# 3620 Assignment 3 Report

Question 10

This part discusses the advancements and improvements made to our initial planning system, a basic encoding model, based on the logistics domain. The objective is to enhance the system's planning efficiency through the inclusion of domain-specific control knowledge.

The assignment was accomplished through a series of steps from Question 1 to Question 8, starting with developing a basic encoding model by implementing action and fluent variables, initial state and goal axioms, preconditions and effect axioms, explanatory frame axioms, serial mutex axioms, interference mutex axioms, reachable action axioms, and fluent mutex axioms. This basic encoding model was further augmented in the final step by applying domain-specific control knowledge. These control rules imposed limitations on the model's variables, constraining the planning problem in ways specific to the logistics domain, such as the movement of trucks and planes.

The code pertains to the implementation of control rules in the Logistics domain. Three constraints were formulated and added to the basic encoding model.

- Package Stays Once at Destination: This rule asserts that once a package arrives at its destination, it cannot leave. If a package is at its destination at time t, it cannot leave at time t+1.
- Package Does Not Return to Origin: This rule prevents a package from returning to its original location after it's been moved. This reduces unnecessary movements, thus increasing efficiency.
- Vehicle Capacity Limit: This rule restricts a vehicle from transporting more packages than its capacity allows.

The clauses corresponding to these constraints were then added to the basic encoding model using the add_clause method.

The application of these domain-specific control rules to the initial encoding model significantly improved the system's performance. Tests indicated that the modified logistic model was consistently 0.05 to 0.2 seconds faster than the model without logistic control. The effectiveness of the rules is evident in the performance boost, with the cost of this improvement being a potential decrease in the model's generality. The rules made the problem easier to solve by limiting the model's flexibility, but they preserved the capability to find some solution. The strategy of introducing control knowledge increased the planner's efficiency and, at the same time, constrained the search space, which ultimately improved the solution's quality.

Question 11

The testing result is here: 3620 Assignment 3 result.xlsx (let me know if you can't access it)

The purpose of this study was to compare different configurations of automated planning

systems across three benchmark domains: Miconic, Depot, and Blocks. The configurations tested included parallel and serial planning, as well as different graph-planning techniques (no graph-plan vs graph-plan, Fluent mutex vs Reachable actions). The objective was to understand which configurations performed best in each domain in terms of time and quality of the produced plans.

Three benchmark domains were chosen for the experiment: Miconic, Depot, and Blocks. For each domain, eight configurations were tested involving variations in planning techniques (serial or parallel), and graph-plan techniques. Each configuration was run on the instances that were chosen in each domain with a time limit of 100 seconds per test. The Miconic domain showed comparable performance for both parallel and serial configurations. However, the parallel configuration used fewer actions to reach the goal within a 30-step limit, indicating a higher plan quality. Therefore, parallel planning is recommended for the Miconic domain. In the Depot domain, the serial configuration encountered time run out issues in some instances (e.g., depot03), indicating a longer computation time for this configuration. In contrast, the parallel configuration did not exhibit such issues. Therefore, the Depot domain is better suited to parallel planning for efficient plan generation. The Blocks domain showed minimal differences in computation time for both parallel and serial configurations. However, the serial configuration used fewer actions within the 30-step limit to reach the goal, indicating a higher quality plan. Therefore, serial planning is recommended for the Blocks domain.

In terms of graph-planning techniques, Fluent mutex consistently outperformed Reachable actions in producing higher-quality plans. This suggests Fluent mutex should be the preferred choice.