2II66 Assignment 2 – Robbert Jongeling

## 1: For which noise type, and with which percentage of noise, does the ‘alpha miner’ provide the ‘worst’ model in terms of fitness and understandability?

Figure 1 shows the distribution of fitness for various noise levels and types. We see that the Remove Head noise type has the lowest performance in terms of fitness. Also, we see that Add Event has fitness 1 or ‘no reliable alignments exist’ for all noise types.

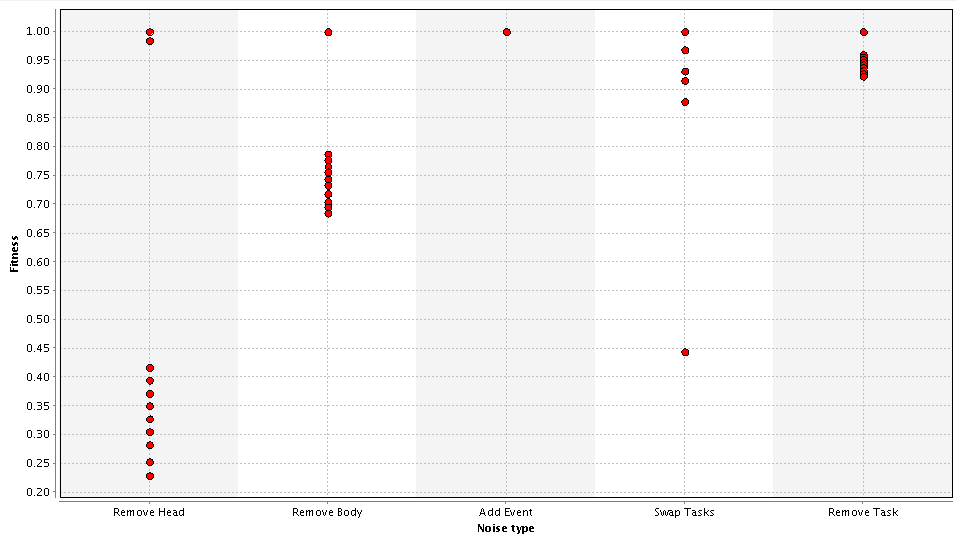


Figure 1: Fitness for various noise types for the alpha miner, 11 dots per noise type represent 0 ,10,..,100% noise

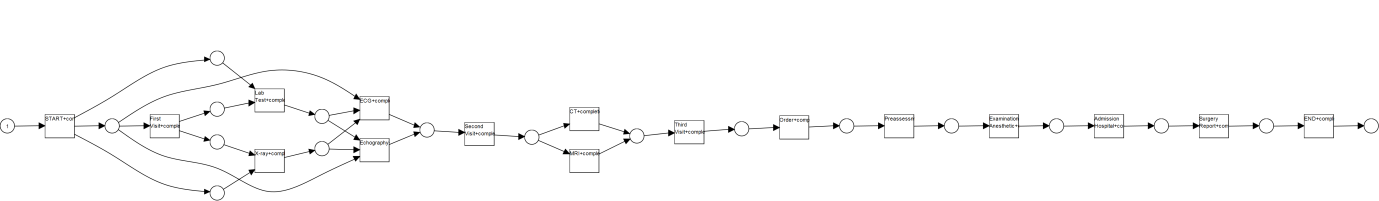
If we look at the generated Petri net for Remove head with 90% noise (the one with the lowest fitness) we see the net in Figure 2. This is a fairly simple to understand net. In comparison, we present the Petri net belonging to swap tasks (with 10% noise) in Figure 3.

Figure 2: Petri net belonging to Remove Head, 90% noise

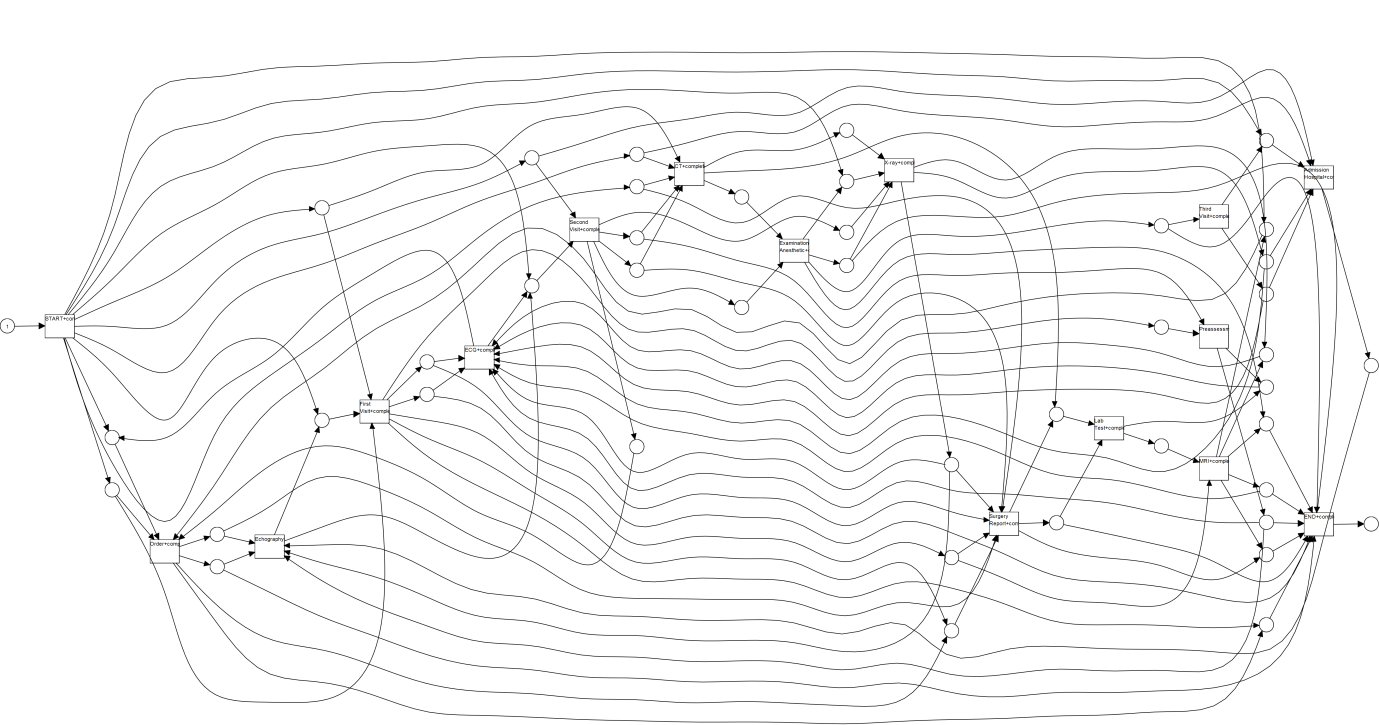


Figure : Petri net obtained by introducing noise type Swap Tasks at 10% using the alpha miner

The shown net is the net with the lowest fitness of noise type Swap Tasks and is a curious outlier considering the other noise percentages. It’s fitness is 0.44 where the others are all in the range 0.87-1. You could consider this to be the worst net in terms of understandability, but as this is an outlier in terms of fitness, we conclude that the Remove Head noise type at 10% noise products the worst model in terms of fitness and understandability.

## 2: For which noise type, and with which percentage of noise, does the ‘ILP miner’ provide the ‘worst’ model in terms of fitness and understandability?

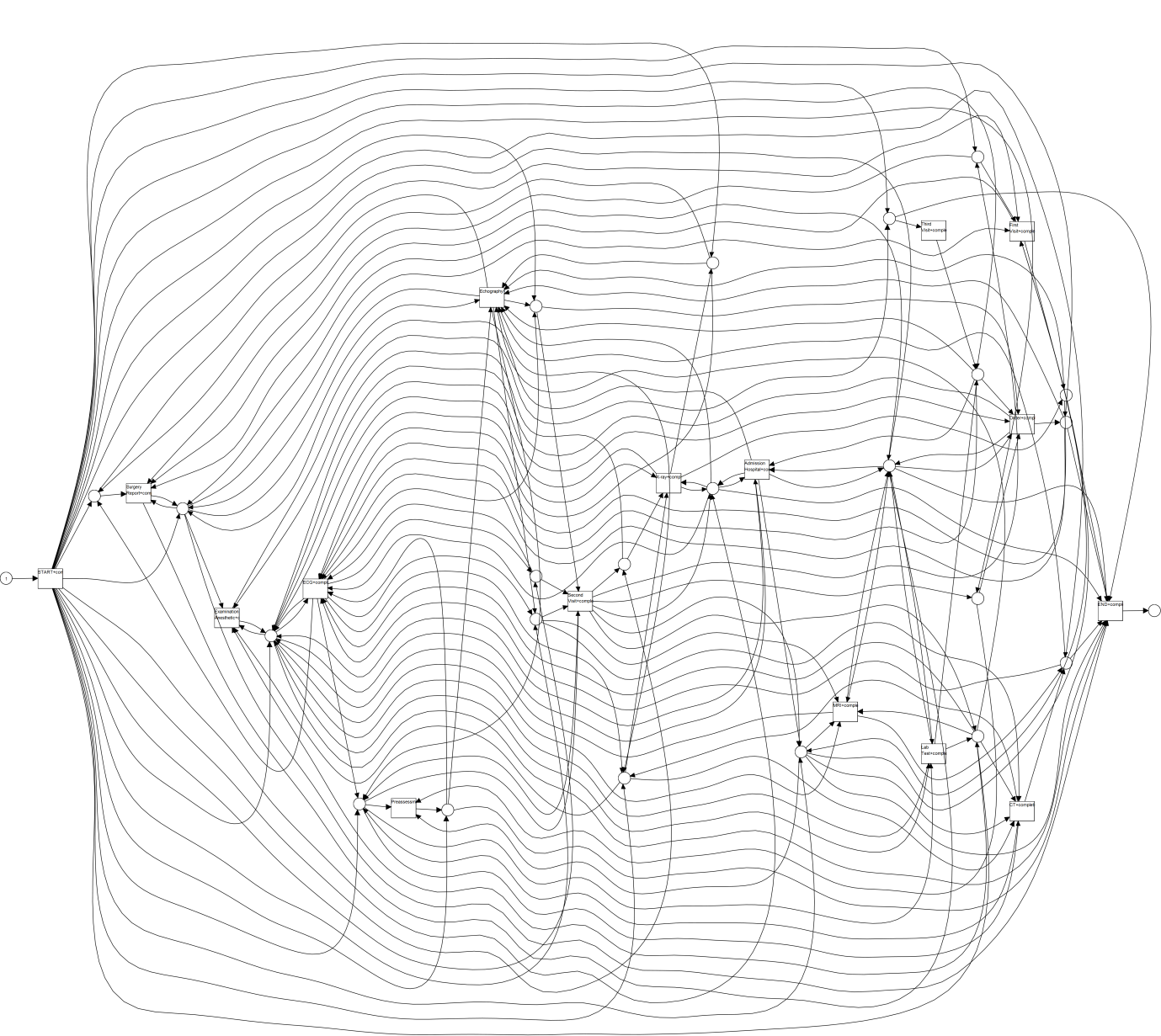
The ILP miner variant *emptyAfterCompletionILPModel* gives a model with fitness 1.0 for each noise type and each percentage tested. In terms of understandability, the Add Event type with 10% noise gives the most chaotic picture, as shown in Figure 4.

Figure : Add Event 10% ILP miner

Using variant *PetrinetILPModel,* we get models with lower fitness than 1.0 for noise types Remove Body, Swap Tasks and Remove Task. We get the distribution as shown in Figure 5. We see that Remove Task noise type yields the worst fitness. At 100% added noise, it also yields a Petri net with low understandability, as shown in Figure 6. We conclude that the ILP miner performs worst on noise type Remove Task and 100% noise percentage.

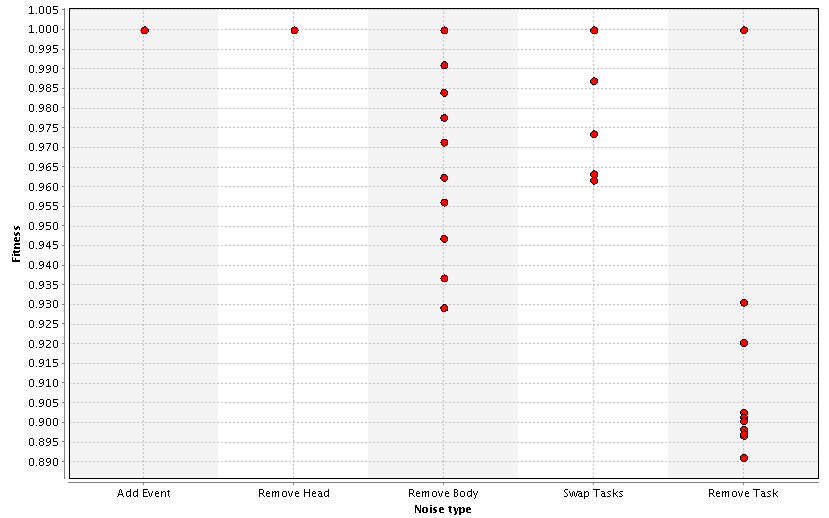


Figure 5: Noise type and fitness obtained by running the ILP miner variant PetrinetILPModel

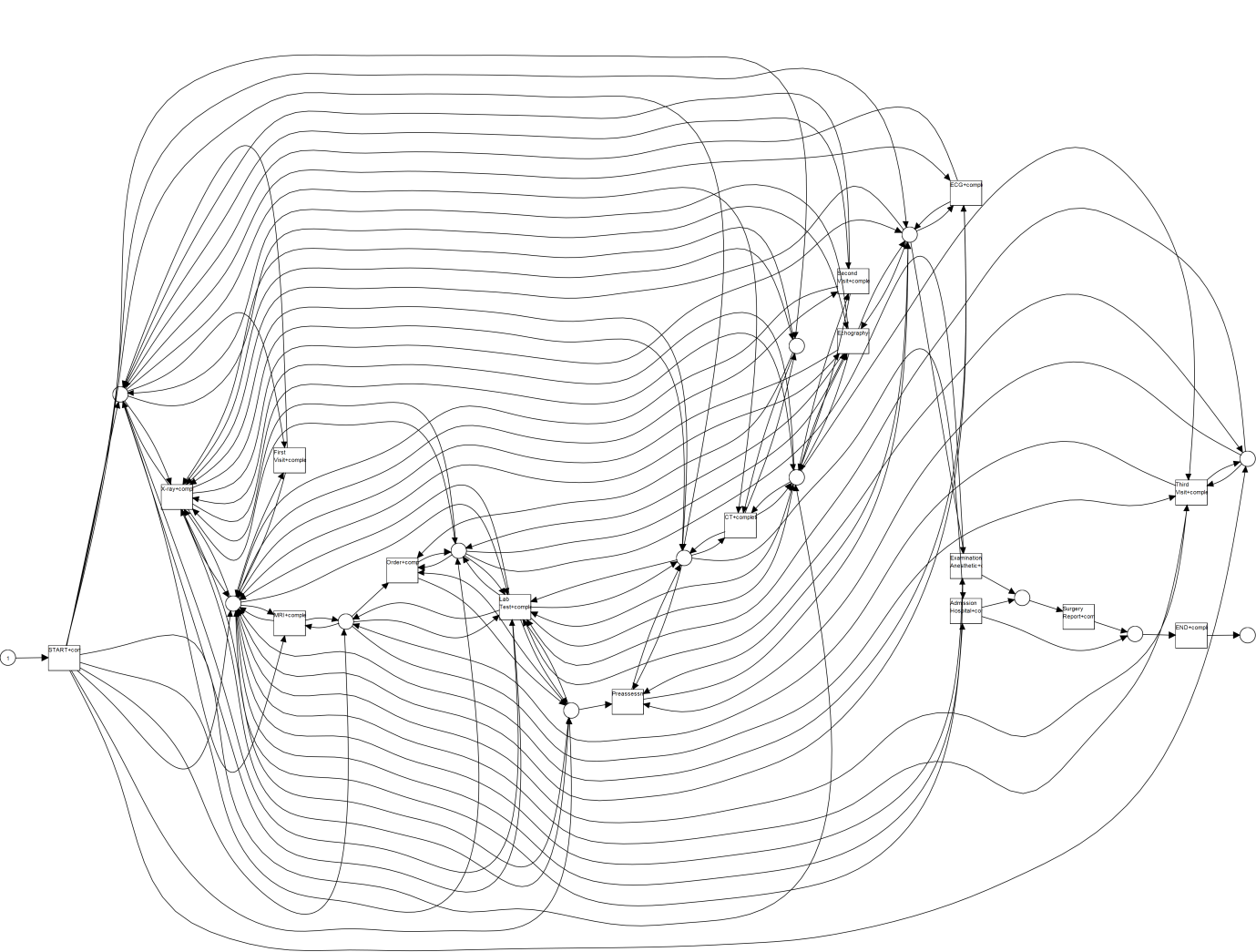


Figure 6: ILP miner with option PetrinetILPModel noise type Remove Task, 100%

## 3: For which noise type, and with which percentage of noise, does the ‘passage miner’ provide the ‘worst’ model in terms of fitness and understandability?

Figure 7shows the distribution of fitness for various noise levels and types. We see that it is very similar to the results of the alpha miner, this can be explained by the fact that we set the property *Mine petri net* of the passage miner to *Alpha Miner.* The other options are *Basic Log relations* and *20.*

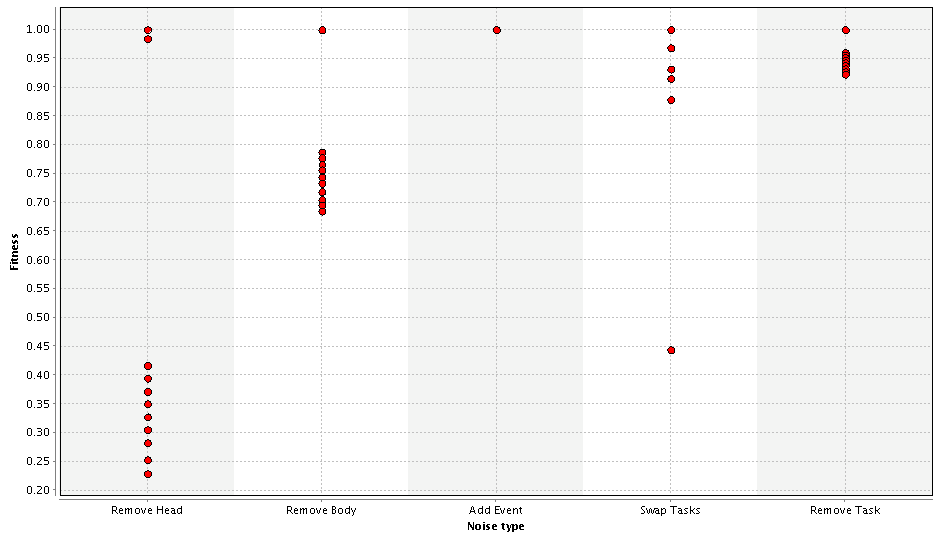


Figure : fitness by noise type passage miner

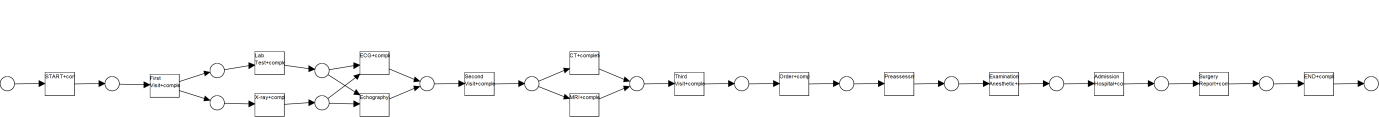
The Petri net for Swap Tasks 10% is not similar to that of the alpha miner though. It is shown in Figure 8. This is a much more understandable image than the same net generated by the alpha miner. 

Figure : Swap Tasks 10% passage miner

Similarly to the alpha miner, the passage miner performs worst on noise type Remove Head. … shows the generated Petri net for that type and 90% noise.

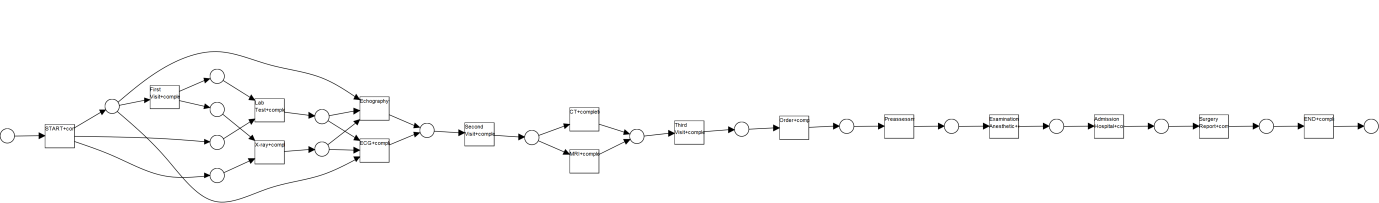


Figure : Remove Head 90% passage miner

We conclude that again, the Remove Head noise type performs worst, this time at 90% noise.

To see what the performance of the passage miner is using a variant we did not try before, we also ran the Passage miner with option *Mine petri net: Exhaustive*. The other options were not altered. Especially for the noise types Remove Head and Remove Body, this option performs bad. Just as seen with the alpha miner option. This option does not change the conclusions drawn after running the alpha miner option.

## 4: For which noise type, and with which percentage of noise, does the ‘inductive miner’ provide the ‘worst’ model in terms of fitness and understandability?

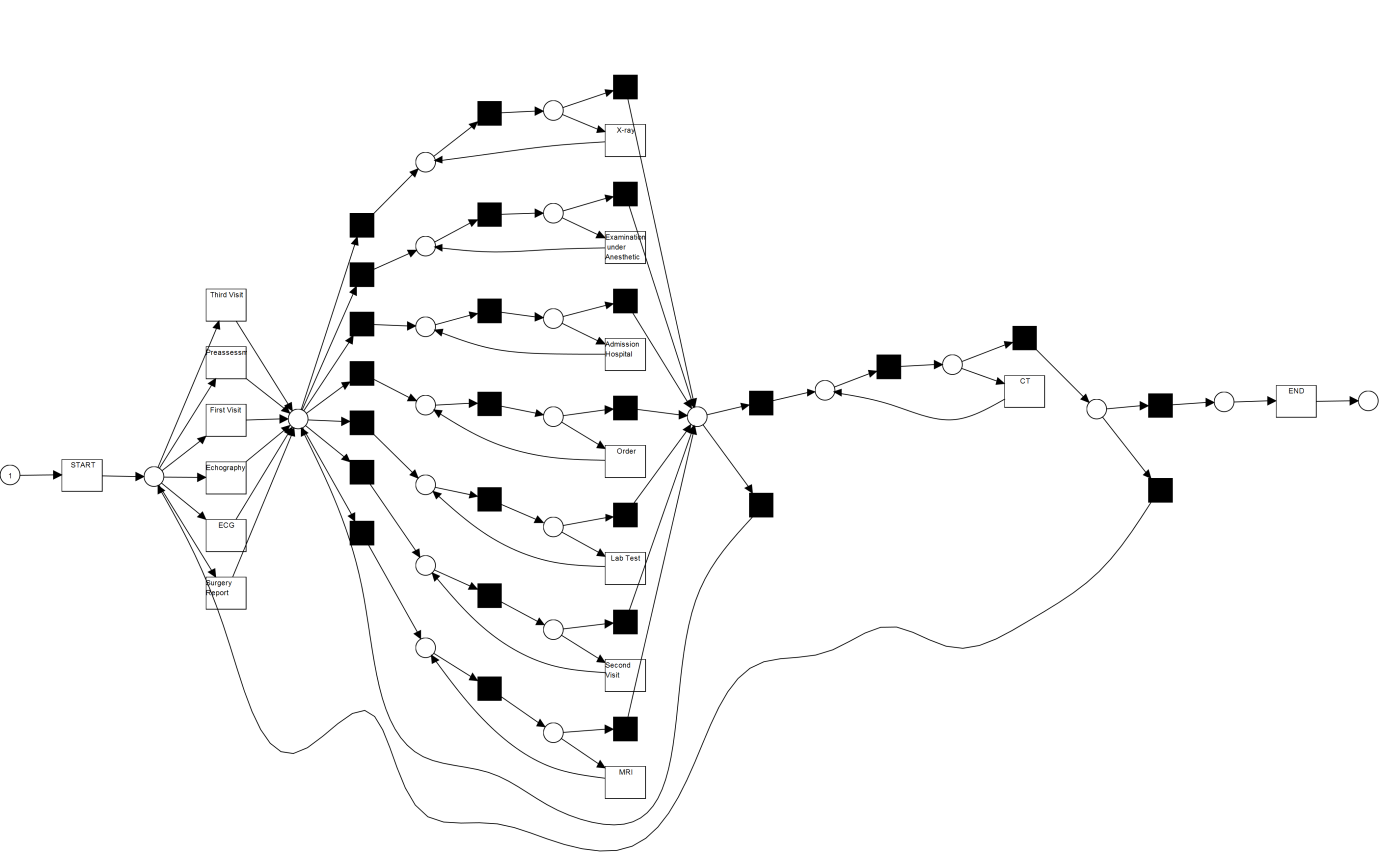
The inductive miner, like the ILP miner generates a net with fitness 1.0 for each noise type and each percentage investigated. We use the setting *Define Inductive Miner: “*Mine Petri Net with Inductive Miner.”We in this experiment did not check the Add Event type for noise percentages 80, 90 and 100. As the nets are all equal in terms of fitness, we have a look at the understandability. As in the test with the ILP miner, the Add Event type gives the most unreadable net, shown in Figure 10.

Figure : Add Event 10% Inductive Miner

The Remove Head task here performs a net with a better understandability. For this miner, we conclude that the Add Event type with 10% noise performs the worst.

We tested this miner also for the option *Define inductive miner: Mine Petri net with inductive miner – frequent.* With noise threshold 1.0. It yields a different distribution of fitness over the noise types compared to the standard option. The yielded distribution is shown in Figure 11. The Remove Body task and Add Event task perform worst of all types. Interestingly, the fitness of these types is worst when the added noise percentage is about 50%. Higher and lower percentages yield models with higher fitness. In terms of understandability, the generated nets are good. To compare with the previous option, we again present a model with noise type Add Event. This time 70%, as this was the noise percentage with the lowest fitness. The resulting net is shown in Figure 12. Add Event at 70% performs the worst in terms of fitness and understandability, but it is still quite good.

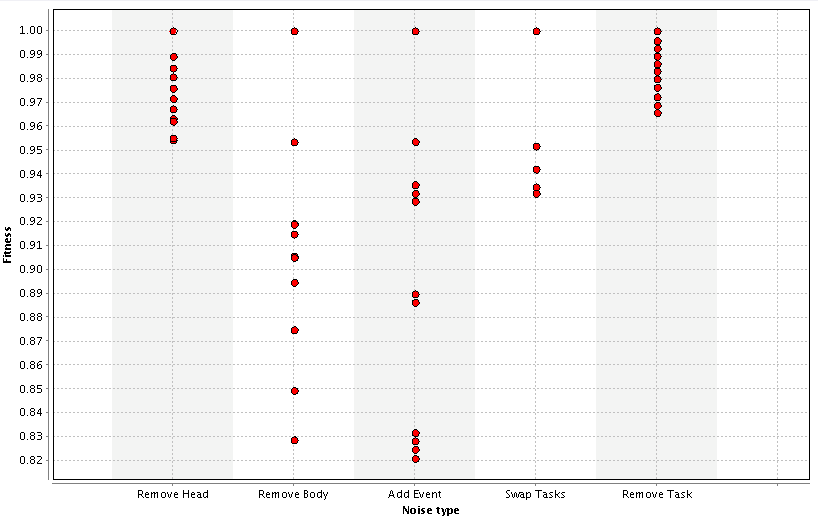


Figure 11: fitness by noise type for inductive miner with option frequent 1.0

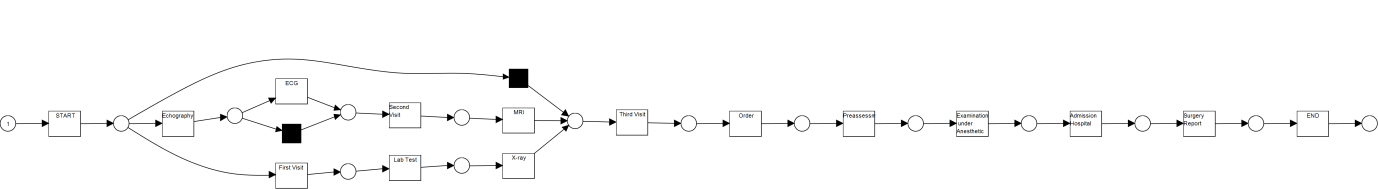


Figure 12: Add Event 70% inductive miner with option frequent

## 5: Based on the results of the previous questions, can you indicate for each type of noise which is the most robust control-flow miner? Additionally, can you indicate which control-flow miner is the most robust against noise and which control-flow miner is the least robust against noise?

In Table 1, we present the most robust control-flow miner per noise type.

|  |  |
| --- | --- |
| Noise Type | Most robust control-flow miner |
| Remove Head | ILP miner |
| Remove Body | ILP miner |
| Add Event | Alpha/ILP/Passage miner |
| Swap Tasks | ILP miner |
| Remove Task | Inductive miner |

Table 1: Most robust control-flow miners by noise type

The ‘best’ control-flow miner, i.e. most robust against noise in this test is the ILP miner, of course its robustness will depend on the variant chosen, but we tested two and both proved very robust.

The Alpha miner seems to be the least robust against noise, perhaps together with the Passage miner, as we noted their similarity earlier.