Computational Problem complexity and Solvability Analysis

We perform a systematic error analysis of existing methods on (optimal)-Blocksworld and Logistics, hypothesizing that the difficulty of problems within each class is strongly correlated with the characteristics of the solution distribution. Specifically, we examine the relationship between the success rate of each method and the solution step complexity, defined as the minimum number of execution steps required to complete each task. The resulting plots, which depict success rate as a function of the minimum required steps to complete a task, are presented in Figures 1 to 5. It is important to note that the PEA method is excluded from the Blocksworld task analysis, as it achieves a perfect success rate across all models and tasks in this domain.

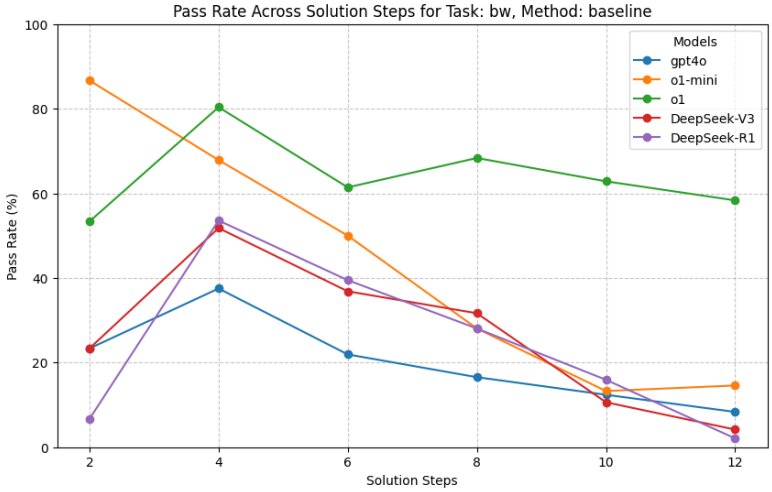

Figure 1. Success rate of direct IO prompting on optimal-Blocksworld tasks, categorized by the minimum steps required to solve each task. The success rate generally decreases as the step complexity increases, except for certain two-step problems where models struggle. Manual examination reveals that these models often produce correct but non-optimal solutions, despite the prompt explicitly requiring optimal solutions.

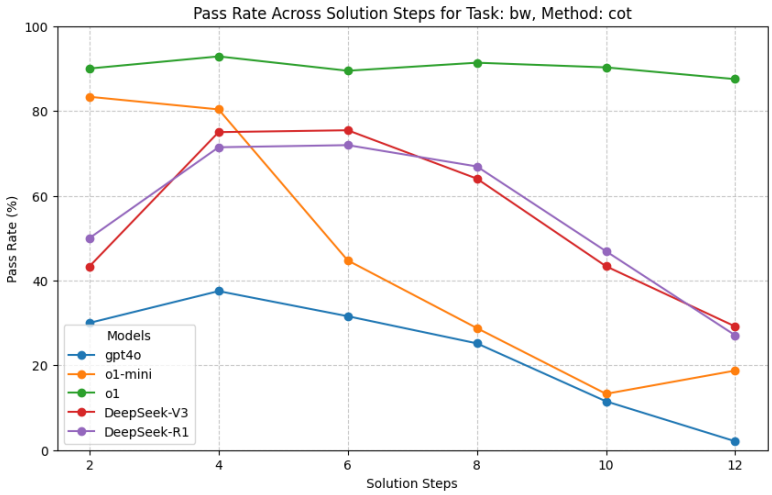

Figure 2. Success rate of CoT prompting on optimal-Blocksworld tasks, categorized by the minimum steps required to solve each task. The success rate decreases as the step complexity increases.

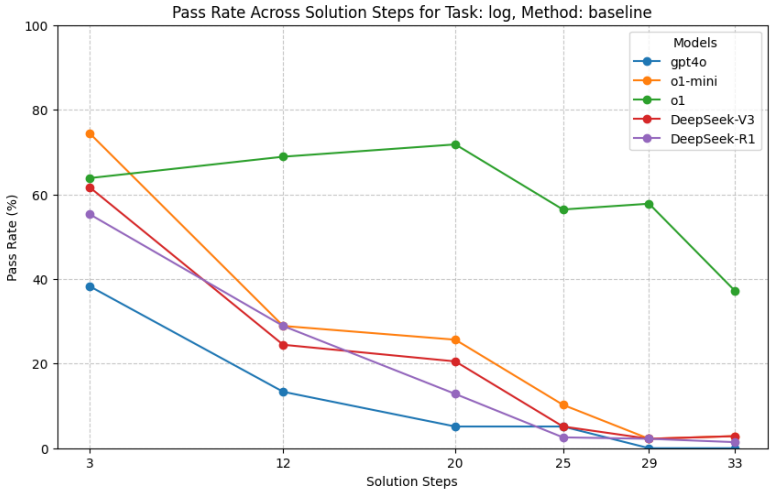

Figure 3. Success rate of direct IO prompting on Logistics tasks, categorized by the minimum steps required to solve each task. The success rate decreases as the step complexity increases.

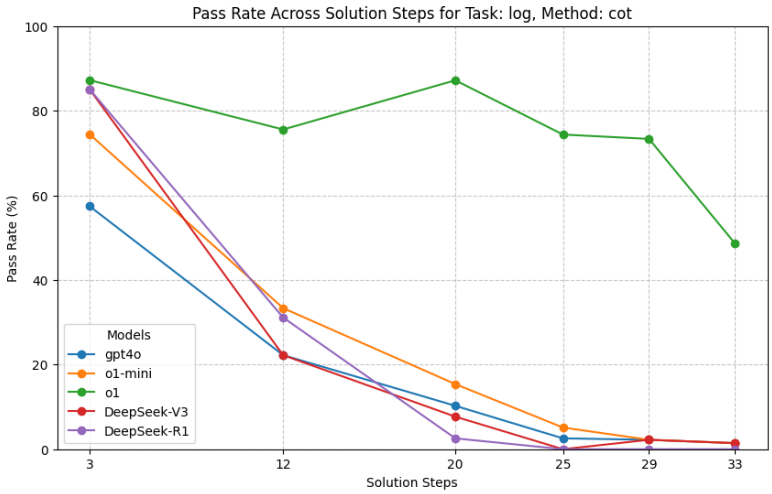

Figure 4. Success rate of CoT prompting on Logistics tasks, categorized by the minimum steps required to solve each task. The success rate decreases as the step complexity increases.

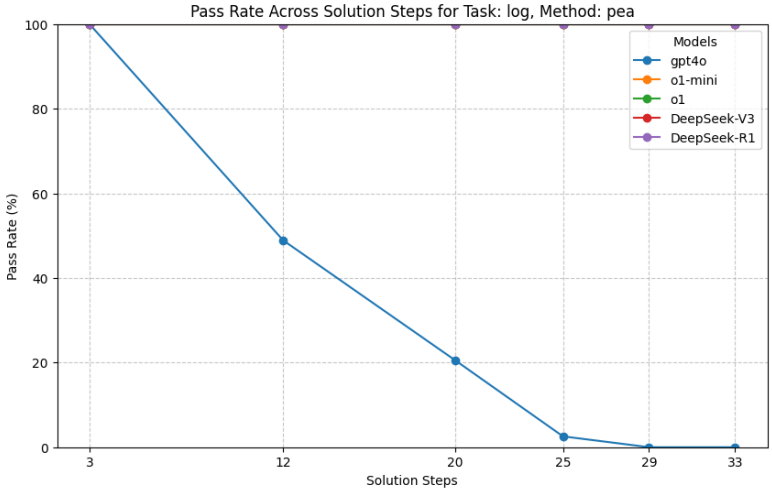

Figure 5. Success rate of PEA prompting on Logistics tasks, categorized by the minimum steps required to solve each task. All models, except GPT-4o, achieved a 100% success rate with optimized code across all tasks. For GPT-4o's unoptimized code, the success rate declines as step complexity increases.