Note: This is a supplemental document for "Contact tracing enhances the efficiency of COVID-19 group testing," submitted to *2021 IEEE International Conference on Acoustics, Speech and Signal Processing* by Ritesh Goenka,* Shu-Jie Cao,* Chau-Wai Wong, Ajit Rajwade, and Dror Baron. RG and SJC have made equal contributions to the paper.

1. ALGORITHMIC DETAILS OF M1

1.1. Family Denoiser

We now formalize a family-based infection mechanism that can be used in designing group testing algorithms for improving the detection accuracy. We define $\mathcal{M}_\mathcal{F}$ as the set of indices of all members of family $\mathcal{F}$. We say that $\mathcal{F}$ is *viral* when there exists viral material in the family. Next, define the infection probability of individual i within viral family $\mathcal{F}$ as $\pi_{\text{ind}} = \Pr(X_i = 1 \mid \mathcal{F} \text{ viral})$, for all $i \in \mathcal{M}_\mathcal{F}$, and $\pi_{\text{vf}} = \Pr(\mathcal{F} \text{ viral})$. Note that the infection status of individuals in a viral family are conditionally independent and identically distributed (i.i.d.).

Under our definition, family $\mathcal{F}$ being viral need not be attributed to any individual $i \in \mathcal{M}_\mathcal{F}$. After all, viral material can be on an infected pet or contaminated surface. For this model, once the family is viral, the virus spreads independently with a fixed probability π_{ind}. Of course, our simplified model may not accurately reflect reality. That said, without a consensus in the literature on how coronavirus spreads, it is unrealistic to create a more accurate model. On the other hand, our model is plausible, and we will see that it is mathematically tractable.

We further assume that individuals cannot be infected unless the family is viral, i.e., $\Pr(X_i = 1 \mid \mathcal{F} \text{ not viral}) = 0$. The family structure serves as SI and allows the group testing algorithm to impose the constraint that people living together have strongly correlated health status.

Next, we derive the exact form of the denoiser (1) by incorporating the family-based infection mechanism. Denote the pseudodata of the members of family $\mathcal{F}$ as $\boldsymbol{v}_\mathcal{F} = (v_i)_{i\in\mathcal{M}_\mathcal{F}}$, the family-based denoiser for ith individual can be decomposed as follows:

$$\begin{aligned}
&g_{\text{in}}^{\text{family}}(\boldsymbol{v}_\mathcal{F}) \\
=&\mathbb{E}\left[X_i \mid \boldsymbol{v}_\mathcal{F}\right] && \text{(7a)}\\
=&\Pr(X_i = 1 \mid \boldsymbol{v}_\mathcal{F}) && \text{(7b)}\\
=&\Pr(X_i = 1, \mathcal{F} \text{ viral} \mid \boldsymbol{v}_\mathcal{F}) && \text{(7c)}\\
=&\Pr(\mathcal{F} \text{ viral} \mid \boldsymbol{v}_\mathcal{F}) \Pr(X_i = 1 \mid \boldsymbol{v}_\mathcal{F}, \mathcal{F} \text{ viral}), && \text{(7d)}
\end{aligned}$$

where the first term of (7d) is

$$\Pr(\mathcal{F} \text{ viral} \mid \boldsymbol{v}_\mathcal{F}) = \frac{f(\boldsymbol{v}_\mathcal{F}, \mathcal{F} \text{ viral})}{f(\boldsymbol{v}_\mathcal{F}, \mathcal{F} \text{ viral}) + f(\boldsymbol{v}_\mathcal{F}, \mathcal{F} \text{ not viral})}. \tag{8}$$

The two quantities in (8) can be further expanded as

$$\begin{aligned}
&f(\boldsymbol{v}_\mathcal{F}, \mathcal{F} \text{ not viral}) && \text{(9a)}\\
=&(1 - \pi_{\text{vf}})\, f(\boldsymbol{v}_\mathcal{F} \mid \mathcal{F} \text{ not viral}) && \text{(9b)}\\
=&(1 - \pi_{\text{vf}}) \prod_{i\in\mathcal{M}_\mathcal{F}} \mathcal{N}(v_i; 0, \Delta), && \text{(9c)}
\end{aligned}$$

and

$$\begin{aligned}
&f(\boldsymbol{v}_\mathcal{F},\ \mathcal{F} \text{ viral}) = \pi_{\text{vf}}\, f(\boldsymbol{v}_\mathcal{F} \mid \mathcal{F} \text{ viral}) && \text{(10a)}\\
=&\pi_{\text{vf}} \sum_{\boldsymbol{x}_k\in\Omega_\mathcal{F}} \prod_{i\in\mathcal{M}_\mathcal{F}} \\
&\Big[f(v_i|X_i = x_{k,i}) \Pr(X_i = x_{k,i}|\mathcal{F} \text{ viral})\Big], && \text{(10b)}
\end{aligned}$$

where $\mathcal{N}(x; \mu, \sigma^2) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{(x-\mu)^2}{2\sigma^2}\right)$, and $\Omega_\mathcal{F} = \{0...00,\ 0...10,\ \ldots,\ 1...11\}$ is a power set comprised of $2^{|\mathcal{M}_\mathcal{F}|}$ distinct infection patterns for family $\mathcal{F}$. The second term of (7d) can be simplified as follows:

$$\begin{aligned}
&\Pr(X_i = 1 \mid \boldsymbol{v}_\mathcal{F}, \mathcal{F} \text{ viral}) \\
=&\Pr(X_i = 1 \mid v_i, \mathcal{F} \text{ viral}) && \text{(11a)}\\
=&\Pr(X_i = 1, v_i \mid \mathcal{F} \text{ viral}) \,/\, \Pr(v_i \mid \mathcal{F} \text{ viral}) && \text{(11b)}\\
=&\frac{\pi_{\text{ind}}\, \mathcal{N}(v_i; 1, \Delta)}{\pi_{\text{ind}}\, \mathcal{N}(v_i; 1, \Delta) + (1 - \pi_{\text{ind}})\, \mathcal{N}(v_i; 0, \Delta)} && \text{(11c)}\\
=&\left(1 + \frac{1 - \pi_{\text{ind}}}{\pi_{\text{ind}}} \cdot \frac{\mathcal{N}(v_i; 0, \Delta)}{\mathcal{N}(v_i; 1, \Delta)}\right)^{-1} && \text{(11d)}\\
=&\left(1 + \left(\pi_{\text{ind}}^{-1} - 1\right) \exp\left[\left(v_i - \tfrac{1}{2}\right)/\Delta\right]\right)^{-1}. && \text{(11e)}
\end{aligned}$$

1.2. Contact Tracing Denoiser

While family structure SI characterizes part of the spread of the disease, individual members of a family will presumably all come in close contact with each other, hence CT SI will include cliques for these individuals. Additionally, CT SI describes inter-family contacts. Therefore, CT SI can characterize the spread of the disease more comprehensively than family SI. To exploit the CT SI, we encode it for each individual i into the prior probability of infection, $\Pr(X_i = 1)$, and use the following scalar denoiser:

$$\begin{aligned}
&g_{\text{in}}^{\text{CT}}(v_i) \\
=&\mathbb{E}\left[X_i|v_i\right] = \Pr(X_i = 1|v_i) && \text{(12a)}\\
=&f(v_i|X_i = 1) \Pr(X_i = 1)/f(v_i) && \text{(12b)}\\
=&\left\{1+\left[\Pr(X_i=1)^{-1}-1\right] \exp\left[\left(v_i - \tfrac{1}{2}\right)/\Delta\right]\right\}^{-1}. && \text{(12c)}
\end{aligned}$$

Here, $\Pr(X_i=1)$ for day $k+1$ can be estimated by aggregating CT information of individual i over a so-called *SI period* from day $k-7$ to day k as follows

$$\widehat{\Pr}^{(k+1)}(X_i = 1) = 1 - \prod_{d=k-7}^{k} \prod_{j=1}^{n} \left(1 - \widehat{p}_{i,j}^{(d)}\right), \tag{13}$$

where $\widehat{p}_{i,j}^{(d)}$ is the estimated probability of infection of individual i due to contact with individual j. This probability, $\widehat{p}_{i,j}^{(d)}$, can be determined by the CT information $(\tau_{ij}^{(d)}, d_{ij}^{(d)})$, as well as their infection status as follows:

$$\widehat{p}_{i,j}^{(d)} = \exp\left(-\left(\lambda\, \tau_{ij}^{(d)}\, d_{ij}^{(d)}\, \Psi_{ij}^{(d)} + \epsilon\right)^{-1}\right), \tag{14}$$

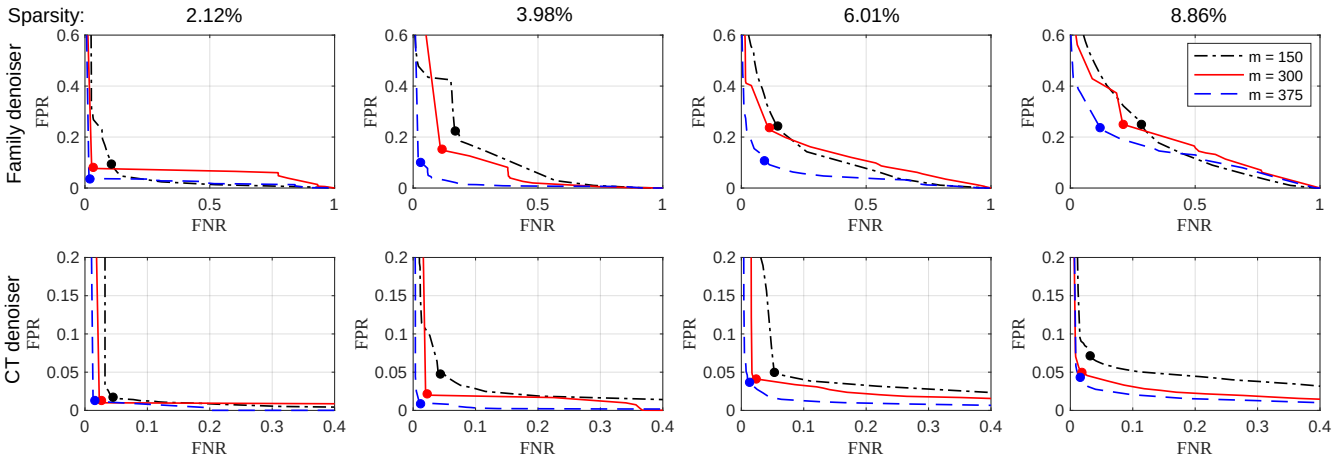

Fig. 4. Performance of **M1** in terms of ROC when family denoiser (top row) and CT denoiser (bottom row) are used. Columns correspond to averaged sparsity levels ranging from 2.12% to 8.86%. Within each plot, the performance under three measurement levels for a population of $n = 1000$ individuals is compared. The dot on each curve corresponds to an operating point that minimizes the sum of FPR and FNR. The CT denoiser significantly outperforms the family denoiser with error rates mostly below 0.05. The estimation problem is more challenging when fewer measurements are used at a higher sparsity level.

where $\Psi_{ij}^{(d)} = 1 - \widehat{\Pr}^{(d)}(X_i = 0)\,\widehat{\Pr}^{(d)}(X_j = 0)$, λ is an unknown Poisson rate parameter, and ϵ is used to avoid division by zero. We estimate λ with maximum likelihood (ML) using the pseudodata of all individuals, i.e.,

$$\widehat{\lambda}^{\text{ML}} = \arg\max_{\lambda} \prod_{i=1}^{n} f(v_i|\lambda), \tag{15}$$

where $f(v_i|\lambda) = f(v_i|X_i = 1)\ \Pr(X_i = 1|\lambda) + f(v_i|X_i = 0)\ \Pr(X_i = 0|\lambda)$. Once $\widehat{\lambda}^{\text{ML}}$ is obtained, it is plugged into (14) for calculating the prior probability [18]. Note that this plug-in strategy is also used for two other denoisers, namely, $\lambda = \rho$ for $g_{\text{in}}^{\text{Bernoulli}}(v_i)$ and $\lambda = (\pi_{\text{vf}}, \pi_{\text{ind}})$ for $g_{\text{in}}^{\text{family}}(\boldsymbol{v})$.

2. ADDITIONAL RESULTS FOR M1

In Sec. 4 of the main paper, we reported the performance of M1 in a compact way, due to space limitations, by choosing a representative operating point on an ROC curve instead of using the whole curve. In this section, we provide complete ROC curves that correspond to the top row of Fig. 3 of the main paper. Fig. 4 illustrates M1's performance for family and CT denoisers at different measurement and sparsity levels. The dot on each curve corresponds to the operating point that minimizes the total error rate, i.e., the sum of FPR and FNR, as reported in Sec. 4 of the main paper. The closer a dot is to the origin of the FPR–FNR plane, the better the performance it reflects. Comparing the ROC curves in the top row to those in the bottom row, we note that the CT denoiser significantly outperforms the family denoiser at all sparsity levels. The CT denoiser, with most of its FNR and FPR $< 5\%$, can achieve as low as 15% of the total error rate of the family denoiser. Across different sparsity levels, the algorithm performs less accurately as the sparsity level increases. In each plot, lower measurement rates make it more challenging for the group testing algorithm.

We also examine the stability of the thresholds corresponding to the operating points we selected to report results in Fig. 3 of the main paper. Our empirical results reveal that at a particular sparsity level, the variation of the threshold due to different design matrices or denoisers is less than 0.003. As the sparsity level increases from 2.12% to 8.86%, the threshold only drops from 0.160 to 0.137. Hence, the threshold for minimizing the total error rate is insensitive to the testing conditions.

3. ADDITIONAL EXPERIMENTS FOR M1

3.1. Using Prior Knowledge of the Infection Status

In this subsection, we examine the advantage that prior knowledge of the population's infection status in the startup phase provides our proposed algorithm for the M1 binary model. As stated in Sec. 3 of the main paper and in (13), our algorithm iteratively uses the updated probability of infection, $\widehat{\Pr}(X_i = 1)$, estimated from an SI period of 8 immediately preceding days. Note that for days $k < 8$, we had to use the ground-truth infection status of each individual in the startup phase to generate the results reported in Sec. 4 of the main paper. However, ground-truth infection data from the startup phase may provide our approach an unfair advantage over the algorithms proposed for M2. Below, we investigate whether this advantage is significant.

We examine how varying the amount of startup information impacts our algorithm's quality. Specifically, we ran-

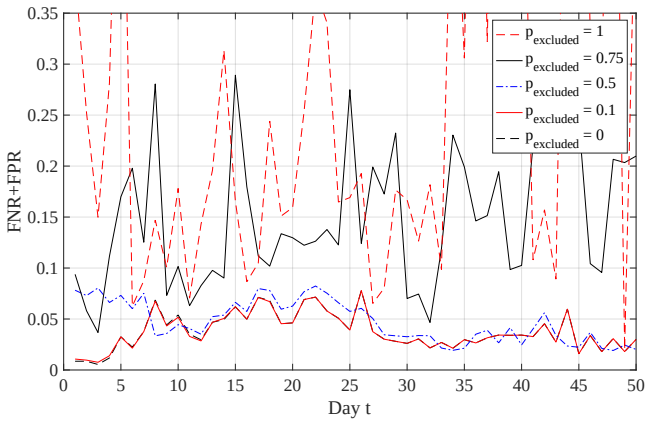

Fig. 5. Performance of M1 when a proportion, p_{excluded}, of the population's health states in the startup phase is unknown. The curves reveal that in the absence of up to 50% prior knowledge of the infection status of the population, the accuracy of M1 is close to that when complete startup information is available.

domly replace a portion, $p_{\text{excluded}} \in \{0, 0.1, 0.5, 0.75, 1\}$, of the population's infection status by an estimated probability of infection, e.g., 5%, for a setup that has a true averaged sparsity level of 7.2%. Using a probability instead of a binary value, 0 or 1, gives the algorithm soft probabilistic information instead of hard ground-truth style information. Fig. 5 shows that even with 50% prior knowledge of the infection status of individuals, our detection accuracy for M1 is close to that when using complete prior information after ramping up for 8 days. The averages of the total error rates across time for increasing p_{excluded} are 0.038, 0.039, 0.046, 0.148, and 0.407, respectively. We also tried to replace the startup infection status by an estimated probability of infection of 10%, but only observed negligible performance differences. The results show that the CT algorithm is robust to the absence of up to 50% of startup infection information.

3.2. Duration of Startup SI Period

We investigated the impact of the duration of the startup SI on estimation performance. In principle, the longer the SI duration, the more accurate we expect the results to be. There is a trade-off between the accuracy of our algorithm and the startup SI infection status information that needs to be precollected before the initialization of the testing algorithm. In our experiment, we tested three startup SI durations, namely, 4 days, 8 days, and 12 days. Our experimental results (omitted for brevity) show that the estimation accuracy is somewhat insensitive to the duration of the SI period. Hence, for the experiments conducted for this paper, we chose 8 days as the SI period.

4. AN ADDITIONAL EXPERIMENT FOR M2

4.1. Data Generation

For this experiment, we use a different and slightly more general contact tracing graph to simulate the spread of infection. Recall that the adjacency matrix of the contact graph has a block diagonal structure with sizes of cliques coming from the distribution of family sizes in India [17, pg. 18]. However, in this case, we allow two consecutive (according to the order in which cliques appear along the diagonal of the contact matrix) non-trivial cliques (i.e., cliques with more than one node) to have an overlap of one node with probability half. This assumption is reasonable since the concept of family encompasses more general groups such as people at the same workplace, students studying in the same classroom, etc. Furthermore, we remove $\alpha = 5\%$ of the edges from this block diagonal structure, thus converting the existing cliques into "almost-cliques." This modified block diagonal structure is kept constant over time while the cross-clique contacts are updated every day. Except for the changes in the underlying contact tracing graph, the rest of the data generation method is the same as that described in Sec. 2 of the main paper.

4.2. Inference

We use the four algorithms (including COMP) for multiplicative noise described in Sec. 3 of the main paper. However, instead of using maximal cliques as groups in COMP-SQRT-OGLASSO, we use the decomposition of the contact tracing graph into overlapping 3-clique communities [23]. An algorithm for detecting k-clique communities can be found in Sec. 1 of [23, Supplementary Notes]. The first step of this algorithm involves finding the maximal cliques in the contact graph, for which we use the Bron-Kerbosch algorithm [21]. In the next step, we detect 3-clique communities and label each of those as groups. Further, we also label as groups the maximal cliques that are not part of any of these communities, in order to ensure that every contact is taken into account. The advantage of using 3-clique communities over just maximal cliques is that the former is able to capture "almost cliques," i.e., cliques with a small fraction of absent pairwise contacts.

4.3. Numerical Results

We present the results in a format similar to that in Sec. 4 of the main paper, but for the contact graph described in Sec. 4.1. Fig. 6 shows the mean values (across 50 signals) of the false negative rate (FNR) and false positive rate (FPR) obtained for four different sparsity levels. The sparsity levels were obtained by varying the amount of cross-clique contacts. We remark that the length of each bar in Fig. 6, FNR + FPR, is equal to $1 -$ Youden's Index. Further, we plot heat maps (Fig. 7) to compare the performance of the four algorithms under consideration—the intensity of gray corresponds to the mean value (across 50 signals) of the Matthews Correlation

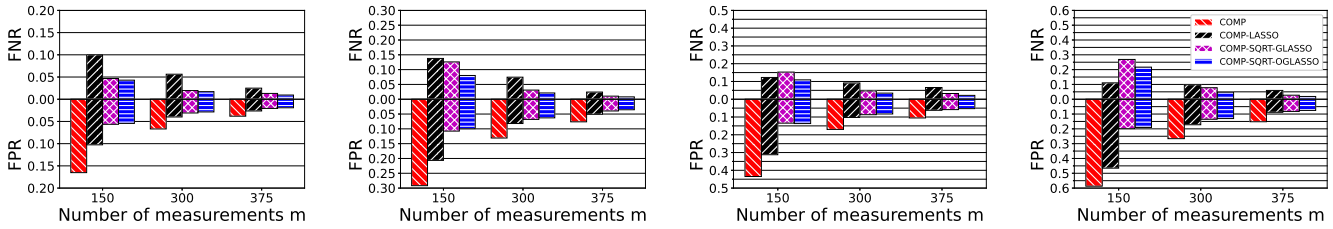

Fig. 6. Figure showing mean FNR and FPR values for the contact graph from Section 4.1, for mean sparsity levels of 3.20%, 4.84%, 6.25%, 8.66% (from left to right).

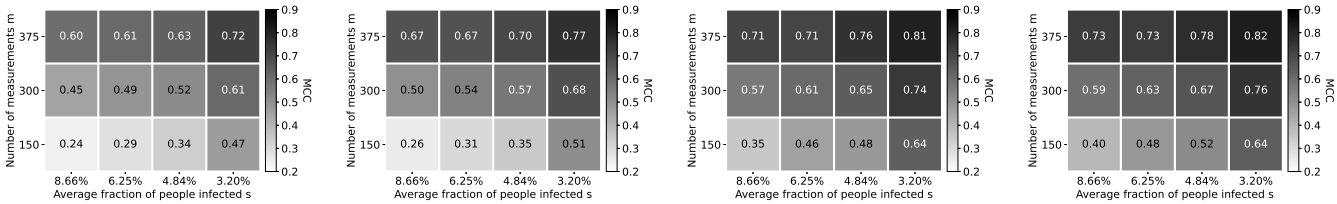

Fig. 7. Figure showing mean MCC values obtained using COMP, COMP-LASSO, COMP-SQRT-GLASSO, COMP-SQRT-OGLASSO (from left to right).

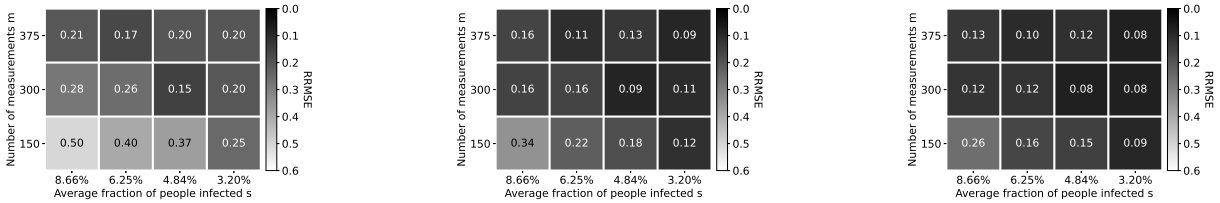

Fig. 8. Figure showing mean RRMSE values obtained using COMP-LASSO, COMP-SQRT-GLASSO, COMP-SQRT-OGLASSO (from left to right).

Coefficient (MCC). The MCC is defined as

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}}, \quad (16)$$

and has been proposed as a comprehensive metric to evaluate the performance of binary classification algorithms [27]. Its values range from -1 to $+1$, where a value closer toward $+1$ is desirable. The RRMSE values can be seen from the heat maps in Fig. 8 (we do not provide a heat map for COMP since it does not estimate viral loads).

5. ADDITIONAL RESULTS FOR M2

For model **M2**, we present a comparison of the four algorithms for the experiment described in Sec. 4 of the main paper. Fig. 9 and Fig. 10 show a comparison of the performance of the algorithms under consideration in terms of mean MCC and mean RRMSE values, respectively. Further, we remark that the true viral loads of the false negatives yielded by COMP-LASSO variants are concentrated toward lower values. For instance, only about 29% of the false negatives given by COMP-SQRT-OGLASSO had viral load values greater than $2^{12} = 4096$.

6. SENSING MATRIX DESIGN

As mentioned in the main paper, we use Kirkman triple matrices as sensing matrices for performing pooling. A Kirkman triple (binary) matrix $\boldsymbol{A}$ can be partitioned into $3n/m$ submatrices of dimensions $m \times m/3$, each of which contains exactly one nonzero entry in each row and three nonzero entries in each column. Further, the dot product of any two columns of the matrix $\boldsymbol{A}$ should not exceed 1. For a given value of n, $m(< n)$ must satisfy the following conditions:

1. m must be of the form $3n_1$, where n_1 divides n, since the number of sub-matrices and the number of columns in each sub-matrix must be integers.

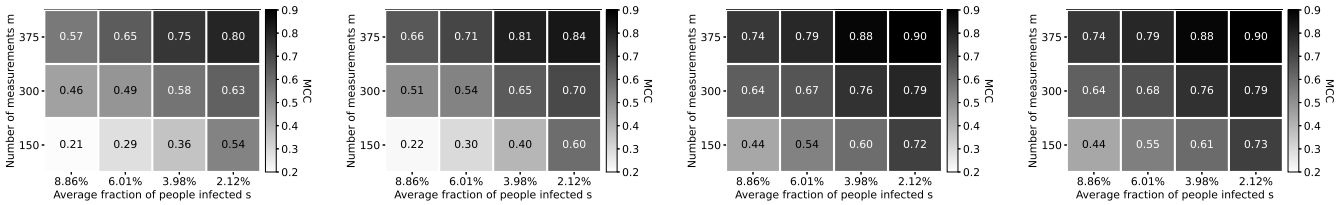

Fig. 9. Figure showing mean MCC values obtained using COMP, COMP-LASSO, COMP-SQRT-GLASSO, COMP-SQRT-OGLASSO (from left to right).

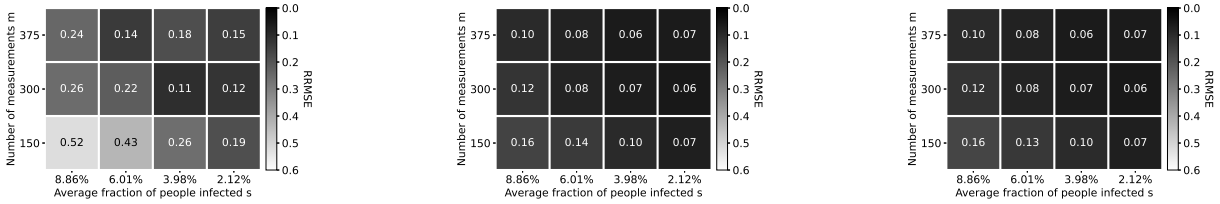

Fig. 10. Figure showing mean RRMSE values obtained using COMP-LASSO, COMP-SQRT-GLASSO, COMP-SQRT-OGLASSO (from left to right).

2. $\binom{3}{2} \cdot n \leq m(m-1)/2$ since a triple contains $\binom{3}{2}$ pairs and a pair must belong to at most one triple.

For $n = 1000$, the only values of m which satisfy the above constraints are 120, 150, 300, 375, 600, and 750. We construct Kirkman triple matrices with $m = 150, 300, 375$ and use them in our experiments. The matrices are constructed based on a few simple rules:

1. The indices of ones in each column form an arithmetic progression (AP).
2. The matrix has a block structure and the common difference of the AP remains constant throughout each block. Furthermore, the sum of all columns in a block yields the vector consisting of all ones.
3. The common difference values $\{d_B : \boldsymbol{B} \text{ is a block}\}$ are chosen such that the multi-set $\{r \cdot d_B : r \in \{n : n \in \mathbb{N}, n < 3\}, \boldsymbol{B} \text{ is a block}\}$ has no duplicate values.

Fig. 11 shows the structure of a 375×1000 Kirkman triple matrix obtained using the above approach. Let $\boldsymbol{B}$ be any block and let d_B denote the common difference of the AP for block $\boldsymbol{B}$ as indicated in the figure. Then, the ith column of $\boldsymbol{B}$ is given by

$$\boldsymbol{B}_i = \sum_{j=1}^{3-1} \boldsymbol{e}_{\beta_i + j d_B}, \quad \beta_i = \text{mod}(i, d_B) + 3d_B[i/d_B], \quad (17)$$

where $[\cdot]$ denoted the greatest integer function and $\boldsymbol{e}_j$ denotes the jth standard basis vector. Clearly, any block $\boldsymbol{B}$ must have dimensions $3n_B \times n_B$, where d_B divides n_B.

As m decreases, it becomes harder to design matrices satisfying all three rules specified earlier. However, it is possible to relax the third rule in such cases and still obtain a matrix satisfying the required constraints. For example, our 150×1000 Kirkman triple matrix does not obey the third rule. We further remark that one may design balanced matrices with a different number of (say k) ones in each column such that the dot product of every pair of columns is bounded by 1, using the above approach. Such matrices would arise from the Steiner systems $S(2, k, m)$ [just as Kirkman matrices arise from $S(2, 3, m)$]. For example, it is straightforward to design a 400×1000 matrix with $k = 4$ using our approach.

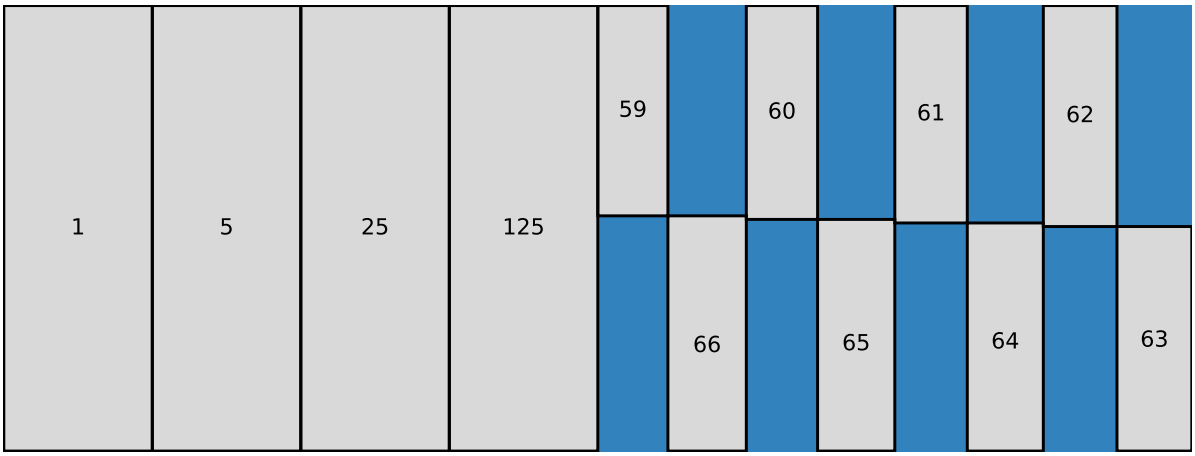

Fig. 11. A 375×1000 Kirkman triple matrix obtained using our approach. The number written within each block is equal to the corresponding common difference value and the blocks without number markings are zero matrices. The blocks having less than $m = 375$ rows have dimensions equal to $3d_B \times d_B$.