Pooling Strategy Optimization for Accelerating Asymptomatic COVID-19 Screening

Keqin Li*

Department of Computer Science
State University of New York
New Paltz, New York 12561, USA

Email: lik@newpaltz.edu

*The author can be reached at phone: (845) 257-3534, fax: (845) 257-3996.
Abstract

Testing has been a major factor that limits our response to the COVID-19 pandemic. The method of sample pooling and group test has recently been introduced. However, it is still not clearly known how to determine the appropriate group size. In this paper, we develop an analytical method and a numerical algorithm to determine the optimal group size, which minimizes the total number of tests, maximizes the speedup of the pooling strategy, and minimizes both time and cost of testing. The optimal group size is determined by the fraction of infected people and independent of the size of the population. Furthermore, both the optimal pooling size and the achieved speedup grow exponentially with the reciprocal of the fraction of infected people, a quite impressive and nontrivial result. Our method is effective in supporting faster and cheaper asymptomatic COVID-19 screening. Our research has important social implications and financial impacts. For example, if the percentage of infected people is 0.001, we can achieve speedup of almost 16, which means that months of testing time can be reduced to days, and over 93% of the testing cost can be saved. Such a result has not been available in the known literature, and is a significant progress and great advance in pooling strategy optimization for accelerating asymptomatic COVID-19 screening.

Keywords: asymptomatic screening, COVID-19, group test, pooling strategy optimization, sample pooling.
1 Introduction

1.1 Background

A coronavirus test requires a number of time consuming steps in the laboratory, which can take several hours. Testing has been a major factor that limits our response to the COVID-19 pandemic [1]. As governments reopen more businesses and public spaces, the number of infected people will surge, especially when there are asymptomatic people [2].

The method of sample pooling and group test has recently been introduced in [3, 4]. The strategy involves pooling samples from multiple people. If the test result of a group of \( k \) \((k \geq 2)\) samples is negative, we know that all the individual samples are negative. If the test result of a group of samples is positive, then the individual samples need to be tested one by one. If the percentage of infected people is low, this pooling method can potentially significantly reduce the required number of tests and substantially save the necessary cost of tests. For example, recently, the City of Wuhan successfully screened 300 asymptomatic individuals from 9,899,828 people in only 19 days (May 14 – June 1, 2020), by using the pooling method with group size of \( k = 5 \), involving 63 testing laboratories, 1,451 scientists and professionals, and 701 examination equipments (24 hours a day without interruption), and reaching a peak testing capacity of 1 million per day\(^1\).

However, it is still not clearly known how to determine the appropriate group size, although some attempt has been made. For instance, it has been recommended that the batch size should be powers of two [5], which depends on \( p_0 \), the frequency of positive samples out of all samples. It is clear that the choice of the best group size can reduce

\(^1\)http://www.xinhuanet.com/local/2020-06/03/c_1126066386.htm
the time and cost of testing to the maximum extent, and therefore, will have tremendous practical impact to COVID-19 detection, prevention, response, and control.

1.2 Contributions

The contributions of the paper can be summarized as follows.

- We develop an analytical method and a numerical algorithm to determine the optimal group size $k^*$, which minimizes the total number of tests for a population of $N$, maximizes the speedup of the pooling strategy, and minimizes both time and cost of testing.

- It is discovered that the optimal group size is determined by the fraction $p_0$ of infected people and independent of the size of the population. Furthermore, both the optimal pooling size and the achieved speedup grow exponentially with the reciprocal of the fraction of infected people.

- Our research has important social implications and financial impacts. For example, if the percentage of infected people is $p_0 = 0.001$, we can achieve speedup of almost 16, which means that months of testing time can be reduced to days, and over 93% of the testing cost can be saved.

Such a result has not been available in the known literature, and is a significant progress and great advance in pooling strategy optimization for accelerating asymptomatic COVID-19 screening.

In Section 2, we present our theory and develop our procedure. In Section 3, we show numerical data. In Section 4, we conclude the paper.
2 The Method

In this section, we develop our method to find the optimal pooling size.

2.1 Theory

We define the following quantities.

- \( p_0 \): the probability that one individual test result is positive;
- \( q_0 \): the probability that one individual test result is negative;
- \( p_1 \): the probability that one group test result is positive;
- \( q_1 \): the probability that one group test result is negative.

The value of \( p_0 \) is given and known in advance. It is clear that \( q_0 = 1 - p_0 \). Furthermore, we have \( q_1 = q_0^k = (1 - p_0)^k \), and \( p_1 = 1 - q_1 = 1 - q_0^k = 1 - (1 - p_0)^k \).

For a single group of samples, if the test result of the group is negative (which happens with probability \( q_1 \)), only one test is enough; if the test result of the group is positive (which happens with probability \( p_1 \)), \((k + 1)\) tests are required, one for group test, and \( k \) for individual tests. Hence, the expected number of tests for one group using the pooling method is

\[
T_{\text{group}} = 1 \times q_1 + (k + 1) \times p_1 = q_0^k + (k + 1)(1 - q_0^k) = k + 1 - kq_0^k.
\]

The total number of tests for a population of \( N \) (which is divided into \( N/k \) groups) is

\[
T_{\text{pooling}} = \binom{N}{k} T_{\text{group}} = \binom{N}{k} (k + 1 - kq_0^k) = N \left( 1 + \frac{1}{k} - q_0^k \right).
\]
Since the number of tests without using pooling is $N$, the speedup of the pooling strategy is

$$\text{Speedup}(k) = S(k) = \frac{N}{T_{\text{pooling}}} = \frac{1}{\frac{1}{k} - q_0^k}.$$ 

Our objective is to maximize the speedup.

It is clear that maximizing $S(k)$ is equivalent to minimizing

$$F(k) = \frac{1}{k} - q_0^k.$$ 

Note that

$$\frac{\partial F(k)}{\partial k} = -\frac{1}{k^2} - q_0^k \ln q_0.$$ 

To have $\frac{\partial F(k)}{\partial k} = 0$, we need

$$\frac{1}{k^2} = q_0^k \ln \frac{1}{q_0},$$

which implies that $k$ satisfies

$$k = \sqrt[2]{\frac{\left(\frac{1}{q_0}\right)^k}{\ln \frac{1}{q_0}}},$$

that is,

$$k = \sqrt[2]{\frac{\left(\frac{1}{1-p_0}\right)^k}{\ln \left(\frac{1}{1-p_0}\right)}}.$$

Unfortunately, there is no analytical and closed-form solution to the above equation of $k$.

### 2.2 Algorithm

We now develop a numerical algorithm to find $k$. 
We define
\[ G(k) = k - \sqrt{\frac{1}{q_0^2 \ln(1/q_0)}} = k - \frac{1}{\sqrt{\ln(1/q_0)}} \left( \frac{1}{\sqrt{q_0}} \right)^k. \]

Our purpose is to solve the equation \( G(k) = 0 \). Noticed that
\[
\frac{\partial G}{\partial k} = 1 - \frac{1}{\sqrt{\ln(1/q_0)}} \left( \frac{1}{\sqrt{q_0}} \right) \left( \frac{1}{\sqrt{q_0}} \right)^k,
\]
and
\[
\frac{\partial^2 G}{\partial k^2} = -\frac{1}{\sqrt{\ln(1/q_0)}} \left( \frac{1}{\sqrt{q_0}} \right)^2 \left( \frac{1}{\sqrt{q_0}} \right)^k < 0,
\]
which implies that \( G(k) \) is a concave function. Figure 1 illustrates \( G(k) \) for \( q_0 = 0.1 \). It is observed that \( G(k) \) is an increasing function of \( k \) when \( k \) is less than 35, and \( G(k) \) is a decreasing function of \( k \) when \( k \) is greater than 35. In other words, there are two solutions to the equation \( G(k) = 0 \). One is between 3 and 4, and the other is between 54 and 55.

Figure 2 illustrates the speedup \( S(k) \) for \( q_0 = 0.1 \). It is observed that as \( k \) increases, \( S(k) \) increases and reaches its maximum value at \( k = 4 \), and then decreases. However, when \( k \) exceeds 55, \( S(k) \) increases again; however, the increment is very little and not noticeable. Furthermore, the speedup beyond \( k = 55 \) is less than 1, i.e., the pooling method is not effective any more. Therefore, we only need to find the smaller solution of \( k \).

Algorithm 1 gives a complete description of our numerical procedure to find \( k^* \) which satisfies \( G(k^*) = 0 \). Our algorithm is essentially the standard bisection method (lines 1–11), based on the observation is that \( G(k) \) is an increasing function of \( k \) around the smaller solution of \( k \) (lines 5–9). Since the \( k \) found is a real value, we round it to the nearest integers, i.e., \( k_1 = \lfloor k \rfloor \) and \( k_2 = \lceil k \rceil \). The value of \( k^* \) is either \( k_1 \) or \( k_2 \), depending on which one yields the larger speedup (lines 12–15).
Figure 1: $G(k)$ vs. pooling size ($p_0 = 0.1$).
Figure 2: $S(k)$ vs. pooling size ($p_0 = 0.1$).
Algorithm 1: Finding $k$

**Input:** $p_0$.

**Output:** $k^*$ which satisfies $G(k^*) = 0$.

Initialize the search interval of $k$; (1)

**while** (the search interval is not small enough) **do** (2)

$k \leftarrow$ the middle point of the search interval; (3)

Calculate $G(k)$; (4)

**if** ($G(k) < 0$) **then** (5)

Change the search interval to the right half; (6)

**else** (7)

Change the search interval to the left half; (8)

**end if** (9)

**end do**; (10)

$k \leftarrow$ the middle point of the search interval; (11)

$k_1 \leftarrow \lfloor k \rfloor$; (12)

$k_2 \leftarrow \lceil k \rceil$; (13)

$k^* \leftarrow k_1$ or $k_2$, whichever gives the larger speedup; (14)

**return** $k^*$. (15)

3 Numerical Results

In this section, we display some numerical data.

In Figure 3, we show the speedup as a function of the pooling size for $p_0 = 0.001$, 0.002, 0.003, ..., 0.010. It is observed that as $k$ increases, Speedup($k$) increases significantly, especially when $p_0$ is small; however, beyond certain point, Speedup($k$) decreases noticeably. Hence, there is an optimal choice $k^*$, such that the speedup is maximized.
Figure 3: Speedup vs. pooling size ($p_0 = 0.001, 0.002, 0.003, ..., 0.010$).
Table 1: Optimal Pooling Size ($p_0 = 0.001, 0.002, 0.003, \ldots, 0.010$)

<table>
<thead>
<tr>
<th>$p_0$</th>
<th>$k$</th>
<th>$k_1 = \lfloor k \rfloor$</th>
<th>$S(k_1)$</th>
<th>$k_2 = \lceil k \rceil$</th>
<th>$S(k_2)$</th>
<th>$k^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>32.13</td>
<td>32</td>
<td>15.93399</td>
<td>33</td>
<td>15.92853</td>
<td>32</td>
</tr>
<tr>
<td>0.002</td>
<td>22.87</td>
<td>22</td>
<td>11.29398</td>
<td>23</td>
<td>11.30195</td>
<td>23</td>
</tr>
<tr>
<td>0.003</td>
<td>18.77</td>
<td>18</td>
<td>9.24211</td>
<td>19</td>
<td>9.24912</td>
<td>19</td>
</tr>
<tr>
<td>0.004</td>
<td>16.32</td>
<td>16</td>
<td>8.02469</td>
<td>17</td>
<td>8.01986</td>
<td>16</td>
</tr>
<tr>
<td>0.005</td>
<td>14.65</td>
<td>14</td>
<td>7.18399</td>
<td>15</td>
<td>7.18919</td>
<td>15</td>
</tr>
<tr>
<td>0.006</td>
<td>13.42</td>
<td>13</td>
<td>6.57134</td>
<td>14</td>
<td>6.56901</td>
<td>13</td>
</tr>
<tr>
<td>0.007</td>
<td>12.47</td>
<td>12</td>
<td>6.09111</td>
<td>13</td>
<td>6.09023</td>
<td>12</td>
</tr>
<tr>
<td>0.008</td>
<td>11.69</td>
<td>11</td>
<td>5.69891</td>
<td>12</td>
<td>5.70711</td>
<td>12</td>
</tr>
<tr>
<td>0.009</td>
<td>11.06</td>
<td>11</td>
<td>5.38874</td>
<td>12</td>
<td>5.37217</td>
<td>11</td>
</tr>
<tr>
<td>0.010</td>
<td>10.52</td>
<td>10</td>
<td>5.11201</td>
<td>11</td>
<td>5.11324</td>
<td>11</td>
</tr>
</tbody>
</table>

Table 2: Optimal Pooling Size ($p_0 = 10^{-1}, 10^{-2}, 10^{-3}, \ldots, 10^{-7}$)

<table>
<thead>
<tr>
<th>$p_0$</th>
<th>$k$</th>
<th>$k_1 = \lfloor k \rfloor$</th>
<th>$S(k_1)$</th>
<th>$k_2 = \lceil k \rceil$</th>
<th>$S(k_2)$</th>
<th>$k^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-1}$</td>
<td>3.75</td>
<td>3</td>
<td>1.65472</td>
<td>4</td>
<td>1.68379</td>
<td>4</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>10.52</td>
<td>10</td>
<td>5.11201</td>
<td>11</td>
<td>5.11324</td>
<td>11</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>32.13</td>
<td>32</td>
<td>15.93399</td>
<td>33</td>
<td>15.92853</td>
<td>32</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>100.50</td>
<td>100</td>
<td>50.12365</td>
<td>101</td>
<td>50.12366</td>
<td>101</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>316.73</td>
<td>316</td>
<td>158.23823</td>
<td>317</td>
<td>158.23859</td>
<td>317</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>1000.50</td>
<td>1000</td>
<td>500.12486</td>
<td>1001</td>
<td>500.12486</td>
<td>1001</td>
</tr>
<tr>
<td>$10^{-7}$</td>
<td>3162.78</td>
<td>3162</td>
<td>1581.26376</td>
<td>3163</td>
<td>1581.26380</td>
<td>3163</td>
</tr>
</tbody>
</table>

12
In Table 1, we demonstrate the optimal pooling size $k^*$ obtained by our numerical algorithm for $p_0 = 0.001, 0.002, 0.003, \ldots, 0.010$. As $p_0$ becomes smaller, the probability $q_1 = q_0^{k^*}$ that a group test result is negative becomes higher. For instances, when $p_0 = 0.01$, the chance for a negative group test result is $q_0^{11} = 0.99^{11} = 0.8953382$. When $p_0 = 0.001$, the chance for a negative group test result is $q_0^{32} = 0.999^{32} = 0.9684911$. Such higher chance will balance the potential higher cost for individual tests in case a group test result is positive.

In Table 2, we demonstrate the optimal pooling size $k^*$ obtained by our numerical algorithm for $p_0 = 10^{-1}, 10^{-2}, 10^{-3}, \ldots, 10^{-7}$. It is observed that as $p_0$ decreases, the optimal pooling size and the achieved speedup increase rapidly. In particular, we have for $p_0 = 10^{-r}$,

$$k^* > 3^r = 3^{\log_{10}(1/p_0)} = (1/p_0)^{\log_{10}3} = (1/p_0)^{0.477}.$$  

Furthermore, we have $S(10^{-(r+1)})/S(10^{-r}) > 3$, and

$$S(10^{-r}) > 0.56 \times 3^r = 0.56 \times 3^{\log_{10}(1/p_0)} = 0.56(1/p_0)^{\log_{10}3} = 0.56(1/p_0)^{0.477}.$$  

That is, both $k^*$ and $S(p_0)$ grow exponentially with $1/p_0$, a quite impressive and nontrivial result.

It is worth to mention that the optimal group size $k^*$ is determined by the fraction $p_0$ of infected people and independent of the size $N$ of the population, since the equation $G(k^*) = 0$ only involves $q_0$ (actually $p_0$), not $N$. 

13
4 Conclusions

We have developed an analytical method and a numerical algorithm to determine the optimal group size, which minimizes the total number of tests, maximizes the speedup of the pooling strategy, and minimizes both time and cost of testing. The optimal group size is determined by the fraction of infected people and independent of the size of the population. Furthermore, both the optimal pooling size and the achieved speedup grow exponentially with the reciprocal of the fraction of infected people. Our method is effective in supporting faster and cheaper asymptomatic COVID-19 screening.

References


**Declarations:**

Competing interests: The author declares no competing interest.

Funding: None.