means that workers who answer incorrectly will tend to produce previously unseen answers. We consider the following variations of tasks: 1) Low difficulty, 2) High difficulty, high \( \theta \), and 3) High difficulty, low \( \theta \). In the first two cases, we see expected behavior. LazySusan is able to use its model to infer correct answers.

However, in the third case, we see some very interesting behavior. Since the difficulty is high, workers more often than not produce the wrong answer. Additionally, since \( \theta \) is low, they also tend to produce the same wrong answer, making it look like the correct answer. If the ratio is large enough, we find that LazySusan is unable to infer the correct answer, because of the unfortunate ambiguity between high difficulty, low \( \theta \) problems and low difficulty problems. In fact, as LazySusan observes more ballots, it becomes more convinced that the common wrong answer is the right answer, because of the model dynamics we mention earlier (Section 3.1). This problem only arises, however, if the model produces adversarial 0, and we see in practice that workers on Mechanical Turk generally do not exhibit such behavior.

### 5.5 Experiments on Mechanical Turk

Next, we compare LazySusan to an agent using majority-vote (MV) using real responses generated by Mechanical Turk workers. We test these agents with 134 math questions with levels of difficulty comparable to those found on the SAT Math section. Figure 4 is an example of one such task and the user interface we provided to workers. We set the utility for an incorrect answer, \( C_W \), to be \(-100\), because with this utility setting, LazySusan requests about 7 jobs on average for each task, and a simple binary search showed this number to be satisfactorily optimal for MV. We find that the workers on Mechanical Turk are surprisingly capable at solving math problems. As Table 3 shows, LazySusan almost completely eliminates the error made by MV. Since the two agents cost about the same, LazySusan achieves a higher net utility, which we find to be statistically significant using a Student’s t-test (\( p < 0.0002 \)).

We examine the sequence of actions LazySusan made to infer the correct answer to the task in Figure 4. In total, it requested 14 ballots, and received the following responses: 215, 43, 43, 43, 5, 215, 43, 3, 55, 43, 215, 215, 215, 215. Since MV takes the majority of 7 votes, it infers the answer incorrectly to be 43. LazySusan on the other hand, uses its knowledge of correlated answers as well as its knowledge from previous tasks that the first three workers who responded with 43 were all relatively poor workers compared to the first two workers who claimed the answer is 215. So even though a clear majority of workers preferred 43, LazySusan was not confident about the answer. While it cost twice as much as MV, the cost was a worthy sacrifice with respect to the utility setting.

Finally, we compare our EM algorithm to MV, using