this follows directly from e.g. Theorem 6.2 in Watson (1980). □

Corollary 1. *There exist an* $\Omega_1^* \subseteq \Gamma_1$ *and a* $w^* \in \Re^K$, *where* $|\Omega_1^*| = t$ *and* $x_i = \sum_{j \in \Omega_1} f_{ij} w_j^*$ *for at least t distinct values of i, such that* $\varphi^* = \varphi(\Omega_1^*, w^*)$.

Proof: Follows immediately from Proposition 2 and the fact that $\phi(\Omega_1) \leq \phi(\Omega_1')$ if $\Omega_1' \subseteq \Omega_1$. □

Corollary 1 suggests that a search for an optimal solution to problem (1) be confined to sets $\Omega_1 \subseteq \Gamma_1$ and $\Omega_2 \subseteq \Gamma_2$, where $|\Omega_1| = t$ and $|\Omega_2| = N - t$, and where $w_j = 0 \quad \forall j \in \Gamma_1 \setminus \Omega_1$ and $r_i^+ = r_i^- = 0 \; \forall i \in \Gamma_2 \setminus \Omega_2$. This has also been fundamental in solution methods for the ℓ_1-approximation problem, including the simplex based method by Barrodale and Roberts (1973) and the descent method by Bartels et al. (1978).

Furthermore, Proposition 1 discourages a search for the optimal solution to problems of realistic size. We therefore propose a local search heuristic, which embeds the ideas used in the ℓ_1-approximation algorithm in the above papers.

3. Exchange neighborhoods

We shall consider pairs $\Omega = (\Omega_1, \Omega_2) \subseteq \Gamma_1 \times \Gamma_2$ for which $|\Omega_1| + |\Omega_2| = N$ and $\{f_j\}_{j \in \Omega_1} \cup \{e_i\}_{i \in \Omega_2}$ is linearly independent. Such pairs are defined as *independent* pairs.

If Ω is independent and $|\Omega_1| = t$, Ω is called a *basis*. In this case we also define the $N \times N$-matrix B_Ω where for all $i \in \Omega_2$ the ith column is e_i, and the other columns are f_j ($j \in \Omega_1$) ordered consistently with the column order in F. Any independent pair satisfies the conditions stated in Proposition 2. We also define the corresponding objective function value $\psi_\Omega = \sum_{i \in \Omega_2} |b_i|$, where $b = B_\Omega^{-1} x$. We say that Ω is *optimal* if $\psi_\Omega = \varphi^*$.

Let $\mathcal{I}$ denote the set of independent pairs, and let $\bar{\Omega}_i = \Gamma_i \setminus \Omega_i$, $i = 1, 2$.

In our local search method, the following neighborhoods will be considered:

Exchange between: $\mathcal{N}_{12}(\Omega) = \{\omega = (\omega_1, \omega_2) : \omega_1 = \Omega_1 \cup \{j\}, j \in \bar{\Omega}_1; \omega_2 = \Omega_2 \setminus \{i\}, i \in \Omega_2\} \cap \mathcal{I}$
Exchange within Ω_1: $\mathcal{N}_{11}(\Omega) = \{\omega = (\omega_1, \Omega_2) : |\Omega_1 \setminus \omega_1| = |\omega_1 \setminus \Omega_1| = 1\} \cap \mathcal{I}$
Exchange within Ω_2: $\mathcal{N}_{22}(\Omega) = \{\omega = (\Omega_1, \omega_2) : |\Omega_2 \setminus \omega_2| = |\omega_2 \setminus \Omega_2| = 1\} \cap \mathcal{I}$

We let $\mathcal{N}(\Omega) = \mathcal{N}_{11}(\Omega) \cup \mathcal{N}_{12}(\Omega) \cup \mathcal{N}_{22}(\Omega)$.

If $\psi_\Omega \leq \psi_\omega$ for all $\omega \in \mathcal{N}_{22}(\Omega)$, we say that Ω is *saturated*, and a sequence of improving $\mathcal{N}_{22}$-moves where Ω_1 is fixed, is referred to as a saturation process.

The possibility of an augmentation of Ω_2 compensated by an elimination from Ω_1 is not considered. This is because such moves cannot be superior to the best move into $\mathcal{N}_{22}(\Omega)$ (Corollary 3 below). Hence if Ω is independent and saturated, then (2)–(6) has a basic optimal solution where all residual variables but those indexed by Ω_2 are zero.

In each iteration of the local search procedure defined by Algorithm 1, we select one neighborhood in which we make a move. All moves are determined by first choosing the entering variable. Keeping this variable fixed, we next determine the leaving variable such that the largest possible reduction in the objective function value is obtained. Below we show how the entering variable is determined, and for each neighborhood we show how the corresponding leaving variable can be computed. The choice of what neighborhood to move in is a matter of search strategy discussed in Section 4.

Algorithm 1

```
$\Omega_1 \leftarrow \emptyset$, $\Omega_2 \leftarrow \Gamma_2$
while $\Omega$ is not $\mathcal{N}$-optimal do
    Choose $(E, L) \in \{(1, 1), (1, 2), (2, 2)\}$ such that $\Omega$ is not $\mathcal{N}_{EL}$-optimal
    Choose an entering variable $\epsilon \in \bar{\Omega}_E$
    Find $\ell^* \in \arg\min_{\ell \in \Omega_L} \{\psi_\omega : \omega \in \mathcal{N}_{EL}, \epsilon \in \omega_E, \ell \notin \omega_L\}$
    $\Omega_L \leftarrow \Omega_L \setminus \{\ell^*\}$
    $\Omega_E \leftarrow \Omega_E \cup \{\epsilon\}$
end while
```

3.1. Determining the entering variable

Consider the LP (2)–(5) written on standard form, i.e. with w replaced by $w^+ - w^-$ and a non-negativity constraint on each of the new variables. Any independent pair Ω defines a feasible basis to this LP, where for all $j \in \Omega_1$ either w_j^+ or w_j^- is basic, and for all $i \in \Omega_2$ either r_i^+ or r_i^- is basic. We let the choice of variable to enter Ω_1 or Ω_2 be guided by this LP-basis, and to this end we write $f_j = e_{j-K}$ ($j = K+1, \ldots, K+N$), and define $a^j = B_\Omega^{-1} f_j$ for all $j = 1, \ldots, K+N$. Now the *reduced cost* of w_j^+ ($j = 1, \ldots, K$) is defined as $c^j = \sum_{i \in \Omega_2} \sigma(b_i) a_i^j$, where $\sigma(y) = \begin{cases} 1, & y \geq 0 \\ -1, & y < 0 \end{cases}$. Similarly, for all $j \in \{K+1, \ldots, K+N\}$ we let $c^j = \sum_{i \in \Omega_2} \sigma(b_i) a_i^j - 1$ be the reduced cost of r_{j-K}^+. Reduced costs of w_j^- and r_{j-K}^- are implicitly available through the expressions $-c^j$ and $-2 - c^j$, respectively.

Following the usual criterion for selecting the entering variable in the primal simplex method, we let, in the case of an $\mathcal{N}_{12}$-move or an $\mathcal{N}_{11}$-move, some $j \in \bar{\Omega}_1$ maximizing $|c^j|$ define the entering variable. For $\mathcal{N}_{22}$-moves, we let some $j - K \in \bar{\Omega}_2$ maximizing $\max\{c^j, -2 - c^j\}$ enter Ω_2.

In either move, we study how a non-zero value δ assigned to the entering variable affects the objective function. For all $j = 1, \dots, K + N$, we let $\rho^j : \Re \mapsto \Re^N$ and $\psi^j : \Re \mapsto \Re$ be defined as $\rho^j(\delta) = B_\Omega^{-1}(x - \delta f_j)$ and $\psi^j(\delta) = \sum_{i \in \Omega_2} |\rho_i^j(\delta)|$, respectively. This means that if the variable to enter is assigned the value δ, while all other currently non-basic variables remain zero, the variable that currently is basic in row i of (2)–(5) is assigned the value $\rho_i^j(\delta)$. The corresponding objective function value is $\psi^j(\delta)$. Since some variable is to leave, we have to choose δ such that $\rho_i^j(\delta) = 0$ for some $i = 1, \dots, N$. Furthermore, Algorithm 1 says that this should be done in such a way that $\psi^j(\delta)$ is minimized.

When computing a best leaving variable, it turns out that only rows i where b_i / a_i^j agrees in sign with c^j need consideration, and therefore we define $\Omega_2^j = \{i \in \Omega_2 : \sigma(a_i^j)\sigma(b_i)\sigma(c^j) = 1\}$ $(j = 1, \dots, K + N)$.

3.2. Moves in $\mathcal{N}_{12}$

Since the leaving variable ℓ^* is to be chosen optimally in Ω_2 for $j \in \bar{\Omega}_1$ fixed, it must for some $\delta^* \in \Re$ satisfy

$$\left(\ell^*, \delta^*\right) \in \arg \min_{\ell \in \Omega_2, \delta \in \Re} \left\{\psi^j(\delta) : \rho_\ell^j(\delta) = 0\right\} \tag{7}$$

Fortunately, (ℓ^*, δ^*) can be determined merely by sorting rows.

Proposition 3. *Let $j \in \bar{\Omega}_1$, and let the rows of (F, x) be ordered such that $|b_1/a_1^j| \leq \cdots \leq |b_N/a_N^j|$. Then $\min_{\delta \in \Re} \psi^j(\delta) = \psi^j(\delta^*)$ and $\rho_{\ell^*}^j(\delta^*) = 0$, where*

$$\ell^* = \min_{\ell \in \Omega_2^j} \left\{ \ell : \sum_{i \in \Omega_2^j, i \leq \ell} \left|a_i^j\right| \geq |c^j|/2 \right\} \tag{8}$$

and

$$\delta^* = b_{\ell^*}/a_{\ell^*}^j \tag{9}$$

Proof: Let $\Omega_2(\delta) = \{i \in \Omega_2 : \sigma(a_i^j)\sigma(b_i)\sigma(\delta) = 1\}$ and $i^\delta = \max_{i \in \Omega_2(\delta)}\{i : |b_i| \leq |\delta a_i^j|\}$. Then we have

$$\begin{aligned}
\psi^j(\delta) &= \sum_{i \in \Omega_2} \left|b_i - \delta a_i^j\right| = \sum_{i \in \Omega_2} \left|\sigma(b_i)|b_i| - \sigma(\delta)\sigma\left(a_i^j\right)\left|\delta a_i^j\right|\right| \\
&= \sum_{i \in \Omega_2(\delta)} \left|\sigma(b_i)\left(|b_i| - \left|\delta a_i^j\right|\right)\right| + \sum_{i \in \Omega_2 \setminus \Omega_2(\delta)} \left|\sigma(b_i)\left(|b_i| + \left|\delta a_i^j\right|\right)\right| \\
&= -\sum_{i \in \Omega_2(\delta), i \leq i^\delta} \left(|b_i| - \left|\delta a_i^j\right|\right) + \sum_{i \in \Omega_2(\delta), i > i^\delta} \left(|b_i| - \left|\delta a_i^j\right|\right)
\end{aligned}$$

$$
\begin{aligned}
&+ \sum_{i\in\Omega_2\setminus\Omega_2(\delta)} \sigma(b_i)\big(b_i - \delta a_i^j\big)\\
&= -\sum_{i\in\Omega_2(\delta), i\leq i^\delta} \sigma(b_i)\big(b_i - \delta a_i^j\big) + \sum_{i\in\Omega_2(\delta), i> i^\delta} \sigma(b_i)\big(b_i - \delta a_i^j\big)\\
&\quad + \sum_{i\in\Omega_2\setminus\Omega_2(\delta)} \sigma(b_i)\big(b_i - \delta a_i^j\big)\\
&= \sum_{i\in\Omega_2} |b_i| - c^j\delta + 2\sum_{i\in\Omega_2(\delta), i\leq i^\delta} \sigma(b_i)\big(\delta a_i^j - b_i\big) \qquad (10)
\end{aligned}
$$

Note that ψ^j is a piecewise linear convex function for which at least one minimum exists. Since the last sum in (10) is non-negative, and zero if $\delta = 0$, we have that $\psi^j(\delta) \geq \psi^j(0)$ if $\sigma(\delta) = -\sigma(c^j)$, and hence ψ^j attains its minimum value in one of the break points b_ℓ/a_ℓ^j, where $\ell \in \Omega_2^j$. Since $s_\ell = -c^j + 2\sum_{i\in\Omega_2^j, i\leq\ell} \sigma(b_i)a_i^j = -c^j + 2\sigma(c^j)\sum_{i\in\Omega_2^j, i\leq\ell} \sigma(a_i^j)a_i^j = -c^j + 2\sigma(c^j)\sum_{i\in\Omega_2^j, i\leq\ell} |a_i^j|$ is a sub-gradient of $\psi^j(\delta)$ at $\delta = b_\ell/a_\ell^j$, and $-c^j$ is a sub-gradient at $\delta = 0$, the function is minimized for the smallest ℓ where s_ℓ and c^j agree in sign, i.e. at $\ell^* = \min\{\ell : \sum_{i\in\Omega_2^j, i\leq\ell} |a_i^j| \geq |c^j|/2\}$. Since $\rho_i^j(\delta) = b_i - \delta a_i^j$, we trivially have $\rho_{\ell^*}^j(\delta^*) = 0$, and the proof is complete. □

Proposition 3 states that δ^* minimizes ψ^j in the unconstrained case, but also satisfies the constraint in (7). It follows that a best solution in $\mathcal{N}_{12}(\Omega)$ provided that j enters Ω_1, is found by eliminating ℓ^* from Ω_2. In LP-terms, this move corresponds to q pivots $(w_j, r_{i_1}^{\pm})$, $(r_{i_1}^{\mp}, r_{i_2}^{\pm}), \ldots, (r_{i_{q-1}}^{\mp}, r_{i_q}^{\pm})$, where $i_q = \ell^*$. However, (8) enables us to spot the terminal point of this process without having to work out the pivots. This is equivalent to the idea of "bypassing intermediate data points" described in Barrodale and Roberts (1973).

3.2.1. Numerical example

Let

$$F = \begin{pmatrix} 1 & 2 & 3\\ 2 & 3 & 4\\ 5 & 2 & 1 \end{pmatrix}, x = \begin{pmatrix} 2\\ 3\\ 1 \end{pmatrix}, t = 2,$$

and consider the independent pair $\Omega = (\{1\}, \{1, 2\})$.

We have that $B_\Omega = \begin{pmatrix} 1 & 0 & 1\\ 0 & 1 & 2\\ 0 & 0 & 5 \end{pmatrix}$,

and

$$\begin{pmatrix} c^1 & \cdots & c^{K+N} & \psi_\Omega\\ a_1^1 & \cdots & a_1^{K+N} & b_1\\ \vdots & \ddots & \vdots & \vdots\\ a_N^1 & \cdots & a_N^{K+N} & b_N \end{pmatrix} = \frac{1}{5}\begin{pmatrix} 0 & 19 & 32 & 0 & 0 & -8 & 22\\ 0 & 8 & 14 & 5 & 0 & -1 & 9\\ 0 & 11 & 18 & 0 & 5 & -2 & 13\\ 5 & 2 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

We now choose $j = 3$ since $\max_{j \in \bar{\Omega}_1} |c^j| = c^3 = 32/5$. According to Proposition 3, $\ell^* = 2$ since $a_1^3 = 14/5 < c^3/2 = 16/5$, whereas $a_1^3 + a_2^3 = 32/5 \geq c^3/2$. If the ratio test of the primal simplex method were applied, r_1^+ rather than r_2^+ would have been chosen to leave. This is reflected by a negative value of b_1 when ℓ^* has replaced j. After an update of Ω we get

$$B_\Omega = \begin{pmatrix} 1 & 1 & 3 \\ 0 & 2 & 4 \\ 0 & 5 & 1 \end{pmatrix}$$

and

$$\begin{pmatrix} c^1 & \cdots & c^{K+N} & \psi_\Omega \\ a_1^1 & \cdots & a_1^{K+N} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_N^1 & \cdots & a_N^{K+N} & b_N \end{pmatrix} = \frac{1}{18} \begin{pmatrix} 0 & 2 & 0 & -36 & -4 & -20 & 4 \\ 0 & -2 & 0 & 18 & -14 & 2 & -4 \\ 18 & 5 & 0 & 0 & -1 & 4 & 1 \\ 0 & 11 & 18 & 0 & 5 & -2 & 13 \end{pmatrix} \tag{11}$$

3.3. Moves in $\mathcal{N}_{22}$

For $\mathcal{N}_{22}$-moves we have that the entering variable is identified by an integer j where $j - K \in \bar{\Omega}_2$. The leaving variable must be some ℓ^* such that (7) holds for some δ^*.

For notational convenience, let $b_0 = 0$ and $a_0^j = 1$.

Proposition 4. *Let $j - K \in \bar{\Omega}_2$ and let the rows of (F, x) be ordered such that $|b_1/a_1^j| \leq \cdots \leq |b_N/a_N^j|$. If $c^j \in [-2, 0]$, then $\min_{\delta \in \Re} \psi^j(\delta) = \psi^j(0)$. Otherwise, $\min_{\delta \in \Re} \psi^j(\delta) = \psi^j(\delta^*)$ and $\rho_{\ell^*}^j(\delta^*) = 0$, where*

$$\ell^* = \min_{\ell \in \Omega_2^j} \left\{ \ell : \sum_{i \in \Omega_2^j, i \leq \ell} |a_i| \geq (|c^j| + \sigma(c^j) - 1)/2 \right\} \tag{12}$$

and

$$\delta^* = b_{\ell^*}/a_{\ell^*}^j \tag{13}$$

Proof: By following the proof of Proposition 3, we get

$$\psi^j(\delta) = \psi^j(0) + \big(\sigma(\delta) - c^j - 1\big)\,\delta + 2 \sum_{i \in \Omega_2(\delta), i \leq i^\delta} \sigma(b_i)\big(\delta a_i^j - b_i\big) \tag{14}$$

where i^δ and $\Omega_2(\delta)$ are defined as in the proof of Proposition 3.

Define the extended reals $\delta_0 = 0, \delta_1 = b_1/a_1^j, \ldots, \delta_N = b_N/a_N^j$. Hence the smallest $\delta \geq 0$ (if any) for which $\psi^j(\delta)$ has a non-negative sub-gradient is δ_{ℓ^+}, where

$$\ell^+ = \min_{\ell \in \{0\} \cup \Omega_2(1)} \left\{ \ell : -c^j + 2 \sum_{i \in \Omega_2(\delta_\ell), i \leq \ell} \left| a_i^j \right| \geq 0 \right\},$$

and the largest $\delta \leq 0$ (if any) for which $\psi^j(\delta)$ has a non-positive sub-gradient is δ_{ℓ^-}, where

$$\ell^- = \min_{\ell \in \{0\} \cup \Omega_2(-1)} \left\{ \ell : -c^j - 2 - 2 \sum_{i \in \Omega_2(\delta_\ell), i \leq \ell} \left| a_i^j \right| \leq 0 \right\}$$

Note that $\psi^j(\delta) \geq 0 \; \forall \delta \in \Re$ implies that both ℓ^+ and ℓ^- exist. If $c^j \in [-2, 0]$, it follows that $\ell^+ = \ell^- = 0$, and 0 is a sub-gradient of $\psi^j(\delta)$ at $\delta = 0$. If $c^j > 0$, we have that 0 is a sub-gradient of $\psi^j(\delta)$ at $\delta = \delta_{\ell^+}$, and that $\rho_{\ell^+}^j(\delta_{\ell^+}) = 0$. If $c^j < -2$, we have that 0 is a sub-gradient of $\psi^j(\delta)$ at $\delta = \delta_{\ell^-}$, and that $\rho_{\ell^-}^j(\delta_{\ell^-}) = 0$. The proof is complete by observing that (12) implies $\ell^* = \begin{cases} \ell^+, & c^j > 0 \\ \ell^-, & c^j < -2 \end{cases}$ □

Corollary 2. *If $c^j \in [-2, 0]$ for all $j - K \in \bar{\Omega}_2$ then Ω is saturated.*

Corollary 3. *If $\Omega \in \mathcal{I}$, then*

$$\min_{\omega \in \mathcal{I}} \{ \psi_\omega : |\omega_2 \setminus \Omega_2| = 1, |\Omega_1 \setminus \omega_1| + |\Omega_2 \setminus \omega_2| = 1 \} = \psi_{\hat{\omega}}$$

for some $\hat{\omega} \in \mathcal{I}$ where $|\Omega_2 \setminus \hat{\omega}_2| = |\hat{\omega}_2 \setminus \Omega_2| = 1$.

Corollary 3 states that among all 1-exchange moves where a residual enters, there is a best move where also the leaving variable is a residual. Hence we do not consider moves where a frame vector is replaced by a residual variable.

3.3.1. Numerical example (continued from Section 3.2.1)

From (11) we see that $c^4, c^5, c^6 \in [-2, 0]$, and it follows that Ω is saturated.

3.4. Moves in $\mathcal{N}_{11}$

When an entering j, where $j - K \in \bar{\Omega}_2$, is given, and the best variable to leave Ω_1 is to be found, we have to solve

$$\min_{\ell \in \bar{\Omega}_2, \delta \in \Re} \left\{ \sum_{i \in \Omega_2} \psi^j(\delta) : \rho_\ell^j(\delta) = 0 \right\} \tag{15}$$

Proposition 3 states that $\sum_{i \in \Omega_2} \left| \psi^j(\delta) \right|$ is minimized at δ^*, where δ^* is as given by (8)–(9). By convexity of ψ^j, (15) can be solved by first computing ℓ^* through (8). Next we locate the closest break point on each side of δ^* for which the constraint in (15) holds, if such a point exists.

Consequently we define $b_0 = 0$, $b_{N+1} = +\infty$, $a_0^j = a_{N+1}^j = 1$, and let $\bar{\Omega}_2^j = \{i \in \bar{\Omega}_2 : \sigma(a_i^j)\sigma(b_i)\sigma(c^j) = 1\}$, $\ell_1 = \max\{\ell \in \bar{\Omega}_2^j \cup \{0\} : \ell < \ell^*\}$ and $\ell_2 = \min\{\ell \in \bar{\Omega}_2^j \cup \{N+1\} : \ell > \ell^*\}$. Depending on the solution to $\min_{\delta \in \{0,\delta_1,\delta_2\}} \psi^j(\delta)$, where $\delta_1 = b_{\ell_1}/a_{\ell_1}^j$ and $\delta_2 = b_{\ell_2}/a_{\ell_2}^j$, the conclusion is either that no $\mathcal{N}_{11}$-move is to be carried out, or that the frame vector indexed by ℓ_1 or ℓ_2 is to leave Ω_1.

3.4.1. *Numerical example (continued from Section 3.2.1)*

From (11) we choose $j = 2$, and after sorting rows we get $a^j = (5/18, 11/18, -1/9)^T$ and $b = (1/18, 13/18, -2/9)^T$. Hence $\ell^* = 3$, $\delta^* = 2$, $\delta_1 = 13/11$ and $\delta_2 = +\infty$, and since also $\psi^j(\delta_1) = 1/11 < \psi^j(0) = 2/9$, the move results in the solution $\Omega = (\{1, 2\}, \{1\})$. It is easily checked that this solution is $\mathcal{N}$-optimal, and in fact also globally optimal.

4. Local search strategies

Since the $\mathcal{N}_{11}$-move is computationally more expensive than the others (see the next section), it will be chosen only when none of the others can provide any improvement. We consider two extreme ways of combining $\mathcal{N}_{12}$- and $\mathcal{N}_{22}$-moves.

In the *saturate first* strategy, any move in the $\mathcal{N}_{22}$-neighborhood yielding a reduction in the objective function value, is preferred to all moves in $\mathcal{N}_{12}(\Omega)$. When Ω is saturated without being a basis, a move in $\mathcal{N}_{12}(\Omega)$ is made.

In the *basis first* strategy, the preference is on $\mathcal{N}_{12}(\Omega)$. This strategy implies that a basis is reached by performing such a move in each of the first t iterations, and next the basis is saturated through $\mathcal{N}_{22}$-moves.

When a saturated basis is achieved by either strategy, we try to improve it by a single $\mathcal{N}_{11}$-move followed by a saturation process. This is repeated until Ω is saturated and $\mathcal{N}_{11}$-optimal.

It is reasonable to believe that *saturate first* yields better solutions than does *basis first*, whereas the latter will require fewer iterations. If we saturate Ω before extending Ω_1, the selection of the new member can be based on the solution to the ℓ_1-approximation problem corresponding to Ω_1. Such a careful choice of entering variable does however cost the price of solving an updated approximation problem between all extensions of Ω_1.

For the $\mathcal{N}_{12}$- and $\mathcal{N}_{22}$-moves, the entering variable is determined by $\max_{j \in \bar{\Omega}_1} |c^j|$ and $\max_{i \in \bar{\Omega}_2} \max\{c^{K+i}, -2 - c^{K+i}\}$, respectively, as explained in Section 3.1. For the $\mathcal{N}_{11}$-move, however, the $j \in \bar{\Omega}_1$ with maximum reduced cost does not necessarily have a corresponding $j' \in \Omega_1$ such that swapping j and j' yields an improvement. Finding an $\mathcal{N}_{11}$-move hence implies that possibly all $j \in \bar{\Omega}_1$ be checked in an order of non-increasing $|c^j|$, until all are checked and $\mathcal{N}_{11}$-optimality is concluded, or until some $(j, j') \in \bar{\Omega}_1 \times \Omega_1$ defining an improving $\mathcal{N}_{11}$-move is found.

In addition to the search strategies mentioned above, we also define an *interlaced* strategy, where we solely use the reduced costs to choose between $\mathcal{N}_{12}(\Omega)$ and $\mathcal{N}_{22}(\Omega)$. As long as Ω is not a basis, we make a move in $\mathcal{N}_{12}(\Omega)$ if $\max_{j \in \bar{\Omega}_1} |c^j| > \max_{i \in \bar{\Omega}_2} \max\{c^{K+i}, -2 - c^{K+i}\}$, whereas we make an $\mathcal{N}_{22}$-move otherwise. When Ω already is a basis, the strategy is identical to *saturate first* and *basis first*.

It is also possible to design local search procedures where Ω is initialized to some feasible independent pair different from $(\emptyset, \Gamma_2)$. There exist fast heuristics aimed to compute good vector selections in the case where $p = 2$, e.g. 'fully orthogonal matching pursuit' (FOMP) (Gharavi-Alkhansari and Huang, 1998), and Ω_1 can for instance be initialized by applying such a method. While keeping Ω_1 fixed, we next solve (2)–(6), and initialize Ω_2 to the set of indices of the $N - t$ basic r^+- and r^--variables. If the initial solution satisfies $|\Omega_1| = t$, which will be true if Ω_1 is generated by FOMP, only $\mathcal{N}_{11}$- and $\mathcal{N}_{22}$-moves will be carried out in the subsequent local search. Hence saturate first and basis first coincide for such initializations.

5. Algorithm analysis and implementation

The local search procedure is implemented by use of techniques from the revised simplex method applied to (2)–(5). That is, we store B_Ω^{-1} explicitly in memory, and update it through pre-multiplication by an elementary matrix. The same operation is applied to the right-hand side b. We compute 'dual variables' $u_1, \ldots, u_N$ by applying the formula $u_i = \sum_{k \in \Omega_2} \sigma(b_k) \left(B_\Omega^{-1}\right)_{ki}$ $(i = 1, \ldots, N)$, such that the reduced costs can be computed through $c^j = \sum_{i=1}^N u_i f_{ij}$ $(j \in \bar{\Omega}_1)$ and $c^{K+i} = u_i - 1$ $(i \in \bar{\Omega}_2)$ whenever needed. Note that due to the presence of $\sigma(b_i)$ in the definition of c^j (Section 3.1), the reduced costs cannot be computed by applying the elementary row operations used to update B_Ω^{-1} and b. This is an immediate consequence of bypassing vertices in the feasible region of (2)–(5).

Moves in any of the neighborhoods will require $\mathcal{O}(N^2)$ operations in order to update B_Ω^{-1}, u and b. Hence the computational burden of sorting at most N elements, required in the minimization of ψ^j, is no more significant than this update. In the worst case, computation of all c^j $(j \in \bar{\Omega}_1)$ requires $\mathcal{O}(KN)$ operations, and is only needed to open an $\mathcal{N}_{11}$- or an $\mathcal{N}_{12}$-move. Furthermore, moves into $\mathcal{N}_{12}(\Omega)$ require that a single column in the simplex tableau (a^j) be calculated $(\mathcal{O}(N^2))$, whereas in the worst $\mathcal{N}_{11}$-case all a^j $(j \in \bar{\Omega}_1)$ may have to be calculated. In summary, the work load for $\mathcal{N}_{11}$-, $\mathcal{N}_{12}$- and $\mathcal{N}_{22}$-moves are hence estimated to $\mathcal{O}(KN^2)$, $\mathcal{O}(KN + N^2)$ and $\mathcal{O}(N^2)$, respectively.

Bounds on the number of pivots carried out in the heuristics, apart from very weak ones based on enumeration of all basic solutions, are not available. Hence we cannot claim polynomial running time of our heuristics, but the experiments reported in the next section show as expected that they are much faster than exact methods.

6. Computational results

The three local search strategies proposed in Section 4 (references *saturate*, *basis* and *interlaced* in the tables below) are tested and compared with FOMP (Gharavi-Alkhansari and Huang, 1998). Since this technique produces a column index set Ω_1 and corresponding weights w_j $(j \in \Omega_1)$ intended to minimize the Euclidean norm, we apply it solely to find Ω_1, whereas the weights are recomputed by solving (2)–(6). The resulting method (reference *fomp1* below), is also used in a local search procedure (reference *fomp1ls* below) where *fomp1* produces the initial solution and $\mathcal{N}_{11}$- and $\mathcal{N}_{22}$-moves are used to improve it (see Section 4, last paragraph).

Note that the second phase of *fomp1*, solving (2)–(6), can be accomplished by applying any of the local search strategies of Section 4 to the sub-frame of F consisting exclusively of columns indexed by Ω_1. Since $|\Omega_1| = t$, the local search strategy will eventually select all these columns, and also the optimal set of non-zero residual variables. We have implemented the second phase of *fomp1* by applying *saturate first* to the sub-frame, while the first phase is a straightforward implementation of the algorithm in Gharavi-Alkhansari and Huang (1998).

The first data set contains three frames, here denoted $F_A, F_B, F_C \in \Re^{32\times 64}$, produced by frame design techniques described in Engan (2000). We consider 1000 different signal segments x, each of which is compressed by use of each frame. These segments are the leading samples of signal MIT100 in the MIT arrhythmia database (Moody, 1992), which contains several recorded electrocardiograms. All data are normalized such that $\|f_1\|_2 = \cdots = \|f_K\|_2 = \|x\|_2 = 1$. Below we report the average number of $\mathcal{N}_{11}$- and $\mathcal{N}_{22}$-moves (the number of $\mathcal{N}_{12}$-moves is always t), along with average CPU-time (seconds) and average objective function value. Note that*fomp1* does not do any $\mathcal{N}_{11}$-moves. We compare the search strategies with respect to these key figures, and repeat the experiment for three different values of t (1, 6 and 10).

Table 1 contains the results from these experiments. We see that the *saturate first* strategy is superior to *basis first* and *interlaced* when $t > 1$. This is what we would expect when

Table 1 Results from experiments with $N = 32$ and $K = 64$

t	Frame	Method	$\mathcal{N}_{11}$	$\mathcal{N}_{22}$	CPU	Objective
1	*A*	saturate	0.952	0.880	0.006	1.13629
1	*A*	basis	0.952	0.880	0.006	1.13629
1	*A*	interlaced	0.952	0.880	0.006	1.13629
1	*A*	fomp1	–	0.000	0.001	1.07996
1	*A*	fomp1ls	0.027	0.023	0.004	1.07410
1	*B*	saturate	0.712	0.651	0.006	1.35560
1	*B*	basis	0.712	0.651	0.006	1.35560
1	*B*	interlaced	0.712	0.651	0.006	1.35560
1	*B*	fomp1	–	0.000	0.001	1.32014
1	*B*	fomp1ls	0.042	0.036	0.004	1.31183
1	*C*	saturate	0.844	0.769	0.006	1.36288
1	*C*	basis	0.844	0.769	0.006	1.36288
1	*C*	interlaced	0.844	0.769	0.006	1.36288
1	*C*	fomp1	–	0.000	0.001	1.32942
1	*C*	fomp1ls	0.054	0.046	0.004	1.32115
6	*A*	saturate	2.549	22.865	0.010	0.418491
6	*A*	basis	3.061	16.630	0.011	0.420865
6	*A*	interlaced	2.938	17.510	0.010	0.420808
6	*A*	fomp1	–	14.835	0.003	0.406671
6	*A*	fomp1ls	0.893	17.372	0.008	0.389549
6	*B*	saturate	2.039	22.539	0.008	0.435232
6	*B*	basis	2.594	15.902	0.009	0.443009
6	*B*	interlaced	2.359	16.725	0.009	0.439292
6	*B*	fomp1	–	16.605	0.003	0.379707
6	*B*	fomp1ls	0.398	17.889	0.007	0.370183
6	*C*	saturate	2.535	24.995	0.009	0.486144
6	*C*	basis	2.947	17.900	0.010	0.491230

(*continued on next page*)

Table 1 (*Continued*)

t	Frame	Method	$\mathcal{N}_{11}$	$\mathcal{N}_{22}$	CPU	Objective
6	C	interlaced	2.801	18.665	0.010	0.490771
6	C	fomp1	–	16.726	0.002	0.429154
6	C	fomp1ls	0.559	18.702	0.006	0.407252
10	A	saturate	4.300	45.758	0.014	0.294181
10	A	basis	4.785	25.327	0.014	0.297015
10	A	interlaced	4.602	28.273	0.013	0.295201
10	A	fomp1	–	32.529	0.004	0.299896
10	A	fomp1ls	2.177	39.173	0.011	0.278892
10	B	saturate	2.957	42.191	0.011	0.304942
10	B	basis	3.601	21.779	0.011	0.304569
10	B	interlaced	3.422	23.955	0.011	0.305657
10	B	fomp1	–	33.873	0.004	0.292208
10	B	fomp1ls	1.719	38.636	0.010	0.279481
10	C	saturate	2.726	46.275	0.010	0.310379
10	C	basis	3.362	23.998	0.010	0.317121
10	C	interlaced	3.323	27.599	0.011	0.314044
10	C	fomp1	–	37.429	0.004	0.275908
10	C	fomp1ls	0.985	40.570	0.009	0.266383

comparing the result produced by the heuristics, whereas the observation is rather unexpected when it comes to computation time. It appears that the relatively small number of $\mathcal{N}_{11}$-moves required by *saturate first*, more than compensates for the larger number of less expensive $\mathcal{N}_{22}$-moves.

However, all these strategies seem to be inferior to *fomp1*, which not only is faster, but also produces better solutions. The only exception to this is F_A in combination with $t = 10$, in which case *fomp1* performs rather poorly. Although the column selection procedure in *fomp1* aims to minimize the Euclidean rather than the unit norm, it seems to perform better than the suggested strategies. This can be explained by the fact that when $p = 2$, it is easy to find an optimal 1-extension of the column set. The method in Gharavi-Alkhansari and Huang (1998) actually is a forward greedy heuristic consisting of t such extensions (starting with $\Omega_1 = \emptyset$). For $p = 1$, our search strategies are capable of finding the best solution in $\mathcal{N}_{12}(\Omega)$, but note that such a move is not necessarily an optimal 1-extension of Ω_1. This would require a more comprehensive neighborhood, like $\bar{\mathcal{N}}_{12}(\Omega) = \{\omega = (\omega_1, \omega_2) : \omega_1 = \Omega_1 \cup \{j\}, j \in \bar{\Omega}_1; |\omega_2| = |\Omega_2| - 1\} \cap \mathcal{I}$, to be explored. Hence the results in Table 1 indicate that optimal 1-extensions of the column selection, with respect to some vector norm, seems to be a better design criterion for heuristics than fidelity to the norm for which the problem is stated.

However, in none of the cases reported in Table 1 is the *fomp1*-solution $\mathcal{N}_{11}$-optimal for all signal segments tested. This is seen from the fact that by performing at least one $\mathcal{N}_{11}$-move (followed by a saturation process), *fomp1ls* is able to improve the mean objective function value in each of the cases. Furthermore, the total running time of *fomp1ls* never exceeds the running time of any of the methods that are not based on *fomp1*.

In Table 2 we give the results obtained by concatenating all the frames studied above. This should consistently give better approximations than what is obtained in the previous

Table 2 Results from experiments with $N = 32$ and $K = 192$

t	Frame	Method	$\mathcal{N}_{11}$	$\mathcal{N}_{22}$	CPU	Objective
1	ABC	saturate	1.315	1.205	0.018	1.07326
1	ABC	basis	1.315	1.205	0.018	1.07326
1	ABC	interlaced	1.315	1.205	0.018	1.07326
1	ABC	fomp1	–	0.000	0.001	1.03852
1	ABC	fomp1ls	0.034	0.028	0.011	1.03372
6	ABC	saturate	2.972	26.664	0.026	0.385874
6	ABC	basis	3.637	19.708	0.031	0.393540
6	ABC	interlaced	3.373	20.305	0.031	0.390517
6	ABC	fomp1	–	16.842	0.002	0.347935
6	ABC	fomp1ls	0.720	19.007	0.018	0.335140
10	ABC	saturate	4.079	47.850	0.034	0.276016
10	ABC	basis	4.895	26.758	0.039	0.279633
10	ABC	interlaced	4.641	28.960	0.039	0.276229
10	ABC	fomp1	–	35.974	0.005	0.261114
10	ABC	fomp1ls	1.839	41.387	0.026	0.248024

example, since now there are 192 rather than 64 columns to choose from. The conclusions in favor of *fomp1ls* are confirmed, since it gives a better solution than all other methods in all cases. It is also faster than *saturate first*, *basis first* and *interlaced* in all cases. However, it spends most of the time improving the initial solution provided by *fomp1*, which in less than one fifth of the time is able to produce solutions that also are superior to those provided by the above search strategies.

The conclusions above are further strengthened by the results in Table 3 from experiments with larger frames and longer target vectors ($K = 5000$, $N = 500$, $t = 50$ represents the largest case). Here all data (frame and signal segments) are drawn randomly from the uniform distribution on $[-1, 1]$, before $\{f_j\}$ and x are normalized. The frames are run on 10 signal segments of consistent length, and average number of moves, CPU-time and objective function value are reported in the table. In all of the cases where $t = 50$, *fomp1* produces an $\mathcal{N}_{11}$-optimal solution, and *fomp1ls* is hence unable to improve it. In one of the cases ($K = 5000$, $N = 500$, $t = 10$), *fomp1ls* does not produce a better solution than does *saturate first*, and is only marginally faster.

Small instances of (1) can be solved to optimality e.g. through a straightforward mixed integer programming model. In order to measure how far from the optimal solution the results from local search are, we implemented such a model using ILOG CPLEX 8.110. We applied it to some of the signal segments used in the above experiments, in combination with frame F_A for $t = 1, 2, 3, 4, 5$. The results from this experiment are summarized in Table 4. Since *fomp1ls* seems to be the best performing local search method, it is the only one considered here.

It should be noted that not even when $t = 1$ does the use of $\mathcal{N}_{11}$-moves guarantee that the optimal solution is found. As stated above, the $\mathcal{N}_{12}$-move is not an optimal extension of Ω_1, and in the same way it can be argued that $\mathcal{N}_{11}(\Omega)$ is not comprehensive enough to guarantee optimal 1-exchanges.

Table 3 Results from large scale experiments

K	N	t	Method	$\mathcal{N}_{11}$	$\mathcal{N}_{22}$	CPU	Objective
1000	100	10	saturate	0.7	94.9	0.6	2.56651
1000	100	10	basis	1.7	32.7	0.8	2.87598
1000	100	10	interlaced	1.1	49.9	0.8	2.75643
1000	100	10	fomp1	–	79.6	0.1	2.46475
1000	100	10	fomp1ls	0.2	81.5	0.6	2.45335
1000	100	50	saturate	0.4	1150.1	1.1	0.437618
1000	100	50	basis	6.4	222.8	1.9	0.781820
1000	100	50	interlaced	2.5	334.8	2.2	0.657388
1000	100	50	fomp1	–	1185.4	0.6	0.116697
1000	100	50	fomp1ls	0.0	1185.4	1.2	0.116697
1000	500	10	saturate	0.3	133.9	13.5	8.57332
1000	500	10	basis	1.8	58.9	18.1	8.78949
1000	500	10	interlaced	0.6	88.6	14.5	8.55117
1000	500	10	fomp1	–	114.7	2.2	8.56526
1000	500	10	fomp1ls	0.4	123.0	14.3	8.51990
1000	500	50	saturate	0.2	2361.0	49.5	6.21900
1000	500	50	basis	6.7	627.3	37.0	7.01508
1000	500	50	interlaced	0.6	1270.4	50.2	6.38987
1000	500	50	fomp1	–	2075.8	33.3	6.19014
1000	500	50	fomp1ls	0.0	2075.8	45.3	6.19014
5000	500	10	saturate	0.8	148.5	67.3	8.16008
5000	500	10	basis	2.2	71.0	83.0	8.53995
5000	500	10	interlaced	0.9	99.5	68.8	8.20483
5000	500	10	fomp1	–	123.9	2.9	8.19809
5000	500	10	fomp1ls	0.1	125.6	64.5	8.19135
5000	500	50	saturate	0.3	2588.1	106.1	5.31017
5000	500	50	basis	4.3	480.2	137.8	6.54263
5000	500	50	interlaced	1.0	1258.3	205.8	5.49467
5000	500	50	fomp1	–	2409.6	40.1	5.01002
5000	500	50	fomp1ls	0	2409.6	98.9	5.01002

Table 4 Local search versus exact method

	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
Cases run	1000	10	10	4	1
Cases solved to optimality	884	2	2	0	0
Objective (fomp1ls)	1.074	0.5084	0.3762	0.4376	0.385375
Optimal objective function value	1.066	0.4592	0.3209	0.3387	0.291769
CPU (fomp1ls)	0.0039	0.0044	0.0050	0.0059	0.0062
CPU (CPLEX)	1.525	22.8	310.4	2844	28022

We did not repeat the comparison with the exact method for values of t beyond 5. It turned out that, given a time limit of 8 CPU-hours, CPLEX did not converge for this big values of t. One reason for such a bad performance lies naturally in the lower bounding phase of Branch and Bound. The MILP-formulation has one binary variable for each frame vector, and as long as no more than $K - N$ of these are fixed to zero, a better bound than 0 is hardly achievable.

All experiments have been run on an Intel Pentium 4 computer with a 2.53 GHz CPU. All data can be downloaded at `http://www.ii.uib.no/~dag/opteng04/data`.

7. Concluding remarks

In the paper we have studied a signal compression model based on frames, and the subspace selection problem with unit norm minimization has been suggested to be the working model. The problem under study is $\mathcal{NP}$-hard, and a local search heuristic based on pivoting in a related linear program is developed, analyzed and implemented. Particular attention has been paid to the problem of exploiting the structure of the ℓ_1-approximation problem in order to define moves in non-trivial neighborhoods.

Computational experiments with three different variants of a local search method indicate that a large number of relatively cheap moves are preferable to a smaller number of more expensive moves. This conclusion is valid when comparing either speed or accuracy of the methods. The moves in question are actually LP-pivots, and the cheapest moves are those where both variables are residuals. Best results seem to be obtained when such pivots are given priority over all other pivots.

Applying the proposed local search methods to an initial zero solution (i.e. the set of selected frame columns is empty), does not seem to perform as well as starting from a more advanced initial solution. Forward greedy heuristics designed for the Euclidean version of the subspace selection problem are fast. By letting such a method produce an initial solution, and from there apply one of our local search strategies, all experiments show that we get better solutions with no more computational effort. In most of our experiments, this initial solution is actually better than the final output of the local search when starting from the zero solution.

Acknowledgment The authors want to thank Associate Professor Kjersti Engan at Dept. of Electrical and Computer Engineering, Stavanger University, for kindly providing parts of the test data.

References

Barrodale I, Roberts FDK (1973) An improved algorithm for discrete ℓ_1 linear approximation. SIAM Journal on Numerical Analysis 10(5):839–848

Bartels RH, Conn AR, Sinclair JW (1978) Minimization techniques for piecewise differentiable functions: the ℓ_1 solution to an overdetermined linear system. SIAM Journal on Numerical Analysis 15(2):224–241

Chen SS, Donoho DL, Saunders MA (1998) Atomic decomposition by Basis Pursuit. SIAM Journal on Scientific Computing 20(1):33–61

Coifman RR, Wickerhauser MV (1992) Entropy-based algorithms for best-basis selection. IEEE Transactions on Information Theory 38:713–718

Couvreur C, Bresler Y (2000) On the optimality of the backward greedy algorithm for the subset selection problem. SIAM Journal on Matrix Analysis and Applications 21(3):797–808

Daubechies I (1988) Time-frequency localization operators: a geometric phase space approach. IEEE Transactions on Information Theory 34:605–612

Engan K (2000) Frame based signal representation and compression. Dr.Ing. Thesis, Stavanger University College

Gharavi-Alkhansari M, Huang TS (May 1998) A fast orthogonal matching pursuit algorithm. In Proc. ICASSP '98, Seattle, Washington, USA, pp 1389–1392

Golub GH, van Loan CF (1989) Matrix computations, 2nd. ed., The Johns Hopkins University Press, Baltimore, MD

Haugland D (2001) An edge-traversal algorithm for the subspace selection problem. Paper presented at The 7th Meeting of the Nordic MPS Section, Copenhagen, Denmark

Mallat S, Zhang Z (1993) Matching pursuit in a time-frequency dictionary. IEEE Transactions on Signal Processing 41:3397–3415

Moody G (1992) MIT-BIH arrhythmia database cd-rom (second edition) overview. Massachusetts Institute of Technology

Natarajan BK (1995) Sparse approximate solutions to linear systems. SIAM Journal on Computing 24(2):227–234

Reeves SJ (1999) An efficient implementation of the backward greedy algorithm for sparse signal reconstruction. IEEE Signal Processing Letters 6:266–268

Ryen T (June 19–21, 2001) Simultaneous selection and quantization in frame based compression. Presented at 2nd IEEE R8-EURASIP Symposium on Image and Signal Processing and Analysis (ISPA'01), Pula, Croatia

Watson GA (1980) Approximation theory and numerical methods. Wiley, Binghamton, New York