# qustop

## *Release 0.0.3*

**Vincent Russo**

**Apr 15, 2022**

# GETTING STARTED

qustop (QUantum STate OPtimizer) is a Python package for investigating various quantum state optimization scenarios including calculating optimal values for quantum state distinguishability, quantum state exclusion, quantum state cloning, and more.

## CHAPTER ONE

# INSTALLING

The qustop package can be installed from PyPi via

```
pip install qustop
```

To build from source, you may also run the following command from the top-level package directory.

```
python setup.py install
```

To test installation, run

```
>>> import qustop
>>> qustop.about()

qustop: Quantum Optimizer: A Python toolkit for computing optimal values of
various convex optimization problems in quantum information.
=============================================================================
Authored by: Vincent Russo, 2021

Core Dependencies
-----------------
CVXPY Version:  1.1.13
NumPy Version:  1.21.0
SciPy Version:  1.6.3
Optional Dependencies
---------------------
Python Version: 3.9.2
Platform Info:  Darwin (x86_64)
```

This prints out version information about core requirements and optional conic optimization software packages that qustop can interface with.

CHAPTER

TWO

# TESTING

The pytest module is used for testing. In order to run and pytest, you will need to ensure it is installed on your machine. Consult the pytest website for more information. To run the suite of tests for qustop, run the following command in the root directory of this project:

```
pytest --cov-report term-missing --cov=qustop tests/
```

# CHAPTER THREE

# CONTRIBUTING

All contributions, bug reports, bug fixes, documentation improvements, enhancements, and ideas are welcome.

A detailed overview on how to contribute can be found in the contributing guide.

CHAPTER

FOUR

# CITING

You can cite qustop using the following DOI: 5595257.

If you are using the qustop software package in research work, please include an explicit mention of qustop in your publication. Something along the lines of:

> To solve problem “X” we used *qustop*; a package for studying quantum state optimization scenarios.

A BibTeX entry that you can use to cite qustop is provided here:

```
@misc{qustop,
   author       = {Vincent Russo},
   title        = {qustop: A {P}ython package for investigating quantum state␣
↪optimization, version 0.1},
   howpublished = {\url{https://github.com/vprusso/qustop}},
   month        = May,
   year         = 2021,
   doi          = {5595257}}
```

CHAPTER

FIVE

# INTRODUCTORY TUTORIAL

This tutorial will illustrate the basics of how to use `qustop`.

This is a user guide for `qustop` and is not meant to serve as an introduction to quantum information. For introductory material on quantum information, please consult "Quantum Information and Quantum Computation" by Nielsen and Chuang or the freely available lecture notes "Introduction to Quantum Computing" by John Watrous.

More advanced tutorials can be found on the main documentation directory.

This tutorial assumes you have `qustop` installed on your machine. If you do not, please consult the getting started instructions.

## 5.1 States, ensembles, and measurements

Quantum states, and collections of those quantum states that form ensembles, are the core building blocks of `qustop`.

### 5.1.1 Quantum states

A *quantum state* is a density operator

$$\rho \in \mathrm{D}(\mathbb{C}^d)$$

where $\mathbb{C}^d$ is a complex Euclidean space of dimension $d$ and where $\mathrm{D}(\cdot)$ represents the set of density matrices, that is, the set of matrices that are positive semidefinite with trace equal to $1$. We will typically represent complex Euclidean spaces using the scripted capital letters $\mathcal{A}, \mathcal{B}, \mathcal{X}, \mathcal{Y}$, etc.

Consider the density matrix corresponding to one of the four Bell states

$$\rho_0 = |\psi_0\rangle\langle\psi_0| = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \in \mathrm{D}(\mathcal{A} \otimes \mathcal{B})$$

where

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \in \mathcal{A} \otimes \mathcal{B}$$

such that $\mathcal{A} = \mathbb{C}^2$ and $\mathcal{B} = \mathbb{C}^2$. We can use `qustop` to encode this state as follows.

```
import numpy as np
from qustop import State

# Define the |0> and |1> ket vectors.
q_0 = np.array([[1, 0]]).T
q_1 = np.array([[0, 1]]).T

# Define the respective dimensions of each complex Euclidean space.
dims = [2, 2]

# Define the Bell state vector.
psi_0 = 1/np.sqrt(2) * np.kron(q_0, q_0) + 1/np.sqrt(2) * np.kron(q_1, q_1)

# Define a `State` object corresponding to the Bell vector.
rho_0 = State(psi_0, dims)
```

Printing the rho_0 variable gives some further information about the state.

```
>>> print(rho_0)
State:
 dimensions = [2, 2],
 spaces = ^2  ^2,
 labels = A_1  B_2,
 pure = True,
 shape = (4, 4),
```

For instance, we see the shape attribute gives information about the size of the density matrix of the state. There is also information about the subsystems along with which party the subsystems belong to (either Alice or Bob), whether the state is pure, etc.

We can use the value property of any State object to obtain the numpy matrix representation of the quantum state

```
>>> print(rho_0.value)
[[0.5 0.  0.  0.5]
 [0.  0.  0.  0. ]
 [0.  0.  0.  0. ]
 [0.5 0.  0.  0.5]]
```

We can also do things like take tensor products of State objects.

```
>>> sigma_0 = rho_0.kron(rho_0)
>>> print(sigma_0)
State:
 dimensions = [2, 2, 2, 2],
 spaces = ^2  ^2  ^2  ^2,
 labels = A_1  B_2  A_3  B_4,
 pure = True,
 shape = (16, 16),
```

It is sometimes convenient to swap the subsystems of a given state. For instance, this example shows how we can swap the second and third subsystems of the sigma_0 state.

```
>>> sigma_0.swap([2, 3])
>>> print(sigma_0)
```

(continues on next page)

(continued from previous page)

```
State:
 dimensions = [2, 2, 2, 2],
 spaces = ^2  ^2  ^2  ^2,
 labels = A_1  A_3  B_2  B_4,
 pure = True,
 shape = (16, 16),
```

Notice how the A_3 and B_2 subsystems are swapped.

### 5.1.2 Ensembles

An *ensemble* is a collection of $N$ quantum states defined over some complex Euclidean space $\mathcal{X}$ as

$$\eta = \{(p_1, \rho_1), \ldots, (p_N, \rho_N)\},$$

where $(p_1, \ldots, p_N)$ is a vector of probability values and where $\rho_1, \ldots, \rho_N \in \mathrm{D}(\mathcal{X})$ are quantum states.

Recall the four two-qubit Bell states

$$\begin{aligned} |\psi_0\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \quad |\psi_1\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}, \\ |\psi_2\rangle &= \frac{|01\rangle - |10\rangle}{\sqrt{2}}, \quad |\psi_3\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}. \end{aligned} \tag{5.1}$$

$$|\psi_3\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}.$$

The corresponding density operators may be defined as

$$\begin{aligned} \rho_0 &= |\psi_0\rangle\langle\psi_0|, \quad \rho_1 = |\psi_1\rangle\langle\psi_1|, \\ \rho_2 &= |\psi_2\rangle\langle\psi_2|, \quad \rho_3 = |\psi_3\rangle\langle\psi_3|. \end{aligned}$$

$$\rho_3 = |\psi_3\rangle\langle\psi_3|.$$

We can define the following ensemble consisting of the Bell states where the probability of selecting any one state from the ensemble is equal to $1/4$:

$$\eta = \left\{ \left(\frac{1}{4}, \rho_0\right), \left(\frac{1}{4}, \rho_1\right), \left(\frac{1}{4}, \rho_2\right), \left(\frac{1}{4}, \rho_3\right) \right\}.$$

In `qustop`, we would define this ensemble like so

```
from toqito.states import bell
from qustop import State, Ensemble

# Construct the corresponding density matrices of the Bell states.
dims = [2, 2]
states = [
    State(bell(0) * bell(0).conj().T, dims),
    State(bell(1) * bell(1).conj().T, dims),
    State(bell(2) * bell(2).conj().T, dims),
    State(bell(3) * bell(3).conj().T, dims)
]
ensemble = Ensemble(states=states, probs=[1/4, 1/4, 1/4, 1/4])
```

Printing out any `Ensemble` object gives us some information about the contents:

```
>>> print(ensemble)
Ensemble:
 num_states = 4,
 states = _0  _1  _2  _3,
 is_mutually_orthogonal = True,
 is_linearly_independent = True,
```

We can see certain pieces of information including how many states are contained in the ensemble, whether the states in the ensemble are all mutually orthogonal, linearly independent, etc.

We can access any of the states from the `Ensemble` object using standard array indexing notation. For instance, here is how we can access the first state in the ensemble.

```
>>> print(ensemble[0])
State:
 dimensions = [2, 2],
 spaces = ^2  ^2,
 labels = A_1  B_2,
 pure = True,
 shape = (4, 4),
```

We may also wish to apply some of the functions that we saw before for `State` objects onto the entire ensemble. For instance, here is an example of how we can swap the first and second subsystems of each state in the ensemble.

```
>>> ensemble.swap([1, 2])
>>> print(ensemble[0])
State:
 dimensions = [2, 2],
 spaces = ^2  ^2,
 labels = B_2  A_1,
 pure = True,
 shape = (4, 4),
```

### 5.1.3 Measurements

A *measurement* is defined as a function

$$\mu : \Sigma \rightarrow \text{Pos}(\mathcal{X})$$

for some finite and nonempty set $\Sigma$ and some complex Euclidean space $\mathcal{X}$ satisfying the constraint that

$$\sum_{a \in \Sigma} \mu(a) = \mathbb{I}_{\mathcal{X}}.$$

There are many different classes of measurements.

## 5.2 Quantum state distinguishability

Given an ensemble of quantum states, we can consider the setting of *quantum state distinguishability*. This setting can be considered as an interaction between two parties–typically denoted as *Alice* and *Bob*.

A more in-depth description and tutorial on this setting in `qustop` can be found in:

- Tutorial : Quantum state distinguishability.

More in-depth descriptions pertaining to quantum state distinguishability under positive, PPT, and separable measurements can be found in:

- Tutorial: Quantum state distinguishability using positive measurements.
- Tutorial: Quantum state distinguishability using PPT measurements.
- Tutorial: Quantum state distinguishability using separable measurements.

## 5.3 Quantum state exclusion

- Tutorial : Quantum state exclusion.

## 5.4 Quantum state cloning

- Tutorial : Quantum state cloning.

# CHAPTER SIX

# QUANTUM STATES AND ENSEMBLES

CHAPTER

SEVEN

# OPTIMAL QUANTUM STATE DISCRIMINATION

CHAPTER

EIGHT

# OPTIMAL QUANTUM STATE EXCLUSION

CHAPTER

NINE

# OPTIMAL QUANTUM STATE CLONING

CHAPTER

TEN

# QUANTUM STATE DISTINGUISHABILITY

In this tutorial we are going to cover the problem of *quantum state distinguishability* (sometimes analogously referred to as *quantum state discrimination*). We are going to briefly describe the problem setting and then describe how one may use `qustop` to calculate the optimal probability with which this problem can be solved when given access to certain measurements.

Further information beyond the scope of this tutorial can be found in the text [WatrousQI] as well as the course [SikoraSDP].

## 10.1 The state distinguishability problem

The quantum state distinguishability problem is phrased as follows.

1. Alice possesses an ensemble of $n$ quantum states:

$$\eta = \left( (p_0, \rho_0), \ldots, (p_n, \rho_n) \right),$$

where $p_i$ is the probability with which state $\rho_i$ is selected from the ensemble. Alice picks $\rho_i$ with probability $p_i$ from her ensemble and sends $\rho_i$ to Bob.

2. Bob receives $\rho_i$. Both Alice and Bob are aware of how the ensemble is defined but he does *not* know what index $i$ corresponding to the state $\rho_i$ he receives from Alice is.

3. Bob wants to guess which of the states from the ensemble he was given. In order to do so, he may measure $\rho_i$ to guess the index $i$ for which the state in the ensemble corresponds.

This setting is depicted in the following figure.

Fig. 1: The quantum state distinguishability setting.

### 10.1.1 Distinguishability methods

- Minimum-error discrimination:

Minimize the average probability of making an error in conclusively identifying the state.

- Unambiguous discrimination:

Never give an incorrect answer, although the answer can be inconclusive.

### 10.1.2 Distinguishability measurements

Depending on the sets of measurements that Alice and Bob are allowed to use, the optimal probability of distinguishing a given set of states is characterized by the following image.

Fig. 2: Measurement hierarchy.

That is, the probability that Alice and Bob are able to distinguish using PPT measurements is a natural upper bound on the optimal probability of distinguishing via separable measurements and so on.

In general:

- LOCC: These are difficult objects to handle mathematically; difficult to design protocols for and difficult to provide bounds on their power.
- Separable: Separable measurements have a nicer structure than LOCC. Unfortunately, optimizing over separable measurements in NP-hard.
- PPT: PPT measurements offer a nice structure and there exists efficient techniques that allow one to optimize over the set of PPT measurements via semidefinite programming.
- Positive: These measurements are the most general and constitute the set of all valid quantum operations that Alice and Bob can perform. The optimal value of distinguishing via positive operations can be phrased as an SDP.

### 10.1.3 Optimal probability of distinguishing a quantum state

The optimal probability of distinguishing using positive measurements serves as an upper bound on the optimal probability of distinguishing using PPT, separable, and LOCC measurements. Specifically, given an ensemble of quantum states, $\eta$, it holds that

$$0 \leq \text{opt}_{\text{LOCC}}(\eta) \leq \text{opt}_{\text{SEP}}(\eta) \leq \text{opt}_{\text{PPT}}(\eta) \leq \text{opt}_{\text{POS}}(\eta) \leq 1$$

where:

- $\text{opt}_{\text{POS}}(\eta)$ represents the optimal probability of distinguishing using positive measurements,
- $\text{opt}_{\text{PPT}}(\eta)$ represents the probability of distinguishing via PPT measurements,
- $\text{opt}_{\text{SEP}}(\eta)$ represents the probability of distinguishing via separable measurements,
- $\text{opt}_{\text{LOCC}}(\eta)$ represents the probability of distinguishing via LOCC measurements.

## 10.2 References

CHAPTER

ELEVEN

# DISTINGUISHING QUANTUM STATES VIA POSITIVE MEASUREMENTS

In this tutorial, we are going to show how to make use of `qustop` to calculate the optimal probability of distinguishing a state from an ensemble of quantum states when Alice and Bob are allowed to use global (positive) measurements on their system.

## 11.1 Minimum-error distinguishability via positive measurements

The optimal probability with which Bob can distinguish the state he is given may be obtained by solving the following semidefinite program (SDP).

**Primal:**

$$\begin{aligned} \text{maximize:} \quad & \sum_{i=0}^{n} p_i \langle M_i, \rho_i \rangle \\ \text{subject to:} \quad & \sum_{i=0}^{n} M_i = \mathbb{I}_{\mathcal{A} \otimes \mathcal{B}}, \\ & M_1, \ldots, M_n \in \text{Pos}(\mathcal{A} \otimes \mathcal{B}). \end{aligned}$$

**Dual:**

$$\begin{aligned} \text{minimize:} \quad & \text{Tr}(Y) \\ \text{subject to:} \quad & Y - \rho_i \in \text{Pos}(\mathcal{A} \otimes \mathcal{B}), \quad \forall i = 1, \ldots, n, \\ & Y \in \text{Herm}(\mathcal{A} \otimes \mathcal{B}). \end{aligned}$$

$\text{Tr}(Y)$subject to:
$Y \in \text{Herm}(\mathcal{A} \otimes \mathcal{B})$.

The `qustop` package solves either of these two optimization problems depending on whether the optimal measurements are required.

For the special case of distinguishing between two states, the probability of optimally distinguishing is exactly

$$\text{opt}_{\text{POS}}(\eta) = \frac{1}{2} + \frac{1}{4} \left\| \eta(0) - \eta(1) \right\|_1$$

where $\|\cdot\|_1$.

A result of [tWalgate00] shows that any two orthogonal pure states can be distinguished perfectly. This result actually applies to LOCC measurements and is a stronger claim than just for positive measurements, but since $\text{opt}_{\text{LOCC}}(\eta) \leq \text{opt}_{\text{POS}}(\eta)$ is true for any ensemble $\eta$, it also holds for positive measurements.

For example, consider the two orthogonal pure states

$$|\psi_0\rangle = \sqrt{\frac{3}{4}}|+\rangle + \sqrt{\frac{1}{4}}|-\rangle \quad \text{and} \quad |\psi_1\rangle = \sqrt{\frac{1}{4}}|+\rangle - \sqrt{\frac{3}{4}}|-\rangle.$$

Since $|\psi_0\rangle$ and $|\psi_1\rangle$ are pure and mutually orthogonal, they are able to be perfectly distinguished.

```python
import numpy as np

from qustop import Ensemble, OptDist, State

# Define single-qubit |+> and |-> basis states.
e_p, e_m = (
    1 / np.sqrt(2) * np.array([[1, 1]]).T,
    1 / np.sqrt(2) * np.array([[1, -1]]).T,
)

dims = [2]
# sqrt(3/4)|+> + sqrt(1/4)|->
psi_1 = State(np.sqrt(3 / 4) * e_p + np.sqrt(1 / 4) * e_m, dims)
# sqrt(1/4)|+> - sqrt(3/4)|->
psi_2 = State(np.sqrt(1 / 4) * e_p - np.sqrt(3 / 4) * e_m, dims)

# Verify that the states `psi_1` and `psi_2` are pure:
print(f"Is psi_1 pure: {psi_1.is_pure}")
print(f"Is psi_2 pure: {psi_2.is_pure}")

ensemble = Ensemble([psi_1, psi_2])

res = OptDist(ensemble, "pos", "min-error")

# 0.999999998061445
res.solve()
print(res.value)

# The closed-form equation yields: 0.9999999999999999 = 1
print(1 / 2 + 1 / 4 * np.linalg.norm(psi_1.value - psi_2.value, ord="nuc"))
```

Consider now the following two mixed states

$$|\phi_1\rangle = \sqrt{\frac{3}{4}}|+\rangle + \sqrt{\frac{1}{4}}|-\rangle \quad \text{and} \quad |\phi_2\rangle = \sqrt{\frac{1}{4}}|+\rangle + \sqrt{\frac{3}{4}}|-\rangle.$$

The following code sample shows that the closed-form equation matches the result obtained from `qustop`, however, since they are mixed states and not pure, we are not able to perfectly distinguish them.

```python
import numpy as np

from qustop import Ensemble, OptDist, State

```

(continues on next page)

(continued from previous page)

```
# Define single-qubit |+> and |-> basis states.
e_p, e_m = (
    1 / np.sqrt(2) * np.array([[1, 1]]).T,
    1 / np.sqrt(2) * np.array([[1, -1]]).T,
)

dims = [2]
# sqrt(3/4) |+> + sqrt(1/4)|->
phi_1 = State(np.sqrt(3 / 4) * e_p + np.sqrt(1 / 4) * e_m, dims)
# sqrt(1/4) |+> + sqrt(3/4)|->
phi_2 = State(np.sqrt(1 / 4) * e_p + np.sqrt(3 / 4) * e_m, dims)

# Verify that the states `phi_1` and `phi_2` are mixed:
print(f"Is phi_1 pure: {phi_1.is_pure}")
print(f"Is phi_2 pure: {phi_2.is_pure}")

ensemble = Ensemble([phi_1, phi_2])

res = OptDist(ensemble, "pos", "min-error")

# 0.7500000000609778
res.solve()
print(res.value)

# The closed-form equation yields: 3/4
print(1 / 2 + 1 / 4 * np.linalg.norm(phi_1.value - phi_2.value, ord="nuc"))
```

On the note of orthogonality, if the ensemble of states provided consist of all mutually orthogonal states, then it is possible to distinguish with perfect probability in this special case.

As a prototypical example, consider the four Bell states

$$
\begin{aligned}
|\psi_0\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \quad |\psi_1\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}, \\
|\psi_2\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}, \quad |\psi_3\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}.
\end{aligned}
$$

$$|\psi_3\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}.$$

```
from toqito.states import bell

from qustop import Ensemble, OptDist, State

```

(continues on next page)

(continued from previous page)

```
dims = [2, 2]
ensemble = Ensemble(
    [
        State(bell(0), dims),
        State(bell(1), dims),
        State(bell(2), dims),
        State(bell(3), dims),
    ]
)

# Verify that states in the ensemble are mutually orthogonal:
print(f"Are states mutually orthogonal: {ensemble.is_mutually_orthogonal}")

res = OptDist(ensemble, "pos", "min-error")

# Mutually orthogonal states are optimally distinguishable--giving
# an optimal value of one.
res.solve()
# 1.0000000000879223
print(res.value)
```

If there are more than two states and those states are not mutually orthogonal, no closed-form equation is known to exist, so we resort to solving the SDP.

For instance, consider the following set of three non-mutually-orthogonal states

$$
\begin{aligned}
&|\phi_1\rangle = \sqrt{\frac{3}{4}}|+\rangle + \sqrt{\frac{1}{4}}|-\rangle, \quad |\phi_2\rangle = \sqrt{\frac{1}{4}}|+\rangle + \sqrt{\frac{3}{4}}|-\rangle, \\
&|\phi_3\rangle = \sqrt{\frac{1}{2}}|+\rangle + \sqrt{\frac{1}{2}}|-\rangle.
\end{aligned}
$$

```
import numpy as np

from qustop import Ensemble, OptDist, State

# Define single-qubit |+> and |-> basis states.
e_p, e_m = (
    1 / np.sqrt(2) * np.array([[1, 1]]).T,
    1 / np.sqrt(2) * np.array([[1, -1]]).T,
)

dims = [2]
ensemble = Ensemble(
```

(continues on next page)

(continued from previous page)

```
    [
        State(np.sqrt(3 / 4) * e_p + np.sqrt(1 / 4) * e_m, dims),
        State(np.sqrt(1 / 4) * e_p + np.sqrt(3 / 4) * e_m, dims),
        State(np.sqrt(1 / 2) * e_p + np.sqrt(1 / 2) * e_m, dims),
    ]
)

# Verify that ensemble consists of non-mutually-orthogonal states.:
print(f"Is ensemble mutually orthogonal: {ensemble.is_mutually_orthogonal}")

# For any set of more than two states that are non-mutually orthogonal,
# no closed-form expression for optimal distinguishability is known to exist.
# Therefore, we must resort to solving the SDP to determine what the optimal
# probability is.
res = OptDist(ensemble, "pos", "min-error")

# 0.5000000000237005
res.solve()
print(res.value)
```

## 11.2 Unambiguous distinguishability via positive measurements

The optimal probability with which Bob can distinguish the state he is given unambiguously may be obtained by solving the following semidefinite program (SDP).

$$\begin{aligned} &\textbf{Primal:} \\ \text{maximize:} \quad & \sum_{i=0}^{n} p_i \langle M_i, \rho_i \rangle \\ \text{subject to:} \quad & \sum_{i=0}^{n+1} M_i = \mathbb{I}_{\mathcal{A} \otimes \mathcal{B}}, \\ & \langle M_i \rho_j \rangle = 0 \quad \forall i \neq j = 1, \ldots, n \\ & M_1, \ldots, M_n \in \text{Pos}(\mathcal{A} \otimes \mathcal{B}). \end{aligned}$$

As an example, consider the set of three non-mutually-orthogonal states we considered earlier

$$\begin{aligned} |\phi_1\rangle &= \sqrt{\frac{3}{4}}|+\rangle + \sqrt{\frac{1}{4}}|-\rangle, \quad |\phi_2\rangle = \sqrt{\frac{1}{4}}|+\rangle + \sqrt{\frac{3}{4}}|-\rangle, \\ |\phi_3\rangle &= \sqrt{\frac{1}{2}}|+\rangle + \sqrt{\frac{1}{2}}|-\rangle. \end{aligned}$$

The probability to distinguish amongst these states unambiguously gives a value of zero, while the minimum-error case, as we saw, gave a value of $3/4$.

```python
import numpy as np

from qustop import Ensemble, OptDist, State

# Define single-qubit |+> and |-> basis states.
e_p, e_m = (
    1 / np.sqrt(2) * np.array([[1, 1]]).T,
    1 / np.sqrt(2) * np.array([[1, -1]]).T,
)

dims = [2]
ensemble = Ensemble(
    [
        State(np.sqrt(3 / 4) * e_p + np.sqrt(1 / 4) * e_m, dims),
        State(np.sqrt(1 / 4) * e_p + np.sqrt(3 / 4) * e_m, dims),
        State(np.sqrt(1 / 2) * e_p + np.sqrt(1 / 2) * e_m, dims),
    ]
)

res = OptDist(ensemble, "pos", "unambiguous")

# Probability of distinguishing unambiguously is zero.
res.solve()
print(res.value)
```

## 11.3 References

CHAPTER

# TWELVE

# DISTINGUISHING QUANTUM STATES VIA PPT MEASUREMENTS

In this section we will be investigation how to make use of the `qustop` package to optimally distinguish quantum states via PPT measurements.

## 12.1 Minimum-error

In [Cosentino13], an semidefinite program formulation whose optimal value corresponds to the optimal probability of distinguishing a quantum state from an ensemble using PPT measurements with minimum error was provided. The primal and dual problems of this SDP are defined as follows.

**Primal:**

$$\begin{aligned} \text{maximize:} \quad & \sum_{j=1}^{k} \langle P_j, \rho_j \rangle \\ \text{subject to:} \quad & P_1 + \cdots + P_k = \mathbb{I}_{\mathcal{A}} \otimes \mathbb{I}_{\mathcal{B}}, \\ & P_1, \ldots, P_k \in \text{PPT}(\mathcal{A} : \mathcal{B}). \end{aligned}$$

$\sum_{j=1}^{k} \langle P_j, \rho_j \rangle$subject to:
$P_1, \ldots, P_k \in \text{PPT}(\mathcal{A} : \mathcal{B}).$

**Dual:**

$$\begin{aligned} \text{minimize:} \quad & \frac{1}{k} \text{Tr}(Y) \\ \text{subject to:} \quad & Y - \rho_j \geq \text{T}_{\mathcal{A}}(Q_j), \quad j = 1, \ldots, k, \\ & Y \in \text{Herm}(\mathcal{A} \otimes \mathcal{B}), \\ & Q_1, \ldots, Q_k \in \text{Pos}(\mathcal{A} \otimes \mathcal{B}). \end{aligned}$$

$\frac{1}{k}\text{Tr}(Y)$subject to:
$Y \in \text{Herm}(\mathcal{A} \otimes \mathcal{B})$,

## 12.2 Unambiguous

In [Cosentino13], an semidefinite program formulation whose optimal value corresponds to the optimal probability of distinguishing a quantum state from an ensemble using PPT measurements unambiguously was provided. The primal and dual problems of this SDP are defined as follows.

**Primal:**

$$\begin{aligned}
\text{maximize:} \quad & \sum_{j=1}^{k} \langle P_j, \rho_j \rangle \\
\text{subject to:} \quad & P_1 + \cdots + P_k = \mathbb{I}_{\mathcal{A}} \otimes \mathbb{I}_{\mathcal{B}}, \\
& P_1, \ldots, P_k \in \text{PPT}(\mathcal{A} : \mathcal{B}), \\
& \langle P_i, \rho_j \rangle = 0, \quad 1 \leq i, j \leq k, \quad i \neq j.
\end{aligned}$$

$\sum_{j=1}^{k} \langle P_j, \rho_j \rangle$subject to:
$P_1, \ldots, P_k \in \text{PPT}(\mathcal{A} : \mathcal{B})$,

**Dual:**

$$\begin{aligned}
\text{minimize:} \quad & \frac{1}{k}\text{Tr}(Y) \\
\text{subject to:} \quad & Y - \rho_j \geq \text{T}_{\mathcal{A}}(Q_j), \quad j = 1, \ldots, k, \\
& Y \in \text{Herm}(\mathcal{A} \otimes \mathcal{B}), \\
& Q_1, \ldots, Q_k \in \text{Pos}(\mathcal{A} \otimes \mathcal{B}).
\end{aligned}$$

$\frac{1}{k}\text{Tr}(Y)$subject to:
$Y \in \text{Herm}(\mathcal{A} \otimes \mathcal{B})$,

### 12.2.1 Distinguishing four Bell states

Consider the following Bell states:

$$
\begin{aligned}
|\psi_0\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}, &\quad |\psi_1\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}, \\
|\psi_2\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}, &\quad |\psi_3\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}.
\end{aligned}
$$

Assuming a uniform probability of selecting from any one of these states, that is, assuming we define an ensemble of Bell states defined as

$$\mathbb{B} = \left\{ \left(|\psi_0\rangle, \frac{1}{4}\right), \left(|\psi_1\rangle, \frac{1}{4}\right), \left(|\psi_2\rangle, \frac{1}{4}\right), \left(|\psi_3\rangle, \frac{1}{4}\right) \right\}$$

it holds that

$$\text{opt}_{\text{PPT}}(\mathbb{B}) = \frac{1}{2}.$$

We can observe this using `qustop` as follows.

```python
from toqito.states import bell

from qustop import Ensemble, OptDist, State

# Construct the corresponding density matrices of the Bell states.
dims = [2, 2]
ensemble = Ensemble(
    [
        State(bell(0), dims),
        State(bell(1), dims),
        State(bell(2), dims),
        State(bell(3), dims),
    ],
    [1 / 4, 1 / 4, 1 / 4, 1 / 4],
)
res = OptDist(ensemble, "ppt", "min-error")
res.solve()
```

(continues on next page)

(continued from previous page)

```python

# 0.50000000000530641
print(res.value)
```

Indeed, a stronger statement is known to hold for $\mathbb{B}$, that is

$$\text{opt}_{\text{LOCC}}(\mathbb{B}) = \frac{1}{2}.$$

Recall that for any ensemble $\eta$, it holds that $\text{opt}_{\text{LOCC}}(\eta) < \text{opt}_{\text{PPT}}(\eta)$.

### 12.2.2 Four indistinguishable orthogonal maximally entangled states

In [YDY12] the following ensemble of states was shown not to be perfectly distinguishable by PPT measurements, and therefore also indistinguishable via LOCC measurements.

$$\begin{aligned} \rho_0 &= |\psi_0\rangle|\psi_0\rangle\langle\psi_0|\langle\psi_0|, \quad & \rho_1 &= |\psi_1\rangle|\psi_3\rangle\langle\psi_1|\langle\psi_3|, \\ \rho_2 &= |\psi_3\rangle|\psi_1\rangle\langle\psi_3|\langle\psi_1|, \quad & \rho_3 &= |\psi_1\rangle|\psi_1\rangle\langle\psi_1|\langle\psi_1|, \end{aligned}$$

While it was known that perfect distinguishability could not be achieved, the actual value and bound of optimal distinguishability was not known. It was shown in [Cosentino13] and later extended in [CR13] that the optimal probability of distinguishing the above ensemble via a PPT measurement should yield an optimal probability of 7/8.

```python
import numpy as np
from toqito.states import bell

from qustop import Ensemble, OptDist, State

# Define the maximally entangled states from arXiv1107.3224
dims = [2, 2, 2, 2]
ensemble = Ensemble(
    [
        State(np.kron(bell(0), bell(0)) * np.kron(bell(0), bell(0)).conj().T),
        State(np.kron(bell(2), bell(1)) * np.kron(bell(2), bell(1)).conj().T),
        State(np.kron(bell(3), bell(1)) * np.kron(bell(3), bell(1)).conj().T),
        State(np.kron(bell(1), bell(1)) * np.kron(bell(1), bell(1)).conj().T),
    ]
)

# The min-error probability of distinguishing via PPT
# is equal to 7/8.
res = OptDist(ensemble, "ppt", "min-error")
res.solve()

# 0.87500000060847
print(res.value)
```

In was also shown in [Cosentino13] that the optimal probability of distinguishing this ensemble unambiguously when making use of PPT measurements was equal to 3/4.

```python
import numpy as np
from toqito.states import bell

from qustop import Ensemble, OptDist, State

# Define the maximally entangled states from arXiv1107.3224
dims = [2, 2, 2, 2]
ensemble = Ensemble(
    [
        State(
            np.kron(bell(0), bell(0)) * np.kron(bell(0), bell(0)).conj().T,
            dims,
        ),
        State(
            np.kron(bell(2), bell(1)) * np.kron(bell(2), bell(1)).conj().T,
            dims,
        ),
        State(
            np.kron(bell(3), bell(1)) * np.kron(bell(3), bell(1)).conj().T,
            dims,
        ),
        State(
            np.kron(bell(1), bell(1)) * np.kron(bell(1), bell(1)).conj().T,
            dims,
        ),
    ]
)

# The unambiguous probability of distinguishing via PPT
# is equal to 3/4.
res = OptDist(ensemble, "ppt", "unambiguous")
res.solve()

# 0.7499999975754753
print(res.value)
```

## 12.3 Entanglement cost of distinguishing Bell states

One may ask whether the ability to distinguish a state can be improved by making use of an auxiliary resource state.

$$|\tau_{\epsilon}\rangle = \sqrt{\frac{1+\epsilon}{2}}|00\rangle + \sqrt{\frac{1-\epsilon}{2}}|11\rangle,$$

for some $\epsilon \in [0, 1]$.

### 12.3.1 Distinguishing four Bell states

It was shown in [BCJRWY15] that the probability of distinguishing four Bell states with a resource state via PPT measurements is given by the closed-form expression:

$$\text{opt}_{\text{PPT}}(\eta) = \text{opt}_{\text{SEP}}(\eta) = \frac{1}{2}\left(1 + \sqrt{1 - \epsilon^2}\right)$$

where the ensemble is defined as

$$\eta = \left\{ |\psi_0\rangle \otimes |\tau_\epsilon\rangle, |\psi_1\rangle \otimes |\tau_\epsilon\rangle, |\psi_2\rangle \otimes |\tau_\epsilon\rangle, |\psi_3\rangle \otimes |\tau_\epsilon\rangle \right\}.$$

Using `qustop`, we may encode this scenario as follows.

```python
import numpy as np
from toqito.states import basis, bell

from qustop import Ensemble, OptDist, State

e_0, e_1 = basis(2, 0), basis(2, 1)

eps = 0.5
tau = np.sqrt((1 + eps) / 2) * np.kron(e_0, e_0) + np.sqrt(
    (1 - eps) / 2
) * np.kron(e_1, e_1)

dims = [2, 2, 2, 2]
ensemble = Ensemble(
    [
        State(np.kron(bell(0), tau), dims),
        State(np.kron(bell(1), tau), dims),
        State(np.kron(bell(2), tau), dims),
        State(np.kron(bell(3), tau), dims),
    ],
    [1 / 4, 1 / 4, 1 / 4, 1 / 4],
)

ppt_res = OptDist(ensemble, "ppt", "min-error")
ppt_res.solve()

sep_res = OptDist(ensemble, "sep", "min-error")
sep_res.solve()

eq = 1 / 2 * (1 + np.sqrt(1 - eps ** 2))

```

(continues on next page)

(continued from previous page)

```
# 0.9330127018922193
print(eq)
# 0.9330127016540999
print(ppt_res.value)
# 0.9330127016540999
print(sep_res.value)
```

Note that [BCJRWY15] also proved the same closed-form expression for when Alice and Bob make use of separable measurements. More on that in the tutorial on distinguishing via separable measurements.

## 12.4 Werner hiding pairs

In [TDL01] and [DLT02], a quantum data hiding protocol that encodes a classical bit in a Werner hiding pair was provided.

A Werner hiding pair is defined by

$$\sigma_0^{(n)} = \frac{\mathbb{I} \otimes \mathbb{I} + W_n}{n(n+1)} \quad \text{and} \quad \sigma_1^{(n)} = \frac{\mathbb{I} \otimes \mathbb{I} - W_n}{n(n-1)}$$

where

$$W_n = \sum_{i,j=0}^{n-1} |i\rangle\langle j| \otimes |j\rangle\langle i| \in \mathrm{U}\left(\mathbb{C}^n \otimes \mathbb{C}^n\right)$$

is the swap operator defined for some dimension $n \geq 2$.

It was shown in [Cosentino15] that

$$\text{opt}_{\text{PPT}}(\eta) = \frac{1}{2} + \frac{1}{n+1},$$

where $\eta = \{\sigma_0, \sigma_1\}$. Using qustop, we may encode this scenario as follows.

```
import numpy as np
from toqito.perms import swap_operator

from qustop import Ensemble, OptDist, State

dim = 2
sigma_0 = (
    np.kron(np.identity(dim), np.identity(dim)) + swap_operator(dim)
) / (dim * (dim + 1))
sigma_1 = (
    np.kron(np.identity(dim), np.identity(dim)) - swap_operator(dim)
```

(continues on next page)

(continued from previous page)

```
) / (dim * (dim - 1))

ensemble = Ensemble([State(sigma_0, [dim, dim]), State(sigma_1, [dim, dim])])

expected_val = 1 / 2 + 1 / (dim + 1)

res = OptDist(ensemble, "ppt", "min-error")
res.solve()

# opt_ppt \approx 0.8333333333668715
print(res.value)
# Closed-form expression is : 1/2 + 1/(dim+1) = 0.8333333333333333
print(expected_val)
```

## 12.5 References

CHAPTER

THIRTEEN

# DISTINGUISHING QUANTUM STATES VIA SEPARABLE MEASUREMENTS

As previously mentioned, optimizing over the set of separable measurements is NP-hard. However, there does exist a hierarchy of semidefinite programs which eventually does converge to the separable value. This hierarchy is based off the notion of symmetric extensions. More information about this hierarchy of SDPs can be found here [Nav08].

## 13.1 Minimum-error distinguishability via separable measurements

**Primal:**

$$\begin{aligned}
\text{maximize:} \quad & \sum_{k=1}^N p_k \langle \rho_k, \mu(k) \rangle, \\
\text{subject to:} \quad & \sum_{k=1}^N \mu(k) = \mathbb{I}_{\mathcal{X} \otimes \mathcal{Y}}, \\
& \text{Tr}_{\mathcal{Y}_2 \otimes \ldots \otimes \mathcal{Y}_s} (X_k) = \mu(k), \\
& \left(\mathbb{I}_{\mathcal{X}} \otimes \Pi_{\mathcal{Y} \mathcal{Y}_2 \ldots \mathcal{Y}_s}\right) X_k \left(\mathbb{I}_{\mathcal{X}} \otimes \Pi_{\mathcal{Y} \mathcal{Y}_2 \ldots \mathcal{Y}_s}\right) = X_k \\
& \text{T}_{\mathcal{X}}(X_k) \in \text{Pos}\left(\mathcal{X} \otimes \mathcal{Y} \otimes \mathcal{Y}_2 \otimes \ldots \otimes \mathcal{Y}_s\right), \\
& \text{T}_{\mathcal{Y}_2}(X_k) \in \text{Pos}\left(\mathcal{X} \otimes \mathcal{Y} \otimes \mathcal{Y}_2 \otimes \ldots \otimes \mathcal{Y}_s\right), \\
& \vdots \\
& \text{T}_{\mathcal{Y}_s}(X_k) \in \text{Pos}\left(\mathcal{X} \otimes \mathcal{Y} \otimes \mathcal{Y}_2 \otimes \ldots \otimes \mathcal{Y}_s\right), \\
& X_1, \ldots, X_N \in \text{Pos}\left(\mathcal{X} \otimes \mathcal{Y} \otimes \mathcal{Y}_2 \otimes \ldots \otimes \mathcal{Y}_s\right).
\end{aligned}$$

maximize:

$\sum_{k=1}^N p_k \langle \rho_k, \mu(k) \rangle$, subject to:

$\text{Tr}_{\mathcal{Y}_2 \otimes \ldots \otimes \mathcal{Y}_s} (X_k) = \mu(k),$

$\text{T}_{\mathcal{X}}(X_k) \in \text{Pos}\left(\mathcal{X} \otimes \mathcal{Y} \otimes \mathcal{Y}_2 \otimes \ldots \otimes \mathcal{Y}_s\right),$

$\vdots$

$X_1, \ldots, X_N \in \text{Pos}\left(\mathcal{X} \otimes \mathcal{Y} \otimes \mathcal{Y}_2 \otimes \ldots \otimes \mathcal{Y}_s\right).$

**Dual:**

$$\begin{aligned}
\text{minimize:} \quad & \text{Tr}(H), \\
\text{subject to:} \quad & H - Q_k \geq p_k \rho_k, \\
& Q_k \otimes \mathbb{I}_{\mathcal{Y}_2 \otimes \ldots \otimes \mathcal{Y}_s} + \left(\mathbb{I}_{\mathcal{X}} \otimes \Pi_{\mathcal{Y}\mathcal{Y}_2 \ldots \mathcal{Y}_{\int}}\right) R_k \left(\mathbb{I}_{\mathcal{X}} \otimes \Pi_{\mathcal{Y}\mathcal{Y}_2 \ldots \mathcal{Y}_{\int}}\right) \\
& \quad - R_k - \text{T}_{\mathcal{X}}(S_k) - \text{T}_{\mathcal{Y}}(Z_k) \in \text{Pos}(\mathcal{X} \otimes \mathcal{Y} \otimes \mathcal{Y}_2 \otimes \ldots \otimes \mathcal{Y}_s) \\
& H, Q_1, \ldots, Q_N \in \text{Herm}(\mathcal{X} \otimes \mathcal{Y}), \\
& R_1, \ldots R_N \in \text{Herm}(\mathcal{X} \otimes \mathcal{Y} \otimes \mathcal{Y}_2 \otimes \ldots \otimes \mathcal{Y}_s), \\
& S_1, \ldots, S_N, Z_1, \ldots, Z_N \in \text{Pos}(\mathcal{X} \otimes \mathcal{Y} \otimes \mathcal{Y}_2 \otimes \ldots \otimes \mathcal{Y}_s).
\end{aligned}$$

minimize:

$\text{Tr}(H)$, subject to:

$Q_k \otimes \mathbb{I}_{\mathcal{Y}_2 \otimes \ldots \otimes \mathcal{Y}_s} + \left(\mathbb{I}_{\mathcal{X}} \otimes \Pi_{\mathcal{Y}\mathcal{Y}_2 \ldots \mathcal{Y}_{\int}}\right) R_k \left(\mathbb{I}_{\mathcal{X}} \otimes \Pi_{\mathcal{Y}\mathcal{Y}_2 \ldots \mathcal{Y}_{\int}}\right)$
$H, Q_1, \ldots, Q_N \in \text{Herm}(\mathcal{X} \otimes \mathcal{Y})$,
$S_1, \ldots, S_N, Z_1, \ldots, Z_N \in \text{Pos}(\mathcal{X} \otimes \mathcal{Y} \otimes \mathcal{Y}_2 \otimes \ldots \otimes \mathcal{Y}_s)$.

## 13.2 Entanglement cost of distinguishing Bell states

One may ask whether the ability to distinguish a state can be improved by making use of an auxiliary resource state.

$$|\tau_\epsilon\rangle = \sqrt{\frac{1+\epsilon}{2}}|00\rangle + \sqrt{\frac{1-\epsilon}{2}}|11\rangle,$$

for some $\epsilon \in [0, 1]$.

### 13.2.1 Distinguishing four Bell states

It was shown in [BCJRWY15] that the probability of distinguishing four Bell states with a resource state via separable measurements is given by the closed-form expression:

$$\text{opt}_{\text{PPT}}(\eta) = \text{opt}_{\text{SEP}}(\eta) = \frac{1}{2}\left(1 + \sqrt{1-\epsilon^2}\right)$$

where the ensemble is defined as

$$\eta = \left\{ |\psi_0\rangle \otimes |\tau_\epsilon\rangle, |\psi_1\rangle \otimes |\tau_\epsilon\rangle, |\psi_2\rangle \otimes |\tau_\epsilon\rangle, |\psi_3\rangle \otimes |\tau_\epsilon\rangle \right\}.$$

Using `qustop`, we may encode this scenario as follows.

```python
import numpy as np
from toqito.states import basis, bell

from qustop import Ensemble, OptDist, State

e_0, e_1 = basis(2, 0), basis(2, 1)

eps = 0.5
tau = np.sqrt((1 + eps) / 2) * np.kron(e_0, e_0) + np.sqrt(
    (1 - eps) / 2
) * np.kron(e_1, e_1)

dims = [2, 2, 2, 2]
ensemble = Ensemble(
    [
        State(np.kron(bell(0), tau), dims),
        State(np.kron(bell(1), tau), dims),
        State(np.kron(bell(2), tau), dims),
        State(np.kron(bell(3), tau), dims),
    ],
    [1 / 4, 1 / 4, 1 / 4, 1 / 4],
)

ppt_res = OptDist(ensemble, "ppt", "min-error")
ppt_res.solve()

sep_res = OptDist(ensemble, "sep", "min-error")
sep_res.solve()

eq = 1 / 2 * (1 + np.sqrt(1 - eps ** 2))

# 0.9330127018922193
print(eq)
# 0.9330127016540999
print(ppt_res.value)
# 0.9330127016540999
print(sep_res.value)
```

Note that [BCJRWY15] also proved the same closed-form expression for when Alice and Bob make use of PPT measurements (which is an upper bound for separable measurements). More on that in the tutorial on distinguishing via PPT measurements.

### 13.2.2 Distinguishing three Bell states

It was also shown in [BCJRWY15] that the closed-form probability of distinguishing three Bell states with a resource state using separable measurements to be given by the closed-form expression:

$$\text{opt}_{\text{SEP}}(\eta) = \frac{1}{3}\left(2 + \sqrt{1 - \epsilon^2}\right)$$

where the ensemble is defined as

$$\eta = \left( |\psi_0\rangle \otimes |\tau_\epsilon\rangle, |\psi_1\rangle \otimes |\tau_\epsilon\rangle, |\psi_2\rangle \otimes |\tau_\epsilon\rangle \right).$$

Using qustop, we may encode this scenario as follows.

```python
import numpy as np
from toqito.states import basis, bell

from qustop import Ensemble, OptDist, State

e_0, e_1 = basis(2, 0), basis(2, 1)

eps = 0.5
tau = np.sqrt((1 + eps) / 2) * np.kron(e_0, e_0) + np.sqrt(
    (1 - eps) / 2
) * np.kron(e_1, e_1)

dims = [2, 2, 2, 2]
states = [
    State(np.kron(bell(0), tau), dims),
    State(np.kron(bell(1), tau), dims),
    State(np.kron(bell(2), tau), dims),
]
probs = [1 / 3, 1 / 3, 1 / 3]
ensemble = Ensemble(states, probs)
ensemble.swap([2, 3])

sep_res = OptDist(ensemble, "sep", "min-error", level=2)
sep_res.solve()

eq = 1 / 3 * (2 + np.sqrt(1 - eps ** 2))

# 0.9553418012614794
print(eq)

# 0.9583970406126399
print(sep_res.value)
```

Note that the value of `sep_res.value` is actually a bit higher than `eq`. This is because the separable value is calculated by a hierarchy of SDPs. At low levels of the SDP, the problem can often converge to the optimal value, but other times it is necessary to compute higher levels of the SDP to eventually arrive at the optimal value. While this is intractable in general, in practice, the SDP can often converge, or at least get fairly close to the optimal value for small problem sizes.

## 13.3 Unextendible product bases and separable measurements

For complex Euclidean spaces $\mathcal{X}$ and $\mathcal{Y}$, an *unextendable product basis* is defined as an orthonormal collection of vectors

$$\mathcal{U} = \{u_1 \otimes v_1, \ldots, u_m \otimes v_m\} \subset \mathcal{X} \otimes \mathcal{Y},$$

for unit vectors $u_1, \ldots, u_m \in \mathcal{X}$ and $v_1, \ldots, v_m \in \mathcal{Y}$ where:

1. $m <$ dim$(\mathcal{X} \otimes \mathcal{Y})$.

2. For every $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ satisfying $x \otimes y \perp \mathcal{U}$ it holds that $x \otimes y = 0$.

All UPBs are known to be indistinguishable by LOCC measurements and all UPBs are distinguishable by PPT measurements. As separable measurements lie in between LOCC and PPT measurements, it is of interest to know which UPBs are distinguishable by separable measurements.

For instance, in [DMSST99], it was shown that all UPBs in $\mathbb{C}^3 \otimes \mathbb{C}^3$ are perfectly distinguishable via separable measurements.

Consider the "Tiles" UPB

$$\begin{aligned}
|\phi_0\rangle &= |0\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right), & |\phi_1\rangle &= |2\rangle \left(\frac{|1\rangle - |2\rangle}{\sqrt{2}}\right), \\
|\phi_2\rangle &= \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) |2\rangle, & |\phi_3\rangle &= \left(\frac{|1\rangle - |2\rangle}{\sqrt{2}}\right) |0\rangle, \\
|\phi_4\rangle &= \frac{1}{3} \left(|0\rangle + |1\rangle + |2\rangle\right), \left(|0\rangle + |1\rangle + |2\rangle\right).
\end{aligned}$$

$$|\phi_3\rangle = \left(\frac{|1\rangle - |2\rangle}{\sqrt{2}}\right) |0\rangle, |\phi_4\rangle = \frac{1}{3} \left(|0\rangle + |1\rangle + |2\rangle\right), \left(|0\rangle + |1\rangle + |2\rangle\right).$$

Note that the "Tiles" states are contained in $\mathbb{C}^3 \otimes \mathbb{C}^3$. We can use `qustop` to indeed verify that these states are perfectly distinguishable via separable measurements.

```python
from toqito.states import basis, tile

from qustop import Ensemble, OptDist, State

# Construct the corresponding density matrices of the Tiles UPB.
```

(continues on next page)

(continued from previous page)

```
dims = [3, 3]
ensemble = Ensemble(
    [
        State(tile(0) * tile(0).conj().T, dims),
        State(tile(1) * tile(1).conj().T, dims),
        State(tile(2) * tile(2).conj().T, dims),
        State(tile(3) * tile(3).conj().T, dims),
        State(tile(4) * tile(4).conj().T, dims),
    ]
)
res = OptDist(ensemble, "sep", "min-error", level=2)
res.solve()

# 0.999999
print(res.value)
```

In [BCJRWY15], it was shown that the 8-state UPB contained in $\mathbb{C}^4 \otimes \mathbb{C}^4$ introduced in [Feng06] defined as

$$\begin{array}{ll}
|\phi_1\rangle = |0\rangle|0\rangle, & |\phi_5\rangle = \left(|1\rangle + |2\rangle + |3\rangle\right)\left(|0\rangle - |1\rangle + |2\rangle\right)/3, \\
|\phi_2\rangle = |1\rangle\left(|0\rangle - |2\rangle + |3\rangle\right)/\sqrt{3}, & |\phi_6\rangle = \left(|0\rangle - |2\rangle + |3\rangle\right)|2\rangle/\sqrt{3}, \\
|\phi_3\rangle = |2\rangle\left(|0\rangle + |1\rangle - |3\rangle\right)/\sqrt{3}, & |\phi_7\rangle = \left(|0\rangle + |1\rangle - |3\rangle\right)|1\rangle/\sqrt{3}, \\
|\phi_4\rangle = |3\rangle|3\rangle, & |\phi_8\rangle = \left(|0\rangle - |1\rangle + |2\rangle\right)\left(|1\rangle + |2\rangle + |3\rangle\right)/3.
\end{array}$$

$|\phi_6\rangle = \left(|0\rangle - |2\rangle + |3\rangle\right)|2\rangle/\sqrt{3}, |\phi_3\rangle = |2\rangle\left(|0\rangle + |1\rangle - |3\rangle\right)/\sqrt{3},$
$|\phi_8\rangle = \left(|0\rangle - |1\rangle + |2\rangle\right)\left(|1\rangle + |2\rangle + |3\rangle\right)/3.$

is not perfectly distinguishable via separable measurements. This can be observed using qustop as follows.

```
import numpy as np
from toqito.states import basis

from qustop import Ensemble, OptDist, State

e_0, e_1, e_2, e_3 = basis(4, 0), basis(4, 1), basis(4, 2), basis(4, 3)

phi_1 = np.kron(e_0, e_0)
phi_2 = np.kron(e_1, (e_0 - e_2 + e_3) / np.sqrt(3))
phi_3 = np.kron(e_2, (e_0 + e_1 - e_3) / np.sqrt(3))
phi_4 = np.kron(e_3, e_3)
phi_5 = np.kron((e_1 + e_2 + e_3), (e_0 - e_1 + e_2) / 3)
phi_6 = np.kron((e_0 - e_2 + e_3), e_2 / np.sqrt(3))
phi_7 = np.kron((e_0 + e_1 - e_3), e_1 / np.sqrt(3))
phi_8 = np.kron((e_0 - e_1 + e_2), (e_1 + e_2 + e_3) / 3)

```

(continues on next page)

(continued from previous page)

```python
dims = [4, 4]
ensemble = Ensemble(
    [
        State(phi_1, dims),
        State(phi_2, dims),
        State(phi_3, dims),
        State(phi_4, dims),
        State(phi_5, dims),
        State(phi_6, dims),
        State(phi_7, dims),
        State(phi_8, dims),
    ]
)
res = OptDist(ensemble, "sep", "min-error", level=2)
res.solve()

# 0.9967296337698935
print(res.value)
```

### 13.3.1 Impossibility to distinguish a UPB plus one extra pure state

It was shown in [BCJRWY15] that if we consider an ensemble of states consisting of a UPB along with a pure state that is orthogonal to all states in said ensemble, then it is impossible to perfectly distinguish this ensemble.

$$\begin{aligned}
|\phi_0\rangle &= |0\rangle \left(\tfrac{|0\rangle - |1\rangle}{\sqrt{2}}\right), & |\phi_1\rangle &= |2\rangle \left(\tfrac{|1\rangle - |2\rangle}{\sqrt{2}}\right), \\
|\phi_2\rangle &= \left(\tfrac{|0\rangle - |1\rangle}{\sqrt{2}}\right) |2\rangle, & |\phi_3\rangle &= \left(\tfrac{|1\rangle - |2\rangle}{\sqrt{2}}\right) |0\rangle, \\
|\phi_4\rangle &= \tfrac{1}{3} \left(|0\rangle + |1\rangle + |2\rangle\right) \left(|0\rangle + |1\rangle + |2\rangle\right), & |\phi_5\rangle &= \tfrac{1}{2} \left(|0\rangle|0\rangle + |0\rangle|1\rangle - |0\rangle|2\rangle - |1\rangle|2\rangle\right)
\end{aligned}$$

$$|\phi_3\rangle = \left(\frac{|1\rangle - |2\rangle}{\sqrt{2}}\right) |0\rangle, |\phi_4\rangle = \frac{1}{3} \left(|0\rangle + |1\rangle + |2\rangle\right) \left(|0\rangle + |1\rangle + |2\rangle\right),$$

```python
import numpy as np
from toqito.states import basis, tile

from qustop import Ensemble, OptDist, State

dims = [3, 3]
e_0, e_1, e_2 = basis(3, 0), basis(3, 1), basis(3, 2)

# Define a pure state to add to the ensemble.
psi = (
    1
```

(continues on next page)

(continued from previous page)

```
    / 2
    * (
        np.kron(e_0, e_0)
        + np.kron(e_0, e_1)
        - np.kron(e_0, e_2)
        - np.kron(e_1, e_2)
    )
)

# Construct the corresponding density matrices of the Tiles UPB + pure state:
ensemble = Ensemble(
    [
        State(tile(0) * tile(0).conj().T, dims),
        State(tile(1) * tile(1).conj().T, dims),
        State(tile(2) * tile(2).conj().T, dims),
        State(tile(3) * tile(3).conj().T, dims),
        State(tile(4) * tile(4).conj().T, dims),
        State(psi * psi.conj().T, dims),
    ]
)

res = OptDist(ensemble, "sep", "min-error", level=2)
res.solve()

# 0.9860588510298623
print(res.value)
```

## 13.4 References

CHAPTER

FOURTEEN

# QUANTUM STATE EXCLUSION

In this tutorial, we are going to cover the problem of *quantum state exclusion*. We are going to briefly describe the problem setting and then describe how one may use `qustop` to calculate the optimal probability with which this problem can be solved for a number of different scenarios.

Quantum state exclusion is very closely related to the problem of quantum state distinguishability. It may be useful to consult the following tutorial that covers quantum state distinguishability:

- Quantum State Distinguishability

Further information beyond the scope of this tutorial can be found in the text [tPBR12] as well as the course [tBJOP14].

## 14.1 The state exclusion problem

The quantum state exclusion problem is phrased as follows.

1. Alice possesses an ensemble of $n$ quantum states:

$$\eta = \left( (p_0, \rho_0), \ldots, (p_n, \rho_n) \right),$$

where $p_i$ is the probability with which state $\rho_i$ is selected from the ensemble. Alice picks $\rho_i$ with probability $p_i$ from her ensemble and sends $\rho_i$ to Bob.

2. Bob receives $\rho_i$. Both Alice and Bob are aware of how the ensemble is defined but he does *not* know what index $i$ corresponding to the state $\rho_i$ he receives from Alice is.
3. Bob wants to guess which of the states from the ensemble he was *not* given. In order to do so, he may measure $\rho_i$ to guess the index $i$ for which the state in the ensemble corresponds.

This setting is depicted in the following figure.

Fig. 1: The quantum state exclusion setting.

**Note:** The primary difference between the quantum state distinguishability scenario and the quantum state exclusion scenario is that in the former, Bob want to guess which state he was given, and in the latter, Bob wants to guess which state he was *not* given.

### 14.1.1 Optimal probability of conclusively excluding a quantum state

(TODO)

### 14.1.2 Optimal probability of unambiguously excluding a quantum state

(TODO)

## 14.2 References

CHAPTER

FIFTEEN

# QUANTUM STATE CLONING

In this tutorial, we are going to cover the problem of *quantum state cloning*.

# CHAPTER SIXTEEN

# OPEN PROBLEMS IN QUANTUM STATE OPTIMIZATION

This page consists of certain problems in the domain of quantum state optimization that may be framed as computational tasks using the `qustop` module. Most of these examples are provided as brute-force computational approaches. It may be possible to refine many of these approaches in a more sophisticated manner, and I welcome any such input and feedback from the community.

## 16.1 Two-copy problem

### 16.1.1 Problem statement:

Let $n \geq 1$ be an integer, let $\mathcal{X}$ be a compelx Euclidean space, let $\rho_i \in \mathrm{D}(\mathcal{X})$ be a pure quantum state represented as a density operator and let

$$\eta = \left\{ \left( \frac{1}{n}, \rho_1 \right), \ldots, \left( \frac{1}{n}, \rho_n \right) \right\} \subset \mathcal{X}$$

be an ensemble of pure and mutually orthogonal quantum states. Define $\eta^{\otimes 2}$ as the two-copy ensemble where

$$\eta^{\otimes 2} = \left\{ \left( \frac{1}{n}, \rho_1 \otimes \rho_1 \right), \ldots, \left( \frac{1}{n}, \rho_n \otimes \rho_n \right) \right\} \subset \mathcal{X} \otimes \mathcal{X}.$$

**Question**: Does there exist a certain ensemble $\eta^{\otimes 2}$ such that

$$\mathrm{opt}_{\mathrm{PPT}} \left( \eta^{\otimes 2} \right) < 1 \quad \text{or} \quad \mathrm{opt}_{\mathrm{SEP}} \left( \eta^{\otimes 2} \right) < 1?$$

### 16.1.2 Computational approach:

The computational approach here is to generate a random ensemble of quantum states and then construct two copies of that ensemble. We take the two-copy ensemble and check whether or not we get a value less than 1 for distinguishing via PPT measurements.

```python
import pickle

import numpy as np
from scipy.stats import unitary_group

from qustop import Ensemble, OptDist, State


```

(continues on next page)

(continued from previous page)

```python
def generate_random_two_copy_ensemble(num_states: int) -> Ensemble:
    # Generate an ensemble of random mutually orthogonal and pure states.
    # In this case, the ensemble consists of four states, but we could
    # potentially have any n >= 1 number of states in the ensemble.
    group = unitary_group.rvs(num_states)
    states = [np.atleast_2d(group[:, i]).T for i in range(num_states)]

    # Create the two-copy ensemble.
    dims = [2] * num_states
    ensemble = [
        State(np.kron(state, state), dims) for _, state in enumerate(states)
    ]
    return Ensemble(ensemble)


def run(num_states: int, num_trials: int) -> None:
    for test in range(num_trials):
        ensemble_2_copies = generate_random_two_copy_ensemble(num_states)

        # Solve the two-copy PPT distinguishability SDPs.
        ppt_2_copy = OptDist(ensemble_2_copies, "ppt", "min-error")
        ppt_2_copy.solve()

        # If the PPT value of the two-copy ensemble is below some threshold
        # of perfect distinguishability, such an example has been found, in
        # which case, we want to ensure we capture the values and states!
        if not np.isclose(ppt_2_copy.value, 1, atol=0.001):
            print(f"PPT 2-copy: {ppt_2_copy.value}")
            with open("FOUND.pickle", "wb") as f:
                pickle.dump([ensemble_2_copies], f)
            break

        # In any case, print out the two-copy values of the ensembles
        # as we progress through the trials.
        print(f"PPT 2-copy: {ppt_2_copy.value}")
```

One can specify the num_states parameter to determine how many states are in the ensemble and can also modify the num_trials parameter to specify for how many random ensembles should be generated and checked.

For instance, here we can randomly generate a 4-state two-copy ensemble 5 different times:

```
>>> run(num_states=4, num_trials=5)
PPT 2-copy: 0.9999999997183334
PPT 2-copy: 1.0000000011197347
PPT 2-copy: 1.000000005640175
PPT 2-copy: 1.0000000003272558
PPT 2-copy: 0.9999999981951567
```

## 16.2 Entanglement cost of distinguishing four arbitrary two-qubit ensemble with resource state

### 16.2.1 Problem statement:

Let $\mathbb{C}^2 = \mathcal{X}_1 = \mathcal{X}_2 = \mathcal{Y}_1 = \mathcal{Y}_2$ be complex Euclidean spaces and let $\mathcal{X} = \mathcal{X}_1 \otimes \mathcal{X}_2$ and $\mathcal{Y} = \mathcal{Y}_1 \otimes \mathcal{Y}_2$. Define the following orthogonal two-qubit basis

$$|\psi_0\rangle = \alpha|00\rangle + \beta|11\rangle, \quad |\psi_1\rangle = \beta|00\rangle - \alpha|11\rangle,$$
$$|\psi_2\rangle = \alpha|01\rangle + \beta|10\rangle, \quad |\psi_3\rangle = \beta|01\rangle - \alpha|10\rangle,$$

where $|\psi_i\rangle \in \mathcal{X}_1 \otimes \mathcal{Y}_1$ for all $i \in [0, 1, 2, 3]$ and where

$$\alpha = \sqrt{\frac{1+n}{2}} \quad \text{and} \quad \beta = \sqrt{\frac{1-n}{2}}.$$

Define the resource state

$$|\tau_\epsilon\rangle = \sqrt{\frac{1+\epsilon}{2}}|00\rangle + \sqrt{\frac{1-\epsilon}{2}}|11\rangle \in \mathcal{X}_2 \otimes \mathcal{Y}_2$$

for some choice of $\epsilon \in [0, 1]$.

Consider the ensemble

$$\eta = \{|\psi_0\rangle \otimes |\tau_\epsilon\rangle, |\psi_1\rangle \otimes |\tau_\epsilon\rangle, |\psi_2\rangle \otimes |\tau_\epsilon\rangle, |\psi_3\rangle \otimes |\tau_\epsilon\rangle\} \subset \mathcal{X} \otimes \mathcal{Y}.$$

**Question**: Assuming a uniform distribution $p_1 = p_2 = p_3 = p_4 = 1/4$, with any state from $\eta$ being selected, what is the closed-form entanglement cost of distinguishing $\eta$ via PPT measurements for any choice of $\alpha$ and $\beta$?

**Note:** When $\alpha = \beta = \frac{1}{\sqrt{2}}$, the ensemble $\eta$ consists of the Bell states. In this case, it is known that the closed-form entanglement cost is

$$\frac{1}{2}\left(1 + \sqrt{1-\epsilon^2}\right).$$

**Follow-up Question**: Assuming a uniform distribution $p_1 = p_2 = p_3 = p_4 = 1/4$, with any state from $\eta$ being selected, what is the closed-form entanglement cost of distinguishing $\eta$ via *separable* measurements for any choice of $\alpha$ and $\beta$?

This question will most likely be *more* difficult than the PPT case, but it's still worth asking.

### 16.2.2 Computational approach:

Consider the following snippet of code that considers this problem:

```python
import numpy as np
from toqito.states import basis

from qustop import Ensemble, OptDist, State

# Define the |0> and |1> basis states:
e_0, e_1 = basis(2, 0), basis(2, 1)

# Define constants "n" and "epsilon":
n = 0.5
eps = 0.0

# Parameters alpha and beta are defined in terms of "n"
alpha, beta = np.sqrt((1 + n) / 2), np.sqrt((1 - n) / 2)

# Define the two-qubit ensemble states:
psi_0 = alpha * np.kron(e_0, e_0) + beta * np.kron(e_1, e_1)
psi_1 = beta * np.kron(e_0, e_0) - alpha * np.kron(e_1, e_1)
psi_2 = alpha * np.kron(e_0, e_1) + beta * np.kron(e_1, e_0)
psi_3 = beta * np.kron(e_0, e_1) - alpha * np.kron(e_1, e_0)

# Define the resource state:
tau_state = np.sqrt((1 + eps) / 2) * np.kron(e_0, e_0) + np.sqrt(
    (1 - eps) / 2
) * np.kron(e_1, e_1)
tau = tau_state * tau_state.conj().T

# Create the ensemble to distinguish:
dims = [2, 2, 2, 2]
rho_0 = State(np.kron(psi_0 * psi_0.conj().T, tau), dims)
rho_1 = State(np.kron(psi_1 * psi_1.conj().T, tau), dims)
rho_2 = State(np.kron(psi_2 * psi_2.conj().T, tau), dims)
rho_3 = State(np.kron(psi_3 * psi_3.conj().T, tau), dims)
ensemble = Ensemble([rho_0, rho_1, rho_2, rho_3])

# Determine the optimal value of distinguishing the ensemble via PPT
# measurements:
ppt_res = OptDist(ensemble, "ppt", "min-error")
ppt_res.solve()

# Print value of "n", "eps", and the optimal value of distinguishing via PPT
# measurements:
print(f"For n = {n} and eps={eps}, the PPT value is {ppt_res.value}")
```

Is it possible to determine a closed form of the optimal value of distinguishing via PPT measurements for any choice of `n` and `eps`? It may be illuminating to alter these values and see if any obvious closed-form expressions emerge.

It is also possible to alter the `ppt` argument to `sep` to get a sense on the follow-up question. However, keep in mind that the computational technique used to compute separable values is an upper bound–a hierarchy of SDPs that will eventually converge, but may take substantial computational resources and time to do so.

## 16.3 Entanglement cost of distinguishing three arbitrary two-qubit ensemble with resource state

### 16.3.1 Problem statement:

Let $\mathbb{C}^2 = \mathcal{X}_1 = \mathcal{X}_2 = \mathcal{Y}_1 = \mathcal{Y}_2$ be complex Euclidean spaces and let $\mathcal{X} = \mathcal{X}_1 \otimes \mathcal{X}_2$ and $\mathcal{Y} = \mathcal{Y}_1 \otimes \mathcal{Y}_2$. Define the following orthogonal two-qubit basis

$$\begin{aligned}|\psi_0\rangle = \alpha|00\rangle + \beta|11\rangle, \quad |\psi_1\rangle = \beta|00\rangle - \alpha|11\rangle,\\ |\psi_2\rangle = \alpha|01\rangle + \beta|10\rangle,\end{aligned}$$

where $|\psi_i\rangle \in \mathcal{X}_1 \otimes \mathcal{Y}_1$ for all $i \in [0, 1, 2]$ and where

$$\alpha = \sqrt{\frac{1+n}{2}} \quad \text{and} \quad \beta = \sqrt{\frac{1-n}{2}}.$$

Define the resource state

$$|\tau_\epsilon\rangle = \sqrt{\frac{1+\epsilon}{2}}|00\rangle + \sqrt{\frac{1-\epsilon}{2}}|11\rangle \in \mathcal{X}_2 \otimes \mathcal{Y}_2$$

for some choice of $\epsilon \in [0, 1]$.

Consider the ensemble

$$\begin{aligned}\eta = \{|\psi_0\rangle \otimes |\tau_\epsilon\rangle, \quad |\psi_1\rangle \otimes |\tau_\epsilon\rangle,\\ |\psi_2\rangle \otimes |\tau_\epsilon\rangle, \subset \mathcal{X} \otimes \mathcal{Y}.\end{aligned}$$

**Question**: Assuming a uniform distribution $p_1 = p_2 = p_3 = 1/3$, with any state from $\eta$ being selected, what is the closed-form entanglement cost of distinguishing $\eta$ via separable measurements for any choice of $\alpha$ and $\beta$?

**Note:** When $\alpha = \beta = \frac{1}{\sqrt{2}}$, the ensemble $\eta$ consists of three out of the four the Bell states. In this case, it is known that the closed-form entanglement cost for separable measurements is

$$\frac{1}{3}\left(2 + \sqrt{1-\epsilon^2}\right).$$

A simpler version of this question is to determine $\text{opt}_{\text{PPT}}(\eta)$ for $n = 0$ for any $\epsilon$.

Since $\text{opt}_{\text{SEP}}(\eta) < \text{opt}_{\text{PPT}}(\eta)$, we have a lower bound on $\text{opt}_{\text{PPT}}(\eta)$ when $n = 0$. It also holds that $\text{opt}_{\text{POS}}(\eta) = 1$ for all $n$ and $\epsilon$, so we have an obvious upper bound.

### 16.3.2 Computational approach:

Consider the following snippet of code that considers this problem:

```python
import numpy as np
from toqito.states import basis

from qustop import Ensemble, OptDist, State

# Define the |0> and |1> basis states:
e_0, e_1 = basis(2, 0), basis(2, 1)

# Define constants "n" and "epsilon":
n = 0.0
eps = 0.75

# Parameters alpha and beta are defined in terms of "n"
alpha, beta = np.sqrt((1 + n) / 2), np.sqrt((1 - n) / 2)

# Define the two-qubit ensemble states:
psi_0 = alpha * np.kron(e_0, e_0) + beta * np.kron(e_1, e_1)
psi_1 = beta * np.kron(e_0, e_0) - alpha * np.kron(e_1, e_1)
psi_2 = alpha * np.kron(e_0, e_1) + beta * np.kron(e_1, e_0)

# Define the resource state:
tau_state = np.sqrt((1 + eps) / 2) * np.kron(e_0, e_0) + np.sqrt(
    (1 - eps) / 2
) * np.kron(e_1, e_1)
tau = tau_state * tau_state.conj().T

# Create the ensemble to distinguish:
dims = [2, 2, 2, 2]
rho_0 = State(np.kron(psi_0 * psi_0.conj().T, tau), dims)
rho_1 = State(np.kron(psi_1 * psi_1.conj().T, tau), dims)
rho_2 = State(np.kron(psi_2 * psi_2.conj().T, tau), dims)
ensemble = Ensemble([rho_0, rho_1, rho_2])

# Determine the optimal value of distinguishing the ensemble via PPT
# measurements:
ppt_res = OptDist(ensemble, "ppt", "min-error")
ppt_res.solve()

# Print value of "n", "eps", and the optimal value of distinguishing via PPT
# measurements:
print(f"For n = {n} and eps={eps}, the PPT value is {ppt_res.value}")
sep_eq = 1 / 3 * (2 + np.sqrt(1 - eps ** 2))
eq = 1 / 3 * (2 + np.sqrt(1.219 - eps ** 2))
print(f"EQ:{eq}")
print(f"SEP_EQ: {sep_eq}")
```

## 16.4 Separation between LOCC and separable value

### 16.4.1 Problem statement:

Does there exist an ensemble, $\eta$, fully composed of orthogonal maximally entangled states for which the following strict inequality holds:

$$\text{opt}_{\text{LOCC}}(\eta) < \text{opt}_{\text{SEP}}(\eta)?$$

The mathematical structure of LOCC is difficult to generalize, but it is possible to computationally analyze the separable case via a hierarchy of semidefinite programs.

## 16.5 The antidistinguishability conjecture

A set of $n$ pure quantum states $|\rho_1\rangle, \ldots, |\rho_n\rangle$ are *antidistinguishable* if there exists an $n$-outcome measurement that never outputs the result $i$ on $|\rho_i\rangle$.

### 16.5.1 Problem statement:

Let $|\rho_1\rangle, \ldots, |\rho_d\rangle$ be $d$ pure quantum states each of dimension $d$. If

$$|\langle \rho_i | \rho_j \rangle| \leq (d-2)/(d-1)$$

for all $i \neq j$, then the states are antidistinguishable.

**Question**: Does this conjecture hold for all integers $d$?

**Note:** This conjecture is known to hold when $d = 2$ and $d = 3$.

### 16.5.2 Computational approach:

We can randomly generate an ensemble of $d$ pure states of dimension $d$ using the following function.

X

Next, we check whether the states are antidistinguishable. This is equivalent to checking whether the optimal value of the semidefinite program for the problem for the problem of unambiguous quantum state exclusion from arXiv:1306.4683 yields a non-zero result.

If the states *are* antidistinguishable, check the bound in the conjecture:

- If the bound is satisfied it implies a satisfaction of the conjecture, so we go back to the beginning and try to generate another random ensemble.
- If the bound is *not* satisfied, this implies a violation of the conjecture.

# BIBLIOGRAPHY

[WatrousQI] Watrous, John “The theory of quantum information” Section: “A semidefinite program for optimal measurements” Cambridge University Press, 2018

[SikoraSDP] Sikora, Jamie “Semidefinite programming in quantum theory (lecture series)” Lecture 2: Semidefinite programs for nice problems and popular functions Perimeter Institute for Theoretical Physics, 2019

[tWalgate00] Walgate, J., Short, A. J., Hardy, L., & Vedral, V “Local distinguishability of multipartite orthogonal quantum states.” Physical Review Letters 85.23 (2000): 4972.

[TDL01] Terhal, Barbara M., David P. DiVincenzo, and Debbie W. Leung. “Hiding bits in Bell states.” Physical review letters 86.25 (2001): 5807.

[DLT02] DiVincenzo, David P., Debbie W. Leung, and Barbara M. Terhal. “Quantum data hiding.” IEEE Transactions on Information Theory 48.3 (2002): 580-598.

[Cosentino15] Cosentino, Alessandro “Quantum state local distinguishability via convex optimization”. University of Waterloo, Thesis https://uwspace.uwaterloo.ca/handle/10012/9572

[Cosentino13] Cosentino, Alessandro, “Positive-partial-transpose-indistinguishable states via semidefinite programming”, Physical Review A 87.1 (2013): 012321. https://arxiv.org/abs/1205.1031

[CR13] Cosentino, Alessandro and Russo, Vincent “Small sets of locally indistinguishable orthogonal maximally entangled states”, Quantum Information & Computation, Volume 14, https://arxiv.org/abs/1307.3232

[YDY12] Yu, Nengkun, Runyao Duan, and Mingsheng Ying. “Four locally indistinguishable ququad-ququad orthogonal maximally entangled states.” Physical review letters 109.2 (2012): 020506. https://arxiv.org/abs/1107.3224

[BCJRWY15] Bandyopadhyay, Somshubhro, Cosentino, Alessandro, Johnston, Nathaniel, Russo, Vincent, Watrous, John, & Yu, Nengkun. “Limitations on separable measurements by convex optimization”. IEEE Transactions on Information Theory 61.6 (2015): 3593-3604.

[Nav08] Navascués, Miguel. “Pure state estimation and the characterization of entanglement.” Physical review letters 100.7 (2008): 070503. https://arxiv.org/abs/0707.4398

[DMSST99] DiVincenzo, David P., et al. “Unextendible product bases, uncompletable product bases and bound entanglement.” Communications in Mathematical Physics 238.3 (2003): 379-410.

[Feng06] Feng, Keqin. “Unextendible product bases and 1-factorization of complete graphs.” Discrete applied mathematics 154.6 (2006): 942-949.

[tPBR12] Pusey, Matthew, Barret, Jonathan, and Rudolph, Terry “On the reality of the quantum state” Nature Physics 8.6 (2012): 475-478. arXiv:1111.3328

[tBJOP14] Bandyopadhyay, Somshubhro, Jain, Rahul, Oppenheim, Jonathan, Perry, Christopher “Conclusive exclusion of quantum states” Physical Review A 89.2 (2014): 022336. arXiv:1306.4683