Self-Service Laundry Scheduling

This project aims to analyze factors on performance in a self-service laundry. There are three main components to this project: the effect of rules on performance for FCFS-like scheduling, using a stopping algorithm to find when a user should choose to do laundry, and the effect of user strategies on performance. They are described in more detail below as Scheduling, Stopping, and Game Theory respectively. Note that this project is not complete and cannot be run.

Definitions and Vocabulary

The following definitions are used throughout this document and the code:

- **Process**: A set of clothes to be washed together in one Segment (e.g. whites, darks, towels).
- **Segment**: An element of a laundry task (e.g. washer, dryer). Segments may contain multiple Processes (e.g. combine whites and darks in dryer).
- **Cycle**: An element from the set of possible Segment lengths (e.g. 30, 45, 60 minutes).
- **Task**: A list of Segments that must be completed in order to finish a Process (e.g. washer and dryer). Segments are ordered and may be repeated (e.g. washer, washer, dryer).
- **User**: A person who has a set of Processes.
- **Wait Time**: The time between when a User submits a Process and when the Process is started.
- **Occupied Idle Time**: The time when a Segment is not running but is not available for a new Process due to a Process that has not been removed.
- **Unoccupied Idle Time**: The time when a Segment is not running and is available for a new Process.
- **Performance**: Execution Time^{-1}

Algorithms

Scheduling

This is partially implemented.

The following are possible rules that may be implemented:

1. Number of concurrent segments of one type per user
2. Time before forced removal of a process
3. Availability window

Stopping

This is not implemented.

This is meant to be a stopping algorithm that a user would use to decide when to do laundry (processes). A user would presumably want to do laundry when the expected wait time is low. However, estimating the wait time has a cost and waiting too long may result in incomplete processes. This is similar to the problem of deciding when to stop looking for a parking spot as the destination approaches.

Game Theory

This is not implemented.

This is meant to be a game theory analysis of the effect of user strategies on performance. The following are possible user strategies:

1. Maximize concurrent processes
2. Wait for a segment to finish before starting another (minimize offsets)
3. Greedily take the first available segment for each process