Online Supplement: Subset Selection for Multiple Linear Regression via Optimization

Young Woong Park [*1] and Diego Klabjan [†2]

[1]Ivy College of Business, Iowa State University, Ames, IA, USA
[2]Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA

September 17, 2019

1 Big M for x_j's in (9)

This section refers to the proof of Proposition 6. We start with the following lemma from Schrijver [30].

Lemma OS 1. (Corollary 3.2b, Schrijver [30]) If $Ax = B$ has a solution, it has one of size polynomially bounded by size of A and B. That is, for a solution $\hat{x}$, we have $size(\hat{x}) \leq size(A) \cdot size(B)$.

Next, we derive a bound for rational number r based on $size(r)$.

Lemma OS 2. We have $|r| \leq 2^{size(r)-1}$.

Proof. Starting from the definition, we derive

$$\begin{aligned} size(r) &= 1 + \lceil \log_2(|r_{num}| + 1) \rceil + \lceil \log_2(r_{den} + 1) \rceil \\ &\geq 1 + \log_2(|r_{num}| + 1) + \log_2(r_{den} + 1) \\ &= 1 + \log_2[|r_{num}| r_{den} + |r_{num}| + r_{den} + 1] \\ &= 1 + \log_2[|r| r_{den}^2 + |r| r_{den} + r_{den} + 1] \\ &> 1 + \log_2[|r| r_{den}^2 + |r| r_{den}] \\ &= 1 + \log_2 |r| + \log_2(r_{den}^2 + r_{den}). \end{aligned}$$

By rearranging, we obtain $|r| \leq 2^{size(r)-1-\log_2(r_{den}^2 + r_{den})} \leq 2^{size(r)-1}$. □

Next we derive an upper bound for $\hat{x}$ using Lemmas OS 1 and 2.

Lemma OS 3. Let $\hat{x}$ be a solution to $Ax = B$. For any j in J, we have $|\hat{x}_j| \leq 2^{size(A) size(B)-1}$.

Proof. We first derive $m + size(\hat{x}_j) \leq m + \sum_{j=1}^{m} size(\hat{x}_j) = size(\hat{x}) \leq size(A) size(B)$, in which the last inequality holds by Lemma OS 1. Combining everything, we obtain $|\hat{x}_j| \leq 2^{size(A) size(B)-1}$. □

Note that optimizing MSE is to select a subset of the columns of a. Let $\bar{a} \in \mathbb{R}^{n \times p}$ be the data matrix that corresponds to a subset of m columns of a, with cardinality p. For simplicity, let us assume that we sort columns of a so that the selected p columns have indices from $j = 1$ to p. Let $\bar{A} = \bar{a}^T \bar{a}$.

Lemma OS 4. We have $size(\bar{A}) \leq size(A)$ for any $\bar{A}$.

Proof. Recall that we have $A = [\alpha_{ij}]_{i=1,\cdots,m, j=1,\cdots,m}$ and $\bar{A} = [\bar{\alpha}_{ij}]_{i=1,\cdots,p, j=1,\cdots,p}$. We derive

$$\begin{aligned} size(\bar{A}) &= p^2 + \sum_{i=1}^{p} \sum_{j=1}^{p} size(\bar{\alpha}_{ij}) \\ &\leq m^2 + \sum_{i=1}^{p} \sum_{j=1}^{p} size(\bar{\alpha}_{ij}) \\ &= m^2 + \sum_{i=1}^{p} \sum_{j=1}^{p} size(\alpha_{ij}) \\ &\leq m^2 + \sum_{i=1}^{m} \sum_{j=1}^{m} size(\alpha_{ij}) \\ &= size(A), \end{aligned}$$

which completes the proof. □

Using Lemmas OS 3 and 4, it is trivial to see that Proposition 6 holds.

[*]ywpark@iastate.edu
[†]d-klabjan@northwestern.edu

2 Proofs for Core Set Algorithms

Proof of Lemma 2
Recall that $\min_{j\in J} \bar{U}_j = -0.5$ and $\max_{j\in J} \bar{U}_j = 0.5$. Hence, it follows

$$\frac{\max_{j\in J} q_j}{\min_{j\in J} q_j} = \frac{\max_{j\in J} e^{-\bar{U}_j}}{\min_{j\in J} e^{-\bar{U}_j}} = \frac{e^{0.5}}{e^{-0.5}} \leq 2.72,$$

which completes the proof. □

Lemma OS 5. We have $\min_{j\in J} q_j^{(t)} \geq \frac{1}{1+2.72(m-1)}$ for any t.

Proof. Let $q_{min} = \min_{j\in J} q_j^{(t)}$. From Lemma 2, we have $q_j^{(t)} \leq 2.72 \cdot q_{min}$ for all $j \in J$. Without loss of generality, let $q_1^{(t)} = q_{min}$. Since $\sum_{j\in J} q_j^{(t)} = 1$, we derive

$$q_{min} = 1 - \textstyle\sum_{j=2}^{m} q_j^{(t)} \geq 1 - \sum_{j=2}^{m} 2.72 \cdot q_{min} = 1 - 2.72 \cdot q_{min}(m-1).$$

By rearranging, we obtain $q_{min} \geq \frac{1}{1+2.72(m-1)}$. Observe also that Lemma OS 5 holds for the renormalized probabilities (28) at any point of the selection iteration, since $q_j^{(t)}$'s are increasing for the explanatory variables that have not been selected. □

Proof of Lemma 3
Note that $q_j^{(t)}$'s are renormalized in Steps 5 - 6 of *Update-Core-Set-Random* and $q_j^{(t)}$'s in Step 2 depends on all of the events happened prior to *Update-Core-Set-Random*. However, each $q_j^{(t)}$ is non-decreasing by the renormalization in Step 6 and $q_j^{(t)}$'s in Step 2 are at least $\frac{1}{1+2.72(m-1)}$ by Lemma OS 5.

Event $\{S^{opt} \subset C_t\}$ happens if C_t can be partitioned into (i) all the explanatory variables in S^{opt} and (ii) $\Theta - |S^{opt}|$ explanatory variables from $J \setminus S^{opt}$. Then, we obtain

$$\begin{aligned} P[S^{opt} \subset C_t | \mathcal{H}_{t-1}] &\geq \tbinom{m-|S_{opt}|}{\Theta-|S_{opt}|} \textstyle\prod_{j\in S^{opt}} \left(\frac{1}{1+2.72(m-1)}\right) \prod_{j=1}^{\Theta-|S^{opt}|} \left(\frac{1}{1+2.72(m-1)}\right) \\ &\geq \left(\frac{1}{1+2.72(m-1)}\right)^{\Theta}, \end{aligned}$$

in which the first inequality holds since we can select $\Theta - |S_{opt}|$ explanatory variables from $J \setminus S_{opt}$. This completes the proof. □

Proof of Lemma 4
We derive

$$\begin{aligned} P\left[\textstyle\bigcap_{k=1}^{t} A_k\right] &= P\left[A_t | \bigcap_{k=1}^{t-1} A_k\right] P\left[A_{t-1} | \bigcap_{k=1}^{t-2} A_k\right] \cdots P[A_2|A_1] P[A_1] \\ &= \left(1 - P\left[S^{opt} \subset C_t | \textstyle\bigcap_{k=1}^{t-1} A_k\right]\right) \cdots \left(1 - P[S^{opt} \subset C_2 | A_1]\right) P[A_1] \\ &\leq \left(1 - P\left[S^{opt} \subset C_t | \mathcal{H}_{t-1}\right]\right) \cdots \left(1 - P[S^{opt} \subset C_2 | \mathcal{H}_1]\right) P[A_1] \\ &\leq (1-\varphi)^t, \end{aligned}$$

where the first equality holds by the chain rule, the first inequality holds since $\bigcap_{k=1}^{t-1} A_k \subseteq \mathcal{H}_{t-1}$, and the last inequality holds by Lemma 3. □

Proof of Lemma 5
Note that $P\left[\bigcap_{k=1}^{t} A_k\right]$ in Lemma 4 is equivalent to $P\left[S^{opt} \not\subset C_k \text{ for all } k \in \{1, \cdots, t\}\right]$. Hence, we derive

$$
\begin{aligned}
P\Big[mae_a(t) = mae_a^{opt}\Big] &= P\Big[S^{opt} \subset C_k \text{ for at least one } k \in \{1, \cdots, t\}\Big] \\
&= 1 - P\Big[S^{opt} \not\subset C_k \text{ for all } k \in \{1, \cdots, t\}\Big] \\
&= 1 - P\Big[\textstyle\bigcap_{k=1}^{t} A_k\Big] \\
&\geq 1 - (1-\varphi)^t,
\end{aligned}
$$

where the inequality holds by Lemma 4. □

Lemma OS 6. The objective function value of S^* in Algorithm 1 is strictly decreasing at each iteration except in the last iteration.

Proof. Let t be the current iteration in Algorithm 1 and let S_t^* be the best subset over core set C_t with objective function value mae_a^t in Step 5 of Algorithm 1. Next in Steps 1 - 11 of Algorithm 2, we check neighboring subsets, iteratively update the best subset, and obtain $\bar{S}_t$ with $\overline{mae}_a^t$ after Step 12. Note that it is possible to have $S_t^* = \bar{S}_t$ if there is no update in Step 7 of Algorithm 2. Then after Step 14 of Algorithm 2, we obtain new core set C_{t+1} and iteration t is over. We execute iteration $t+1$ similarly and obtain (i) S_{t+1}^* with mae_a^{t+1} in Step 5 of Algorithm 1, and (ii) $\bar{S}_{t+1}$ with $\overline{mae}_a^{t+1}$ after Step 12 of Algorithm 2. Hence, the current best objective function values after Step 6 of Algorithm 1 are $\overline{mae}_a^t$ and $\overline{mae}_a^{t+1}$ for iterations t and $t+1$, respectively. For a contradiction, let us assume $\overline{mae}_a^t \leq \overline{mae}_a^{t+1}$ and $t+1$ is not the last iteration in Algorithm 1. We have two cases.

1. $\overline{mae}_a^t < \overline{mae}_a^{t+1}$
   First, note that $\bar{S}_t \subset C_{t+1}$ due to Step 14 of Algorithm 2 and that S_{t+1}^* is an optimal subset to (24) over C_{t+1}. This implies $mae_a^{t+1} \leq \overline{mae}_a^t$. Second, due to Steps 1 - 12 of Algorithm 2, we must have $\overline{mae}_a^{t+1} \leq mae_a^{t+1}$. Combining all three inequalities, we obtain

   $$\overline{mae}_a^{t+1} \leq mae_a^{t+1} \leq \overline{mae}_a^t < \overline{mae}_a^{t+1},$$

   which is a contradiction.

2. $\overline{mae}_a^t = \overline{mae}_a^{t+1}$
   This implies that Algorithm 1 terminates after iteration $t+1$ since we are violating the criterion in Step 4. This contradicts the assumption that $t+1$ is not the last iteration.

Hence, mae_a^* is strictly decreasing except in the last iteration in Algorithm 1. □

Lemma OS 7. Algorithm 1 does not visit the same core set except in the last two iterations.

Proof. For a contradiction, let us assume that Algorithm 1 visits the same core set C in two different iterations but they are not the last two iterations. By Lemma OS 6, we must have different subsets and objective function values for the two iterations. Let S_1^* and S_2^* be the subsets with corresponding objective function values mae_a^1 and mae_a^2 such that $mae_a^1 > mae_a^2$. However, $mae_a^1 > mae_a^2$ implies that S_1^* is not an optimal subset to (24) over core set C. This is a contradiction. □

3 Stepwise Algorithm

Most of the publicly available implementations of stepwise algorithm consider AIC, BIC, or criteria other than MAE and MSE. The R Statistics package Leaps, Lumley [19], supports the adjusted r^2 objective, which is equivalent to optimizing MSE. However, Leaps cannot handle the fat case, and there are no publicly available packages for the MAE objective. Further, we consider new objectives MAE_a and MSE_a for computational experiments. Hence, we implement a stepwise algorithm that works for all of the objective functions we are considering for both the thin and fat cases.

Algorithm 1 closely follows the standard stepwise selection procedure except that it considers p^{max}, the maximum number of explanatory variables allowed, given selection criterion OBJ. It starts with empty set in Step 1. Then in Step 2, it iteratively adds an explanatory variable based on a greedy strategy (one that minimizes the objective function most) until there is no improvement or we reach p^{max} explanatory variables.

Next in Step 3, we consider both the forward and backward direction to check if a neighboring set of S is better. Step 3 continues until there is no improvement.

Finally, we call *Stepwise*(m,OBJ) with OBJ $\in \{MSE, MAE, MAE_a, MSE_a\}$ for the thin case, as $m \leq n-2$ is guaranteed for this case. For the fat case, we call *Stepwise*(p^{max},OBJ) with $p^{max} \leq n-2$ and OBJ $\in \{MSE, MAE, MAE_a, MSE_a\}$.

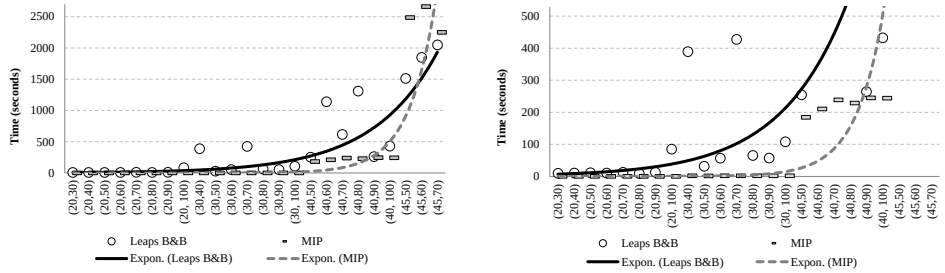
Algorithm 1 Stepwise(p^{max}, OBJ)

Input: p^{max} (maximum cardinality of subset), OBJ (selection criteria)
Output: S (subset of explanatory variables)

1: $S \leftarrow \emptyset$
2: Forward selection based on OBJ and update S, until (i) there is no improvement or (ii) $|S| = p^{max}$
3: Forward selection and backward elimination based on OBJ, and update S as long as $|S| \leq p^{max}$, until there is no improvement

4 The Empirical Time Complexity of Leaps-and-bound and (9)

Leaps, R package by Lumley [19], supports exhaustive search based on the leaps-and-bound algorithm (Leaps B&B) of Furnival and Wilson [12]. In this section, we investigate the empirical time complexities of Leaps B&B and (9) by checking the execution time to get an optimal solution for the instances used in Section 4 of the paper. If it takes more than 1 hour, we quit the algorithm and record 3600 seconds instead. In Figure OS 1, we present the average computational time of Leaps B&B and MIP for MSE. Figure OS 1(a) is the full size plot and Figure OS 1(b) is with a truncated y axis. Axes x and y are instance sets and time in seconds. The circles and rectangles represent the average computational time of the MIP model and Leaps B&B, respectively. The plain and dotted lines are fitted exponential curves for MIP and Leaps, respectively. Observe that the computation time for both algorithms increases superlinearly, yet the computational time of Leaps B&B grows much faster.



(a) (b)

Figure OS 1: Average computational time of Leaps B&B and the MIP for MSE over the instance sets

5 Performance of MIP for Different Big M Values for x_j's

In Section 2.1.3, two approaches are proposed to derive a valid value of big M for regression coefficients x_j's for (6) and (8). In this section, we present an experiment to support the necessity of valid big M values by checking the solution quality and execution time of MIP model (6) over various big M values. Let M_{valid} be the valid big M value derived by the proposed approach. Using M_{valid}, we obtain the optimal regression coefficients x^* by solving (6). Let $M_{\text{tight}} = \max_{j \in J} |x_j^*|$ be the tightest valid big M value for x_j's.

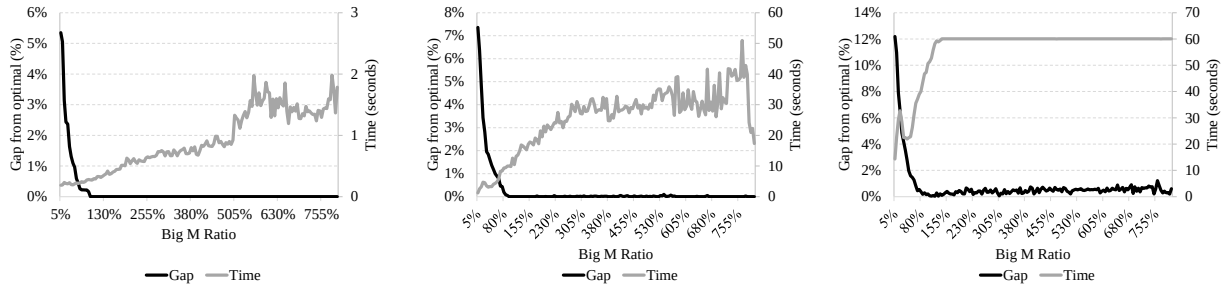
We first check the ratio between M_{tight} and M_{valid} to assess the quality of the big M values derived by the proposed approach. In Table OS 1, the summary statistics of the ratios ($\frac{M_{\text{valid}}}{M_{\text{tight}}}$) are presented using the

synthetic instances with $n = 100$ and $m \in \{20, 30, 40, 50\}$. If the ratio is small, then the proposed approach returns a tight big M value. If the ratio is large, then a relaxed big M value is returned. Overall, M_{valid} is 7.84 times large than M_{valid} on average. There are larger variations in the ratio when m is small but the variations decrease as m increases.

Statistics	$m = 20$	$m = 30$	$m = 40$	$m = 50$	All
Average	9.76	6.14	7.71	7.75	7.84
Median	4.94	6.12	7.42	7.60	6.36
StdDev	16.10	2.31	1.97	2.00	8.03

Table OS 1: Ratio between M_{tight} and M_{valid}

Next, we check the performance of MIP over increasing big M values using the same instances. In detail, we check the big M values from 5% to 800% of the M_{tight} value for each instance. The upper bound 800% is set based on the average ratio for all instances in Table OS 1. The result for the instances with $m = 50$ are excluded because the result is similar to those for the instances with $m = 40$. In Figure OS 2, the gap from the optimal solution (in %) and execution time (in seconds) are presented for each $m \in \{20, 30, 40\}$. In each plot, the horizontal axis is for the ratio between the big M value used and M_{tight}, the vertical axes are for the gap from the optimal solution and execution time.



(a) $m = 20$ (b) $m = 30$ (c) $m = 40$

Figure OS 2: Big M result

From the the results presented in Figure OS 2, we observe the followings.

1. When big M value is smaller than M_{tight} (the ratio is smaller than 100%), the solution quality drastically changes. Hence, using an arbitrary big M value can affect the solution quality.
2. When big M value is larger than M_{tight} (the ratio is larger than 100%), the solution quality remains near zero. The fluctuations in the objective function gap (e.g., in Figure OS 2(c)) are due to the fact that the algorithm terminated after 60 seconds. As the ratio increases, the fluctuations become severe, which implies that having an arbitrary large big M value can return a bad solution. Hence, we conclude that using a valid big M value returns optimal or near-optimal solutions as long as the ratio is in a reasonable range.
3. When the big M value is larger than M_{tight} (the ratio is larger than 100%), the execution time tends to increase as the big M ratio increases. When $m = 20$ or 30, there exists a clear trend showing a large big M value increases the execution time. However, as the ratio approaches 800%, the execution time does not increase rapidly.

Combining the observations, we conclude that it is important to have a valid value of big M (i.e., larger than M_{tight}) to guarantee optimality. Once validity is guaranteed, big M values within a reasonable range do not affect the performance significantly while an arbitrary large value can decrease the solution quality and it can cause numerical stability issues. Hence, the use of the proposed big M value calculation algorithm is justified.

6 Comparison of Big M and Logical Constraint-based Models

In Section 2 of the main document, mathematical programming models (6) and (8) are proposed based on the valid big M values derived, and in Section 5 of the online supplement, the usefulness of the valid big M values are shown. In this section, we show the advantage of the big M-based models over alternative modeling techniques available in most commercial optimization solvers. This also supports the necessity of tight and valid big M values. In detail, we compare the proposed big M-based models with *indicator constraints* or *logical constraints* of CPLEX. The logical constraints do not require a valid big M value, which may improve numerical robustness of the solver or decrease the solution time. In the experiment in this section, we compare the two approaches, big M-based and logical constraint-based formulations, for solving (6).

The logical constraints replace big M constraints by combining linear constraints based on logical operators and conditional statements. In the following logical constraint-based formulation, the big M-based constraints (6d)-(6g) of (6) are replaced by the logical constraints (OS1d) - (OS1g). Note that the constraints $x_j^+ = 0$, $x_j^- = 0$, $v_j = 0$, and $v_j = u$ are conditional and depend on the z_j value in the logical constraints.

$$\min \quad u \tag{OS1a}$$

$$s.t. \quad \sum_{i=1}^{n}(t_i^+ + t_i^-) = (n-1)u - \sum_{j\in J} v_j, \tag{OS1b}$$

$$t_i^+ - t_i^- = \sum_{j=1}^{m} a_{ij}(x_j^+ - x_j^-) + (y^+ - y^-) - b_i, \qquad i \in I, \tag{OS1c}$$

$$z_j = 0 \Rightarrow x_j^+ = 0, \qquad j \in J, \tag{OS1d}$$

$$z_j = 0 \Rightarrow x_j^- = 0, \qquad j \in J, \tag{OS1e}$$

$$z_j = 0 \Rightarrow v_j = 0, \qquad j \in J, \tag{OS1f}$$

$$z_j = 1 \Rightarrow v_j = u, \qquad j \in J, \tag{OS1g}$$

$$x_j^+ \geq 0, x_j^- \geq 0, y^+ \geq 0, y^- \geq 0, v_j \geq 0, u \geq 0, t_i^+ \geq 0, t_i^- \geq 0, z_j \in \{0,1\} \tag{OS1h}$$

In the computational experiment, we compare the performances of (6) and (OS1) with the one minute time limit. The synthetic instances from Section 5 of the online supplement with $n = 100$ and $m \in \{20, 30, 40, 50\}$ are used. The performances are measured using the criteria presented in Section 4: execution time, GAP_{IP}, and GAP_{sol}. Note that each (n, m) pair includes ten instances and the average results are presented for each m in Table OS 2.

	Formulation (6)			Formulation (OS1)		
m	Time	GAP_{IP}	GAP_{sol}	Time	GAP_{IP}	GAP_{sol}
20	1.4	0.0%	2.6%	3.6	0.0%	2.6%
30	33.6	0.2%	3.2%	49.9	1.0%	3.1%
40	61.7	4.6%	6.0%	61.4	9.0%	5.4%
50	62.6	13.2%	6.6%	62.0	18.7%	6.5%

Table OS 2: Performances of big M and logical constraint-based formulations

When $m = 20$, GAP_{IP} values are 0 and GAP_{sol} values are identical for the two formulations implying that all of the instances are solved optimally by the two approaches. The average execution time for (6) is 2.57 times faster than (OS1), but the absolute difference is not significant. When $m = 30$, formulation (6) solves 70% of the instances optimally, whereas formulation (OS1) returns an optimal solution for 40% of the instances. Further, the smaller GAP_{IP} and larger GAP_{sol} values show that (6) performs better. For all other instances with $m = 40$ and 50, both formulations do not return optimal solutions in one minute. However, because (6) provides smaller optimality gaps (GAP_{IP}) and larger relative gaps (GAP_{sol}) than (OS1), we conclude that (6) outperforms the logical constraint-based formulation.

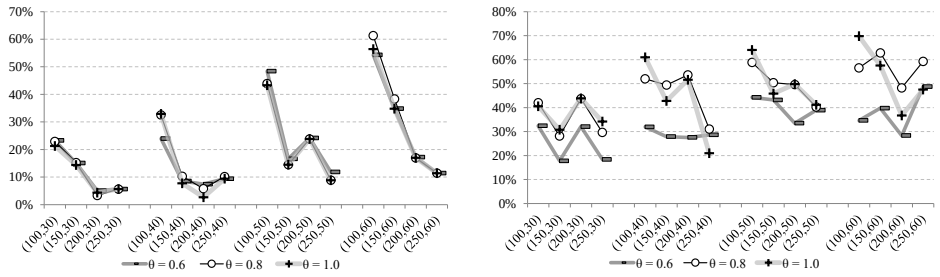
7 Best θ for Core Set Algorithms

Recall that both *Core-Heuristic* and *Core-Rand* take θ as input to decide the core set cardinality in (23). To decide the best θ value for each core set algorithm, we compare the performance of the algorithms with several θ values for selected instance sets.

For the *Core-Heuristic*, we test all 160 instances generated. In Figure OS 3, we plot the average GAP_{sol} across 16 instance sets for different θ values.

1. For MSE_a, there is no big difference between the three θ values. This is because the algorithm did not update the best solution after the first iteration and thus terminates the algorithm immediately. For this reason, the shape of the lines are very similar to those of the thin case result in Figure 2(a). We conclude that $\theta = 0.8$ is best for the *Core-Heuristic*, as it gives a slightly larger average improvement.

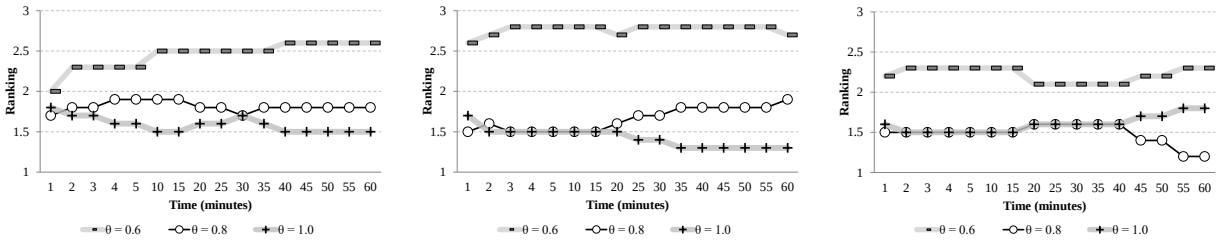
2. For MAE_a, on the other hand, the shape of the lines seems random although GAP_{sol} are generally increasing as instance size increases. We observe that $\theta = 1.0$ is best for (100,40),(100,50),(100,60) instances sets, and $\theta = 0.8$ is best for the other instance sets. Hence, we conclude that $\theta = 1$ is the best for instance sets satisfying $\{\frac{n}{m} \geq 0.4, n \leq 40\}$ or $\{\frac{n}{m} \geq 0.5, n > 40\}$. For all other instances, we conclude $\theta = 0.8$ is the best.



(a) MSE_a (b) MAE_a

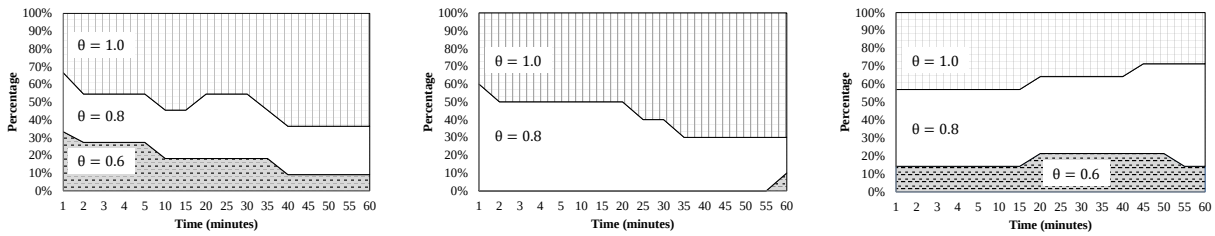
Figure OS 3: Average GAP$_{sol}$ of *Core-Heuristic* with different θ values

For *Core-Random*, we tested 30 instances in $\{(m,n)|(100,30),(150,40),(200,50)\}$, where each instance set contains 10 instances. The three instance sets are selected to represent varying instance size. We refer the instance sets $(100,30),(150,40),(200,50)$ as small, medium, large size, respectively. We only tested the algorithm with the MAE_a objective, since the algorithm behaves similarly for the MSE_a objective. Although *Core-Random* is designed to iterate infinitely, we executed it only for one hour since we are interested in if *Core-Random* defeats the MIP model with the same one hour time limit. We rank the θ values over time based on the best objective function value the algorithm gives with each θ.



(a) Small size (100,30) (b) Medium size (150,40) (c) Large size (200,50)

Figure OS 4: Average rankings of *Core-Random* over time with different θ values



(a) Small size (100,30) (b) Medium size (150,40) (c) Large size (200,50)

Figure OS 5: Precentage of the top-ranked of *Core-Random* over time with different θ values

In Figure OS 4, the average ranking for each θ is plotted. Note that the ranking is close to 1 if the algorithm with the corresponding θ gives the best objective function value out of the three θ values, and the ranking is close to 3 for the opposite case. Observe that $\theta = 1.0$ is best in general for the small and medium size instances. However, $\theta = 0.8$ outperforms $\theta = 1.0$ for the large size instances, especially after 40 minutes. This is because we have a larger number of iterations with $\theta = 0.8$ than with $\theta = 1.0$ in 1 hour, and $\theta = 0.8$ starts to take advantage after some time.

In order to check the performance from a different view, in Figure OS 5, we plot the area charts over time. Each area represents the percentage of the top ranked for each θ. The figure shows that $\theta = 1.0$ is best for the small and medium size instances, whereas $\theta = 0.8$ starts to outperform as iterations increases for the large size instances. This is in line with the observation from Figure OS 4.

Based on the observations from Figures OS 4 and OS 5, we conclude the following universal rule for the selection of θ for *Core-Random*.

1. With a 10 minute time limit, $\theta = 1.0$ is best for all sizes.
2. With a 1 hour time limit, $\theta = 0.8$ is best for large instances. Hence, with the one hour time limit, we use $\theta = 0.8$ if $mn \geq 9000$ and $\theta = 1.0$ otherwise.

8 Instance Generation Procedure

We generate synthetic instances by the following procedure.

1. We generate response variable $b_i \sim N(0, 5)$ for $i = 1, \cdots, n$.
2. Next, $\frac{m}{5}$ explanatory variables are generated, in which each variable is correlated to b with correlation coefficient $\rho = 0.2$.
3. For each already generated explanatory variable, we generate four explanatory variables that are correlated to the already generated variable with correlation coefficient $\rho \sim Uniform(0.5, 0.8)$.

This procedure generates $\frac{m}{5}$ groups of explanatory variables, in which five explanatory variables in each group are highly correlated to each other.

9 Study of Thin Case ($m < n$) for mRMR with Synthetic Data

In this section, we conduct two experiments for mRMR using the thin case synthetic instances from Section 4.2. Because there are four parameters to control for synthetic data, in the first experiment, we check the results with various p and λ values by fixing the data size m and n. In the second experiment, by fixing p to 10, the thin case synthetic instances with $m \in \{20, 30, 40\}$ and $n \in \{50, 60, \cdots, 100\}$ are used to test the performances with various m, n, and λ values. All of the performance measures in Section 4.3 of the main document are used.

The results are reported in Figures OS 6 - OS 9. In all figures, heatmaps are presented for each (m, n, λ) tuple. The rows are defined for each (m, n) pair and the columns are for λ values for MIP_{mix}. Each cell represents the average across ten instances except for Figure 7(b), which presents the percentages of optimally solved cases out of ten instances.

In the first experiment, ten random instances with $m = 30$ and $n = 100$ are used with parameters $p \in \{2, 4, \cdots, 14, 16\}$ and $\lambda \in \{0.05, 0.1, \cdots, 0.45, 0.5\}$ and the results are reported in Figures OS 6 - OS 9. For all MIP models, it takes less than one minute to terminate with an optimal solution. Hence, we do not report the execution times for the first experiment.

p \ λ	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	Avg
2	0.4	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
4	1.5	1.4	1.6	1.6	1.6	1.7	1.7	1.7	1.7	1.6	1.6
6	3.2	3.1	3.1	3.2	3.1	3.1	3.0	3.2	3.2	3.2	3.1
8	4.5	4.4	4.8	4.5	4.7	4.7	4.6	4.4	4.4	4.4	4.5
10	5.3	6.1	5.7	5.7	6.0	6.1	6.1	6.1	6.1	6.1	5.9
12	6.3	6.7	7.0	6.8	6.9	7.0	7.0	7.0	7.1	7.1	6.9
14	5.8	6.1	6.9	7.0	7.2	7.1	7.2	7.5	7.6	7.6	7.0
16	5.7	6.3	6.5	7.1	7.1	7.3	7.3	7.2	7.2	7.4	6.9
Avg	4.1	4.3	4.5	4.6	4.6	4.7	4.7	4.7	4.7	4.7	4.6

(a) Heatmap of SD_{mrmr}

p \ λ	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	Avg
2	0.5%	1.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	2.1%	1.8%
4	1.0%	2.3%	4.1%	5.7%	12.9%	13.7%	13.7%	13.7%	13.7%	17.4%	9.8%
6	2.0%	5.9%	6.2%	7.4%	10.3%	10.3%	11.1%	11.8%	11.8%	11.8%	8.9%
8	2.6%	5.8%	9.8%	11.9%	13.4%	13.4%	15.3%	16.9%	16.9%	16.9%	12.3%
10	3.2%	6.6%	9.9%	12.4%	17.3%	18.3%	19.6%	19.6%	19.6%	19.6%	14.6%
12	3.0%	8.0%	10.1%	15.2%	16.5%	17.5%	18.8%	18.8%	21.1%	21.1%	15.0%
14	3.9%	7.8%	11.6%	15.6%	16.3%	17.2%	19.4%	22.4%	23.7%	23.7%	16.2%
16	3.7%	8.3%	11.6%	13.4%	15.8%	17.7%	17.7%	19.2%	19.2%	20.9%	14.8%
Avg	2.5%	5.7%	8.2%	10.5%	13.1%	13.8%	14.7%	15.6%	16.0%	16.7%	11.7%

(b) Heatmap of GAP_{mrmr}

p \ λ	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	Avg
2	0.5	0.5	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
4	1.5	1.2	1.4	1.3	0.7	0.7	0.7	0.7	0.7	0.6	1.0
6	2.0	1.0	1.1	0.8	0.6	0.6	0.5	0.6	0.6	0.7	0.9
8	2.4	2.2	1.7	0.8	0.5	0.5	0.4	0.0	0.0	0.0	0.9
10	4.9	2.6	2.4	2.0	1.1	1.3	0.5	0.5	0.5	0.6	1.6
12	4.3	3.2	3.1	1.5	1.1	1.1	0.9	0.9	0.8	0.8	1.8
14	3.9	3.1	2.3	1.9	1.7	1.6	1.3	0.7	0.3	0.3	1.7
16	4.7	2.9	2.1	1.6	1.3	1.1	1.1	1.2	1.2	0.7	1.8
Avg	3.0	2.1	1.8	1.3	0.9	0.9	0.7	0.6	0.6	0.5	1.2

(c) Heatmap of SD_{sae}

p \ λ	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	Avg
2	1.3%	1.2%	0.9%	0.9%	0.9%	0.9%	0.9%	0.9%	0.9%	0.9%	1.0%
4	2.9%	2.4%	2.3%	2.2%	1.5%	1.3%	1.3%	1.3%	1.3%	1.2%	1.8%
6	2.4%	1.2%	1.2%	1.0%	0.7%	0.7%	0.2%	0.1%	0.1%	0.1%	0.8%
8	1.7%	1.6%	0.9%	0.7%	0.2%	0.2%	0.2%	0.0%	0.0%	0.0%	0.6%
10	2.7%	1.5%	0.8%	0.7%	0.3%	0.2%	0.1%	0.1%	0.1%	0.1%	0.7%
12	2.3%	1.1%	0.6%	0.3%	0.2%	0.1%	0.1%	0.1%	0.1%	0.1%	0.5%
14	2.1%	1.2%	0.8%	0.3%	0.3%	0.3%	0.2%	0.1%	0.0%	0.0%	0.5%
16	1.9%	0.9%	0.4%	0.3%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.4%
Avg	2.2%	1.4%	1.0%	0.8%	0.5%	0.5%	0.4%	0.3%	0.3%	0.3%	0.8%

(d) Heatmap of GAP_{sae}

Figure OS 6: Comparison of MIP_{mrmr} and MIP_{mix} with fixed $m = 30$ and $n = 100$

The heatmaps in Figure OS 6 show that SD_{mrmr} and GAP_{mrmr} increase as p and λ increase because larger p and λ values allow and force S_{mix} to deviate from S_{mrmr}. On the other hand, GAP_{sae} decreases in increasing p and λ because larger p and λ values allow and force S_{mix} to be close to S_{sae}. When λ is small, SD_{sae} increases as p increases because there exist more alternative subsets.

In the second experiment, we use fixed $p = 10$ and penalty constant $\lambda \in \{0.05, 0.1, \cdots, 0.45, 0.5\}$, and one hour time limit. Note that small λ value means the solution to (20) should have closer mRMR objective to $\bar{\Omega}$ while sacrificing SAE, and large λ value means the solution to (20) focuses more on minimizing SAE. For each (m, n, λ) tuple, ten random instances are solved. For both of MIP_{mrmr} and MIP_{sae}, CPLEX terminated optimality within one hour for all cases. On the other hand, MIP_{mix} terminated without optimality within one hour for several cases. Hence, we report the percentage of non-optimal cases for MIP_{mix}.

In Figure OS 7(a), we observe that the execution times of all three models increase drastically as m and n increase. The execution time of MIP_{mix} increases in increasing λ values. This is because the feasibility space of MIP_{mix} is larger when λ is large. As λ increases, constraint (20d) becomes less restrictive. In Figure IS 7(b), we can also observe that the percentage of cases where MIP_{mix} cannot be solved optimally increases as m, n, and λ increase. Among all parameters m, n, and λ, we conclude that the number of features (m) affects the execution time most. Among the three models, MIP_{mrmr} is the fastest and MIP_{mix} is the slowest.

In Figure OS 8, MIP_{mrmr} and MIP_{mix} are compared. As m and λ increase, SD_{mrmr} increases and the selected subsets become more distinct. When m is large, there are more alternative subsets given fixed λ and this increases SD_{mrmr}. When λ is large, the feasible space of (20) is larger and S_{mix} can be significantly different from S_{mrmr}. The heatmap for GAP_{mrmr} in Figure OS 8(b) also shows the same trend. Increasing λ decreases the mRMR value.

In Figure OS 9, MIP_{sae} and MIP_{mix} are compared. As m increases, SD_{sae} increases and the selected subsets become more distinct, because there are more alternative subsets given fixed λ. As λ increases, constraint (20d) becomes less restrictive and SD_{sae} and GAP_{sae} values approach zero.

(m,n)	MIP mrmr	MIP sae	MIP_{mix} (λ values)										
			0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	Avg
(20,50)	0.4	0.1	0.4	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
(30,50)	0.0	1.7	0.4	0.7	1.1	1.9	3.0	3.6	3.7	3.9	4.3	4.6	2.7
(40,50)	9.0	180.6	114.9	209.3	301.2	429.3	468.4	497.9	556.4	898.7	670.9	643.8	479.1
(20,60)	0.4	0.1	0.5	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
(30,60)	4.0	2.7	12.0	8.3	9.2	10.2	9.6	8.9	7.8	7.6	7.6	7.7	8.9
(40,60)	11.9	337.4	319.7	359.2	482.2	362.1	743.5	1013.0	776.3	842.9	1004.1	973.1	687.6
(20,70)	0.7	0.2	0.3	0.3	0.3	0.4	0.4	0.3	0.3	0.3	0.3	0.3	0.3
(30,70)	4.2	3.0	18.4	15.6	8.0	9.5	9.3	8.4	7.8	7.9	7.6	7.8	10.0
(40,70)	22.9	220.1	409.5	508.9	422.5	711.2	454.0	839.2	692.0	555.7	714.2	861.0	616.8
(20,80)	0.5	0.2	0.5	0.4	0.4	0.4	0.5	0.4	0.4	0.5	0.4	0.4	0.4
(30,80)	5.4	6.7	13.3	11.1	14.0	12.0	13.9	15.9	15.5	17.2	17.3	18.8	14.9
(40,80)	20.4	417.0	401.5	910.7	705.2	942.6	977.0	1476.9	995.9	1533.7	1200.7	1209.8	1035.4
(20,90)	0.5	0.2	0.5	0.4	0.4	0.4	0.3	0.3	0.3	0.3	0.4	0.3	0.4
(30,90)	4.6	6.3	19.3	14.6	14.1	15.7	12.8	14.3	12.2	14.9	15.5	14.1	14.8
(40,90)	22.7	239.0	257.0	791.4	689.2	844.4	780.6	795.9	808.2	904.8	1087.1	1008.7	796.7
(20,100)	0.6	0.2	0.6	0.6	0.5	0.4	0.4	0.4	0.5	0.4	0.5	0.5	0.5
(30,100)	6.2	5.9	25.7	15.2	13.2	13.1	13.2	15.1	10.9	12.8	12.8	14.3	14.6
(40,100)	16.8	462.5	272.5	409.9	495.7	759.2	910.1	1044.8	1068.2	1117.1	1141.3	1147.2	836.6
Avg	7.3	104.7	103.7	181.0	175.4	228.5	244.3	318.7	275.4	328.9	327.0	328.5	251.1

(a) Heatmap of Execution times

$(m,n)\backslash\lambda$	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	Avg
(20,50)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
(30,50)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
(40,50)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
(20,60)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
(30,60)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
(40,60)	0%	0%	0%	0%	10%	0%	0%	10%	10%	10%	4%
(20,70)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
(30,70)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
(40,70)	0%	0%	0%	0%	0%	10%	0%	0%	0%	0%	1%
(20,80)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
(30,80)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
(40,80)	0%	10%	0%	10%	0%	10%	10%	20%	10%	10%	8%
(20,90)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
(30,90)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
(40,90)	0%	0%	0%	10%	0%	0%	0%	0%	0%	0%	1%
(20,100)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
(30,100)	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
(40,100)	0%	0%	0%	0%	10%	20%	20%	20%	20%	20%	11%
Avg	0%	1%	0%	1%	1%	2%	2%	3%	2%	2%	1%

(b) Heatmap of Non-Optimal MIP_{mix}

Figure OS 7: Execution times and Optimality of MIP_{mrmr}, MIP_{sae}, and MIP_{mix} with fixed $p = 10$

$(m,n)\backslash\lambda$	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	Avg
(20,50)	3.9	4.5	4.7	4.8	5.4	5.0	5.1	5.0	5.1	5.0	4.9
(30,50)	1.4	1.9	2.2	2.9	3.2	3.7	3.8	3.9	4.1	4.2	3.1
(40,50)	6.1	6.0	6.0	6.4	6.3	6.5	6.3	6.2	6.4	6.5	6.3
(20,60)	4.2	4.1	4.3	4.4	4.6	4.7	4.7	4.6	4.2	4.8	4.5
(30,60)	4.8	4.8	5.1	5.1	5.3	5.2	5.4	5.2	5.4	5.5	5.2
(40,60)	4.7	5.7	5.7	5.8	6.1	6.3	6.5	6.6	6.6	6.7	6.1
(20,70)	3.6	4.0	4.0	4.2	4.2	4.3	4.4	4.6	4.6	4.6	4.3
(30,70)	4.1	5.0	5.5	5.6	5.8	5.7	5.7	5.9	5.9	5.9	5.5
(40,70)	6.1	6.4	6.5	6.6	6.7	6.7	6.5	6.9	7.0	7.1	6.7
(20,80)	3.2	3.5	3.8	3.8	3.8	4.0	4.1	4.1	4.1	4.1	3.9
(30,80)	5.4	5.9	5.5	5.4	5.8	5.9	5.9	5.9	5.9	5.9	5.8
(40,80)	5.7	6.2	6.1	5.9	6.1	6.0	6.1	6.1	6.2	6.2	6.1
(20,90)	4.4	4.5	4.5	4.4	4.4	4.6	4.6	4.6	4.9	5.0	4.6
(30,90)	4.9	5.6	6.0	6.3	6.3	6.3	6.6	6.6	6.6	6.6	6.2
(40,90)	5.7	6.0	6.1	6.2	5.9	6.0	6.0	6.1	6.0	6.1	6.0
(20,100)	4.5	4.7	4.6	5.0	4.9	4.9	5.0	5.0	5.0	5.0	4.9
(30,100)	5.3	6.1	5.7	5.7	6.0	6.1	6.1	6.1	6.1	6.1	5.9
(40,100)	5.3	5.7	5.8	5.6	6.0	6.1	6.2	6.3	6.3	6.4	6.0
Avg	4.6	5.0	5.1	5.2	5.4	5.4	5.5	5.5	5.6	5.7	5.3

(a) Heatmap of SD_{mrmr}

$(m,n)\backslash\lambda$	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	Avg
(20,50)	3%	7%	9%	12%	15%	15%	18%	18%	19%	19%	13%
(30,50)	3%	8%	13%	18%	21%	25%	27%	29%	32%	33%	21%
(40,50)	3%	7%	9%	14%	17%	21%	23%	24%	27%	28%	17%
(20,60)	3%	6%	8%	10%	13%	19%	17%	19%	18%	20%	13%
(30,60)	2%	7%	10%	12%	18%	22%	24%	25%	26%	31%	18%
(40,60)	4%	8%	12%	17%	19%	24%	26%	26%	28%	30%	19%
(20,70)	2%	4%	8%	9%	9%	17%	18%	20%	21%	21%	13%
(30,70)	3%	7%	12%	12%	15%	20%	22%	24%	24%	24%	16%
(40,70)	4%	7%	13%	16%	20%	21%	25%	27%	28%	28%	19%
(20,80)	3%	5%	9%	12%	13%	15%	17%	18%	18%	18%	13%
(30,80)	4%	7%	9%	13%	15%	16%	16%	16%	16%	16%	13%
(40,80)	4%	8%	10%	14%	17%	18%	20%	20%	22%	22%	15%
(20,90)	1%	6%	7%	9%	10%	13%	13%	13%	15%	19%	11%
(30,90)	3%	7%	9%	13%	17%	20%	21%	21%	21%	21%	15%
(40,90)	4%	7%	11%	14%	18%	19%	20%	21%	22%	22%	16%
(20,100)	2%	5%	6%	9%	10%	11%	13%	13%	13%	13%	9%
(30,100)	3%	7%	10%	12%	17%	18%	20%	20%	20%	20%	15%
(40,100)	3%	7%	12%	16%	17%	20%	22%	25%	25%	26%	17%
Avg	3%	7%	10%	13%	16%	19%	20%	21%	22%	23%	15%

(b) Heatmap of GAP_{mrmr}

Figure OS 8: Comparison of MIP_{mrmr} and MIP_{mix} with fixed $p = 10$

$(m,n)\backslash\lambda$	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	Avg
(20,50)	2.8	1.9	1.7	1.7	1.3	1.2	1.3	1.3	1.2	0.9	1.5
(30,50)	4.0	3.4	3.4	2.8	2.2	1.8	1.5	0.8	0.3	0.4	2.1
(40,50)	5.5	4.5	4.1	2.7	1.6	1.5	0.9	0.8	0.6	0.5	2.3
(20,60)	3.2	2.6	2.0	1.7	1.8	1.4	0.9	0.8	1.1	0.8	1.6
(30,60)	5.1	4.2	3.3	3.0	2.7	2.0	1.4	1.1	1.2	1.1	2.5
(40,60)	5.2	5.5	4.9	3.1	2.7	2.1	0.8	1.4	1.0	0.3	2.7
(20,70)	2.0	2.0	1.8	1.3	1.3	1.1	1.1	0.9	0.9	0.9	1.3
(30,70)	4.4	3.3	2.4	2.3	2.1	0.6	0.5	0.2	0.2	0.2	1.6
(40,70)	5.3	5.4	4.6	4.4	2.6	2.7	1.6	0.7	0.5	0.8	2.9
(20,80)	2.5	1.2	1.2	0.8	0.8	0.4	0.3	0.2	0.2	0.2	0.8
(30,80)	3.3	2.6	1.7	0.9	0.5	0.0	0.0	0.0	0.0	0.0	0.9
(40,80)	4.1	3.8	3.2	1.8	1.2	1.0	0.7	0.6	0.2	0.3	1.7
(20,90)	2.1	1.8	1.6	1.0	0.8	0.6	0.6	0.6	0.3	0.0	0.9
(30,90)	4.5	3.1	2.9	2.1	1.2	0.6	0.4	0.4	0.4	0.4	1.6
(40,90)	4.6	3.8	3.0	1.7	1.0	0.6	0.3	0.1	0.1	0.0	1.5
(20,100)	1.6	1.3	1.2	0.8	0.8	0.7	0.6	0.6	0.6	0.6	0.9
(30,100)	4.9	2.6	2.4	2.0	1.1	1.3	0.5	0.5	0.5	0.6	1.6
(40,100)	5.3	4.6	3.9	2.2	1.3	0.7	0.5	0.1	0.1	0.0	1.9
Avg	3.9	3.2	2.7	2.0	1.5	1.1	0.8	0.6	0.5	0.4	1.7

(a) Heatmap of SD_{sae}

$(m,n)\backslash\lambda$	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	Avg
(20,50)	6%	2%	2%	2%	1%	1%	1%	1%	1%	1%	2%
(30,50)	10%	6%	4%	2%	1%	1%	1%	0%	0%	0%	3%
(40,50)	8%	6%	5%	4%	2%	2%	1%	1%	1%	0%	3%
(20,60)	4%	3%	2%	2%	2%	1%	1%	1%	1%	1%	2%
(30,60)	9%	5%	4%	4%	3%	2%	2%	2%	1%	1%	3%
(40,60)	9%	5%	4%	2%	2%	1%	1%	1%	0%	0%	3%
(20,70)	2%	1%	1%	1%	1%	0%	0%	0%	0%	0%	1%
(30,70)	5%	3%	1%	1%	1%	0%	0%	0%	0%	0%	1%
(40,70)	7%	5%	3%	3%	2%	1%	1%	0%	0%	0%	2%
(20,80)	3%	1%	1%	1%	1%	0%	0%	0%	0%	0%	1%
(30,80)	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%
(40,80)	4%	2%	2%	1%	1%	1%	1%	1%	0%	0%	1%
(20,90)	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%
(30,90)	3%	2%	1%	1%	1%	0%	0%	0%	0%	0%	1%
(40,90)	4%	3%	2%	1%	1%	0%	0%	0%	0%	0%	1%
(20,100)	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
(30,100)	3%	2%	1%	1%	0%	0%	0%	0%	0%	0%	1%
(40,100)	4%	3%	1%	0%	0%	0%	0%	0%	0%	0%	1%
Avg	5%	3%	2%	1%	1%	1%	0%	0%	0%	0%	1%

(b) Heatmap of GAP_{sae}

Figure OS 9: Comparison of MIP_{sae} and MIP_{mix} with fixed $p = 10$