4_SymmetricExt

May 31, 2021

1 Werner state symmetric extensions

We symmetrize the computation in

P. D. Johnson, "Compatible quantum correlations: Extension problems for Werner and isotropic states", PRA, vol. 88, no. 3, 2013

https://journals.aps.org/pra/abstract/10.1103/PhysRevA.88.032323

Initialize the RepLAB toolbox (be in the /replab directory or use **`run path/replab/replab_init.m`**)

[1]:
```
replab_init
replab.globals.useReconstruction(1); % use new algorithms for decomposition
```

```
Adding RepLAB to the path
Initializing dependency vpi
Initializing dependency YALMIP
Initializing dependency sdpt3
Adding embedded SDPT3 solver to the path
Initializing dependency MOcov
Initializing dependency MOxUnit
Initializing dependency cyclolab
```

We declare the symmetry group G as a direct product of those two groups:

- $\mathcal{U}(2)$ acting on the subsystems
- $\mathcal{S}(2)$ acting on the copies

[2]:
```
U2 = replab.U(2);
S2 = replab.S(2);
G = U2.directProduct(S2)
```

```
G =

Direct product group with 2 factors
 identity: {[1, 0; 0, 1], [1, 2]}
factor(1): Unitary group U(2)
factor(2): Symmetric group acting on 2 elements
```

We now describe the action of the two factors on - the original state being tested - the symmetric extension we test the existence of

[3]:
```
% The action of U2 on the original two qubit state
U2_rep2 = kron(U2.definingRep, U2.definingRep);

% The action of U2 on the symmetric extension
U2_rep3 = kron(U2.definingRep, U2.definingRep, U2.definingRep);

% The action of S2 is trivial when there is a single copy of Bob
S2_rep2 = S2.trivialRep('C', 4);

% The action of S2 permutes the two copies of Bob (what we call a index␣
↪relabeling)
S2_rep3 = kron(S2.trivialRep('C', 2), S2.indexRelabelingRep(2).
↪complexification);
```

We construct the representations of the direct product using the representations above of the factor, which commute (important!).

[4]:
```
rep2 = G.commutingFactorRepsRep('C', 4, {U2_rep2 S2_rep2});
rep3 = G.commutingFactorRepsRep('C', 8, {U2_rep3 S2_rep3});
H2 = rep2.hermitianInvariant;
H3 = rep3.hermitianInvariant;
```

We define the partial trace operation (trust us).

[5]:
```
ptFun = @(X) reshape(reshape(permute(reshape(X, [2 4 2 4]), [2 4 1 3]), [16␣
↪4])*[1; 0; 0; 1], [4 4]);
pt = replab.equiop.generic(H3, H2, ptFun);
```

We define

- the singlet state and noise as equivars (note that they are not variables)
- the threshold `t` as a sdpvar
- the symmetric extension we try to find `symExt`

[6]:
```
singlet = replab.equivar(H2, 'value', [0 0 0 0; 0 1 -1 0; 0 -1 1 0; 0 0 0 0]/2);
noise = replab.equivar(H2, 'value', eye(4)/4);
t = sdpvar;
rho = singlet*t + noise*(1-t);
symExt = replab.equivar(H3);
```

[7]:
```
C = [pt(symExt) == rho
     issdp(symExt)
     issdp(rho)]
optimize(C, -t, sdpsettings('solver', 'sdpt3')) % force SDPT3 the default␣
↪Octave solver has problems
```

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|   ID|                          Constraint|      Coefficient range|
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|   #1|             Equality constraint 1x1|      4.2432e-16 to 1.5|
|   #2|   Equality constraint (complex) 1x1|   4.5615e-49 to 1.3333|
|   #3|         Element-wise inequality 1x1|                 1 to 1|
|   #4|         Element-wise inequality 1x1|                 1 to 1|
|   #5|         Element-wise inequality 1x1|                 1 to 1|
|   #6|         Element-wise inequality 1x1|           0.25 to 0.75|
|   #7|         Element-wise inequality 1x1|           0.25 to 0.25|
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

 num. of constraints =  4
 dim. of linear var  =  5
 dim. of free   var  =  3
 *** convert ublk to linear blk
*******************************************************************************
************
   SDPT3: homogeneous self-dual path-following algorithms
*******************************************************************************
************
 version  predcorr  gam  expon
   HKM      1      0.000   1
it pstep dstep pinfeas dinfeas  gap     mean(obj)    cputime    kap   tau
theta
-------------------------------------------------------------------------------
------------
 0|0.000|0.000|8.8e+00|1.8e+01|5.1e+02| 2.604023e+00|
0:0:00|5.1e+02|1.0e+00|1.0e+00| chol 1  1
 1|0.268|0.268|8.1e+00|1.7e+01|5.3e+02| 1.965330e+00|
0:0:00|4.6e+02|1.0e+00|9.2e-01| chol 1  1
 2|0.969|0.969|1.4e+00|2.8e+00|7.5e+01| 4.006016e+00|
0:0:00|7.0e+01|1.1e+00|1.7e-01| chol 1  1
 3|0.933|0.933|1.7e-01|3.5e-01|8.1e+00| 2.020528e+00|
0:0:00|1.0e+00|1.3e+00|2.5e-02| chol 1  1
 4|0.995|0.995|1.2e-02|2.7e-02|3.9e-01| 6.486893e-01|
0:0:00|6.6e-01|1.7e+00|2.3e-03| chol 1  1
 5|0.978|0.978|1.9e-03|8.3e-03|5.7e-02| 6.814912e-01|
0:0:00|9.6e-02|1.7e+00|3.7e-04| chol 1  1
 6|0.967|0.967|2.0e-04|6.2e-03|2.6e-03| 6.723013e-01|
0:0:00|1.7e-02|1.8e+00|4.0e-05| chol 1  1
 7|0.995|0.995|1.8e-05|2.4e-03|1.5e-04| 6.687989e-01|
0:0:00|1.9e-03|1.8e+00|3.6e-06| chol 1  1
 8|1.000|1.000|1.8e-06|9.6e-04|1.8e-05| 6.674999e-01|
0:0:00|1.8e-04|1.8e+00|3.5e-07| chol 1  1
 9|1.000|1.000|1.9e-07|3.8e-04|2.6e-06| 6.669985e-01|
0:0:00|1.8e-05|1.8e+00|3.7e-08| chol 1  1
10|1.000|1.000|2.1e-08|1.5e-04|3.8e-07| 6.667993e-01|
```

```
0:0:00|1.8e-06|1.8e+00|4.2e-09| chol 1  1
11|1.000|1.000|4.7e-09|6.2e-05|2.0e-07| 6.667197e-01|
0:0:00|2.1e-07|1.8e+00|9.4e-10| chol 1  1
12|1.000|1.000|7.0e-10|1.2e-06|2.2e-08| 6.666677e-01|
0:0:00|4.7e-08|1.8e+00|1.4e-10| chol 1  1
13|1.000|1.000|1.1e-10|2.5e-07|3.6e-09| 6.666669e-01|
0:0:00|7.1e-09|1.8e+00|2.2e-11| chol 1  1
14|0.998|0.998|2.1e-11|2.9e-09|4.7e-10| 6.666667e-01|
0:0:00|1.1e-09|1.8e+00|3.5e-12|
  Stop: max(relative gap,infeasibilities) < 1.00e-07
-------------------------------------------------------------------
 number of iterations   = 14
 primal objective value =  6.66666667e-01
 dual   objective value =  6.66666672e-01
 gap := trace(XZ)       = 4.66e-10
 relative gap           = 2.79e-10
 actual relative gap    = -1.99e-09
 rel. primal infeas     = 2.12e-11
 rel. dual   infeas     = 2.86e-09
 norm(X), norm(y), norm(Z) = 5.2e+00, 8.3e-01, 9.1e-01
 norm(A), norm(b), norm(C) = 2.3e+01, 1.0e+00, 6.1e-01
 Total CPU time (secs)  = 0.36
 CPU time per iteration = 0.03
 termination code       =  0
 DIMACS: 2.1e-11  0.0e+00  2.9e-09  0.0e+00  -2.0e-09  2.0e-10
-------------------------------------------------------------------
ans =

  scalar structure containing the fields:

    yalmipversion = 20200930
    matlabversion = 6.2.0
    yalmiptime = 0.092821
    solvertime = 0.7255
    info = Successfully solved (SDPT3-4)
    problem = 0
```

[8]:
```
double(t)
```

```
ans = 0.6667
```