Implementation through the Mathematica package of butterfly algorithms for determining the parameters of a linear code

Paskal Piperkov

8 dec 2020

Calculation of characteristic spectrum

Worked 2016-2020

Pubished 29 Oct 2020

Iliya Bouyukiev, Stefka Bouyukieva, Tatsuya Maruta, Paskal Piperkov. Characteristic vector and weight distribution of a linear code. – In: *Cryptography and Communications*. https://doi.org/10.1007/s12095-020-00458-8

```
q = 3; k = 3; tita = (q^k - 1)/(q - 1);
x = {0, 4, 3, 2, 0, 8, 5, 1, 1, 4, 3, 2, 3};
CH = Table[If[v == 1, x[[u]], 0], {u, 1, tita}, {v, 0, q - 1}];
step = 1;
Do[u = 0; newstep = q*step + 1; a = Table[0, {k}];
  While[u < tita, su = u; u = u + newstep; h = Table[0, {q + 1}, {q}];
    Do[h[[q + 1, v]] = CH[[u, v]]; CH[[u, v]] = 0, {v, q}];
    Do[su = su + 1; ssu = su;
      Do[Do[h[[t, v]] = CH[[ssu, v]]; CH[[ssu, v]] = 0, {v, q}];
        ssu = ssu + step, {t, q}];
      h[[2, 1]] = h[[2, 1]] + h[[q + 1, 2]]; ssu = su;
      Do[
        Do[
          Do[CH[[ssu, Mod[(col - 1) + (r - 1)*t, q] + 1]] =
            CH[[ssu, Mod[(col - 1) + (r - 1)*t, q] + 1]] +
            h[[r, col]], {col, q}],
          {r, q}];
        ssu = ssu + step, {t, 0, q - 1}];
      If[cor == 1,
        Do[Do[CH[[u, r]] = CH[[u, r]] + h[[r, col]], {col, q}],
          {r, q}]],
      {cor, step}];
    a[[i]] = a[[i]] + 1; j = i;
    While[a[[j]] == q, u = u + 1; a[[j]] = 0; j = j + 1;
      a[[j]] = a[[j]] + 1]];
  step = newstep, {i, 2, k}];
MatrixForm[CH]
```

On Walsh transform and matrix factorization

Iliya Bouyukliev, Paskal Piperkov. On Walsh transform and matrix factorization. – In: *Eighth International Workshop on Optimal Codes and Related Topics*, July 10-14, 2017, Sofia, Bulgaria, pp. 55-60.

```
kronekerProd[a_,b_]:= (
  a1=Dimensions[a][[1]];a2=Dimensions[a][[2]];
  b1=Dimensions[b][[1]];b2=Dimensions[b][[2]];
  m=Table[0,{a1*b1},{a2*b2}];
  Do[
   m[[i1*b1+i2,j1*b2+j2]]=a[[i1+1,j1+1]]*b[[i2,j2]],
   {i1,0,a1-1},{i2,b1},{j1,0,a2-1},{j2,b2}];
  m)
```

Vilenkin-Chrestenson transform

```
kernel[m_]:=(r=Table[0,{3},{3}];
  r[[1,1]]=m[[1,1]]+m[[2,1]]+m[[3,1]];
  r[[1,2]]=m[[1,2]]+m[[2,2]]+m[[3,2]];
  r[[1,3]]=m[[1,3]]+m[[2,3]]+m[[3,3]];
  r[[2,1]]=m[[1,1]]+m[[2,3]]+m[[3,2]];
  r[[2,2]]=m[[1,2]]+m[[2,1]]+m[[3,3]];
  r[[2,3]]=m[[1,3]]+m[[2,2]]+m[[3,1]];
  r[[3,1]]=m[[1,1]]+m[[2,2]]+m[[3,3]];
  r[[3,2]]=m[[1,2]]+m[[2,3]]+m[[3,1]];
  r[[3,3]]=m[[1,3]]+m[[2,1]]+m[[3,2]];
  r)
```

Vilenkin-Chrestenson transform

```
hh=Table[{h[[i]],0,0},{i,Length[h]}];
Do[sz=3^l;pos=0;
  While[pos<Length[h],
    Do[m=Table[0,{3}];
      Do[m[[k+1]]=hh[[pos+i+k*sz]],{k,0,2}];
      res=kernel[m];
      Do[hh[[pos+i+k*sz]]=res[[k+1]],{k,0,2}],
      {i,1,sz}];
    pos=pos+3*sz],
  {l,0,Log[3,Length[h]]-1}];
Flagn=True;
Do[Flagn=Flagn&&(hh[[i,2]]==hh[[i,3]]),
  {i,Length[hh]}];
Flagn
```

Calculations over a composite field

```
elements=
  Table[ai=IntegerDigits[i,2];
       While[Length[ai]<4,ai=Prepend[ai,0]];
       ai[[1]]*x^3+ai[[2]]*x^2+ai[[3]]*x+ai[[4]],
       {i,0,15}];
mult=Table[
   PolynomialMod[
       PolynomialRemainder[
             Expand[elements[[i]]*elements[[j]]],
             x^4+x^3+1,x],
       2],
   {i,16},{j,16}];
Do[r=mult[[i,j]];
  Do[If[r==elements[[a]],mult[[i,j]]=a],{a,16}],
  {i,16},{j,16}];mult
trace[y_]:=(a=y;s=elements[[a]];
  Do[a=mult[[a,a]];s=s+elements[[a]],{3}];
  PolynomialMod[s,2])
```