Overview:

The goal of this project is to outperform existing techniques in the literature related to nonmonotone missing data.

Initial Simulations:

- **Implemented simulation of monotone MAR data**: This is correspondingly easier than the subsequent nonmonotone MAR simulation. For this simulation we use the following approach:

  1. Generate X, Y_1, and Y_2 for elements $i = 1, \ldots, n$.
  2. Using the covariate X, determine the probability p_1 of Y_1 being observed for each element i.
  3. Based on p_1, determine if $R_1 = 1$.
  4. If $R_1 = 0$, then $R_2 = 0$. Otherwise, using variables X and Y_1, determine the probability p_{12}.
  5. Based on p_{12} determine if $R_2 = 1$.

  At the end of the algorithm, we have determined the values of binary variables R_1 and R_2 for each i and if either of them are equal to 1, the corresponding level of Y_k. As is common in this literature, the values of R_1 and R_2 determine if the corresponding variable Y_1 or Y_2 is missing or observed with $R = 1$ indicating Y being observed.

- **Implemented simulation of nonmonotone MAR data**: Following the approach of [1], I construct a nonmonotone MAR simulation with two response variables Y_1 and Y_2 and one covariate X. The algorithm to generate the data is the following:

  1. Generate X, Y_1, and Y_2 for elements $i = 1, \ldots, n$.
  2. Using the covariate X_i, generate probabilities for each element i p_0, p_1, and p_2 such that $p_0 + p_1 + p_2 = 1$.
  3. Select one option based on the three probabilities for each element i. If 0 is selected: $R_1 = 0$ and $R_2 = 0$. If 1 is selected $R_1 = 1$. If 2 is selected, $R_2 = 1$.
  4. We take the next step in multiple cases. If 0 was selected, we are done. If 1 was selected, we generate probabilities p_{12} based on X and Y_1. Then based on this probability, we determine if $R_2 = 1$. In the same manner, if 2 was selected in the previous step, we generate probabilities p_{21} based on X and Y_2. Then based on this probability, we determine if $R_2 = 1$.

  Like the monotone MAR simulation this algorithm produces similar final results with the determination of binary variables R_1 and R_2 and variables X, Y_1, and Y_2. Unlike the monotone MAR case, the nonmonotone MAR includes observations with Y_2 observed and Y_1 missing.

- **Simulation 1 with Monotone MAR**: Following the algorithm described in the monotone MAR simulation bullet, we first generate data from the following distributions:

$$X_i \overset{iid}{\sim} N(0,1)$$
$$Y_{1i} \overset{iid}{\sim} N(0,1)$$
$$Y_{2i} \overset{iid}{\sim} N(\theta,1)$$

  Then, we create the probabilities $p_1 = \text{logistic}(x_i)$ and $p_{12} = \text{logistic}(y_{1i})$. Since, both x_i and y_1 are standard normal distributions, each of these probabilities is approximately 0.5 in expectation.

  The goal of this simulation is to estimate θ. Alternatively, we can express this as solving the estimating equation:

$$g(\theta) \equiv Y_2 - \theta = 0.$$

  We estimate θ using the following procedures:

  - Oracle: This computes $\bar{Y}$ using *both* the observed and missing data.
  - IPW-Oracle: This is an IPW estimator using only the observed values of Y_2. The weights (inverse probabilities) use the actual probabilities.
  - IPW-Est: This is an IPW estimator using the probabilities that have been estimated by a logistic model.
  - Semi: This is the monotone semiparametric efficient estimator from Slide 11 (Equation 2) of Dr. Kim's Nonmonotone Missingness presentation.

  We run this simulation with different values of θ, sample size of 2000, and 2000 Monte Carlo replications. Each algorithm for each replication generates $\hat{\theta}$. In the subsequent tables, we compute the bias, standard deviation (sd), t-statistic (where we test for a significant difference between the Monte Carlo mean $\hat{\theta}$ and the true θ) and the p-value of the t-statistic.

Table 1: True Value is -5

algorithm	bias	sd	tstat	pval
oracle	0.001	0.033	0.680	0.248
ipworacle	-0.012	0.392	-0.973	0.165
ipwest	0.007	0.186	1.178	0.120
semi	0.001	0.074	0.538	0.295

Table 2: True Value is 0

algorithm	bias	sd	tstat	pval
oracle	-0.001	0.031	-1.091	0.138
ipworacle	-0.001	0.085	-0.201	0.420
ipwest	0.000	0.085	-0.029	0.488
semi	0.000	0.079	0.112	0.455

Table 3: True Value is 5

algorithm	bias	sd	tstat	pval
oracle	0.000	0.033	-0.468	0.320
ipworacle	0.010	0.383	0.857	0.196
ipwest	-0.006	0.176	-1.020	0.154
semi	0.000	0.077	-0.049	0.481

Overall, these results are mostly what I would have expected. All of the algorithms estimate the true value of θ correctly in each case, with the oracle estimate having the smallest variance followed by the semiparametric algorithm. If there is anything surprising it is that the IPW estimator has better performance with the estimated weights compared to the true weights. However, I think that this is a known phenomenon.

- **Simulation 1 with Nonmonotone MAR**:

  We generate variables (X, Y_1, Y_2) using the following setup:

$$\begin{bmatrix} X_i \\ \varepsilon_{1i} \\ \varepsilon_{2i} \end{bmatrix} \overset{iid}{\sim} N\left(\begin{bmatrix} 0 \\ 0 \\ \theta \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \sigma_{yy} \\ 0 & \sigma_{yy} & 1 \end{bmatrix} \right).$$

  Then,

$$y_{1i} = x_i + \varepsilon_{1i} \text{ and } y_{2i} = x_i + \varepsilon_{2i}.$$

  Since we have nonmonotone data, our "Stage 1" probabilities are different. We compute the true Stage 1 probabilities being proportional to the following values:

$$p_0 = 0.2$$
$$p_1 = 0.4$$
$$p_2 = 0.4$$

  However, we keep the same structure for the Stage 2 probabilities with: $p_{12} = \text{logistic}(y_1)$ and $p_{21} = \text{logistic}(y_2)$. The goal remains to estimate θ. We continue to use the Oracle algorithm and the IPW-Oracle algorithm. Since we have nonmonotone MAR data, we use the "Proposed" algorithm that is described on Slide 25 (Equation 12) of Dr. Kim's presentation. The outcome models were estimated using logistic regression and OLS and correctly specified. The response model used the oracle estimates of the probabilities. This yields the following results:

Table 4: True Value is -5. Cor(Y1, Y2) = 0

algorithm	bias	sd	tstat	pval
oracle	0.000	0.032	0.285	0.388
ipworacle	-0.003	0.381	-0.318	0.375
proposed	0.000	0.038	0.492	0.311

Table 5: True Value is 0. Cor(Y1, Y2) = 0

algorithm	bias	sd	tstat	pval
oracle	0.000	0.032	0.285	0.388
ipworacle	0.000	0.076	-0.237	0.406
proposed	0.001	0.038	0.894	0.186

Table 6: True Value is 5. Cor(Y1, Y2) = 0

algorithm	bias	sd	tstat	pval
oracle	0.000	0.032	0.285	0.388
ipworacle	-0.001	0.098	-0.479	0.316
proposed	0.000	0.037	0.505	0.307

- **Simulation 2 with Nonmonotone MAR**: We also want to simulate data that is correlated. For this simulation, we focus on $\text{Cov}(Y_1, Y_2)$. The data generating process now has $\sigma_{yy} \neq 0$. We are still interested in $\bar{Y}_2$ and we still run 2000 simulation with 2000 observations. In all of the next simulations the true value of $\theta = 0$. The results are the following:

Table 7: True Value is 0. Cor(Y1, Y2) = 0.1

algorithm	bias	sd	tstat	pval
oracle	0.001	0.031	1.623	0.052
ipworacle	0.001	0.077	0.762	0.223
proposed	0.001	0.037	1.366	0.086

Table 8: True Value is 0. Cor(Y1, Y2) = 0.5

algorithm	bias	sd	tstat	pval
oracle	0.001	0.032	1.486	0.069
ipworacle	0.004	0.086	1.890	0.029
proposed	0.000	0.041	0.172	0.432

Table 9: True Value is 0. Cor(Y1, Y2) = 0.9

algorithm	bias	sd	tstat	pval
oracle	0.001	0.032	0.706	0.240
ipworacle	0.003	0.098	1.395	0.082
proposed	-0.002	0.062	-1.339	0.090

- **Simulation 3 with Nonmonotone MAR**: This simulation aims to see if the proposed algorithm is doubly robust. First, we check with a misspecified outcome model. In this case the data generating procedure is the following:

$$\begin{bmatrix} X_i \\ \varepsilon_{1i} \\ \varepsilon_{2i} \end{bmatrix} \overset{iid}{\sim} N\left(\begin{bmatrix} 0 \\ 0 \\ \theta \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \sigma_{yy} \\ 0 & \sigma_{yy} & 1 \end{bmatrix} \right).$$

Then,

$$y_{1i} = x_i + x_i^2 \varepsilon_{1i} \text{ and } y_{2i} = -x_i + x_i^3 + \varepsilon_{2i}.$$

This procedure causes X to influence both Y_1 and Y_2 and we still have correlation in the error terms of Y_1 and Y_2. However, since neither Y_1 nor Y_2 are linear in X, the model will be misspecified. The response mechanisms are first generated MCAR with a probability of either Y_1 or Y_2 being the first variable observed to be 0.4. (There is a 0.2 probability neither is observed.) Then the probability of the other variable being observed is proportional to logistic(y_k) where y_k is the y that has been observed. To ensure that the proposed method has the correct propensity score we use the oracle probabilities instead of estimating them. This yields the following:

Table 10: True Value is 0. Cor(Y1, Y2) = 0

algorithm	bias	sd	tstat	pval
oracle	0.000	0.075	0.014	0.494
ipworacle	0.002	0.107	0.876	0.191
proposed	-0.002	0.084	-1.063	0.144

Table 11: True Value is 0. Cor(Y1, Y2) = 0.1

algorithm	bias	sd	tstat	pval
oracle	-0.002	0.074	-1.479	0.070
ipworacle	0.000	0.106	-0.196	0.422
proposed	-0.003	0.083	-1.464	0.072

Table 12: True Value is 0. Cor(Y1, Y2) = 0.5

algorithm	bias	sd	tstat	pval
oracle	-0.003	0.074	-1.567	0.059
ipworacle	-0.002	0.108	-0.818	0.207
proposed	-0.003	0.083	-1.633	0.051

Thus, the proposed method is unbiased with a misspecified outcome model. We now show a simulation where the outcome model is correctly specified but the response model is not.

- **Simulation 4 with Nonmonotone MAR**: Continuing to test if the proposed algorithm is doubly robust, this simulation checks a misspecified response model. Instead of using oracle weights as in Simulation 3, we estimate the weights for the proposed method. However, unlike the true probabilities of being proportional to logistic(y_k), this simulation has the true probabilities being proportional to logistic(x_i). The algorithms to which we compare still use the oracle weights.

Table 13: True Value is 0. Cor(Y1, Y2) = 0

algorithm	bias	sd	tstat	pval
oracle	0.000	0.032	-0.318	0.375
ipworacle	0.002	0.066	1.179	0.119
proposed	0.000	0.036	0.012	0.495

Table 14: True Value is 0. Cor(Y1, Y2) = 0.1

algorithm	bias	sd	tstat	pval
oracle	0	0.031	0.394	0.347
ipworacle	0	0.065	-0.230	0.409
proposed	0	0.035	-0.082	0.467

Table 15: True Value is 0. Cor(Y1, Y2) = 0.5

algorithm	bias	sd	tstat	pval
oracle	0	0.031	0.318	0.375
ipworacle	0	0.065	-0.094	0.462
proposed	0	0.036	-0.056	0.478

Thus, there is strong evidence that the proposed method is doubly robust. Previously, we did not get this result and the reason is the way that the selection probabilities work. When they were proportional to logistic(y_k), the estimator is biased (see the next section); however, when the probabilities are proportional to logistic(x_i) then it works.

Missingness Mechanism

- First I am going to reproduce the proof of double robustness that we talked about during our last meeting. I think it is insightful for future comments:

$$
\begin{aligned}
E[\hat{\theta}_{eff} - \theta_n] &= E\left[n^{-1}\sum_{i=1}^{n} E[g_i \mid X_i] - g_i\right] \\
&\quad + E\left[n^{-1}\sum_{i=1}^{n} \frac{R_{1i}}{\pi_{1+}(X_i)}(b_2(X_i, Y_{1i}) - E[g_i \mid X_i])\right] \\
&\quad + E\left[n^{-1}\sum_{i=1}^{n} \frac{R_{2i}}{\pi_{2+}(X_i)}(a_2(X_i, Y_{2i}) - E[g_i \mid X_i])\right] \\
&\quad + E\left[n^{-1}\sum_{i=1}^{n} \frac{R_{1i}R_{2i}}{\pi_{11}(X_i)}(g_i - a_2(X_i, Y_{2i}) - b_2(X_i, Y_{1i}) + E[g_i \mid X_i])\right] \\
&= n^{-1}\sum_{i=1}^{n}(E[E[g_i \mid X_i]] - E[g_i]) \\
&\quad + n^{-1}\sum_{i=1}^{n} E\left[E\left[\frac{R_{1i}}{\pi_{1+}(X_i)}(b_2(X_i, Y_{1i}) - E[g_i \mid X_i]) \mid X_i\right]\right] \\
&\quad + n^{-1}\sum_{i=1}^{n} E\left[E\left[\frac{R_{2i}}{\pi_{2+}(X_i)}(a_2(X_i, Y_{2i}) - E[g_i \mid X_i]) \mid X_i\right]\right] \\
&\quad + n^{-1}\sum_{i=1}^{n} E\left[E\left[\frac{R_{1i}R_{2i}}{\pi_{11}(X_i)}(g_i - a_2(X_i, Y_{2i}) - b_2(X_i, Y_{1i}) + E[g_i \mid X_i]) \mid X_i\right]\right]
\end{aligned}
$$

Since $R_{1i} \perp Y_{1i} \mid X_i$, $R_{2i} \perp Y_{2i} \mid X_i$, $(R_{1i}, R_{2i}) \perp (Y_{1i}, Y_{2i}) \mid X_i$ and π_{1+}, π_{2+} and π_{11} are all free of Y_{1i} and Y_{2i}.

$$
\begin{aligned}
&= n^{-1}\sum_{i=1}^{n} E\left[\frac{R_{1i}}{\pi_{1+}(X_i)} E\left[(E[g_i \mid X_i, Y_{1i}] - E[g_i \mid X_i]) \mid X_i\right]\right] \\
&\quad + n^{-1}\sum_{i=1}^{n} E\left[\frac{R_{2i}}{\pi_{2+}(X_i)} E\left[E[g_i \mid X_i, Y_{2i}] - E[g_i \mid X_i] \mid X_i\right]\right] \\
&\quad + n^{-1}\sum_{i=1}^{n} E\left[\frac{R_{1i}R_{2i}}{\pi_{11}(X_i)} E\left[(g_i - E[g_i \mid X_i, Y_{2i}] - E[g_i \mid X_i, Y_{1i}] + E[g_i \mid X_i]) \mid X_i\right]\right]
\end{aligned}
$$

Since $E[E[g_i \mid X_i, Y_{ki}] \mid X_i] = E[g_i \mid X_i] = 0$,

$$
= 0.
$$

Thus, if the outcome models are correctly specified $\hat{\theta}_{eff}$ is unbiased. If the response models are correctly specified it is easy to see that $\hat{\theta}_{eff}$ is also unbiased. This means that $\hat{\theta}_{eff}$ is doubly robust.

- However, one of the key steps is that *all* of the response models are free of Y. In a previous iteration of Simulation 4, we had adopted the framework of [1] where we first to observe the first variable and see if we observe the second variable. In this case, the second step can depend on the result of the first step and this is what we did. However, this makes it the case that $\pi_1 1$ is a function of X_i and Y_1 and Y_2. In this case $\hat{\theta}_{eff}$ is not unbiased if the reponse model is misspecified (or even just estimated).

- If we modify Simulation 4, such that the second step of observed the second varible is proportional to logistic(y_k) then we get the same result as before:

Table 16: True Value is 0. Cor(Y1, Y2) = 0

algorithm	bias	sd	tstat	pval
oracle	0.000	0.032	-0.318	0.375
ipworacle	-0.001	0.079	-0.475	0.317
proposed	0.002	0.037	2.851	0.002

Table 17: True Value is 0. Cor(Y1, Y2) = 0.1

algorithm	bias	sd	tstat	pval
oracle	0.000	0.031	0.394	0.347
ipworacle	0.001	0.082	0.560	0.288
proposed	0.008	0.037	9.204	0.000

Table 18: True Value is 0. Cor(Y1, Y2) = 0.5

algorithm	bias	sd	tstat	pval
oracle	0.000	0.031	0.318	0.375
ipworacle	0.001	0.093	0.683	0.247
proposed	0.017	0.039	19.062	0.000

Estimating the Response Model

- In most (critically *not* Simulation 4) of the simulations of the previous section, we used the oracle weights when estimating our proposed method. The reasoning for this was straightforward–we wanted to ensure that the response model was correctly specified and the best case scenario is to use the oracle weights which we did.
- This section focuses more on estimating these response weights. Instead of focusing on the proposed method, we will actually be working on estimation of the complete case IPW estimator. This model is less complex and will thus make it easier to understand where we are making mistakes.
- We use the following simulation study:

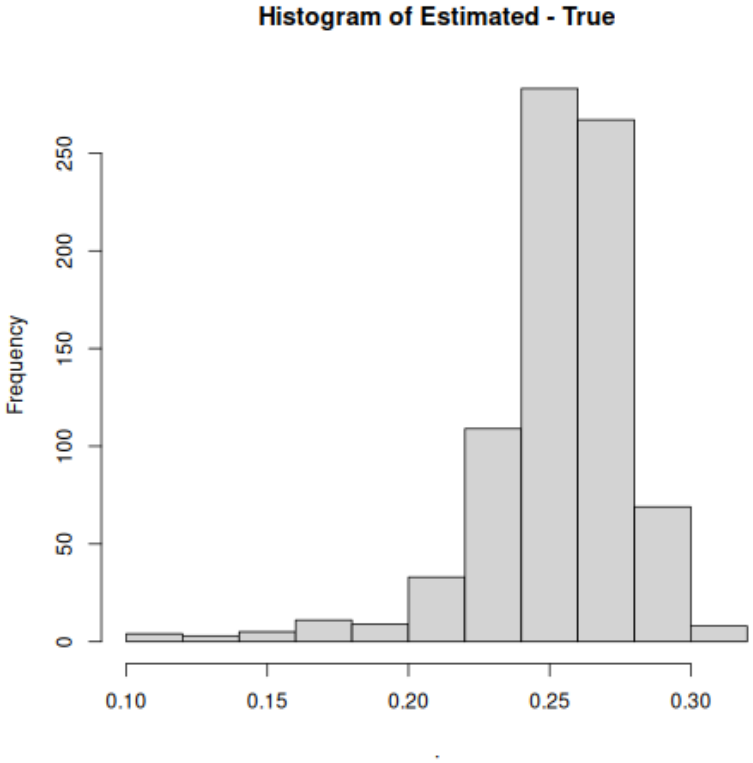
$$
\begin{aligned}
x_i &\overset{iid}{\sim} N(0,1) \\
\varepsilon_{1i} &\overset{iid}{\sim} N(0,1) \\
\varepsilon_{2i} &\overset{iid}{\sim} N(0,1) \\
y_{1i} &= x_i + \varepsilon_{1i} \\
y_{2i} &= x_i + \varepsilon_{2i}
\end{aligned}
$$

To select a missingness pattern for each i, we have sequence: first, we select the first variable to observe (or neither), then we either select the second variable or we do not. In the first step, we select $R_{1i} = 1$ with probability 0.4, $R_{2i} = 1$ with probability 0.4 and neither variable with probability 0.2. For the second step, we have the probability of observing the other variable be logistic(x_i). This yields the following:

Table 19: True Value is 0. Cor(Y1, Y2) = 0

algorithm	bias	sd	tstat	pval
oracle	0.000	0.032	-0.318	0.375
ipw.or	0.002	0.067	1.140	0.127
ipw.0	0.083	0.032	116.911	0.000

The problem with this is that our estimate is biased. For one particular realization of the simulation, here is the distribution of the difference between the estimated and true probabilities of π_{11}:


Histogram of Estimated - True
Frequency
0
50
100
150
200
250
0.10
0.15
0.20
0.25
0.30

Proposal: Full Nonmonotone Estimator

- When constructing estimates of the response model, we estimate π_{11} using $\pi_{A_2=1|A_1=1}$ and we never include information about $\pi_{A_1=1|A_2=1}$. I would like to create a full estimator where we are able to use this information.
- The current estimator is the following (See Slide 24 of Non-monotone Presentation):

$$\begin{aligned}\hat{\theta}_{eff} &= n^{-1}\sum_{i=1}^{n} E[g_i \mid X_i] \\ &+ n^{-1}\sum_{i=1}^{n} \frac{R_{1i}}{\pi_{1+}(X_i)}(b_2(X_i, Y_{1i}) - E[g_i \mid X_i]) \\ &+ n^{-1}\sum_{i=1}^{n} \frac{R_{2i}}{\pi_{2+}(X_i)}(a_2(X_i, Y_{2i}) - E[g_i \mid X_i]) \\ &+ n^{-1}\sum_{i=1}^{n} \frac{R_{1i}R_{2i}}{\pi_{11}(X_1, Y_{1i})}(g_i - b_2(X_i, Y_{1i}) - a_2(X_i, Y_{2i}) + E[g_i \mid X_i])\end{aligned}$$

- The problem with this is that we assume $\pi_{11}(X, Y_1) = \pi_{2|1}(X, Y_1)\pi_{1+}(X)$ when we could have $\pi_{11}(X, Y_2) = \pi_{1|2}(X, Y_2)\pi_{2+}(X)$. In other words, the previous result implicitly assumes that π_{11} is a function of X and Y_1 when it could be a function of X and Y_2 as well.
- I think that we could use the following (small modification) instead,

$$\pi_{11}(X, Y_1, Y_2) = \alpha\pi_{1|2}\pi_{2+} + (1-\alpha)\pi_{2|1}\pi_{1+}$$

  for some $\alpha \in [0, 1]$.
- The reasoning behind this is two-fold. First, it allows us to think of the nonmonotone missingness case as a linear combination of two monotone cases, which I think is useful. Second, we can make explicit our choice of α, which in the existing estimator is simply $\alpha = 0$.
- The problem with adding another parameter α is that it makes the overall model unidentifiable. We cannot estimate α and the conditional and marginal distributions that we have without additional assumptions. So for now, I think that we should just assume that α is known and then we apply this method to a data integration problem we can review what α makes sense or figure out an additional assumption that can help us estimate α (for example if there is a variable correlated with 1|2 versus 2|1).
- This is not just a pure Bayes' rule. Formally, let $L = (X, Y_1, Y_2)$. By Bayes' rule we have

$$\begin{aligned}
&\Pr(R_1 = 1 \mid R_2 = 1, L) \Pr(R_2 = 1 \mid L) \\
&\quad = \Pr(R_1 = 1, R_2 = 1 \mid L) \\
&\quad = \Pr(R_2 = 1 \mid R_1 = 1, L) \Pr(R_1 = 1 \mid L).
\end{aligned}$$

Yet to estimate this model, we need to assume something like the following:

$$\begin{aligned}
&\Pr(R_1 = 1 \mid R_2 = 1, X, Y_2) \Pr(R_2 = 1 \mid X, Y_2) \\
&\quad = \Pr(R_1 = 1, R_2 = 1 \mid L) \\
&\quad = \Pr(R_2 = 1 \mid R_1 = 1, X, Y_1) \Pr(R_1 = 1 \mid X, Y_1)
\end{aligned}$$

in which case the two sides are clearly different.

- Unfortunately, early simulation results are not promising because we cannot distinguish points that should have π_{11} conditional on Y_1 and points that should be conditional on Y_2.

Table 20: True Value is 0. Cor(Y1, Y2) = 0

algorithm	bias	sd	tstat	pval
oracle	0.001	0.032	1.269	0.102
ipworacle	0.000	0.078	-0.141	0.444
prop.or	0.002	0.038	2.201	0.014
prop.0	0.004	0.037	5.037	0.000
prop.half	0.008	0.037	9.056	0.000
prop.1	0.010	0.038	12.058	0.000

- Dr. Kim, please let me know what you think about this idea and if it is worth pursuing more.

References

[1] James M Robins and Richard D Gill. "Non-response models for the analysis of non-monotone ignorable missing data". In: *Statistics in medicine* 16.1 (1997), pp. 39–56.