# Project Enhancement: Combining Robot Types and Working Shifts in CSP

1. Overview

In this enhanced design, we introduce a more realistic and intelligent warehouse system simulation. Robots are not only differentiated by their capabilities (handling fragile, heavy, or general goods) but also by their working shifts (day, night, or 24/7 availability).

This improvement integrates multiple Constraint Satisfaction Problems (CSPs) into one unified framework for intelligent robot-task assignments.

2. Problem Formulation

2.1 Variables

* Each **Task** (defined as the movement of a good from one location to another) is a **variable** in the CSP.

2.2 Domains

* For each task, the **Domain** is the set of **robots** capable and available to perform that task, filtered based on:
  + Robot type compatibility.
  + Robot working shift compatibility.
  + Robot current availability (no task conflicts).

2.3 Constraints

|  |  |
| --- | --- |
| **Constraint Name** | **Description** |
| Type Compatibility | Robot must be capable of handling the assigned good (e.g., fragile items handled only by fragile robots). |
| Shift Timing Compatibility | Robot must handle available during the task’s scheduled time. |
| Single Task Constraint | A robot can only handle one task at a time unless tasks are sequential. |
| Collision Avoidance | Prevent two robots from occupying the same space at the same time. |
| Distance Optimization | Prefer robots closer to the pickup location to reduce travel time. |

3. System Design

3.1 Robot Attributes

* ID
* Robot Type ('fragile', 'heavy', 'general')
* Working Shift ('24/7', 'day', 'night')

3.2 Task Attributes

* ID
* Good Type ('fragile', 'heavy', 'general')

4. Example Table

4.1 Robots

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Robot ID** | **Type** | **Shift** | | R1 | fragile | 24/7 | | R2 | heavy | day | | R3 | General | night | | R4 | fragile | day | |  |  |

4.2 Tasks

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Task ID** | **Good Type** | | T1 | fragile | | T2 | heavy | | T3 | general | |
|  |

4.3 Sample Matching

* T1: R1, R4 (fragile + day time)
* T2: R2 (heavy + day time)
* T3: R1, R3 (general + night time)

5. Use of Arc Consistency (AC-3)

To enhance the efficiency of the CSP solving process, Arc Consistency 3 (AC-3) algorithm will be applied:

* AC-3 will be used to prune the domains of the tasks by removing robots that cannot satisfy the task constraints based on type and shift availability.
* Arcs are created between each Task and its possible Robot assignments.
* For each arc, consistency checks are performed. If a robot is found incompatible for a task, it is removed from the task's domain.
* If the domain of a task becomes empty during this process, the CSP is deemed unsolvable under the current configuration.
* AC-3 ensures early detection of impossible assignments and reduces the overall search space, leading to a more efficient and faster CSP solution.

6. Expected Outcomes

* Intelligent task assignment considering robot specialization and working hours.
* Enhanced simulation realism by modeling different robot behaviors.
* Efficient warehouse task management using CSPs.
* Improved performance through domain pruning with AC-3.

7. Future Expansion Possibilities

* Integrate robot battery management and charging schedules.
* Add overtime penalties for robots operating outside regular shifts.
* Use machine learning to predict high-demand times and pre-assign robots.

8. Conclusion

This enhanced CSP model provides a much richer, more realistic, and flexible warehouse management simulation. It intelligently combines robot capabilities and time availability, making the system suitable for further expansion into real-world complexity in future work.