Multi-Agent House-Cleaning Environment: Vacuum-Cleaner and Dustbin

Alexey Markin
Anugrah Saxena
XinXin Yang

Agenda

- Introduction
- Environment Model
- Vacuum Cleaner
- Dustbin
- Experimental Evaluation
- Summary
- Future work

Introduction

Major Vacuum features:

- Remote control
- Self-charging
- Sensor

What we want:

- Automatic disposal of trash
- Integration with Smart Home

Smart Dustbin

- Ability to move!
- Helper agent for Vacuum Cleaner
- Larger storage capacity
- Integration with Smart Home

Assumptions

- Smart Home Environment

    Undirected Graph of rooms

    Sensors (Dirt sensor, Occupancy sensor)

    Constant cleaning time for all rooms

    Designated “Dump” room

    2 Autonomous agents

    Each agent action takes one minute

- Restrictions

    Vacuum Cleaner could not clean occupied rooms

- Trash Transfer

    Vacuum cleaner can transfer Trash to Dustbin

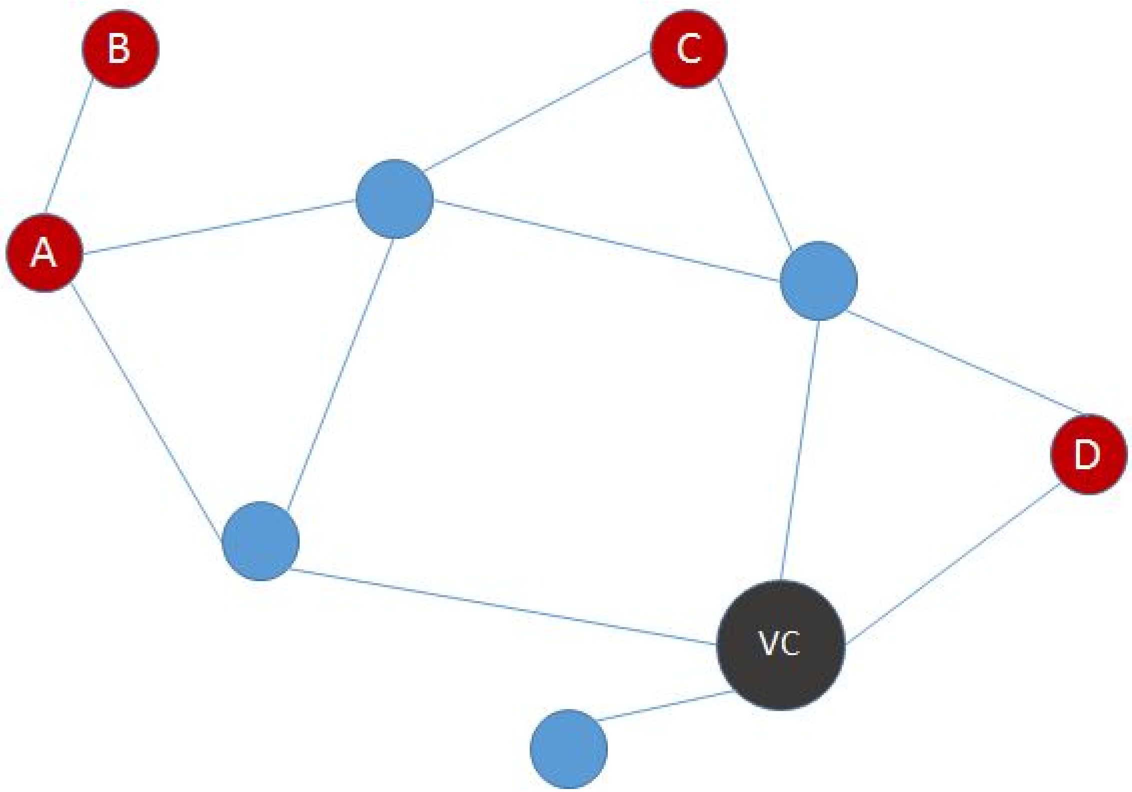
Environment Model

Environment Type

- Fully Observable: Smart Home sensors provide relevant information (occupancy/dirt)
- Stochastic: Occupancy is controlled by human factor
- Dynamic: Room states change over time
- Multi-agent: Vacuum Cleaner and Dustbin agents
- Discrete: Percepts and actions are discrete

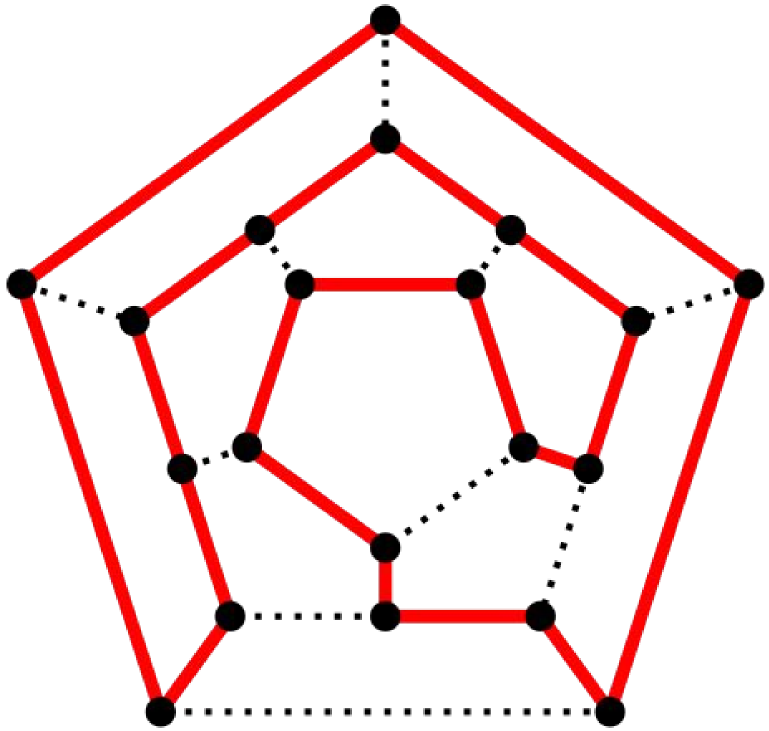
Vacuum Cleaner

- Planning: Find a shortest path connecting "dirty" nodes



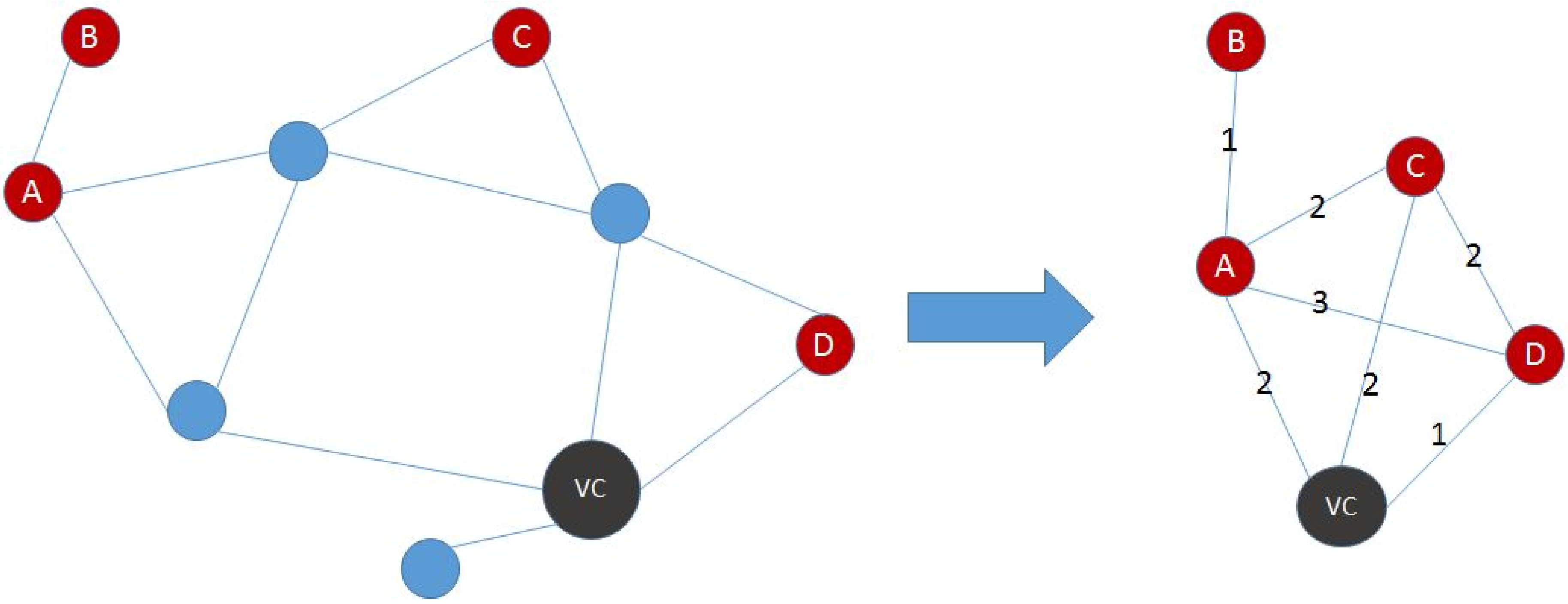
Vacuum Cleaner: hardness

- Planning: Find a shortest path connecting “dirty” nodes
- Problem is NP-hard! Reduction from Hamilton Path:



Hamilton path is a shortest path connecting all the nodes

Vacuum Cleaner: TSP



Find optimal path using Travelling Salesman Problem

Can use polynomial approximation algorithms

AI approach: A*

State:

- current load
- current room
- occupancy states for each room
- remaining # of clean-actions for each room

Goal state: all rooms are clean

State Space: $O(Cn(2A)^n)$

Heuristic function:

*(rooms_to_clean - 1) * move_cost +*
*+ remaining_clean_steps * clean_cost*

Issues with A*

- Size of state space

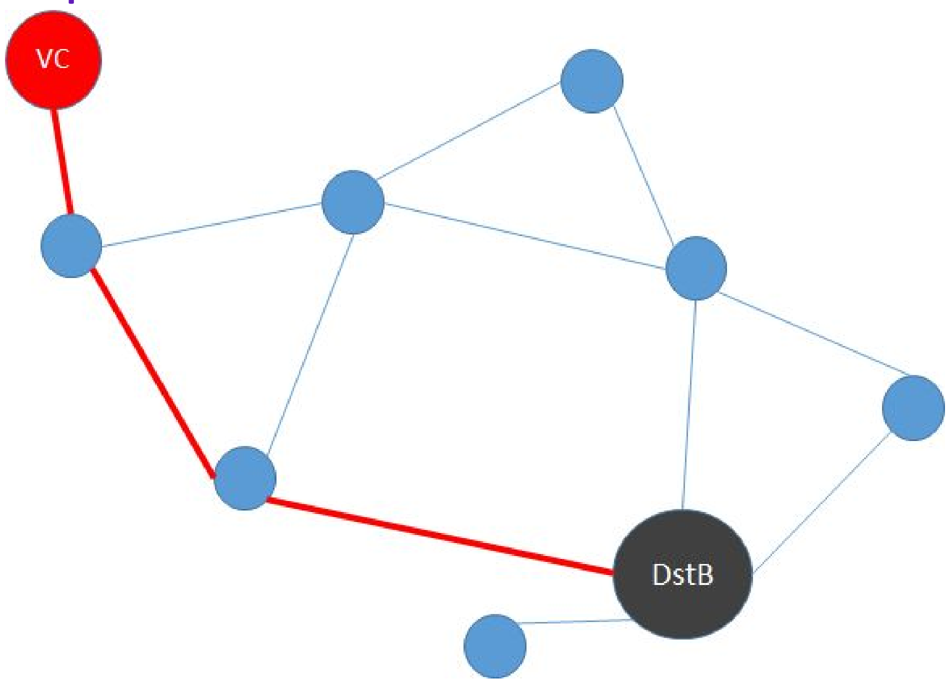
  solution: do not run A to the "end"*

- Dynamic environment: some actions become available/unavailable over time

  solution: Lifelong planning. Incremental Heuristic Search

Dustbin

- Follows BFS search to reach VC or Dump Room.
- DstB reaches for VC on receipt of Dump collection percept.
- Goes to Dump Room when it reaches close to its capacity.



Both agents are defined formally: percepts, actions

Evaluation: House Graph

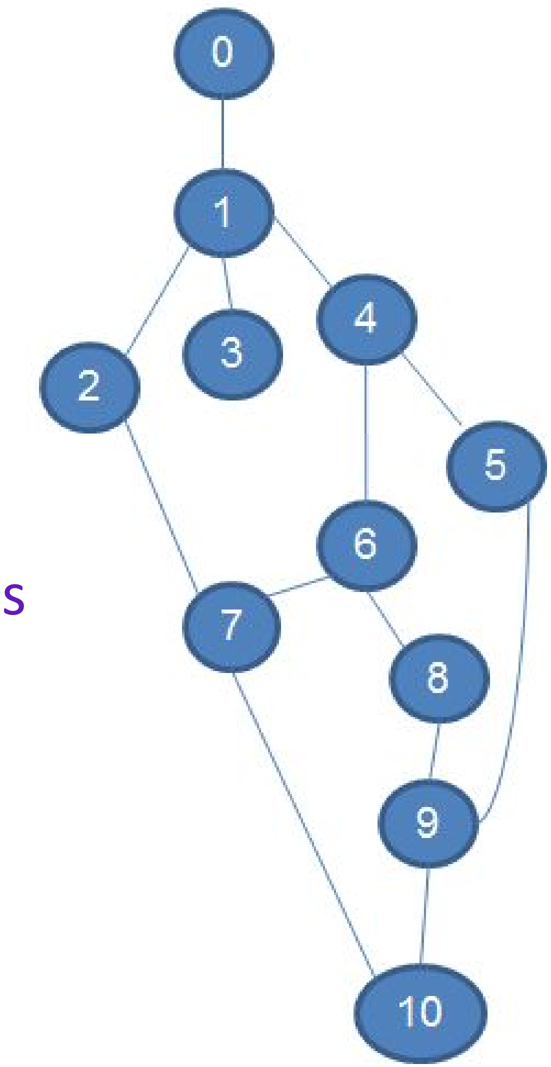
System with Restrictions

- Vacuum cleaner cannot go into an occupied room.

System without Restrictions

- Vacuum cleaner can visit occupied rooms

Room Adjacency Graph



Evaluation: Schedules

- Occupancy schedule for each room: non-overlapping time intervals (time-step is one minute)
- Occupancy might result in room becoming dirty

Schedule Generation:

- $P(O(t+1) \mid O(t)) \approx \mathbf{0.9}$
  Probability of **continuing occupancy**

- $P(O(t+1) \mid \neg O(t)) \approx \mathbf{0.05}$
  Probability of room **becoming occupant**

- $P(D(t) \mid O(t)) \approx \mathbf{0.05}$
  Probability of room **becoming dirty** at a certain occupation step

Schedule example

1 30-31 41-79D 97-204D 212-220 257-269D 275-277 288-289 311-346D 355-362 366-380D

2 11-20 52-107D 112-124D 178-261D **315-315** 321-340D 365-382D 431-532D 541-564D 566-584D

3 2-7 54-71D 106-161D 197-228D 236-238 247-274D 312-318 340-350 354-359D 367-383D

4 2-4 10-11 54-85D 91-96D 98-103D 113-129D 168-198D 221-223 282-298D 313-315 363-374D

5 6-10 31-61D 67-78D 83-93D 149-160D 232-246D 251-289 300-300 313-362D 369-392 **416-478D**

Evaluation setup

Capacity Parameters

- VC: 100 (10 cleans per room, each clean increases load by 2)
- Dustbin: 500

Action costs:

- VC: Stay(0), Clean(10), Move(10), Dump(1)
- Dustbin: Stay(0), Move(8), Load(1), Dump(1)

Schedules

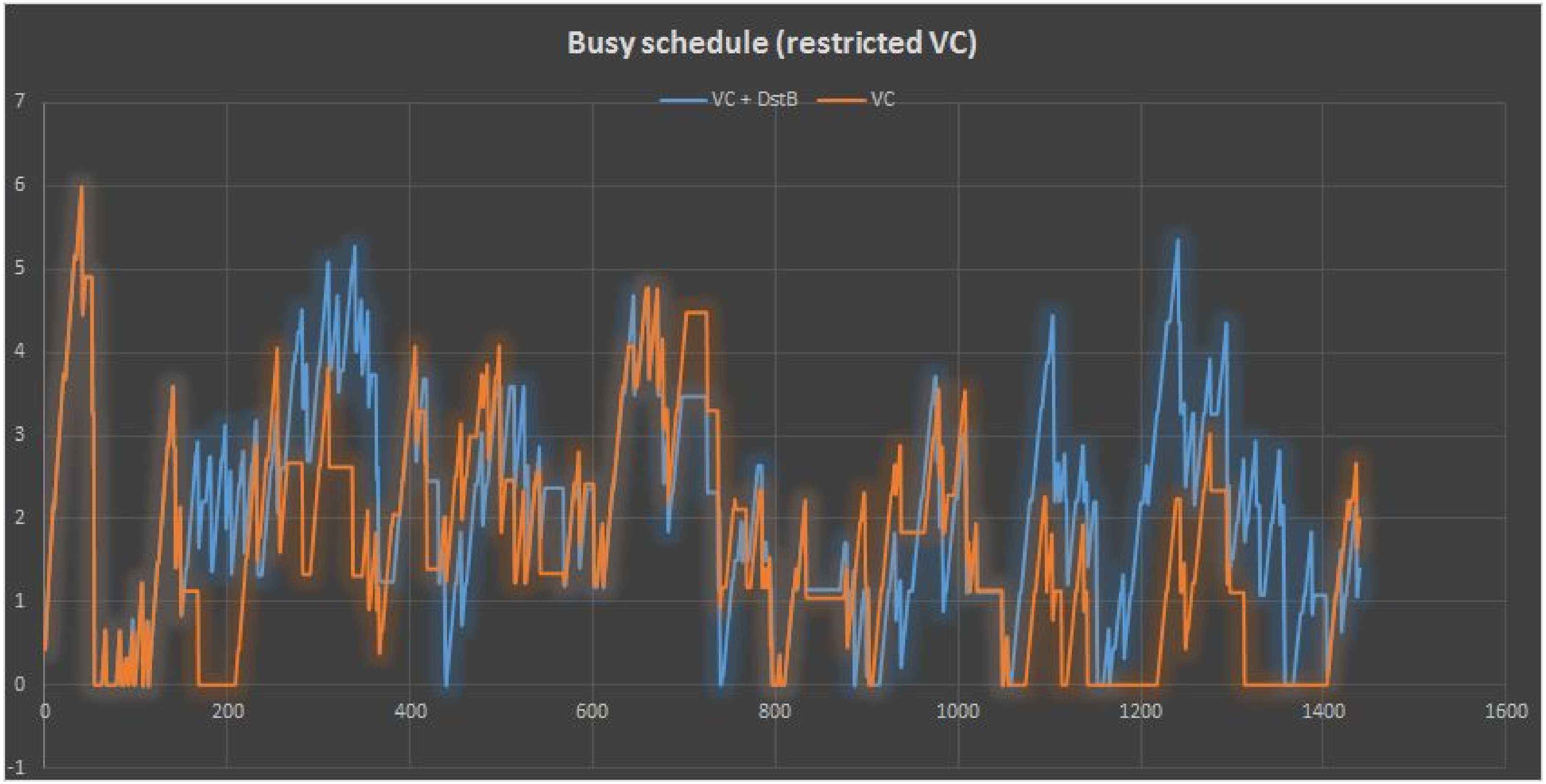
- "Relaxed": $P(O(t+1) \mid O(t)) = \mathbf{0.9}$; $P(O(t+1) \mid \neg O(t)) = \mathbf{0.03}$; $P(D(t) \mid O(t)) = \mathbf{0.05}$
- "Busy": $P(O(t+1) \mid O(t)) = \mathbf{0.95}$; $P(O(t+1) \mid \neg O(t)) = \mathbf{0.05}$; $P(D(t) \mid O(t)) = \mathbf{0.1}$;

Performance Measure

$$\frac{n * cleanStepsPerRoom - remainingCleanSteps}{\log(totalActionCost)}$$

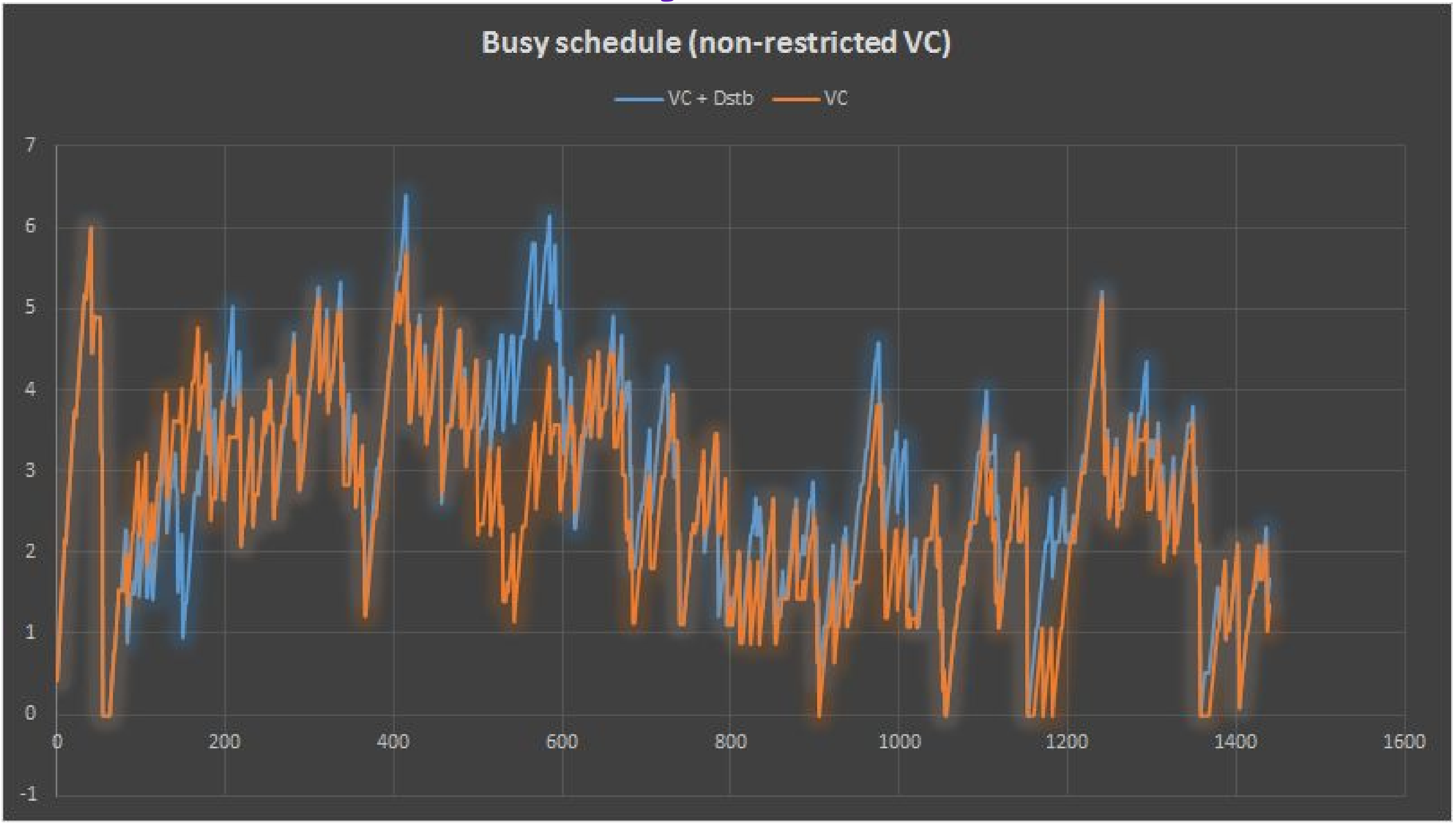
n = number of room in the house (excluding dump room);
cleanStepsPerRoom = total number of actions needed to clean a room;
remainingCleanSteps = remaining cleaning actions required to clean the whole house;
totalActionCost = cost of all actions performed till now.

Evaluation: Busy, Restricted



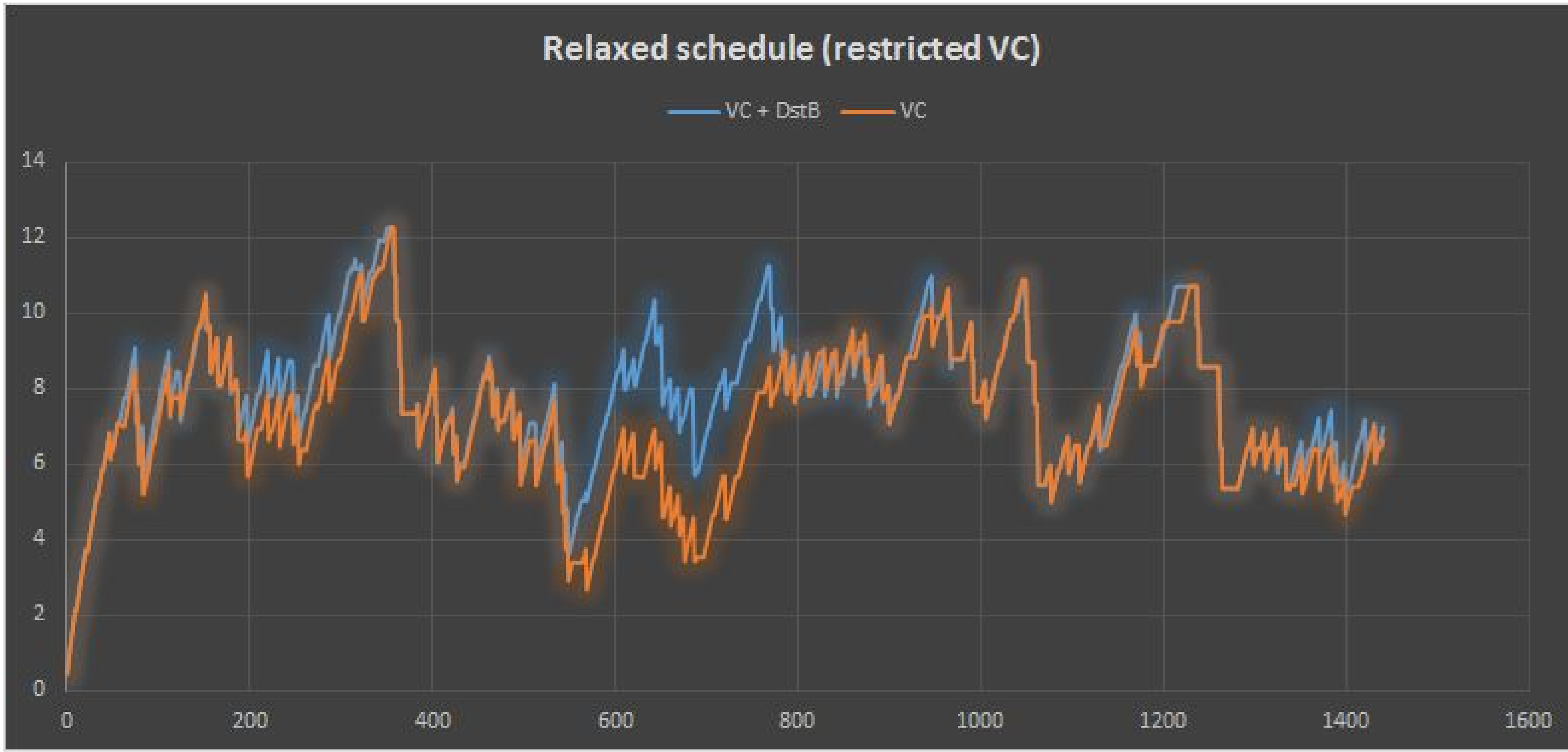
On average for each time step the system gives us ~29% better performance in a Busy Schedule when VC+DstB work together.

Evaluation: Busy, non-restricted



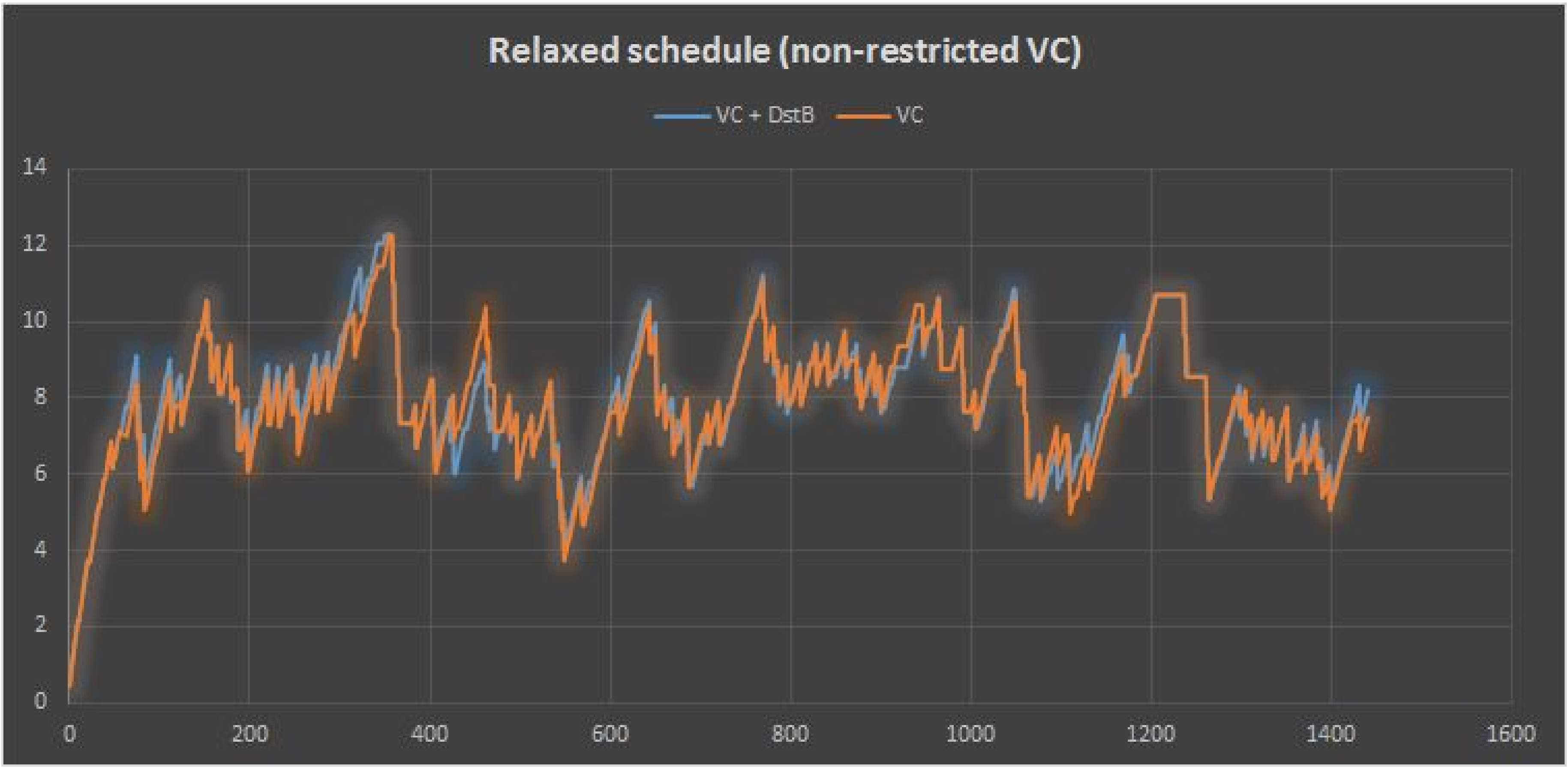
On average for each time step the system gives us ~22.6% better performance in a Busy Schedule when VC+DstB work together.

Evaluation: Relaxed, restricted



On average for each time step the system gives us ~10% better performance in a Relaxed Schedule when VC+DstB work together.

Evaluation: Relaxed, non-restricted



On average for each time step the system gives us ~1% better performance in a Relaxed Schedule when VC+DstB work together.

Summary

1. Developed a general Vacuum Cleaner strategy

2. Introduced and evaluated a novel idea of moving dustbin

3. Obtained promising experimental results

- Busy schedule:
  performance shoots up to ~29% (~23% with non-restricted VC) in a busy schedule.
- Relaxed schedule:
  performance of VC+DstB is ~10% better (~1% with non-restricted VC) in a relaxed schedule.

Future Work

- Enhance the performance of Dustbin using different strategies e.g., DstB follows VC in a pattern to maintain a specific radial distance from VC.
- Work on increasing the performance of VC by considering other heuristics and algorithms.
- Go to higher scale e.g., a big organization as an environment for agents.

Thank you!