8-2015

Integration Of Maintenance Into Design And Sustainability Of Buildings

Dinesh Reddy Patlolla
University of Tennessee - Knoxville, dpatlol2@vols.utk.edu

Recommended Citation

Patlolla, Dinesh Reddy, "Integration Of Maintenance Into Design And Sustainability Of Buildings. " Master's Thesis, University of Tennessee, 2015.
https://trace.tennessee.edu/utk_gradthes/3449

To the Graduate Council:

I am submitting herewith a thesis written by Dinesh Reddy Patlolla entitled "Integration Of Maintenance Into Design And Sustainability Of Buildings." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Industrial Engineering.

Rapinder Sawhney, Major Professor

We have read this thesis and recommend its acceptance:

Robert Keyser, Frank M. Guess

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Integration Of Maintenance Into Design And Sustainability Of Buildings

A Thesis Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Dinesh Reddy Patlolla
August 2015

In loving memory of my grandfather Patlolla Ranga Reddy who has been my source of inspiration throughout my life.

Acknowledgments

Firstly, I take this opportunity to express gratitude to all of the professors at UT for instructing me at the graduate level. A special thanks to my advisor, Dr.Rapinder Sawhney for conceiving the idea of this research and guided me through the course of my work and also completion of my Masters degree. I would like to extend my sincere gratitude to Dr. Robert Keyser and Dr. Frank Guess for reviewing my work and serving on my committee.

I would take this opportunity to thank the entire Department of Industrial and Systems Engineering who have not only been great colleagues but also good friends throughout. I place on record, my sense of gratitude to one and all, who directly or indirectly, have lent their hand in this venture. I am also grateful and indebted to Dr.Ninad Pradhan, for his sincere contributions, valuable guidance and encouragement extended to me. I am extremely thankful to UT facilities services for their co-operation and openness for contributing to this research. Special regards to Penny and Debra who took time off their regular schedule for my work and played a vital role in completion of work. Also special mentions to Bharadwaj Venkatesan, Enrique Macias, Gurudatt Sanil, Harshitha Muppaneni, Kaveri Thakur, Lavanya Marella and Mostafa Gafoorivarzaneh who invested time in getting this thesis started.

Family isn't always blood. I need to express my gratitude to two people who took care of me like my parents during my work. Thank you Dilip Patlolla and Deepthi Kondakindhi. I could not have made it without you! They say friends become our chosen family, but I am lucky enough to find a best friend in my family. Thank you Manisha Patlolla for not having

a problem in telling me that I'm acting like an idiot or I'm not that funny and for the never ending support.

Friends are those you know are always there though you don't see them. This is true in the case of Arun, Bhavana, Manasa, Nagesh, Pramod and Prateek who are great gifts of my life. I would like to thank Bharad, Guru, Harshitha, Kaveri, Ninad, Snigdha, Dhanush, Abhishek and all other friends at UT and Knoxville for all the time, memories and who made Knoxville a better place to stay.

I would like to thank my grandparents Late Sri. Ranga Reddy, Smt. Lakshmi and Sri. Lakshma Reddy and Smt. Rangamma for their love and care. I am most indebted to my parents, Sri. Yadav Reddy and Smt. Prabhavathi and my sister Ms. Harisha Reddy. I should have been blessed to have such amazing parents and take this opportunity to express my thanks for their unconditional love and unrelenting support, and encouragement. I also wish to express my thanks to all my uncles and aunts for the love and affection they have always showered onto me specially Anoosha Patlolla and Vinay Reddy.

Finally, I wish to thank the Almighty and my ancestors for their love and blessings, without which none of this would have been possible.

Abstract

Custodial maintenance is an important aspect of operational maintenance in a facility. Custodial maintenance improves an organization's discipline, performance and keeps surroundings healthy. That being said maintaining clean surroundings involves considerable cost. Custodial maintenance cost forms significant portion of building budgets, however these costs are often neglected. This research deals with the identification of variables that affect maintenance costs in a facility and reduction of maintenance costs. The minimization of cost is done by giving the administrator or facilities manager the option of selecting alternatives in frequency of maintenance, level of maintenance and the number of people required to complete a maintenance task. This allows the administrator to develop maintenance strategies to accommodate the custodial maintenance budget. An optimization model has been built to achieve the goals of the research. Furthermore, the custodial management system (CMS) developed based on an optimization model allows the administrator to design new buildings from the perspective of reduced custodial maintenance cost and to sustain these costs over time. A case study is presented to validate the working of the model and the software. A sensitivity analysis has also been presented to identify the best alternative for the case study.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Custodial maintenance affects an organization's profitability and performance and is necessary for clean facilities. It improves an organization's image and discipline. Organizations spend a significant component of their total operational budget in maintenance costs. However, the reduction and optimization of these costs has only been studied to a limited extent. We present a custodian management system which focuses on maintenance cost reduction.

Custodial maintenance costs are next only to staffing costs in an organization and directly impact its turnover and profits (Williams, 1996). In Norway, for example, maintenance costs are 33% of the total operational expenditure in public buildings (Bjoberg et al., 2007). Wauters (2005) found that maintenance in commercial buildings consumes up to 16% of the total facilities management budget. The UK Department of Health estimated that more than 480 million pounds were spent on hospital maintenance in England in 1997 (Al-Zubaidi and Christer, 1997). Table 1.1 shows that the high percentage of maintenance costs is a common theme across facility types and across countries. The ability to manage and control these costs is thus an industry-wide concern.

Maintenance costs can be broadly classified into two components: labor costs and material costs (Figure 1.1). Labor costs are the more dominant component of the two, accounting for 68% of total maintenance expenditure (Al-Zubaidi, 1997). The components of labor cost are hourly labor cost and burden costs (due to employee benefits). Hourly labor costs comprise 75% of the total labor cost (Frank D, 2010). While burden costs are a fixed component, hourly labor costs for a facility vary according to the number of custodian working hours. Since working hours ultimately influence an organization's profitability, a reduction in working hours is desirable.

This can be achieved by improving output per man-hour (Christer, 1990) through optimal scheduling and routing of custodians. A scheduling system can be implemented to give the facility manager the flexibility to change the assignment of tasks to rooms and the frequency with which these tasks are executed. Routing algorithms can be used to minimize travel time between rooms and to inform custodians about their cleaning routes and schedules. Designing simple and intuitive interfaces for scheduling and routing deliver a system which is accessible to all end users. A custodian scheduling system implemented with these core features has the additional advantage of being applicable not just to existing facilities, but also to budgeting and planning for planned facilities.

Table 1.1: Cost of maintenance based on building categories

No.	Category of Building	Country/Region	Cost of Maintenance	Reference
1.	Healthcare	UK	480 million pounds per year	Al-Zubaidi, Christer (1997)
2.	Commercial Buildings (Hotel)	Russia	$3,994,617 approx.	Wauters (2005)
3.	Public Buildings	United States	$7.5 million	Williams (1996)
4.	Public Buildings	Norway	33% of total maintenance cost	Bjoberg et. al (2007)

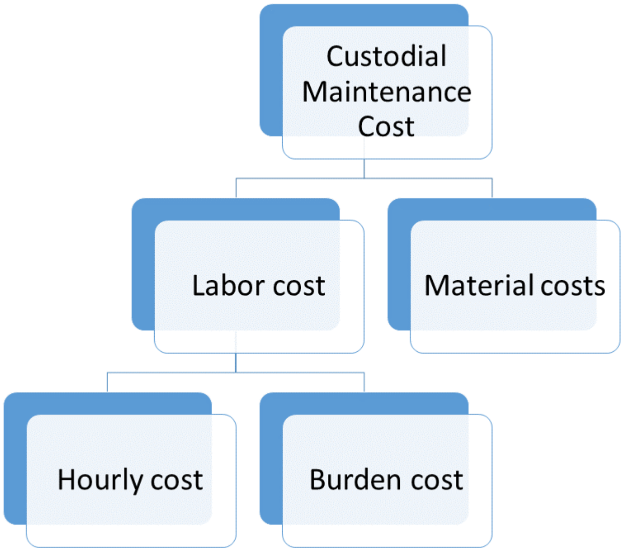


Figure 1.1: Custodial maintenance's cost components

1.1 Objective

The objectives of this thesis are as follows:

1. Create a custodial management system that focuses on labor driven costs for new and old buildings
2. Allow the user to evaluate alternative maintenance strategies by:
    - Altering the frequency of maintenance
    - Altering the level of maintenance
    - Optimizing the routing of the custodians
3. To develop a software prototype that allows the user to alter the frequency of maintenance, alter the level of maintenance, and optimize the routing of the custodians.
4. Estimate the time and labor cost savings after routing
5. Align the alternatives to accommodate the facility's budget

1.2 Modeling Approach

1.2.1 Variables identification

Variables influencing maintenance costs

There are several variables which contribute to labor and material cost components of custodial maintenance. The variables affecting costs have been identified through interviews conducted with the management and staff of the University of Tennessee's facilities services. The variables affecting a building's maintenance costs are represented in Figure 1.2. The connections on the diagram indicate which variables affect which type of cost (i.e., material cost or labor cost). As Figure 1.2 shows, working hours of a custodian are highly influenced by routing, scheduling, and the rate of work (i.e., production rates). The material costs are determined by the cleaning material utilization rate and the type of material allocated to a particular area.

Variable components

The variables influencing the total maintenance costs consist of multiple sub-variables. Figure 1.3 contains each of the 7 variables and 48 sub variables.

Area variable (V1) consists of information describing the building/area. These include square footage, number of rooms, distances between each of these rooms and number of floors in that building.

Priority variable (V2), consists of the frequency at which maintenance activities must be conducted in a particular area and the cleanliness level needed. Five priority variables can be changed based on the room type.

The day variable (V3) is the day of the week on which the maintenance tasks must be completed in a building.

Custodian variable (V4) is the number of custodians available to conduct maintenance activities in a building.

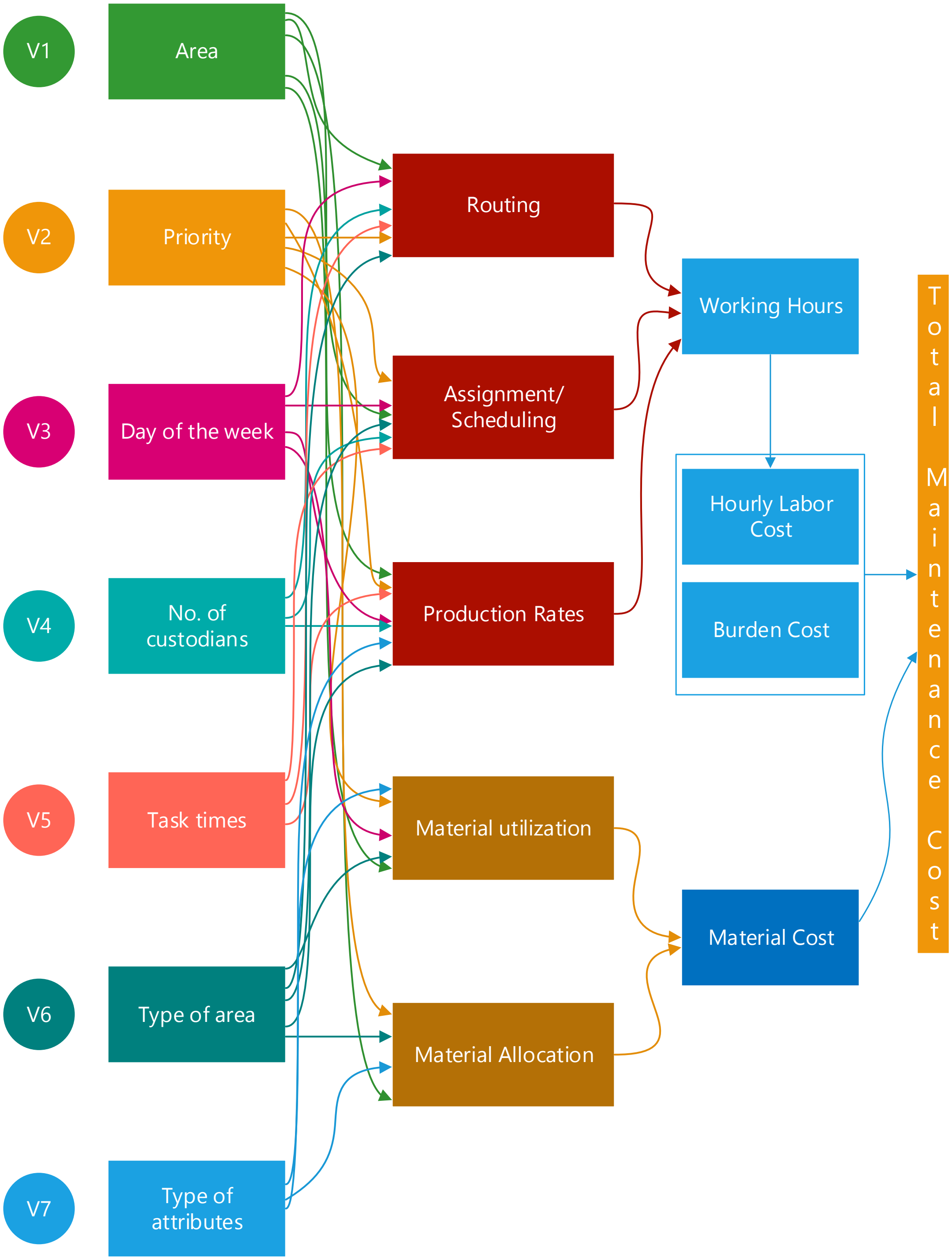


Figure 1.2: Identified variables and their effect on the costs

Task time variable (V5) is task times for 14 different types of tasks performed in a building. Each task time is independent of the others, with all tasks forming the major component of labor hours.

Area type variable (V6) gives the information on classification of an area. According to University of Tennessee's facilities management, there are two types of areas are: public and non- Public. Public areas are those which are most frequently used by multiple number of users. Classrooms and restrooms fall into this category. Non-public areas are used less frequently or are used by fewer people at any given time; these areas include office rooms, conference rooms and store rooms.

Attributes variable (V7) gives information on the type of attributes present in a room or area. There are two types of attributes for each area need to be cleaned: primary and other. The primary attributes are the square footage of floor and wall areas, while the other attributes are the entities present in a room as shown in the Figure 1.3.

1.2.2 Facility management hierarchy

Facility management has a well-defined hierarchy as shown in 1.4. Typically, facilities management in any organization has a facilities manager at the hierarchy's apex. According to Fayol facilities manager is responsible for planning, scheduling and organizing manpower involved in maintenance activities (Tay and Ooi, 2001). Under the facilities manager are area managers, who supervise a cluster of buildings or a cluster of areas in a building. Area managers are responsible for transporting and replenishing supplies, distributing schedules to the custodians, and supervising the custodians.

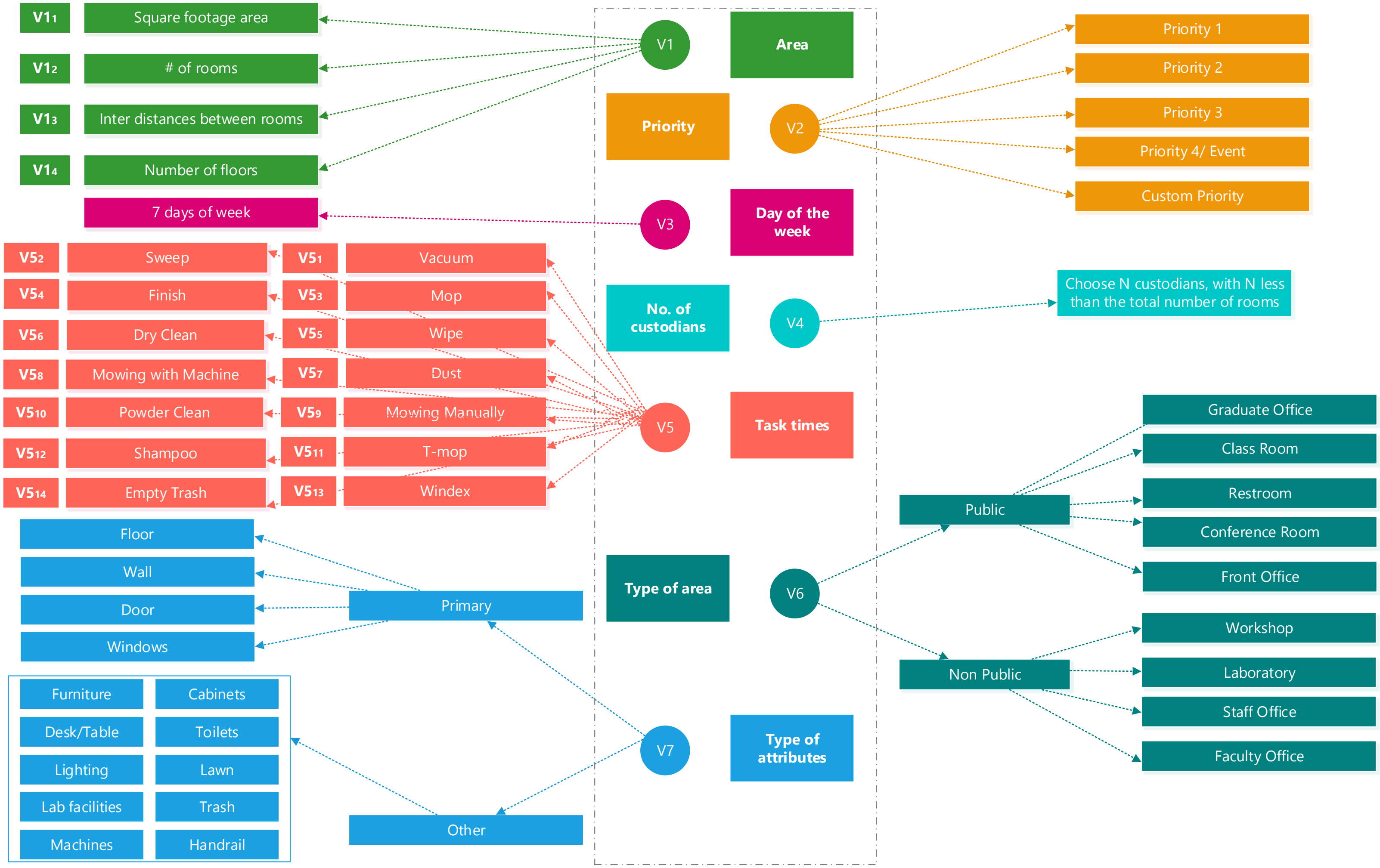


Figure 1.3: Identified variables and variable sub components

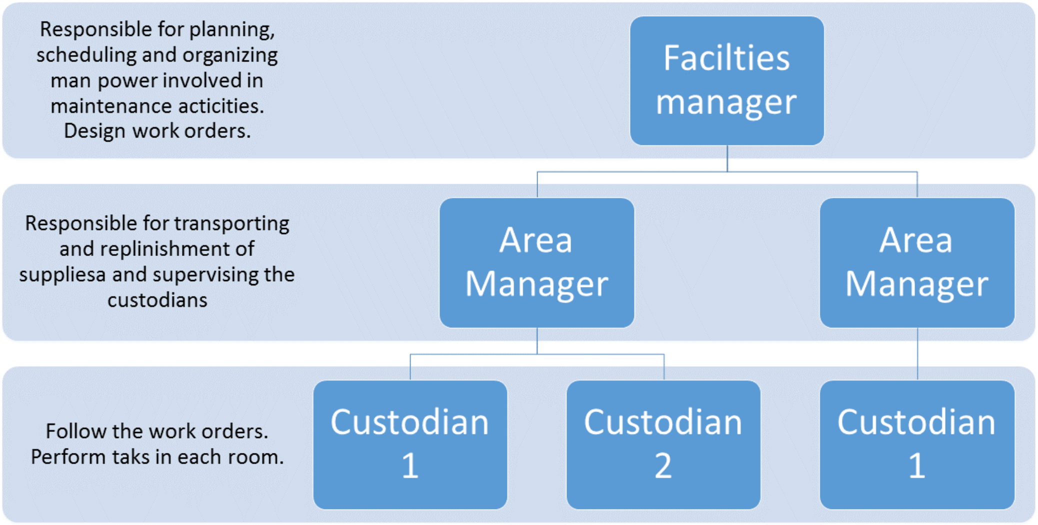


Figure 1.4: Custodial maintenance hierarchy and responsibilities

The custodians are the main workers and are ultimately responsible for a facility's cleanliness. The facilities manager and area managers design work orders, which are given to the custodians.

Figure 1.5 depicts the flow of variables across people in facilities management. The solid lines represent the data flow from higher-level to lower-level employees. The dotted lines represent the flow of corresponding color-coded variables from lower-level to higher-level employees. Custodians follow the work orders to carry out tasks in each area. Typically in a building an "area" is a room. Cleaning tasks depend on the room's properties and entities. The custodian begins at the storage location and moves into each assigned room or area. The order of rooms to be visited is not specified in the work order provided to custodians. A generic routing model in the present system is shown in Figure 1.6. The custodial management system is designed for facility manager and the custodians on the floor. The facility manager will be the administrator of custodial management system. Custodians use the custodial management system to read routes and work orders through the interface.

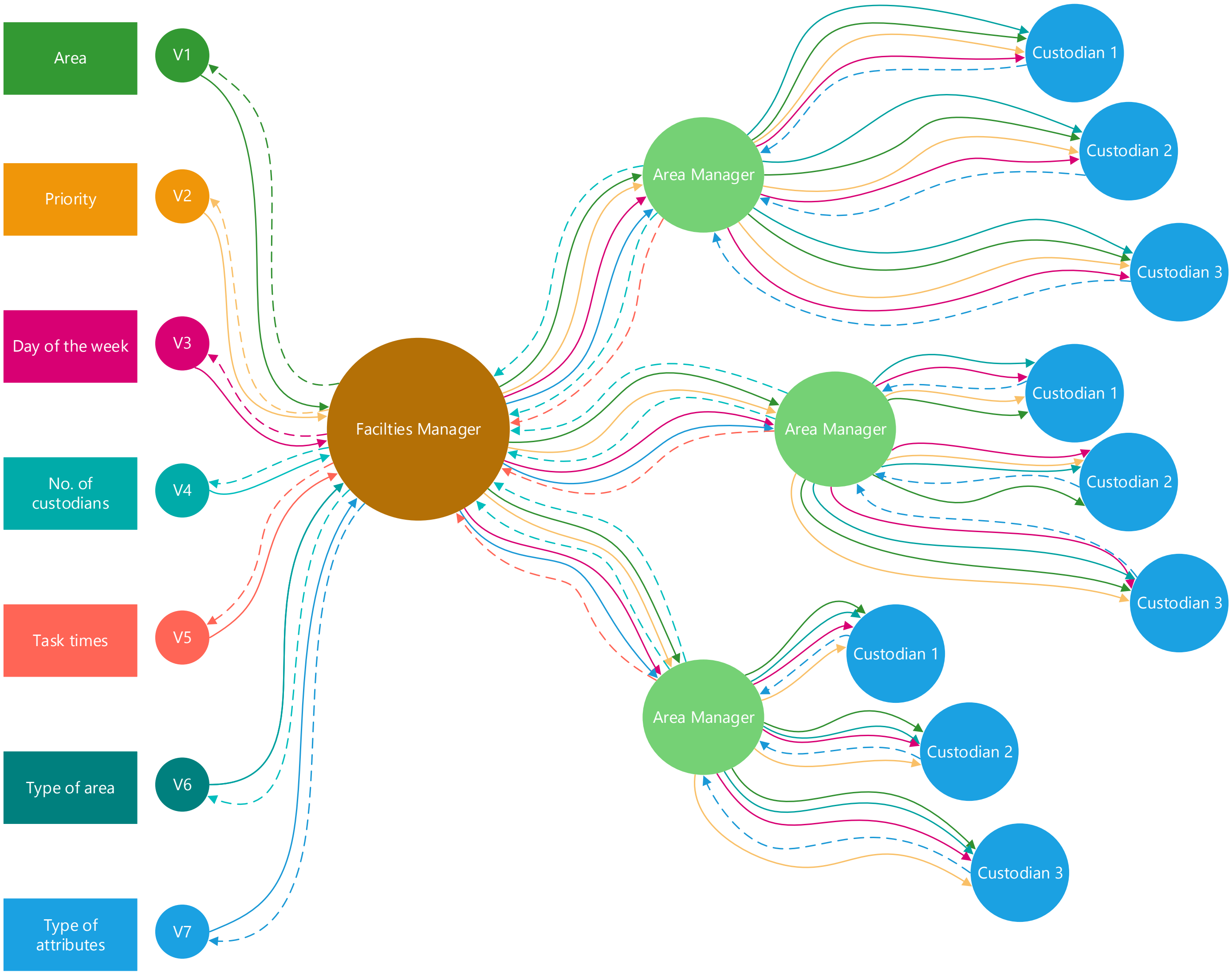


Figure 1.5: Current system depicting the flow of variables between different entities of facilities management

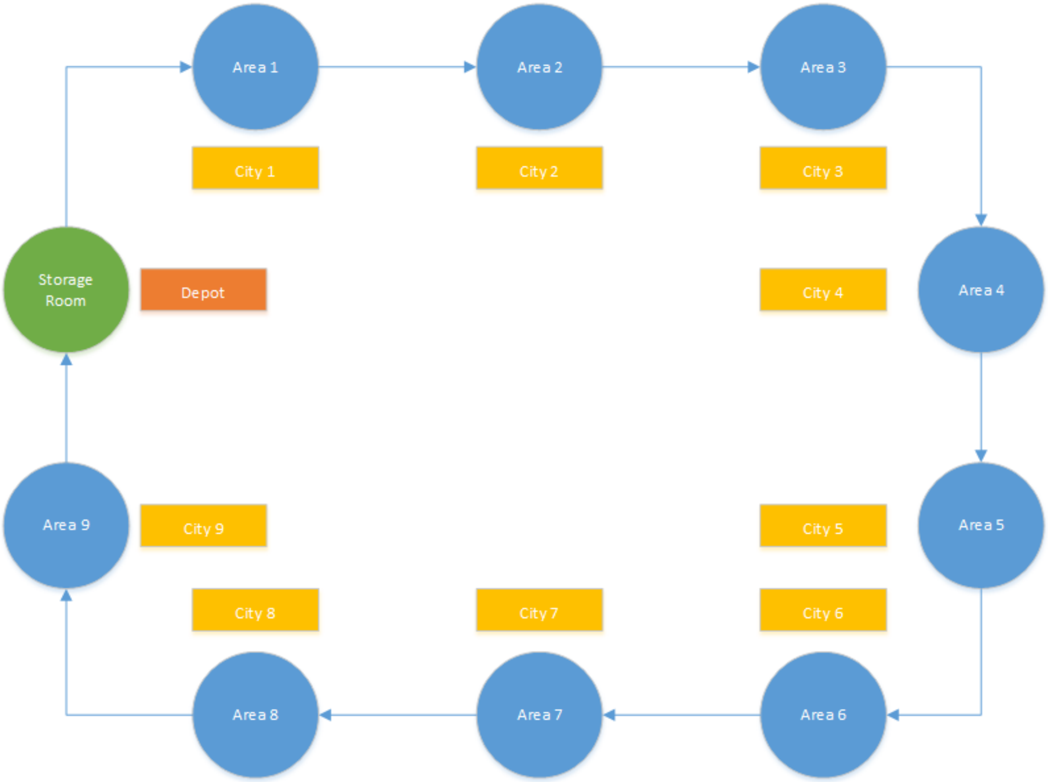


Figure 1.6: Generic route of custodians

1.2.3 Database system creation

An organized database is necessary in the custodial management system for a secure and robust implementation. The database is designed to allow the administrator to securely store and retrieve variables affecting maintenance costs and other relevant data. The database system allows the administrator to establish priorities in the administrator setting system.

1.2.4 Administrator setup system

The administrator setup system is designed in such a way that the administrator has the ability to set up maintenance schedules according to an organization's maintenance strategy. The schedule may also be changed to constrain costs within allocated yearly budgets. The system allows the following variables to be controlled by the administrator:

- Frequency of maintenance
- Number of custodians
- Level of maintenance

1.2.5 Optimization system creation

The optimization system uses a maintenance schedule to route the custodians each day. The purpose of optimization is to minimize the total labor cost by minimizing the total custodian working hours through routing. The variables affecting the custodians' routing are considered to build the mathematical model. Figure 1.7 shows the variables necessary for routing custodians. Variables V1, V4, V5 are used in the optimization model. A custodian's total labor time is the sum of the total task time and travel time. Therefore, total labor time is

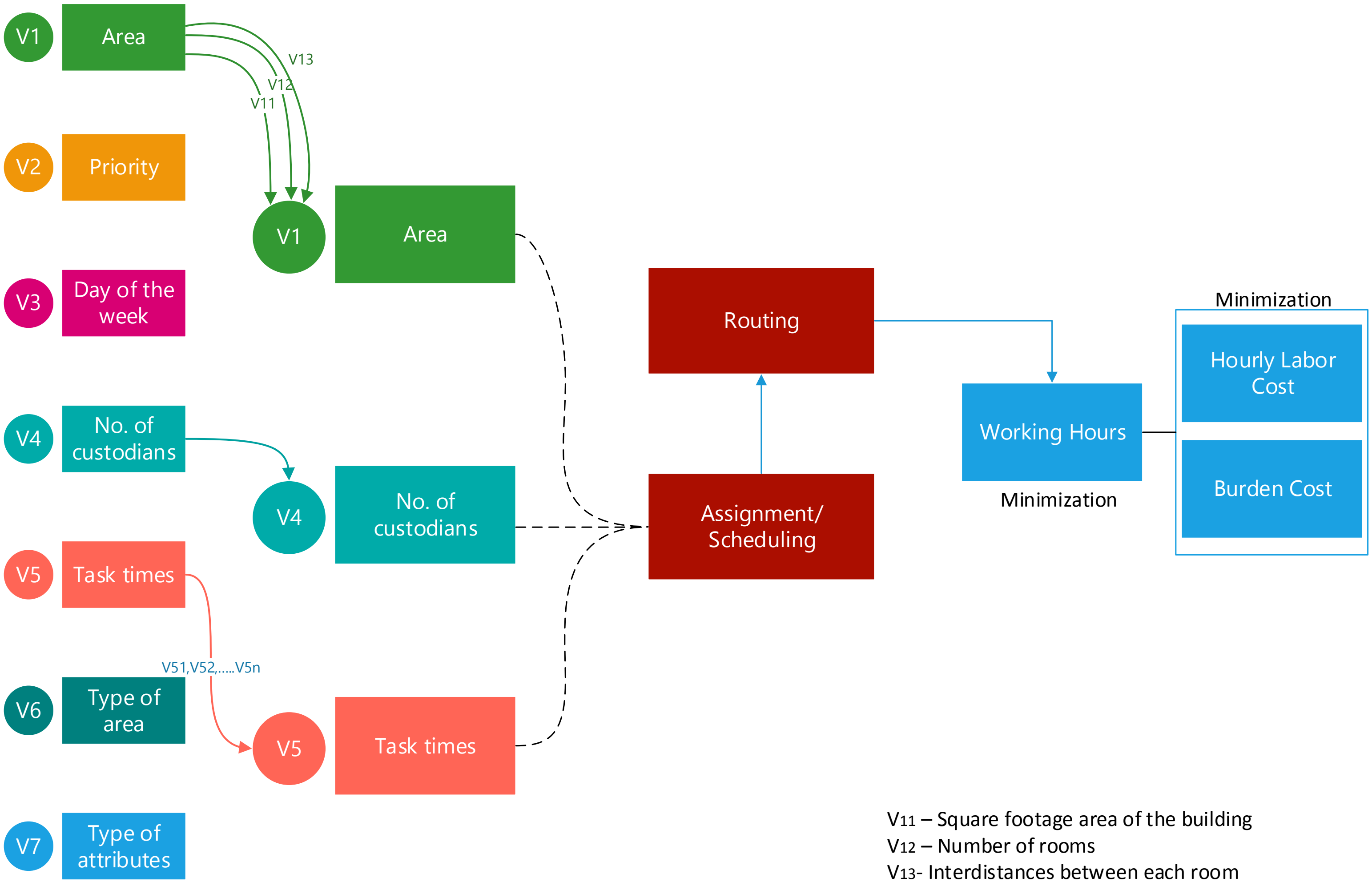
$$t = V5 + (V1_3/\vartheta) \tag{1.1}$$

where $V5 = V5_1 + V5_2 + ...V5_{14}$ which is the sum of individual task times from Figure 1.3 and ϑ is the average speed of each custodian.

Constraints

The following constraints are applied to the model:

- Each custodian starts from one depot and returns to the same depot.
- A custodian's shift time in 8 hours, including a lunch break and two short breaks. Custodians cannot work for more than 8 hours.
- A custodian can visit a room only once.
- A custodian should finish all the tasks in a room before moving to the next room to be cleaned.



V11 – Square footage area of the building
V12 – Number of rooms
V13- Interdistances between each room

Figure 1.7: Variables used in optimization

Relation of Vehicle Routing Problem (VRP) to the Current Problem

Formulating this problem is similar to a typical Vehicle Routing Problem (VRP). The relationship between the entities in a VRP can be related to the entities in this formulation. The carts custodians use can be correlated to vehicles in the Vehicle. The custodians can be considered the drivers of each vehicle and the storage room from which custodians start their duty can be considered a depot. In VRP, each vehicle has to travel to one city, fulfill the customer's demand in that particular city, and move to the next corresponding city for the next customer. All the vehicles have to start from and return to the same depot. The depots in the present case are the storage rooms for cleaning materials and are where these carts are kept. Figure 1.6 is a graphical representation of custodial routing in relation to VRP. In a typical VRP, the total distance a truck travels is minimized. In the current problem, a custodians total travel time should be minimized. Assuming constant speed for all custodians, minimizing total time minimizes the distance custodians travel, hence minimizing cost.

1.3 Scope and assumptions

1.3.1 Modeling assumptions

Single depot assumption

All custodians on a floor are assumed to start and end at a specified storage room, referred to as a depot. This depot is the start and end point of the routing model. This defines the formulation as a Single Depot Vehicle Routing Problem. The system calculates the optimal routes relative to the depot.

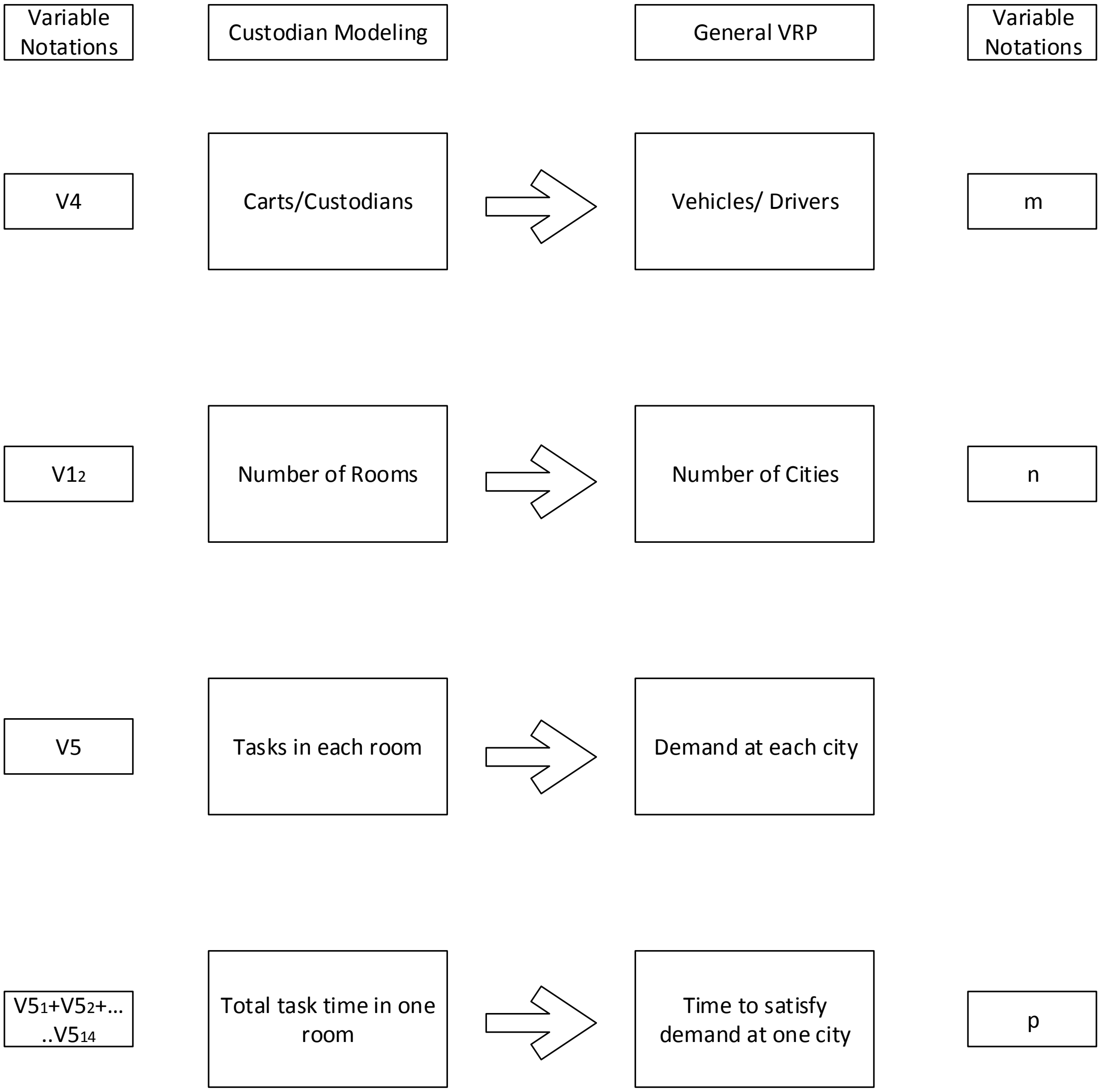


Figure 1.8: Relationship of attributes to VRP

Task simplification assumptions

The data is collected for 14 tasks and task times, the number of variables going into the model had to be decreased for simplification. The 14 different tasks were combined into 4 tasks: vacuuming, cleaning windows, emptying trash, and floor cleaning. All tasks except vacuuming, cleaning windows, and emptying trash have been encapsulated into the floor clean task. All the task times are based on the International Sanitary Supply Association's (ISSA's) standard times based on square footage area.

1.3.2 Floor plan assumptions

Calibration assumption

The optimization system uses Portable Network Graphic (PNG) images for routing and scheduling user interfaces. For distance and time calculations, 1 pixel on the png image is assumed to equal one square foot of area.

Scaling assumptions

For a building with multiple floors, the current system's vehicle routing problem for each floor must be solved; the results would be equally good for solving the vehicle routing problem for all the floors combined. The current system has all the types of rooms associated with an educational institution. This system can also be used for office and commercial buildings, but not for health care facilities, which require more information and which involve specialized tasks and more time. Also, more tasks and priorities have to be added for the current system to work in an industry environment. However, the new system has a provision for including additional tasks or removing existing tasks for different or specialized environments.

1.4 Organization of thesis

The present thesis is organized as shown in Figure 1.9. Chapter two provides a literature review to understand the current custodial maintenance models and routing problems.

Chapter three describes methodology and the development of four subsections, discussing the newly designed system's four parts. Figure 1.10 is an overview of the prototype, users, and variables' data flow through the system. Chapter four describes prototype testing and presents the results and analysis. Chapter 5 provides conclusions and a proposal for future work.

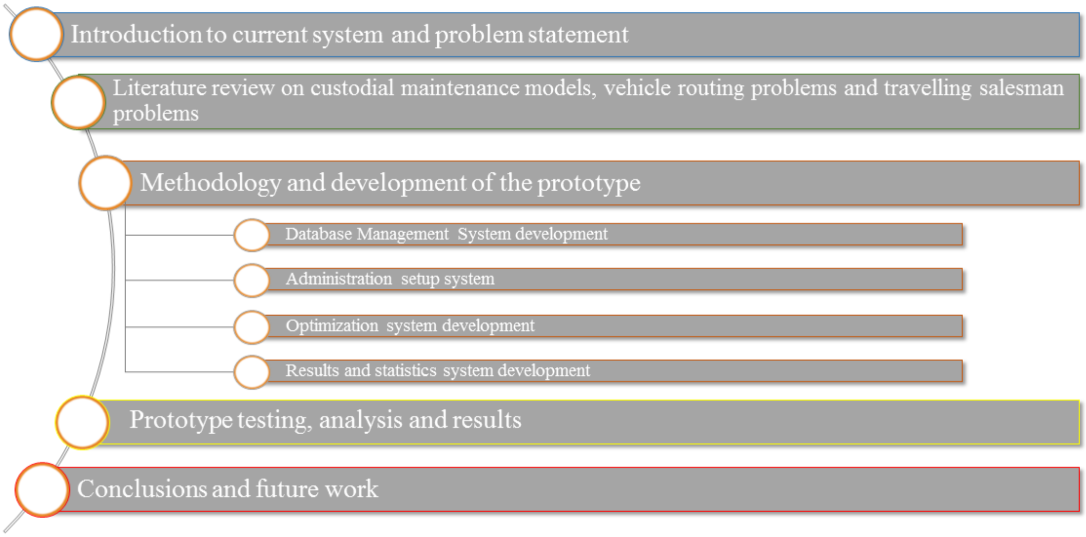


Figure 1.9: Roadmap of presented work

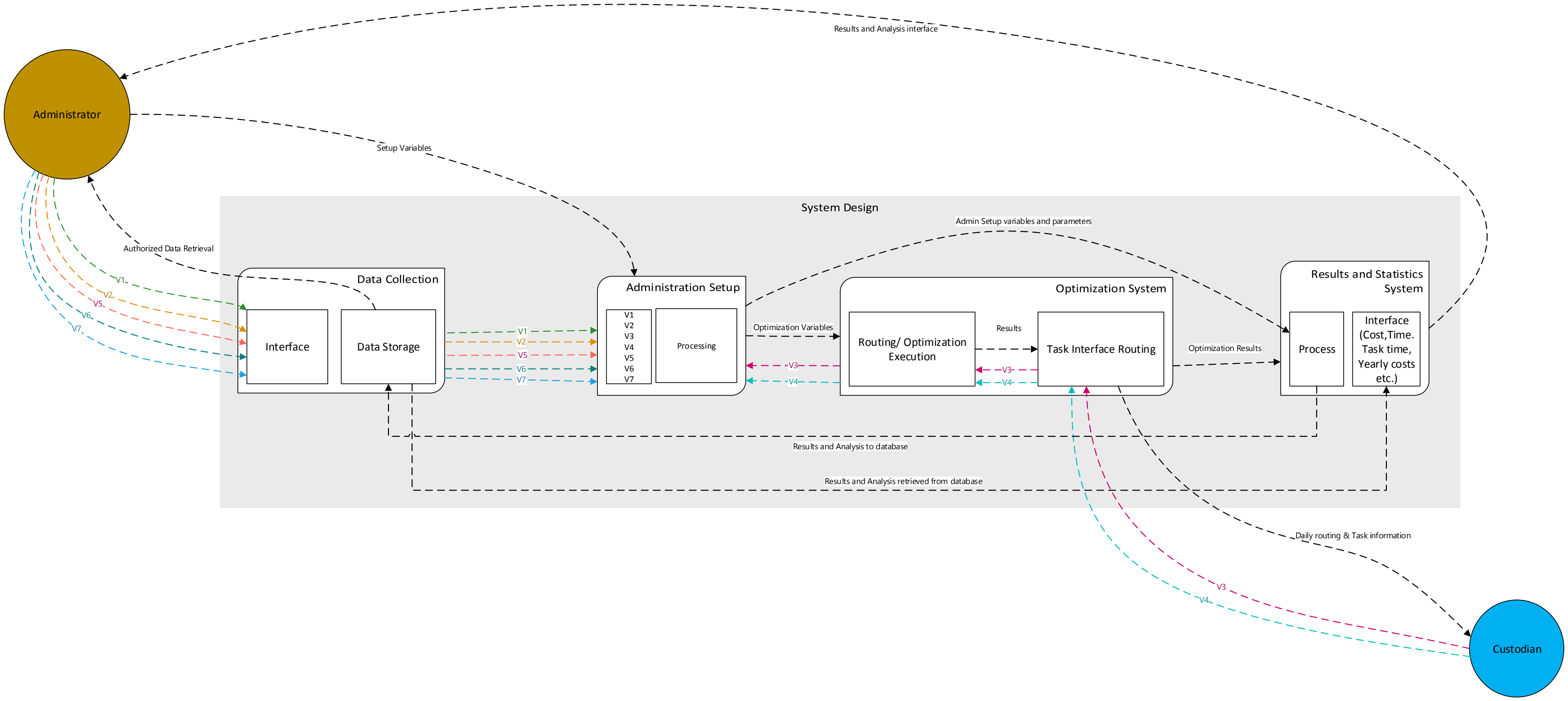


Figure 1.10: The new system overview and components

Chapter 2

Literature Review

2.1 Mathematical models in custodial routing

From literature mathematical models in custodial maintenance are few and far between. Of these, optimization models in custodial maintenance are even more sparse. Table 2.1 lists existing research work in custodial maintenance optimization modeling.

Duffuaa and Raouf (1992) have built a simulation model to determine the size of a maintenance crew in an industrial environment. Custodial maintenance is correlated with this model, since the number of custodians required to maintain the premises is a function of number of rooms in that particular premises. But the model is a specialized simulation model which has been developed specifically for an industrial environment. Similarly, Al-Zubaidi and Christer (1997) have designed a model for determining maintenance requirement of a hospital. The model can be applied only to that particular health care facility and cannot be adapted to any other facility.

Attempts have been made to build mathematical models to the literature however each one has its own limitations. Figure 2.1 shows limitations of models on the x-axis of the plot and percentage of current models to which a limitation applies on the y-axis. About 35% of the models are limited by a lack of data availability, inconsistent data or improper data collection methods.

For operational maintenance models the system has to have a database in which relevant information is stored and retrieved. Data has to be collected under strict rules using a well-defined system component structure(Rommert Dekker, 1998). Therefore for a model to sustain for longerperiod of time a proper database system has to be designed for the user to retrieve and store relevant information.

Due to complexity of models, their adaptation to practical environments has been slow (Dekker, 1996). About 17% of the models have not been used in practice due to their complexity. Models are not easy to apply and understand for users who have limited knowledge of the optimization models. Application of models also requires good formulation of the problem.

Around 25% of models and applications have limited adaptability as they were built for one particular environment and cannot be used in any other environment. Further, it is advantageous to know the cost of maintenance of building even before it is built using the floor plans. This would allow the user to consider various alternatives such as altering the materials used in it within the limitations of a given budget in order to reduce the cost of maintenance of that building. There is no known precedent in literature for this kind of capability in maintenance systems.

Also, custodian scheduling involves maintenance setting priorities to efficiently use available man power (Dekker, 1996). Priorities of maintenance include frequency and type of maintenance. User of any optimization model should be able to tweak priorities, plan routing and also combine maintenance activities.

In summary, an optimization model in custodial maintenance requires a good database system to be built to collect and retrieve data. Moreover, the optimization model and its results should be easily understood by the user. This would enable the user to customize priority settings for accurate estimation of maintenance costs.

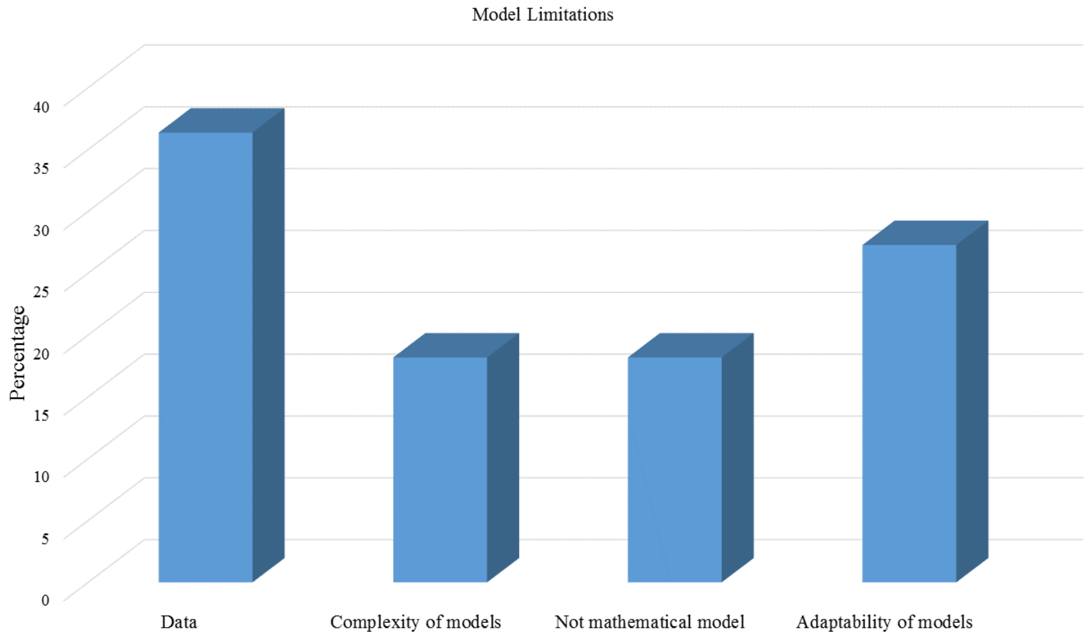


Figure 2.1: Limitations of mathematical models in literature

2.2 Success of Vehicle Routing Problem (VRP) in various areas of application

The custodian routing problem in this thesis is a generalized "Vehicle Routing Problem" (VRP). VRP is concerned with determination of the route that results in minimum time for visiting each of the nodes in a given problem. It has many applications and can be formulated to needs of the problem and its constraints. This makes VRP "a truly one of the greatest success stories of research" as said by Laporte and Osman (1995). Several successful implementations of computerized routing softwares have been documented in literature. These successes can be attributed in part to algorithmic advances in the field of vehicle routing and also to the development of new software and computer technologies.

Table 2.1: Summary of previous works on optimization models in custodial maintenance

Author	**Title**	**Area**	**Inference**	**Drawbacks listed**
Duffuaa, and Rauf (1992)	Simulation model for determining maintenance staffing in an industrial environment	Simulation, Maintenance models in industry	Simulation model to determine the size of maintenance personnel based on	Custodial maintenance accounts for 33% of the cost of managing and maintaining Norway's public building stocks.
Zubaidi, and Christer (1997)	Maintenance manpower modelling of hospital building complex	Simulation	Simulation in healthcare facilities and manpower modelling.	This model has been developed for specific hospital building and cannot be used anywhere else.
Dessouky and Bayer (2002)	A simulation design of experiments modeling approach to minimize building maintenance cost.	Building maintenance estimation techniques	Model to determine the funds allocated over a period of time for maintaining buildings.	Does not focus on custodian routing or scheduling. Poor data availability.
Dekker, R. (1996)	Applications of maintenance optimization models: a review and analysis	Maintenance optimization modeling	Review paper for application of optimization models in the field of maintenance.	Due to complexity of optimization models, applications have come slowly off ground. Adequate data is often lacking and models are not easy to understand for real time users
Dekker,R and Philip (1998)	On the impact of optimization models in maintenance decision making: the state of the art	Maintenance optimization modeling	For operational maintenance the maintenance application system has to have a database in which relevant information is stored and retrieved.	Importance of a database in a system to accurately collect data and use the data for modelling and optimization purposes.
Dekker, R and Scarf, P. (1995)	Integrating optimization, priority setting, planning and combining of maintenance activities	Maintenance optimization models	Scheduling involves maintenance priority setting using available manpower efficiently.	User of any optimization model should be able to tweak priorities, plan the routing and also combine the maintenance activities which was seldom attempted in earlier works.

2.2.1 Areas of impact of application for VRP

Mobil Oil Corporation has implemented VRP through its Computer Assisted Dispatch (CAD), which is a collection of integer programming methods used in real time, transaction driven information management system for the dispatch of their tanker trucks. An interface system has been used to control the distribution of light petroleum products to customers in the continental United States and has substantially reduced costs and staff while improving customer service (Brown et al., 1987). Evans and Norback (1985) in their research implemented a heuristic based decision support system, which utilizes computer graphic pictures of routes in a large food service distribution network. An implementation of the interface called distribution decision support system (DDSS) was tested and results indicated a 10.7% cost savings. In a pilot study conducted by Mathews and Waters (1986) it was found that the proportion of time community nurses spent on traveling was around 22%. The total travel time of nurses was reduced by applying VRP. A VRP has been implemented for optimizing the schedule and routing of the distribution of a newspaper. The scheduling of newspaper distribution has substantially decreased the staff costs. VRP was also extended to soft drink industry. Case studies of Golden and Wasil (1987) in this area focus on critical vehicle routing issues of the soft drink industry. The focus was on inter-facility transfer of soft drink products, delivery of products that have been ordered in advance by customers, and delivery of soft drink products that are sold by drivers to customers. Table 2.2 shows the various applications of VRP and computer based programs that have been developed and tested and their association with the present research.

Summary of previous VRP models and its association with current study

Most of the existing VRP models have made use of computer graphic images to get inputs for optimization models. It is noticeable that VRP was applicable in a wide range of industries like the oil industry, food industry, healthcare etc., thus making it an important methodology for optimizing costs of staffing through proper routing and scheduling. Routing has been conducted on personnel teams which makes the problem relevant to the current study.

2.2.2 Solution Methods for VRP

The solutions to VRP can be generalized by adding conditions as required to it. For finding a shortest route which passes through each of n given points once (assuming that each pair of points is joined by a link), the total number of different routes through n points is $(n/2)!$ which is very large even for a small n (Dantzig and Ramser, 1959). Laporte (1992) has surveyed the broad literature of VRP solution algorithms and provided a classification of various VRP solutions.

Exact methods

In 1986 Laporte, Mercure and Nobert in their research 'An exact algorithm for the asymmetrical vehicle routing' have proposed the assignment lower bound and a related branch-and-bouund algorithm (Laporte et al., 1992). The algorithm exploits the relationship between the VRP and its relaxation m-TSP. It establishes m least cost vehicle routes starting and ending at the depot, and every remaining vertex is visited only once (Laporte et al., 1992). Using this methodology Laporte et al. (1992) have solved to optimality asymmetrical VRPs involving up to 260 vertice or routes. The extensions to this methodology were made through several side constraints (Laporte, 1992). Christofides et al. (1981) have developed the k-degree center tree and a related algorithm for symmetrical VRPs with fixed number of vehicles. The authors have embedded the lower bound in the branch-and-bound algorithm and solved VRPs ranging in size from 10-20 vertices or routes. Solution for VRP using dynamic programming was first proposed by Eilon et al. (1971) in their book 'Distribution Management' in 1971. The ratio of the lower bound to that of the optimum solution using this method varied from 93.1% to 100%. But it could only solve problems containing 10 to 25 vertices.

Table 2.2: Summary of VRP applications and the association of current study

Author	**Title**	**Field of impact**	**Interface**	**Impact**	**Association with current study**
Brown,G., Ellis, C., Graves, G., Ronen, D. (1987)	Real time wide area dispatch of mobile tank trucks	Petroleum industry/ Interfaces	Computer Assisted Dispatch (CAD)	Routing and dispatching of trucks to reduce costs through VRP.	Building interface for VRP and generating solutions from data collected.
Evans,S., Norback, J. (1985)	The impact of decision support system for vehicle routing in food industry	Food industry/ Interfaces	DDSS	Decision support system , utilizing computer graphic pictures for cost saving.	Building a budget monitoring system utilizing computer graphic pictures to route the custodians for cost savings.
Mathews,B., Waters, C. (1986)	Computerized routing for community nurses	Healthcare services	N/A	Minimization of time of travel of nurses and improving the efficiency	Minimization of time of travel of custodians to minimize total time.
Holt, JN., Watts, AM.	Vehicle routing problem in newspaper industry	Newspaper industry	N/A	Scheduling and routing of newspaper distribution to reduce costs	Scheduling and routing of custodians to reduce total working time and total cost of maintenance.

Fisher and Jaikumar (1978) have developed a three index vehicle flow formulation with VRP capacity constraints and no stopping times. This algorithm doesn't work for the present case because there are stopping times in the custodian modeling at each node. Table 2.3 shows a summary of literature review of all the exact algorithms for VRP.

Heuristics

Clarke and Wright (1964) developed an iterative procedure that enables the rapid selection of an optimum or near-optimum route after considering certain theoretical aspects. It starts with vehicle routes containing the depot and one other vertex. Each iteration two routes are merged according to the largest savings that can be generated. The sweep algorithm is cluster first and route second algorithm which was initially proposed by Wren and Holliday (1972) and later on attributed to Gillett and Miller (1974) who gave its name. It is an efficient algorithm, for solving medium as well as large-scale vehicle-dispatch problems with load and distance constraints for each vehicle. The locations that are assigned to each route are determined according to the polar-coordinate angle for each location. An iterative procedure is then used to improve the total distance traveled over all routes (Gillett and Miller, 1974).

Later on Fisher and Jaikumar (1981) came up with an algorithm which uses Generalized Assignment Problem to cluster the nodes and solve the TSP based on GAP. It presents a heuristic algorithm in which an assignment of customers to vehicles is obtained by solving a generalized assignment problem with an objective function that approximates delivery cost (Fisher and Jaikumar, 1981). Recently, solutions to multi depot vehicle routing problems have been obtained through genetic algorithms by Surekha and Sumathi (2011). The customers are grouped based on distance to their nearest depots and then routed through Clark and Wright saving method (Surekha and Sumathi, 2011) using Matlab as a solver.

Table 2.3: Summary of solution algorithms to VRP

Author	**Title**	**Software**	**Type/ Method**	**Inference**
Laporte, G., Mercure, H., Nobert, Y. (1986)	An exact algorithm for asymmetrical capacitated vehicle routing problem	N/A	Exact Method	Exploits the relationship between VRP and its relaxation, m-TSP. It establishes least cost vehicle routes starting and ending at a depot.
Laporte, G., Mercure, H., Nobert, Y. (1991)	A branch and bound algorithm for a class of asymmetrical vehicle routing problems	N/A	Exact Method	A branch and bound algorithm was deigned for AVRP and solved to optimality involving up to 260 vertices or routes.
Christofides, N.	Vehicle routing	N/A	Exact Method	K-degree center tree and a related algorithm.
Eilon, S., Watson-Gandy, C.D.T., and Christofides, N. (1971)	Distribution Management : Mathematical Modelling and Practical Analysis	N/A	Exact Method	Dynamic programming was proposed. VRP solutions were obtained containing 10-25 vertices or routes.
Fisher,M.L., and Jaikumar, R, (1978)	A decomposition algorithm for large scale vehicle routing	N/A	Exact Method	The algorithm forms a feasible and nearly optimal solution for non identical vehicles even though if it is not run to completion.

Kovàcs (2008) has introduced a genetic algorithm through which customers or cities have to be visited and packages have to be transported to each of them, starting from a basis point on the map. The algorithm also uses Matlab as a solver to solve the VRP. Table 2.4 shows a summary of literature in heuristic solutions to VRP.

2.2.3 Summary of solution methods to VRP

From the literature of exact solutions and heuristic solutions it is clear that exact algorithms can only solve relatively small problems (Laporte, 1992). Since the current problem of custodian routing can become large depending on the number of nodes selected, exact algorithms will not be used. Heuristics can solve relatively medium sized problems to large sized problems (Gillett and Miller, 1974). However, results show that sweep based algorithms solutions provide excellent results in short computing times (Renaud and Boctor, 2002). The sweep based heuristic produced best known solutions to certain problems and is sometimes better than the tabu search proposed by Laporte (1992) in heuristics (Renaud and Boctor, 2002). Hence, the custodian routing model uses the sweep algorithm to solve the VRP.

Table 2.4: Summary of solution algorithms to VRP and softwares used

Author	**Title**	**Software**	**Type/Method**	**Inference**
Wren, A., and Holliday, A. (1972)	Computer scheduling of vehicles from one or more depots to a number of delivery points	N/A	Heuristics	Uses the sweep algorithm to solve the VRP with single and multi depot and multiple nodes.
Gendrau, M., Hertz, A., Laporte, G. (1991)	A tabu search heuristic for vehicle routing problem	N/A	Heuristics	Constructs a sequence of solutions and then executes improvements.
Clarke,G., and Wright, J.W, (1964)	Scheduling of vehicles from a central depot to a number of delivery points	N/A	Heuristics	This classical algorithm ignores vehicle fixed costs and fleet size.
Surekha P, S.Sumathi (2011)	Solution to Multi-Depot Vehicle Routing Problem Using Genetic Algorithms	Matlab	Genetic Algorithms/ Heuristics	Customers are grouped based on nearest distance to their depots and then routed using Clark and Wrights algorithm.
Akos Kovacs(2008)	Solving the Vehicle Routing Problem with Genetic Algorithm and Simulated Annealing	Matlab	Genetic Algorithms	Uses Matlab as the software to solve the genetic algorithm VRP.

Chapter 3

Methodology

The methodology has been divide in to two phases namely:

1. Formulation
2. Implementation

The formulation phase consists of the mathematical formulation and techniques used in formulating the routing problem for custodian modeling. The implementation phase describes the techniques used to implement the formulated model and the technical implementation of the 4 sub systems and the software.

3.1 Formulation Phase

The formulation phase has been divided into 3 parts as shown in Figure 3.1. The first part is finding the shortest path between the nodes. While there might be multiple ways of traveling from one room to the other, calculating the shortest path minimizes travel time. The second part is clustering of rooms and their assignment to multiple custodians. The third part is the formulation and solution of the optimization model.

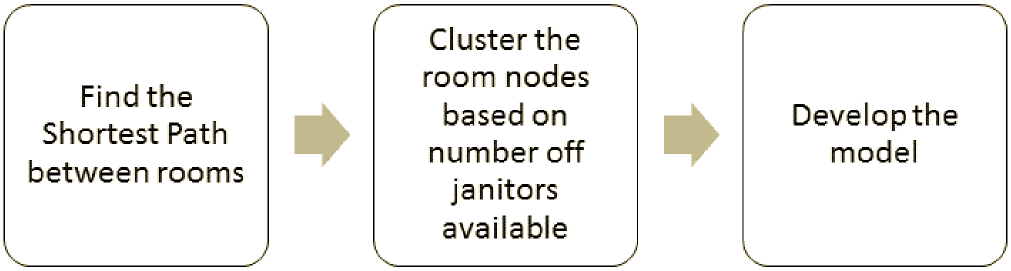


Figure 3.1: Parts of the formulation phase

Shortest Path Algorithm

In a given floor plan there may be multiple ways or paths to travel from one node to another. For minimization and optimization it is essential that the shortest of all possible paths from node N_i to node N_j is chosen. Djikstra's algorithm is used to determine the shortest path between two nodes in the floor plan.

The notations used in this problem have been adopted from "Transportation Networks: Aqualitative treatment" by Teodorovic (1986). Shortest path algorithms assume that all link lengths joining the nodes are non-negative. Every node is assigned a label with two components: the shortest distances between nodes and also the shortest paths from a particular node to all other nodes. A label could be either permanent or temporary. A node is assigned a permanent label if the shortest distance from the source node is achieved and there are no more shortest paths possible. The algorithm stops when all nodes turn permanent.

The notations used in the algorithm are as follows:

$l(N_i, N_j)$: length of link joining node N_i to node N_j.

a : starting node for which the shortest path are to be calculated to all other nodes.

d_{aN_i} : the shortest known path from node a to node i in the network.

q_i : the immediate predecessor node of i on the shortest known path from node a to node i found so far.

c : the last node to have moved to being in closed state.

$x : x = d_{aN_i}$

$y : y = q_{N_i}$

Dijkstra's algorithm consists of five steps to calculate the shortest path as follows:

1. The process starts from node a. The length of shortest path from node a to node a is 0, therefore $d_{aa} = 0$.

   $$d_{aa} = 0, \forall N_i = a \tag{3.1}$$

   The immediate predecessor node of the selected node will be denoted by symbol $+$. Therefore $q_a = +$. The lengths of the shortest paths from node a to all other nodes $N_i \neq a$ in the shortest path are not known at this point. The only node up to this point which is in closed state is node a. Therefore $c = a$.

2. The transformation of temporary labels into permanent labels for a node is done by checking the branches (c, N_i) that exit from last node which is in closed state (node c). If node N_i is in closed state then, the algorithm proceeds to the next node. If node N_i is in open state then it is labeled d_{aN_i} based on the equation:

   $$d_{aN_i} = min[d_{aN_i}, d_{ac} + l(c, N_i)] \tag{3.2}$$

   In (3.2) the left side of the equation is the new label of node N_i. The d_{aN_i} is the old label for node N_i.

3. The values of d_{aN_i} of all nodes which are in open state are compared. The node with smallest d_{aN_i} value say N_j is chosen. Node N_j passes from an open state to closed state if there is no path from a to N_j shorter than d_{aN_j}.

4. Once it is ascertained that node N_j is the next node to pass from an open state to closed state, the immediate predecessor node of node N_j is determined. The shortest path which leads from node N_a to node N_j is determined by taking into account the

lengths of all branches (N_i, N_j) which lead from closed nodes to node N_j satisfying the equation:

$$d_{aN_i} - l(N_i, N_j) = d_{aN_i} \tag{3.3}$$

If a node N_t satisfies the above equation, then node N_t is the immediate predecessor of node N_j on the shortest path which leads from node a to node N_j. Now, $q_{N_j} = N_t$

5. When all the nodes in the network are closed i.e., all labels are permanent, the algorithm stops and gives the shortest path . If there are any more open state nodes then the algorithm repeats from step 2.

Clustering

The custodian scheduling problem is framed in the form of a Vehicle Routing Problem (VRP), in which vehicles (cleaning carts) guided by drivers (custodians) complete a tour of the network (floor of a building) of nodes (rooms) before returning to the depot (custodian storage room). In case of a single custodian, the solution of the VRP is a route in which that custodian visits each room once before returning to the depot. A single custodian, multiple room setup is called a Traveling Salesman Problem (TSP).

When multiple custodians work on the same floor, the solution of the VRP also needs to include the assignment of rooms to be cleaned by each custodian. This problem is NP-hard and an exact optimal solution for such a setup cannot be found analytically. Several VRP algorithms, all of which approximate the solution, exist. The algorithm implemented in this case is known as the Sweep Algorithm.

The Sweep Algorithm applies to planar cases of VRP, i.e. environments in which all nodes lie in a single plane. This constraint is satisfied by the custodian scheduling problem and hence the Sweep Algorithm is applicable to the VRP outlined here. The first step in Sweep Algorithm is clustering, which results in the allocation of rooms to custodians. Once rooms have been allocated, each custodian is routed from the depot to the rooms and back by solving a TSP for allocated rooms.

The classic form of k-means clustering was implemented initially. However, k-means does not allocate rooms in a manner where each custodian works approximately for the same number of hours. Equal time allocation is an important consideration for custodian scheduling from the administrator's standpoint. Hence, a modified clustering algorithm similar to k-means was developed and used in the model. Both clustering implementations are explained ahead.

k-means clustering

The objective of k-means clustering was to divide the set of room nodes on a floor into k clusters, where k is the number of custodians working on that particular floor. K-means algorithms begin with an arbitrary initialization of cluster centers and progressively assign nodes to each cluster based on their proximity to the cluster mean. The assignment of a node to a cluster moves the mean location of the cluster and consequently changes the distances of cluster means from unassigned nodes. The algorithm iteratively assigns nodes in this manner until a stopping condition is reached where the cluster centers are not shifted after node assignment.

K-means clustering is set up as follows. Let,
M be the total number of custodians
N be the total number of rooms
(x_i,y_i) be the planar coordinates of the ith node

Figure 3.2 shows a sample spatial arrangement of planar nodes, each representing a room in the floor plan. The nodes are labeled N_i,$i = 1 \ldots n$, with coordinates (x_i, y_i). Node N_0, with coordinates (x_0, y_0), is defined as the depot node for this example, and clustering is carried out based on coordinates of other nodes relative to this node. The depot node is itself not included in any cluster. A shift of origin of the coordinate system is needed to find coordinates of nodes relative to the depot. Relative coordinates $(\overline{x}_i, \overline{y}_i)$, $i = 1 \ldots n$, are given by:

$$\overline{x}_i = x_i - x_0 \tag{3.4}$$

$$\overline{y}_i = y_i - y_0 \tag{3.5}$$

The nodes with their revised coordinates are shown in Figure 3.3 .

The number of clusters in k-means is equal to the number of custodians working on a floor. Cluster means are initialized with arbitrary coordinates. K-means iteratively operates on each room node N_i, assigning it to the closest cluster mean. The output of clustering is shown in Figure 3.4.

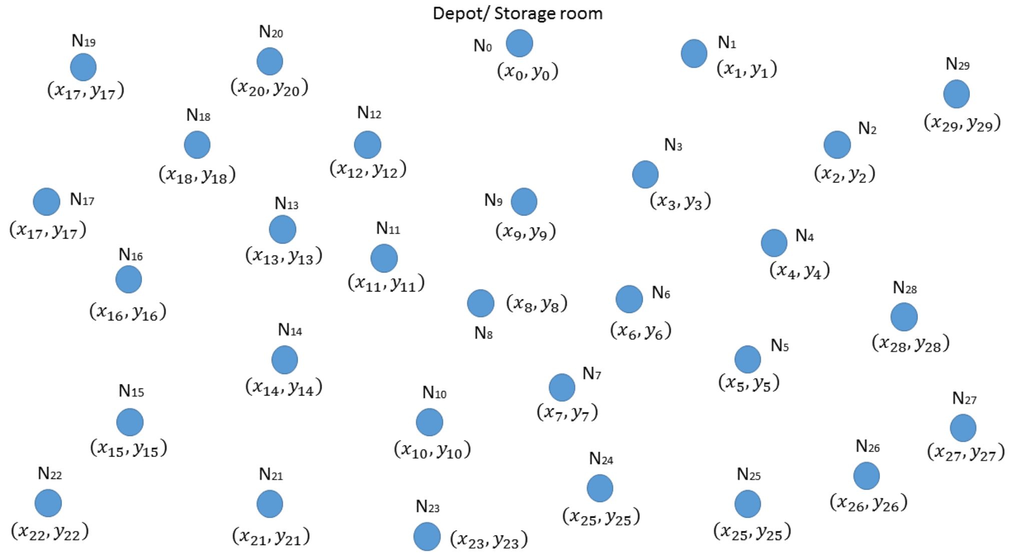


Figure 3.2: Room nodes represented in a 2D space

However, k-means only operates on relative coordinates of room nodes and their distances from the depot. Since the algorithm does not inherently consider the task time at a node, it is possible that the total task time for a cluster (and hence for the custodian working on that cluster) is disproportionately high compared to other custodians. The potential for disproportionate allocation of work to one employee is not tenable from the perspective of an administrator. There is a need for a modified algorithm which explicitly attempts to allocate equal work hours to all custodians.

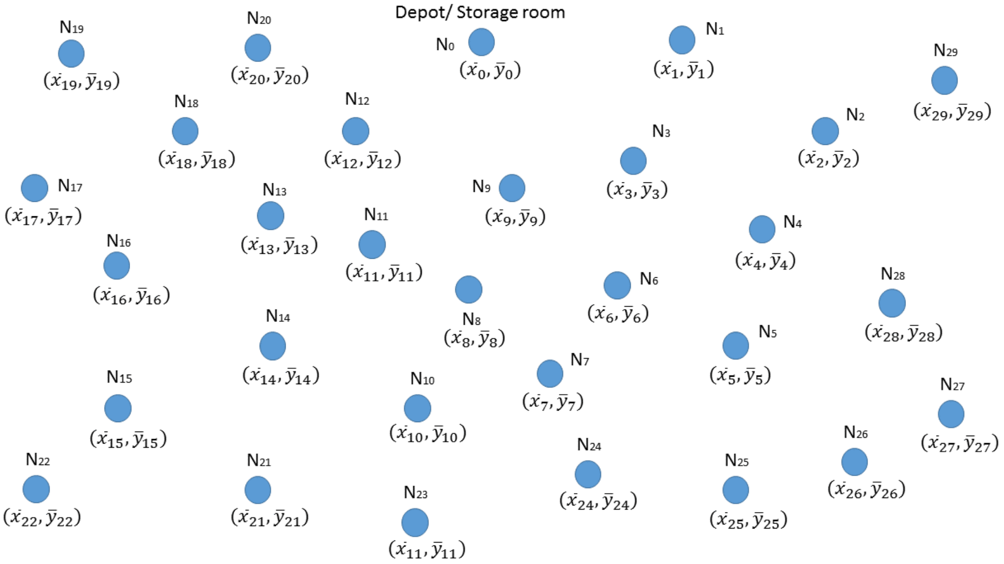


Figure 3.3: Modified room nodes with changed origin represented in a 2D space

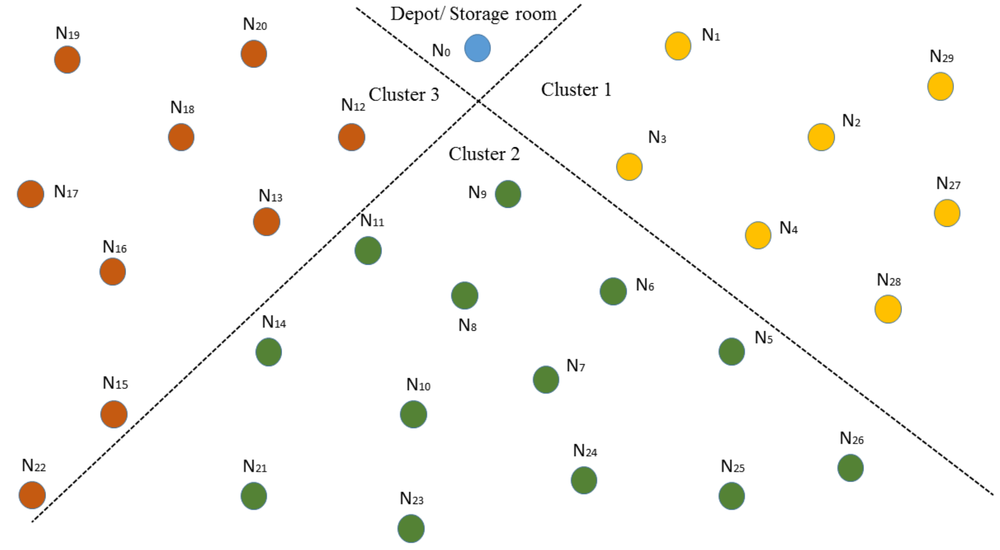


Figure 3.4: Nodes divided into 3 clusters

Modified clustering

Modified clustering has some similarities with k-means: it starts with a known number of clusters and requires an origin shift before room node assignments to clusters are made. However, the task time p_i at a room N_i is also taken into consideration and approximately equal working hours are allocated to all custodians.

As a preliminary step, the average task time per custodian is calculated:

$$p_{avg} = \sum_{i=1}^{n} p_{N_i}/k$$

where k is the number of custodians, n is the number of rooms, and p_{N_i} is the task time at room N_i. The average task time per custodian p_{avg} serves as the upper bound for total task time that may be allocated to each cluster.

Assignment for each cluster starts with the farthest unassigned room node relative to the depot, referred to as the active node. The node nearest to the active node is added to the cluster and becomes the new active node. This process continues until the stopping condition for a cluster, given in equation (3.6), is satisfied. When this happens, the next active node is assigned to a new cluster. Alternately, if nodes are being assigned to the final cluster, then the stopping condition is relaxed and all available nodes are allocated to the final cluster.

$$p_k \leq p_{avg} \tag{3.6}$$

It should be noted that the upper bound on task time cannot be a strict upper bound or hard constraint in practical scenarios. The task time at an individual room is a non-divisible entity, i.e. custodians cannot be assigned a fraction of work in a room. This makes it impossible to guarantee that every custodian will work exactly the same number of hours. However, using the constraint in equation (3.6) reduces the disparity in working time allocation by making task time an integral part of the clustering process.

3.1.1 Optimization Model

The custodian modeling uses the Dantzig et al. (1954) classical VRP formulation with modifications to constraints according to present requirements. Dantzig et al. (1954) formulation is the most cited formulations in the literature (Matai et al., 2010).

Variables

Let, x_{ij} be the decision of whether a custodian travels from room i to room j

$$x_{ij} = \begin{cases} = 1 & \text{if the custodian travels from room } i \text{ to room } j, \\ = 0 & \text{if the custodian does not travel from room } i \text{ to room } j \end{cases}$$

Parameters

n is the total number of rooms of a floor

m is set of custodians

s_{ij} is the traveling time from room i to room j

p_j is the task time in room j

t_{ij} is the traversing time between room i and room j and the time of tasks in room j

$$t_{ij} = s_{i,j} + p_j \tag{3.7}$$

Objective Function

The total task time and the travel time between the rooms has to be minimized.

$$Min \sum_{j=1}^{n} \sum_{i=1}^{n} t_{ij} x_{ij} \tag{3.8}$$

Constraints

$$\sum_{j=2}^{n} x_{1j} = m, \tag{3.9}$$

This constraint ensures that exactly m custodians (specified by the user) depart from the depot or storage room.

$$\sum_{j=2}^{n} x_{j1} = m, \tag{3.10}$$

This constraint ensures that exactly m custodians return to the depot or storage room.

$$\sum_{i=1}^{n} x_{ij} = 1, \tag{3.11}$$

for $j = 2, 3, ...n$. This constraint ensures that a custodian can travel to any other room j from only one room , i.e., the custodian can travel to any room from only one room, he cannot travel from two or more different rooms to one room. In other words there can be only one incoming route for any room.

$$\sum_{j=1}^{n} x_{ij} = 1, \tag{3.12}$$

for $i = 2, 3, ...n$. This constraint ensures that a custodian can travel to only one room from any other room i, i.e., the custodian can travel to only one room from a given room.

$$x_{ij} \in \{0, 1\}, \tag{3.13}$$

This constraint represents the binary nature of the variable x.

Equations (3.11),(3.12) and (3.11) are the assignment constraints. Constraints (3.9) and (3.10) ensure that exactly m number of custodians departing from depot return back to the depot.

$$u_i - u_j + (n - m)x_{ij} \leq n - m - 1, \tag{3.14}$$

for $2 \leq i \neq j \leq n$. Equation (3.14) represent the sub-tour elimination constraints (SECs). The constraint prevents sub-tours, which are degenerate tours formed between intermediate rooms and not connected to the depot or storage room.

3.2 Implementation Phase

A user friendly application has been designed to demonstrate custodian scheduling and routing. Its implementation has been divided into six different parts for simplicity, as shown in Figure 3.5. The interface utilization and users are described in further sections of the chapter.

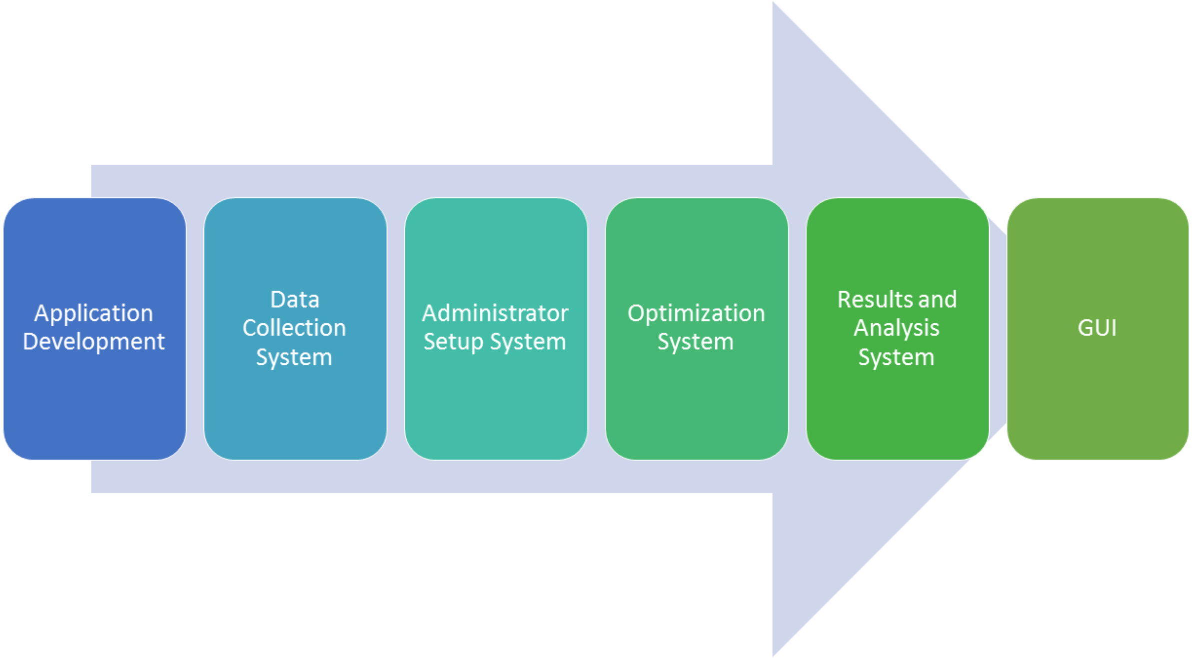


Figure 3.5: Flow of implementation phase

3.2.1 Application Development

An application has been developed using Matlab for the optimization system interface and Visual studio for the data collecting interface. The user interface for data collection system has been programmed in C# and its data is stored in a SQL sever while the optimization system interface has been developed using Matlab.

3.2.2 Data Collection System

The data collection system application requires single time entry to build the database by entering values for all the variables discussed in Section 1.3 that have to be collected. The database application has been designed for ease of understanding and hassle free usage.

Data Collection Process and Choice of Attributes

One of the important tasks in developing a custodial maintenance application was creating a customized database in which the administrator could easily input, store, edit and retrieve data. In a database a ‘query’ is a command processed in the user interface when user enters, retrieves or edits certain information. The type of information to form a database is shown in the following tables. Table (3.1) shows user queries for general information of building and area to be cleaned. Table (3.2) shows queries for square feet information of the area to be cleaned. Table (3.3) shows the queries for fixtures present a room that have to be cleaned. Tables (3.4) and (3.5) show queries pertaining to fixing the unit cost of each task and unit labor time for all the 14 tasks. Tables (3.6) and (3.7) show the queries an administrator can have for total cost of cleaning and total labor time for a building. The cleaning tasks are collected from the ISSA (Frank D, 2010). The cleaning tasks of UT facilities services are studied and related to ISSA standard tasks. These shortlisted tasks are made available for the user for selection in the data collection system interface. The data collection system makes the user the ability to edit and alter data fields related to these cleaning tasks.

Table 3.1: Queries for general information of building areas

Type of Information	**Table name**	**Description**
Building Zone/ Building Number/ Building Name	Main_DivisionName	The name or the number of the building to be maintained.
Area Number/ Room Number	Main_AreaNumber	Area/room in the building that has to be maintained
Floor Number	Main_Floor	The floor number in which the room/area is located
Area Type	Main_Area Type	The type of area/room to be maintained. E.g. classroom, conference room etc.
Priority	Main_Priority	The priority schedule of the room.

Table 3.2: Queries for square feet information of areas

Query/ Type of Information	**Table name**	**Field Name**	**Description**
Carpet area	Areas	Area_FlCarpet	The area (sq.ft.) of carpet flooring in the room/area to be maintained
Wood floor area	Areas	Area_FlWood	The area (sq.ft.) of wooden flooring in the room/area to be maintained
Tile floor area	Areas	Area_FlTile	The area (sq.ft.) of tile flooring in the room/area to be maintained
Other floor area	Areas	Area_Flother	The area (sq.ft.) of any other hard floor in the room/area to be maintained
Wall tile area	Areas	Area_WTile	The area (sq.ft.) of the tile wall in the room/area to be maintained
Wall other area	Areas	Area_WOther	The area (sq.ft.) of the tile wall in the room/area to be maintained
Window area	Areas	Area_Window	The area (sq.ft.) of the windows in the room/area to be maintained
Ceiling area	Areas	Area_Ceiling	The area (sq.ft.) of the ceiling in the room/area to be maintained
Door area	Areas	Area_Door	The area (sq.ft.) of the doors in the room/area to be maintained

Table 3.3: Queries for fixtures in the area/room

Query/ Type of Information	**Table name**	**Field Name**	**Description**
Furniture	Areas	Area_Furniture	Number of furniture fixtures in the room/area
Lighting	Areas	Area_Lighting	Number of lighting fixtures in the room/area
Tables & Desks	Areas	Area_TableDesk	Number of tables/desks in the room/area
Lab Fixtures	Areas	Area_Lab	Number of lab fixtures available in the room/area (E.g. chemical containers)
Chairs	Areas	Area_Chair	Number of chairs available in the room/area
Lawn area	Areas	Area_Lawn	Area (sq.ft) of lawn if present in the area
Sink	Areas	Area_Sink	Number of sinks available in the room/area
Trash	Areas	Area_Trash	Number of trash bins available
Toilet	Areas	Area_Toilet	Number of toilets to be cleaned
Machines	Areas	Area_Machines	Number of machines to be maintained in the room/area
Handrail & Cabinets	Areas	Area_Cabinet	Number of cabinets/handrails

Table 3.4: Queries for fixing unit costs of tasks

Query/Type of Information	Table name	Column name	Description
Unit Vacuum Cost	ArLbt	Cost_Vaccum	Setup unit vacuum cost per square foot area
Unit Sweep Cost	ArLbt	Cost_Sweep	Setup unit sweeping cost per square foot area
Unit Finish Cost	ArLbt	Cost_Finish	Setup unit finishing cost per square foot area
Unit Tmop Cost	ArLbt	Cost_Tmop	Setup Tmop cost per square area
Unit Wipe Cost	ArLbt	Cost_Wipa	Setup wiping cost per unit square foot area
Unit Shampoo Cost	ArLbt	Cost_Shampoo	Setup shampoo cost per unit square foot of carpet
Unit Washing Cost	ArLbt	Cost_ClnWash	Setup washing cost per unit square foot
Unit Mopping Cost	ArLbt	Cost_Mop	Setup mopping cost per square foot
Unit Windex Cost	ArLbt	Cost_Windex	Setup windex cost per unit window
Unit Dusting Cost	ArLbt	Cost_Dust	Setup dusting cost per unit fixture
Unit Mowing (machine) Cost	ArLbt	Cost_MowD	Setup Mowing cost by machine per unit square foot area
Unit Mowing (Manual) Cost	ArLbt	Cost_MowM	Setup manual mowing cost per unit square foot area
Unit Trash Empty Cost	ArLbt	Cost_Trash	Setup trash emptying cost per one trash bin
Unit Wipe Object Cost	ArLbt	Cost_WipeObj	Setup wiping cost per one object
Unit Cleaning Powder Cost	ArLbt	Cost_ClnPow	Setup cleaning powder cost per one square foot area

Table 3.5: Queries for fixing unit labor time of tasks

Query/Type of Information	**Table name**	**Column name**	**Description**
Unit Vacuum Labor Time	ArLbt	Lbt_Vaccum	Setup unit vacuum labor time per square foot area
Unit Sweep Labor Time	ArLbt	Lbt_Sweep	Setup unit sweeping labor time per square foot area
Unit Finish Labor Time	ArLbt	Lbt_Finish	Setup unit finishing labor time per square foot area
Unit Tmop Labor Time	ArLbt	Lbt_Tmop	Setup Tmop labor time per sq. ft.
Unit Wipe Labor Time	ArLbt	Lbt_Wipa	Setup wiping labor time per unit square foot area
Unit Shampoo Labor Time	ArLbt	Lbt_Shampoo	Setup shampoo labor time per unit square foot of carpet
Unit Washing Labor Time	ArLbt	Lbt_ClnWash	Setup washing labor time per sq. ft.
Unit Mopping Labor Time	ArLbt	Lbt_Mop	Setup Mopping labor time per sq. ft.
Unit Windex Labor Time	ArLbt	Lbt_Windex	Setup windex labor time per unit window
Unit Dusting Labor Time	ArLbt	Lbt_Dust	Setup dusting labor time per unit fixture
Unit Mowing (machine) Labor Time	ArLbt	Lbt_MowD	Setup Mowing labor time on machine per unit square foot area
Unit Mowing (Manual) Labor Time	ArLbt	Lbt_MowM	Setup manual mowing labor time per unit square foot area
Unit Trash Empty Labor Time	ArLbt	Lbt_Trash	Setup trash emptying labor time per one trash bin
Unit Wipe Object Labor Time	ArLbt	Lbt_WipeObj	Setup wiping labor time per one object
Unit Cleaning Powder Labor Time	ArLbt	Lbt_ClnPow	Setup cleaning powder applying labor time per sq. ft. area

Table 3.6: Queries for total cost of tasks

Query/Type of Information	**Table name**	**Field name**	**Description**
Total Vacuum Cost	Cost	Cost_Vaccum	Total vacuum cost of selected room/area in dollars
Total Sweep Cost	Cost	Cost_Sweep	Total sweeping cost of selected room/area in dollars
Total Finish Cost	Cost	Cost_Finish	Total finishing cost of selected room/area in dollars
Total Tmop Cost	Cost	Cost_Tmop	Total Tmop cost of room/area in dollars
Total Wipe Cost	Cost	Cost_Wipa	Total wiping cost of selected room/area in dollars
Total Shampoo Cost	Cost	Cost_Shampoo	Total shampoo cost of selected room/area in dollars
Total Washing Cost	Cost	Cost_ClnWash	Total washing cost of selected room/area in dollars
Total Mopping Cost	Cost	Cost_Mop	Total mopping cost in dollars
Total Windex Cost	Cost	Cost_Windex	Total windex cost of selected room/area
Total Dusting Cost	Cost	Cost_Dust	Total dusting cost of selected room/area
Total Mowing (machine) Cost	Cost	Cost_MowD	Total machine mowing cost of selected area
Total Mowing (Manual) Cost	Cost	Cost_MowM	Total manual mowing cost of selected area
Total Trash Empty Cost	Cost	Cost_Trash	Total cost of emptying trash
Total Wipe Object Cost	Cost	Cost_WipeObj	Total cost of wiping objects in selected room/area
Total Cleaning Powder Cost	Cost	Cost_ClnPow	Total cost of cleaning powder used in selected room/area

Table 3.7: Queries for total labor time of tasks

Query/Type of Information	Table name	Field name	Description
Total Vacuum Labor Time	Labor Time	Lbt_Vaccum	Total vacuum labor time of selected room/area in hours
Total Sweep Labor Time	Labor Time	Lbt_Sweep	Total sweeping labor time of selected room/area in hours
Total Finish Labor Time	Labor Time	Lbt_Finish	Total finishing labor time in hours
Total Tmop Labor Time	Labor Time	Lbt_Tmop	Total Tmop labor time in hours
Total Wipe Labor Time	Labor Time	Lbt_Wipa	Total wiping labor time in hours
Total Shampoo Labor Time	Labor Time	Lbt_Shampoo	Total shampoo labor time of selected room/area in hours
Total Washing Labor Time	Labor Time	Lbt_ClnWash	Total washing labor time of selected room/area in hours
Total Mopping Labor Time	Labor Time	Lbt_Mop	Total mopping labor time of selected room/area in hours
Total Windex Labor Time	Labor Time	Lbt_Windex	Total windex labor time of selected room/area in hours
Total Dusting Labor Time	Labor Time	Lbt_Dust	Total dusting labor time of in hours
Total Mowing(machine) Labor Time	Labor Time	Lbt_MowD	Total machine mowing labor time of selected area in hours
Total Mowing (Manual) Labor Time	Labor Time	Lbt_MowM	Total manual mowing labor time of selected area in hours
Total Trash Empty Labor Time	Labor Time	Lbt _Trash	Total labor time of emptying trash selected room/area in hours
Total Wipe Object Labor Time	Labor Time	Lbt_WipeObj	Total labor time of wiping objects in selected room/area in hours
Total Cleaning Powder Labor Time	Labor Time	Lbt_ClnPow	Total labor time of cleaning powder used in selected room/area in hours

System Architecture

System architecture of the data collection system consists of the database, its components and flow of data between the components. Figure (3.6) shows the architecture of the data collection system. The data collecting system was built on C#.NET using Visual Studio interface and SQL Server Management Studio 2008 for building the database. The interface allows access to SQL database connectivity, enabling data to be entered, edited and retrieved. The database was designed to provide authorized access to the administrator. The information entered through the VB form is transferred to SQL database through connection string and retrieved for editing and viewing purposes. A connection string is a command line input which enables secure transactions of data between the interface and database. The systems functionality of the flow of data between the interface and the database server was written in C#. The database server stores information and retrieves it when the VB form user interface sends a requests for information.

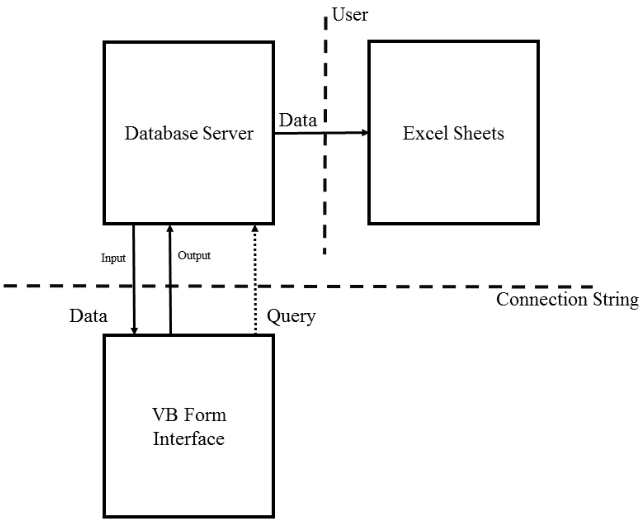


Figure 3.6: Data collection system architecture

Conceptual Design

The database comprises of tables categorized according to queries. The system requirements were gathered using SQL database and an entity relationship (ER) diagram was developed to check the process of data collection. ER diagrams are used to design database, model data and create capability of multiple views (Muppaneni, 2014). For designing the ER model based on significant interactions with UT facilities services and ISSA task lists were taken into account.

Primary keys are allocated to the data in the database for optimizing space allocation. Primary keys are identification keys and are unique to each cleaning area. Figure 3.8 shows room data modeling. A Primary key was assigned to the room number/ area number and the secondary keys to other components of the database for optimizing the space. The identifier Area number is related to other information of the area like the floor area, fixtures etc., as shown in Figure 3.8.

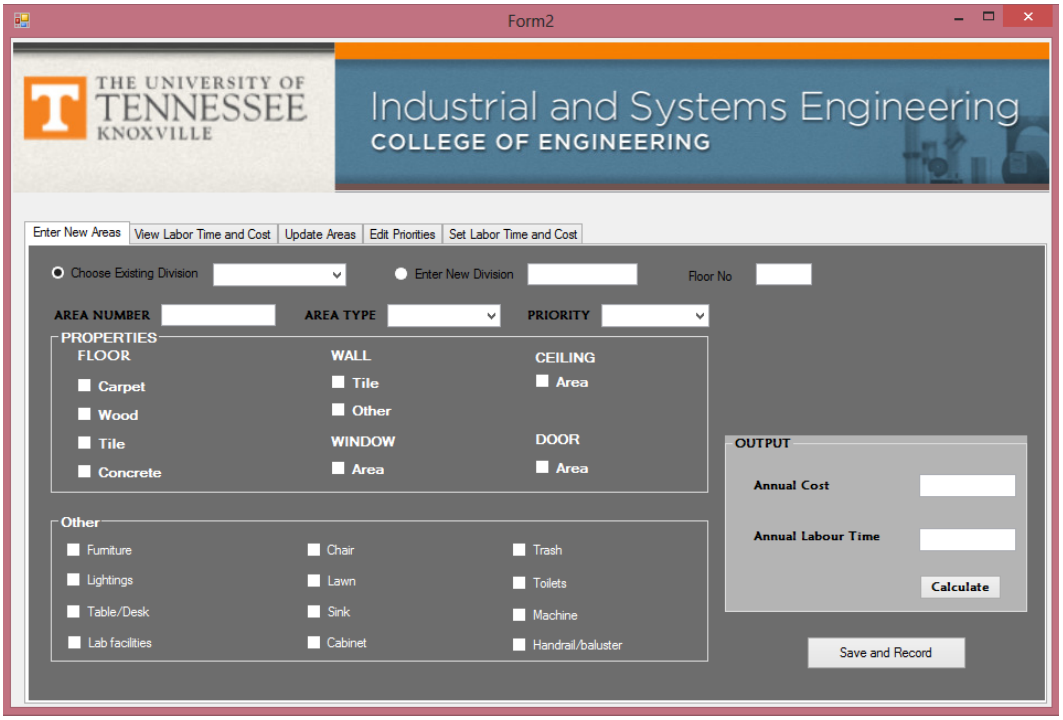


Figure 3.7: A screen capture of the data collecting application

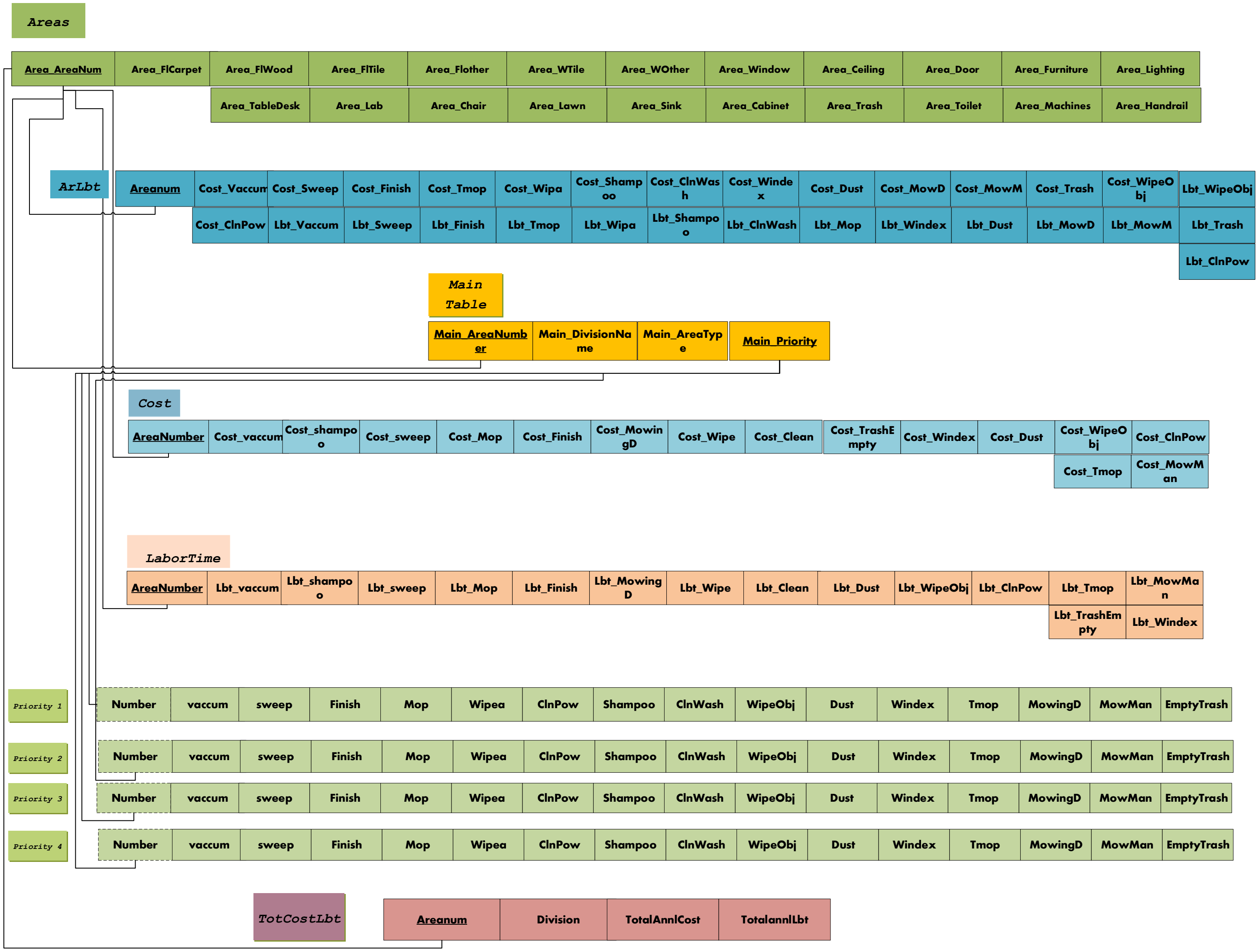


Figure 3.8: Relational model depicting primary keys used to generate data in the database

3.2.3 Administrator Setup System

Floor Plan

The application was built to accept floor plan images of the buildings as input so that room nodes could be labeled. The routing and scheduling of the custodians is shown on the floor plan as output after the algorithm has finished running. The application accepts PNG (Portable Network Graphic) and other standard image formats as input. Each pixel of the image represents 1 square feet of the floor dimension. The rooms and the hallways are marked on the floor plan image to simplify selection of nodes.

Figure 3.9: Example of a floor plan used in the application

Selection of Nodes

There are three kinds of nodes that an administrator essentially has to select in order for the program to run. The first set of nodes are corner nodes, the second set of nodes are room nodes and the third kind of node is the depot node. As the name suggests the room nodes are used to indicate rooms or areas to be cleaned and the corner nodes are those which connect the intersections and the corners of the hallway. The depot node is where the custodians start where the carts are stored. There can be multiple room nodes and corner nodes but only one depot node per floor plan image. Figure 3.10 shows the selection of room nodes and corner nodes. The room nodes are represented in pink color and the corner nodes are in black. The depot node is shown in green.

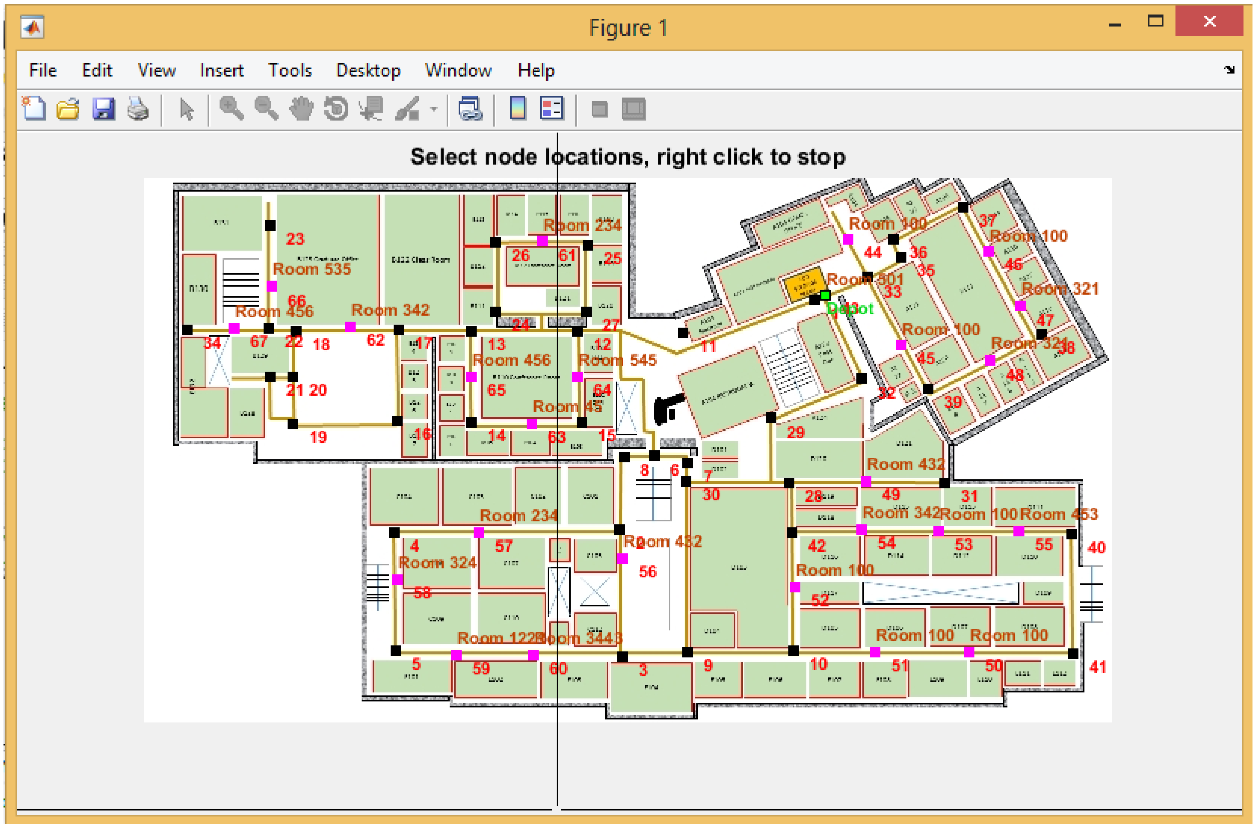


Figure 3.10: A screen capture of selection of room nodes and corner nodes

Priorities

Priorities are the preferences of an administrator related to frequency of maintenance and type of maintenance setup for a particular room or area. The priorities of the UT facilities services have been learned and adopted for the application. The administrator setup system uses the Excel file from the database system (Figure 3.6) created by the administrator. There are 5 types of 'priorities' from which one priority per room has to be selected by the user as shown in the Figure 3.11. Each priority assigns a specific schedule for a day of the week to a room.

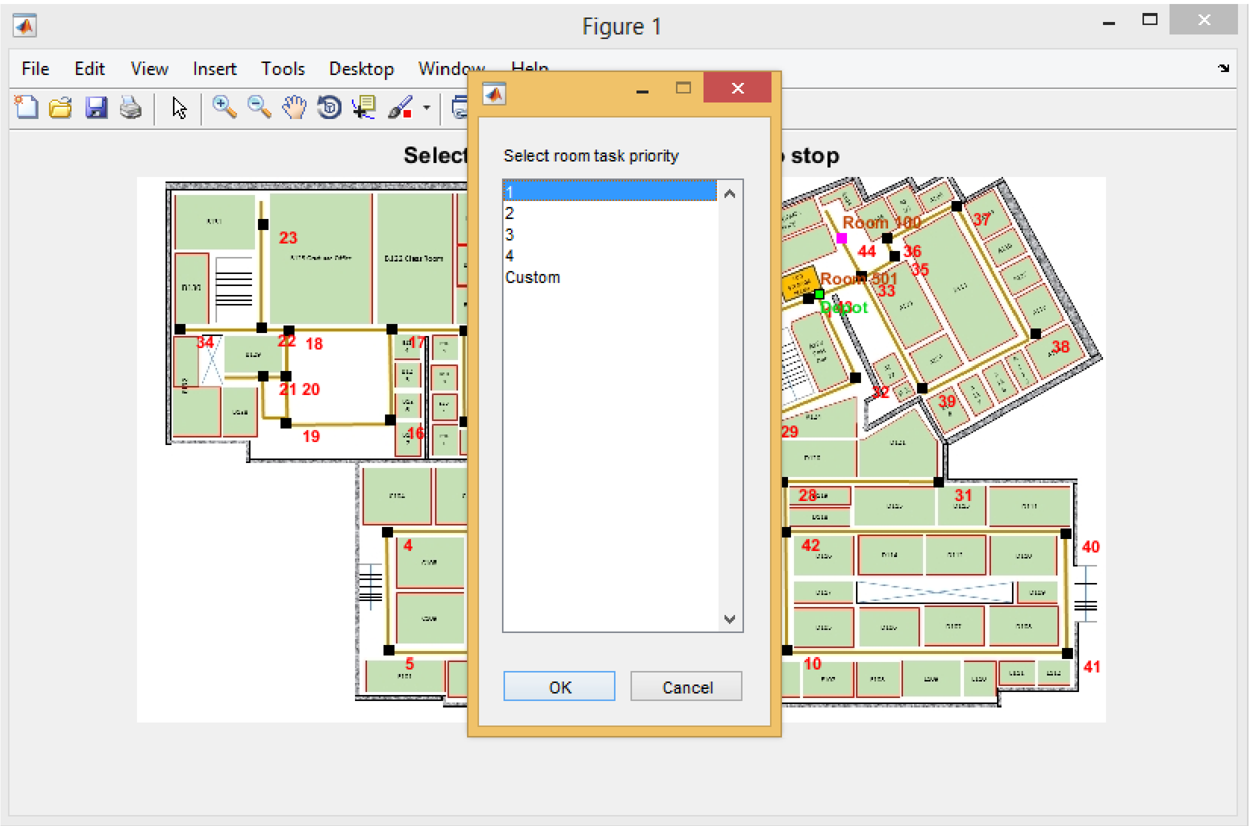
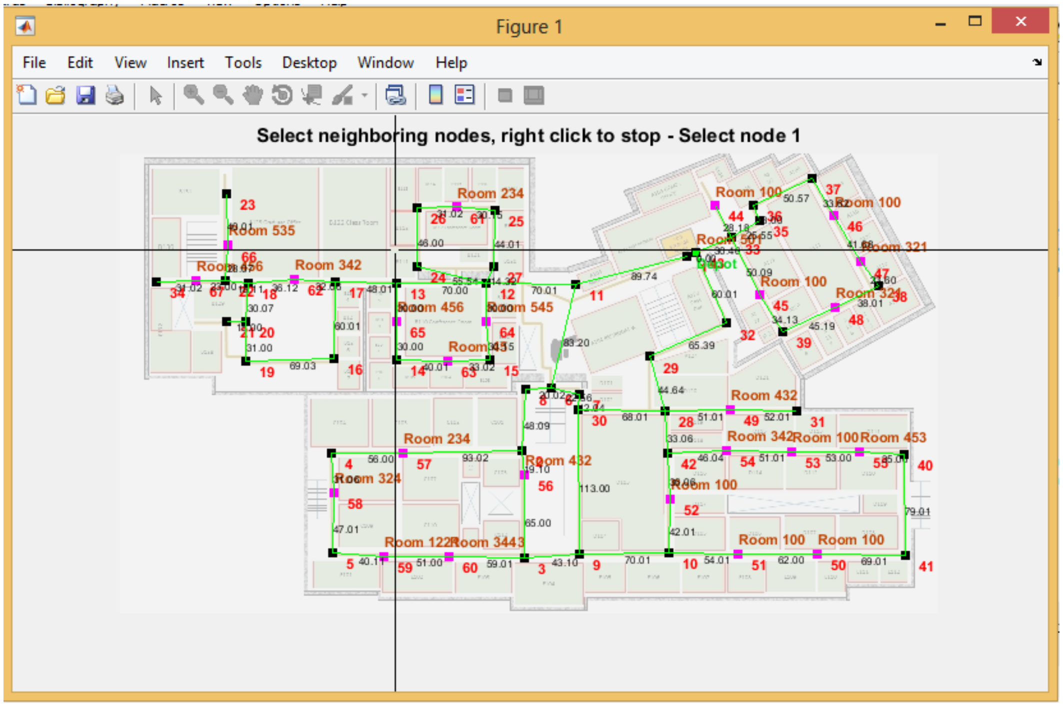


Figure 3.11: A screen capture of selection of priority for room node

Neighboring Nodes

The neighboring node selection is a part of the application where the administrator has to select the neighboring nodes on the hallway path. Neighboring nodes are consecutive nodes which have no nodes in between them. This enables the application to calculate the distances between each node and allows the path to be traced on the hallway itself. Figure 3.12 shows the neighboring nodes in the floor plan image. The green lines on the floor plan image represent the path between two adjacent nodes and the black text shows the distance between any two neighboring nodes.



Select neighboring nodes, right click to stop - Select node 1

Figure 3.12: A screen capture of selection of neighboring nodes

3.2.4 Optimization System

The optimization system has been divided into 3 closely integrated subsystems. The first subsystem finds the shortest path between the room nodes, since this reduces overall maintenance time. The second subsystem is using cluster analysis and assigning rooms to multiple custodians. The third subsystem routes custodians using the algorithm formulation discussed in section 3.1.2.

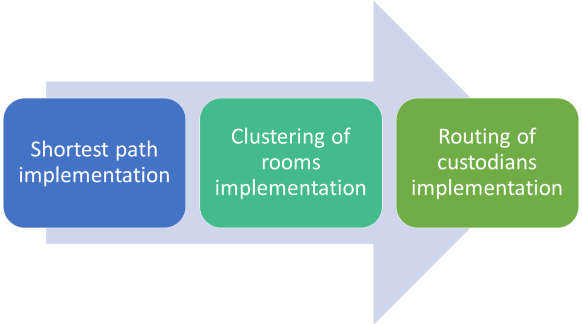


Figure 3.13: The three parts of optimization system

Shortest Path Implementation

After selection of the nodes in the administrator setup system the next step in the application would be calculating the shortest path between the rooms or specific areas of the floor as provided in the floor plan.There may be multiple ways to travel from Room N_i to Room N_j on a given floor plan. To minimize maintenance time and maintenance cost, it is essential that we choose the most economical path or shortest path out of all the possible paths from Room N_i to Room N_j. The current model uses Dijkstra's algorithm to determine the shortest path between the nodes in the floor plan discussed in section 3.1.1. Matlab's implementation of Dijkstra's algorithm is used to calculate the shortest path between each selected room node. The path is saved for the custodian side optimization. Figure 3.14 shows the Matlab interface for shortest path.

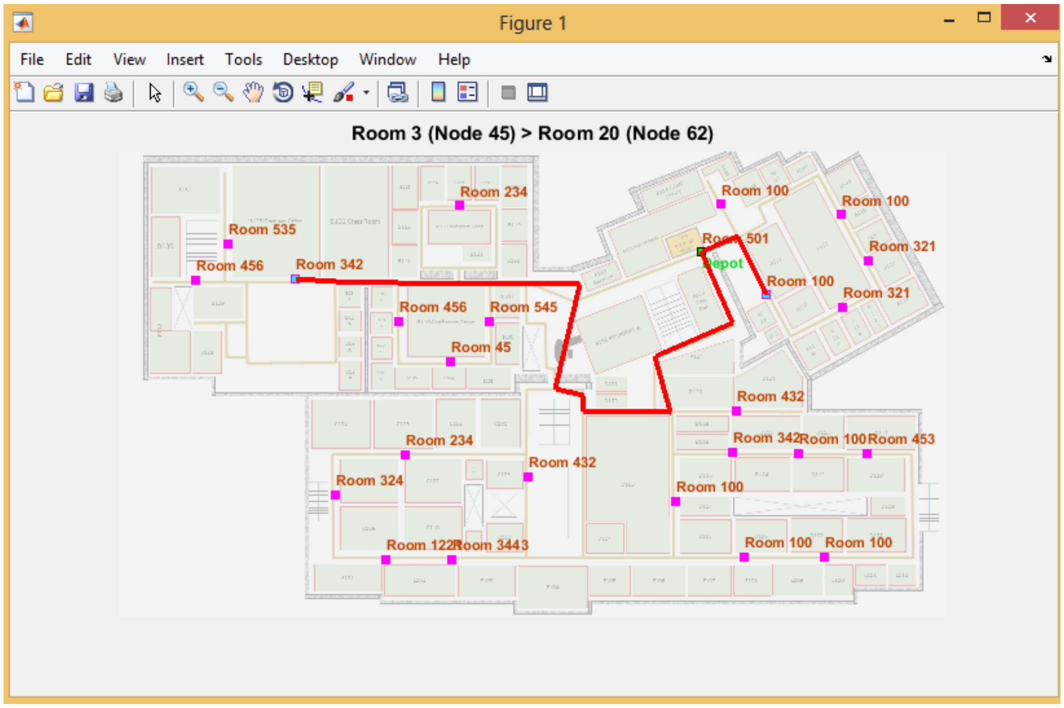


Figure 3.14: A screen capture of shortest path shown in the Matlab interface

Clustering Implementation

The clustering is carried out after the user specifies the number of custodians for a particular floor plan. The clustering has been programmed in Matlab and follows the modified clustering algorithm discussed in section 3.1.1. After the custodian selects the option for the "number of custodians" for the floor on a particular day, the Matlab interface clusters the room nodes into the number as per "the number of custodians" specified by the custodian. Figure (3.15) shows a 2 custodian clustering output represented in two different colors. Then clusters based on the task times of each room or area.

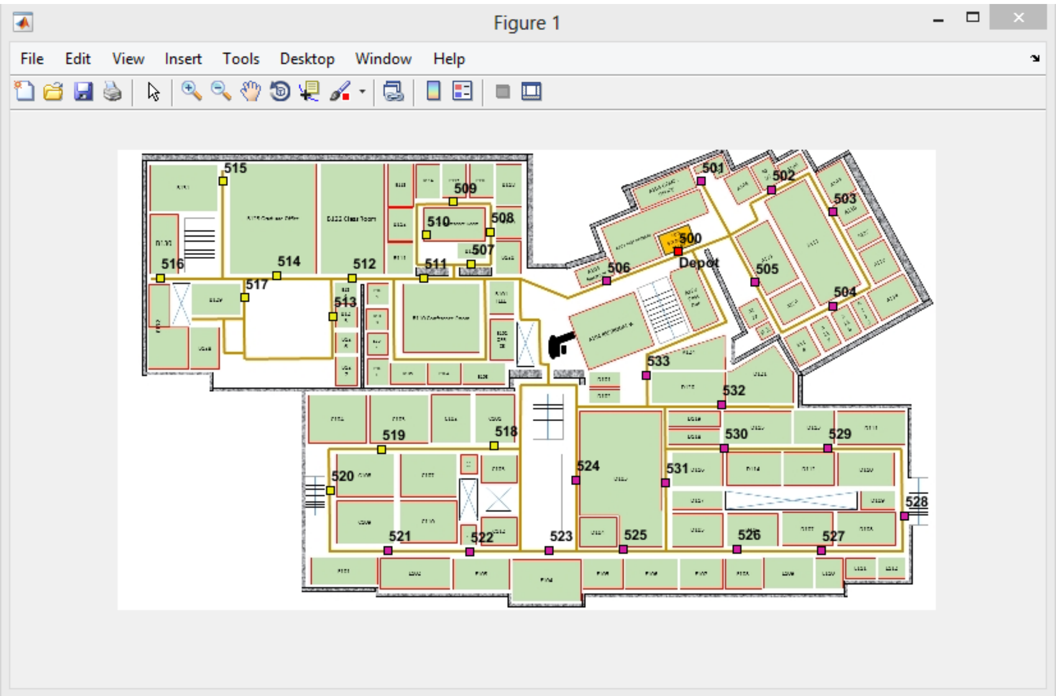
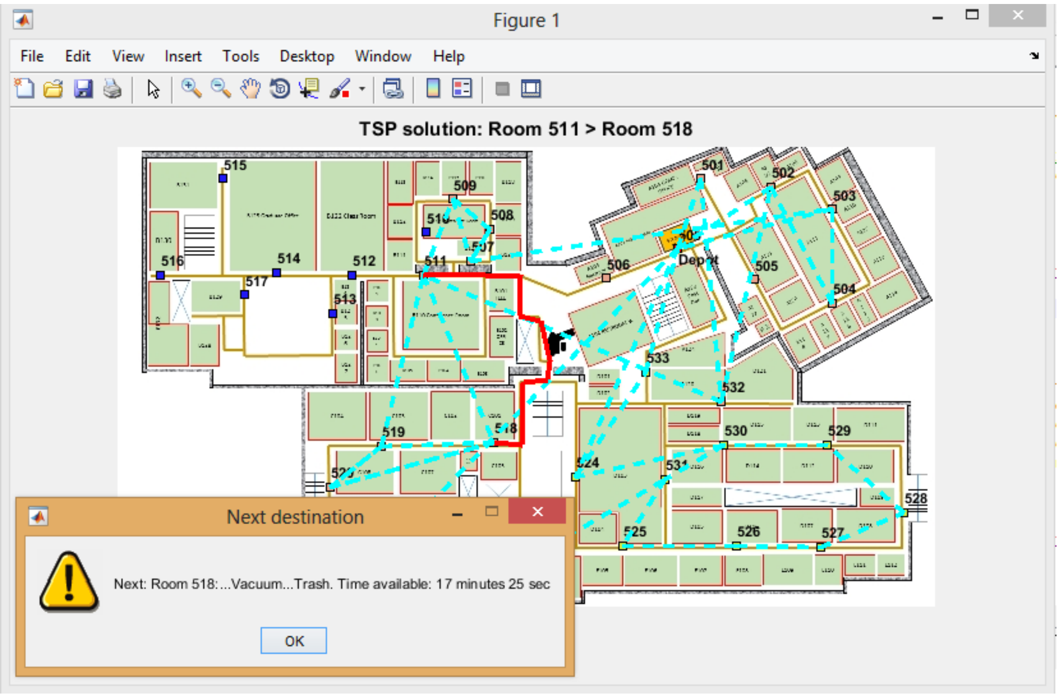


Figure 3.15: Room nodes divided into 2 clusters represented in different colors in the application

Routing Implementation

After clustering, the route for the custodian for that day is shown on the Matlab's interface. The custodian selects the date and the number of custodians working on that floor. This triggers the clustering explained previously and is followed by the routing of the custodian. The custodians can see the route for the day and all the other custodian routes. The user interface shows the custodian the rooms to be visited that day and also the tasks to be carried out in those particular rooms. The routing results are saved to Excel file and can be later accessed by the administrator. Figure 3.16 shows a screen capture of the optimization routing shown on the Matlab interface.



TSP solution: Room 511 > Room 518

Next: Room 518:...Vacuum...Trash. Time available: 17 minutes 25 sec

Figure 3.16: A screen shot of optimization application showing route for one of the 3 custodians

3.2.5 Results and Analysis System

The results and analysis system is designed for the administrator to monitor various facility costs: daily, monthly and yearly labor costs, labor time, production rates, resource costs, and total costs. This system provides feedback necessary for adjusting the priorities and other variables in the data collection system and the administrator setup system. The administrator can plan the maintenance budget with the assistance of the results and analysis monitoring system.

Personnel cost monitoring

The personnel cost monitoring system allows the administrator to observe the costs accrued in labor and labor overhead, over a calendar year. The overhead costs are generally 25% of the total hourly labor cost. Figure 3.17 is one of the outputs of the applications which shows the sample personnel cost for a selected year with labor and overhead cost components.

Personnel cost modeling

Personnel cost or labor cost is a sum of total hourly cost of labor and the burden cost or overhead cost for the working hours.

Let,

C_{tl} be the total labor cost of cleaning a facility in \$

H_w be the total working time of a custodian in hours

$H_w = Min \sum_{j=1}^{n} \sum_{i=1}^{n} t_{ij} x_{ij}$ from equation (3.8)

C_h be the labor cost per hour in \$

$$C_{tl} = H_w \cdot C_h + 0.25 \cdot (H_w \cdot C_h)$$

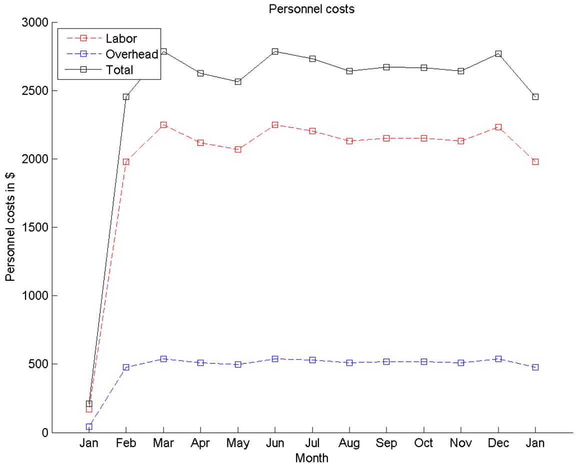


Figure 3.17: Sample plot of labor costs for 1 calendar year

Resource cost monitoring

Resource cost is the total material cost for cleaning a facility. After the rooms to be cleaned have been determined from the administration setup system, the resources used for tasks in that room are estimated based on the square footage area. Resource cost monitoring allows the administrator to analyze the utilization of material and also the tasks in which most of the cost is being accrued. The administrator can alter the priority setting in the administrator setup to minimize the total maintenance cost and stay within budget limits.

Resource cost modeling

The material cost or resource cost modeling is based on day to day routing obtained from the optimization model. The material cost or resource cost is calculated from the tasks obtained from the best possible routing solved by the optimization system. The individual task resource costs are calculated as below.

Let,

$C_{totalvacuum}$ be the total cost of vacuuming a facility in \$

$C_{totalwindow}$ be the total cost of cleaning windows in \$

$C_{totaltrash}$ be the total cost of emptying trash in the facility in \$

$C_{floorclean}$ be the total cost of cleaning the floor in \$

A_{vacuum} be total vacuuming area in sq. ft.

A_{window} be the total area of windows to be cleaned in sq. ft.

A_{trah} be the total number of trash bins to be emptied

$A_{floorclean}$ be the total area to be cleaned in sq.ft.

$C_{uvacuum}$ be the unit vacuuming cost of vacuuming 1 sq. ft. in \$

$C_{uwindow}$ be the cost of cleaning 1 sq.ft. of window in \$

$C_{utrassh}$ be the cost of emptying one trash bin in \$

C_{ufloor} be the cost of cleaning 1 sq.ft. of floor in \$

The unit task costs are obtained from the SQL data server and the total areas of vacuum, window, floor cleaning and number of trash bins are setup by the administrator in the administrator setup system. Figure 3.18 and 3.19 show the sample daily resource costs over a week and monthly resource costs over a selected year respectively. The task specific resource costs or material costs are calculated as below.

$$C_{totalvacuum} = A_{vacuum} \cdot C_{uvacuum}$$
$$C_{totalwindow} = A_{window} \cdot C_{uwindow}$$
$$C_{totaltrash} = A_{trash} \cdot C_{utrash}$$
$$C_{floorclean} = A_{floorclean} \cdot C_{ufloor}$$

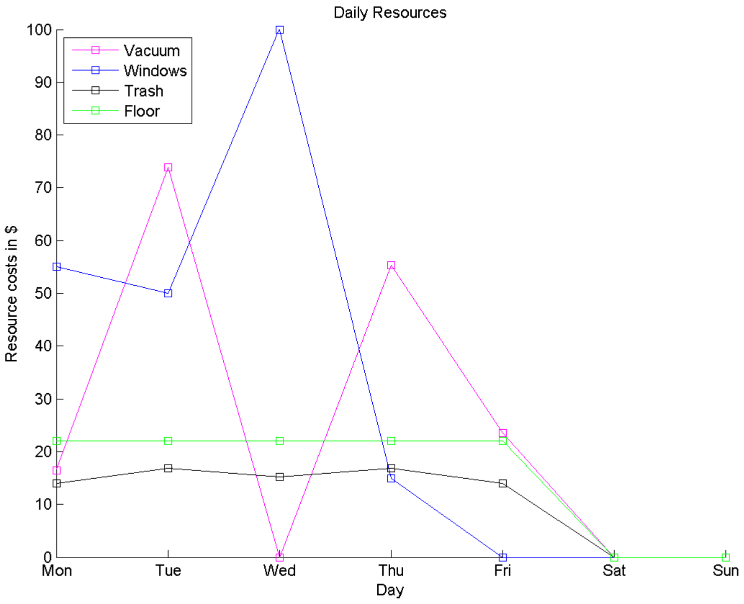


Figure 3.18: Sample plot of task specific resource costs (daily)

The total resource costs are a summation all the individual task resource costs. Figure 3.20 shows a sample of resource costs plot shown in the application over 1 year period. The total resource costs C_r are calculated as below.

$$C_r = C_{totalvacuum} + C_{totalwindow} + C_{totaltrash} + C_{floorclean}$$

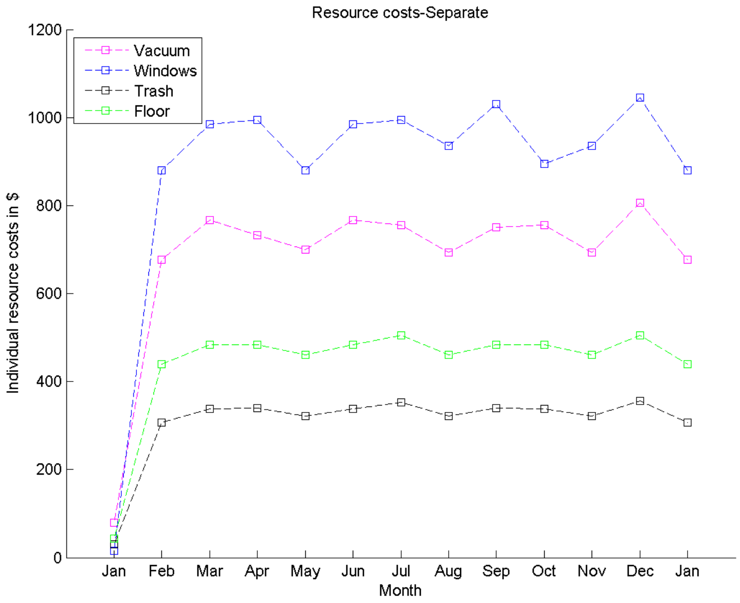


Figure 3.19: Sample plot of task specific resource costs

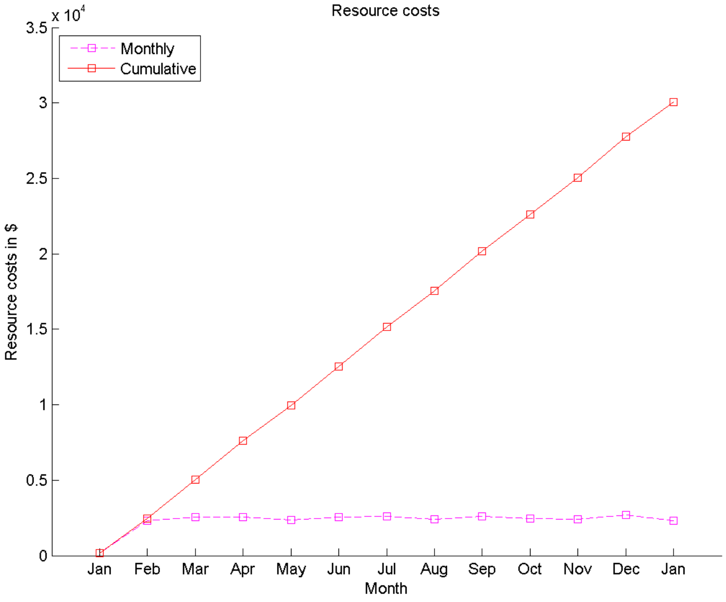


Figure 3.20: Sample plot of total resource costs

Total monthly maintenance costs

The results and analysis system provides a monitoring of the total monthly costs for cleaning a facility. The administrator can select the year in which the total maintenance costs have to be monitored and alter the setup configuration in order to fit the budget.

Monthly maintenance cost modeling

Total monthly maintenance cost is the summation of monthly labor cost and monthly material or resource cost. Figure 3.21 shows a sample of monthly maintenance cost over a selected year by the administrator. The monthly maintenance cost modeling uses the following equation.

$$C_{maintenance/month} = C_{tl} + C_r$$

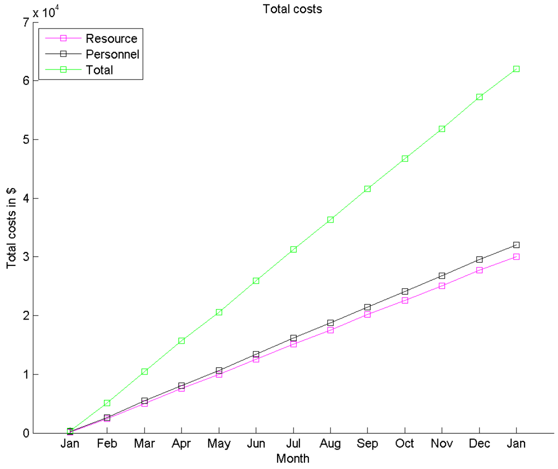


Figure 3.21: Sample plot of total maintenance costs

Cost share monitoring

The total monthly maintenance costs over a month are split into labor cost and material resource cost. Their percentage distribution is shown in the Figure 3.22. This pie chart provides the information to the administrator on the share of costs.

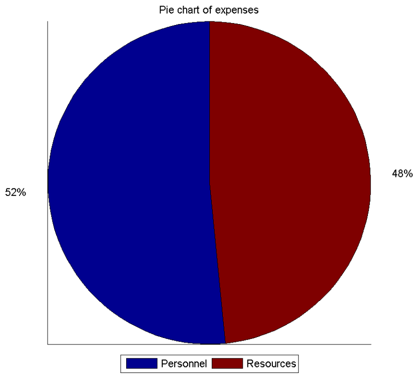


Figure 3.22: Sample pie chart of cost share

Production rates monitoring

Production rates are essential for an administrator to assess the amount of work taking place on a daily basis. The production rate of any task is the total task time over the total square feet area of the task. This monitoring system delivers information regarding the task time and production rates for tasks over a week. Figure 3.23 shows the sample plot of production rates for a week as shown in the application.

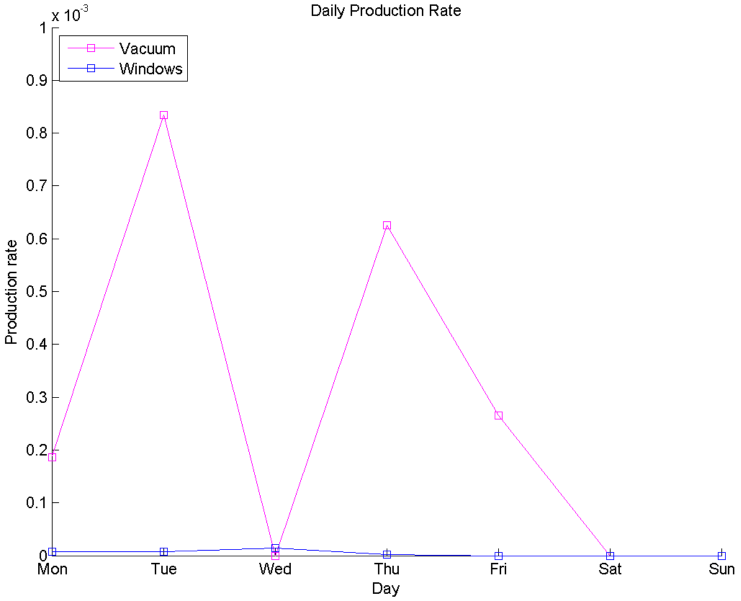


Figure 3.23: Sample plot of daily production rates

Budget monitoring system

The budget monitoring system allows the administrator to view budgets for the optimized routing for multiple custodians. Budget monitor system is thus designed to enable the administrator set budget limits. It also helps review by how much the budget is over or under the cost that the setup configuration acquires over a selected year. The Matlab interface allows the administrator to enter budget and then determine if the total costs accrued with the priority setting is under or over the budget.

3.3 GUI

The application GUI consists of the administrator user interface and the custodian user interface. Figure 3.24 shows the data flows between the 4 systems and the users. The numbers on each of the flow lines are explained below.

1. The administrator or facilities manager enters data for building variables (Figure 1.3) into the data collecting system through VB interface.

2. From the VB interface the variables pass through secure connection string into the SQL database and the data entered by the administrator at the VB interface end is stored in SQL server. The access to the server is secured by a password and can be only accessed by the administrator.

3. For reviewing or editing data to edit, the administrator triggers a query from the VB interface to the SQL server. The SQL server system sends the necessary data back to the VB interface according to the query provided by the user. All the possible queries of an administrator are listed from Table 3.1 to Table 3.7.

4. The VB interface allows the administrator to view and edit the requested data sent by the server.

5. To complete setup of priorities, the administrator copies the priorities from the SQL database tables into an Excel sheet. The Excel sheet is saved in the folder of Matlab

interface # 1. Matlab uses the excel sheet #1 to import the priorities set by the administrator.

6. The administrator selects the floor plan image file, selects floor geometry (nodes), selects room node, selects weekly schedule, enters the room type, identifies the room numbers on the floor plan image, and selects the building floor number through the Matlab interface in the administrator setup system. This completes the setup.
7. The Excel sheet #2 configuration of the floor plan image is sent into the optimization system. The optimization system solves the TSP for each day of the selected year and processes the routing from the configuration sent into it. The shortest paths are calculated between all the nodes and the setup configuration is written automatically from the Matlab interface # 1 to Excel sheet # 2. The Excel sheet #2 can be viewed and modified by the administrator if any room configuration is changed in the future.
8. The results and cost calculations are written to Excel sheet #3 for editing by the administrator.The optimized routing information along with the tasks is sent into the Matlab interface #3 in the results and analysis system. The results and analysis system calculates the total labor cost, resource costs and production rates and plots the relevant graphs for each day of the year.
9. The budget, labor costs, resource costs and production rates are monitored by the administrator on a daily, weekly, monthly and yearly basis as selected through the Matlab interface #3.
10. The costs calculations and room information can be accessed by the administrator from excel sheet #3.
11. The optimized routing is sent to custodian interface after the routing has been calculated for all the days of a selected year.
12. The custodian selects the building number, the floor number where the custodian has to clean and also the day of the year of cleaning through the custodian Matlab interface.

13. The custodian Matlab interface shows the route for the day the custodian has selected and also the tasks to be done in each of the rooms in order.

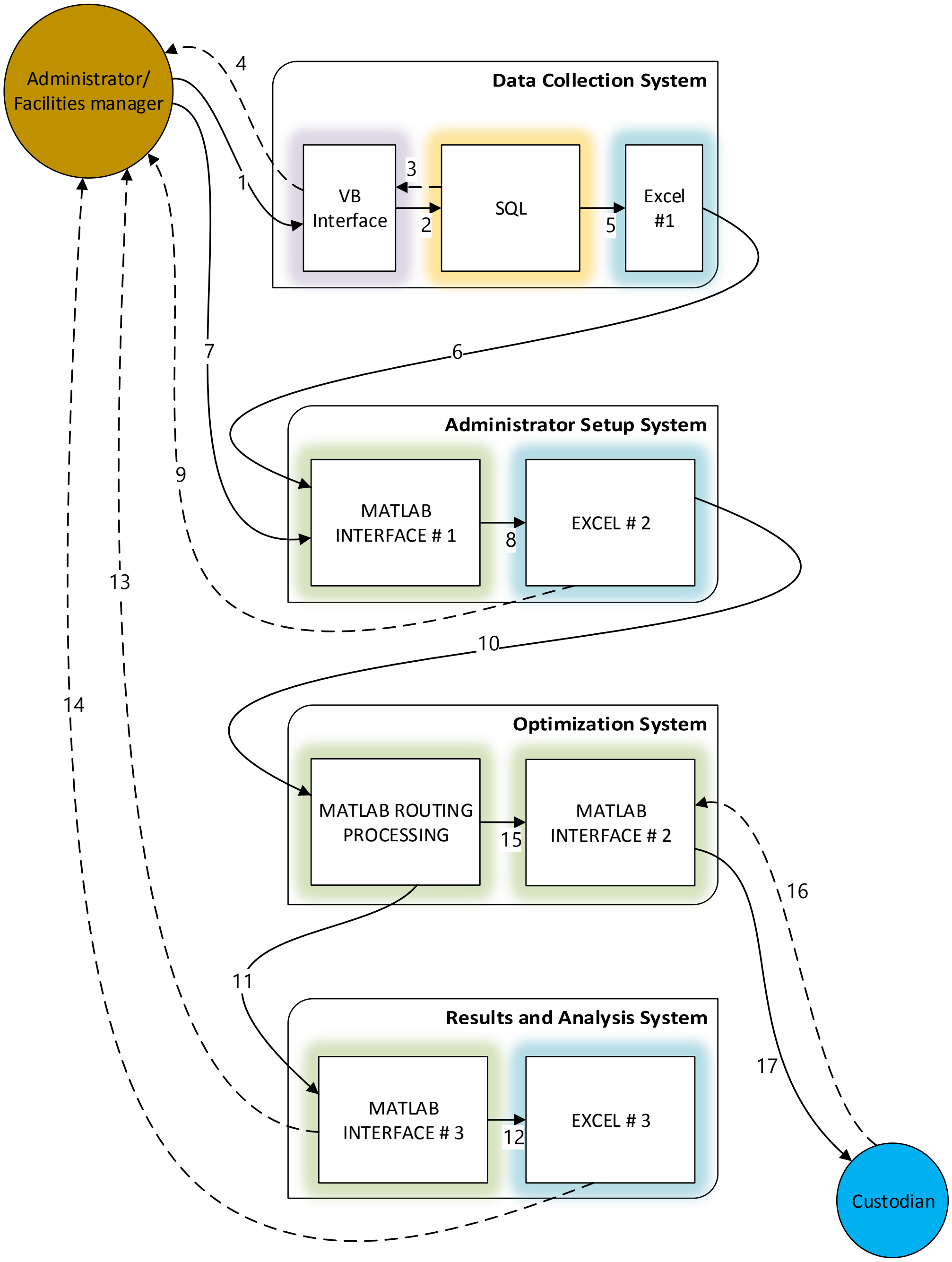


Figure 3.24: Software development for custodian scheduling

Chapter 4

Case Study

This chapter presents the case study demonstrating both the software and the models mechanism. This case study was developed to test and evaluate the application and present its results. The results of this case study form a basis for validating the objectives of this thesis. The case study allows a comparison of the new and old systems and the optimization model. It is divided in to four parts: characteristics of the building, database setup, evaluation of modeling, and the results and analysis.

4.1 Characteristics of building

A replica floor-plan image of the Min H. Kao Building on the University of Tennessee, Knoxville's campus was used for the case study. This building's area is 150,000 square feet. Consisting of offices, class rooms, laboratories and conference rooms, it is an educational and research facility that has been in full operation since 2007. The building is being used by the Department of Electrical Engineering and Computer Science and Center for Ultra-Wide-Area Resilient Electric Energy Transmission Networks (CURENT). The case study has been conducted on the replica floor plan image for the week the January 19-through January 23, 2015. The University of Tennessee facilities services is responsible for the maintenance of this building with three custodians.

Criteria for floor plan selection

The design of the floor plan image was based on the following criteria:

- The floor plan should consist of rooms on either sides of the hallway.
- The floor plan should contain rooms distinctly visible and clearly separated from each other.
- The hallways should be distinctly marked on the floor plan, enabling the user to mark the room nodes.

Figure 4.1 shows the floor plan used in the case study. This floor plan was modified because the current software can work on only one floor plan and while the maximum number of rooms on a floor needed to be accommodated. Thirty-six rooms have been labeled, including the room nodes and corner nodes, totaling 82 nodes. The definition and selection procedures for the room nodes and corner nodes were included in the previous chapter. The administrator enters fields for the room numbers and the room types while selecting the nodes, including the depot node.

4.2 Database setup

The data fields related to the room's entities were entered into the database system. The procedure for entering the fields was discussed in the methodology chapter. The following are the fields related to the rooms:

- Room number (Variable $V1$).
- Room type (Variable $V1$).
- Priority of the room (Variable $V2$).
- Number of trash bins in the room (Variable $V7$).
- Area of window in the room in square feet (Variable $V6$).

- Total area of floor of the room in square feet (Variable $V7$).

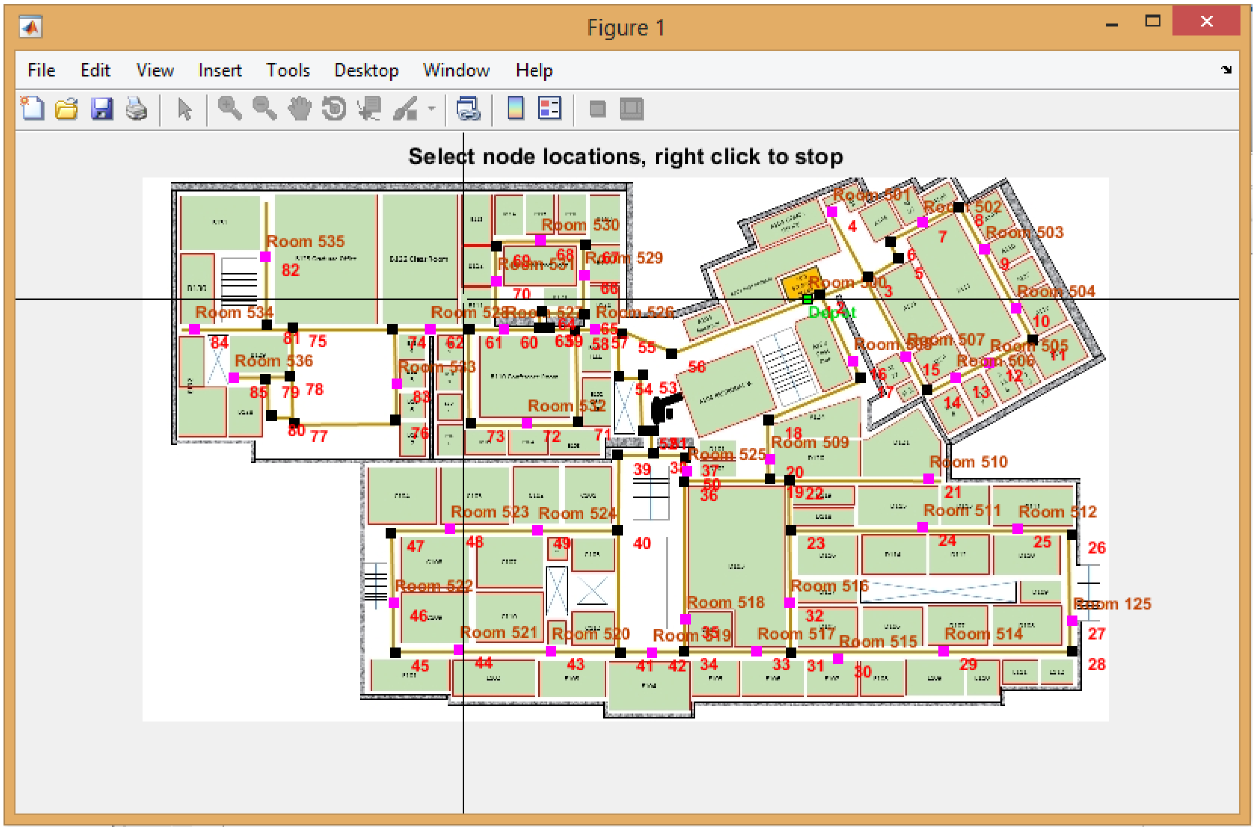


Figure 4.1: Floor plan used in the application for case study representing room nodes (in pink), non room nodes (in black) and depot nodes (green) along room number and node numbers labeled

Figure 4.2 shows the following for each room: room number, room type, number of trash bins, area of window, and area of the floor. The room's entities, mentioned earlier in the methodology are entered into the database system for each room.

4.2.1 Priority setup

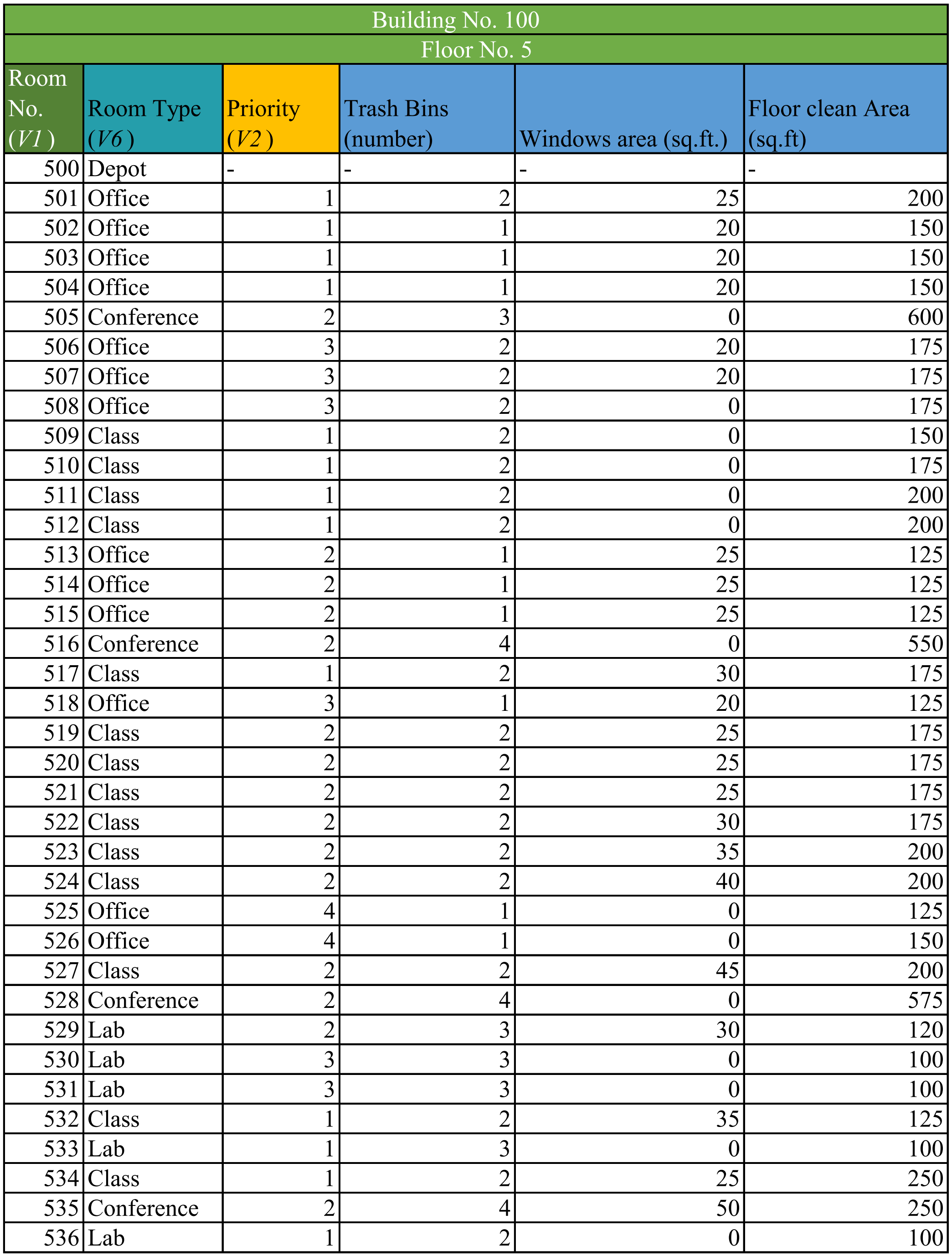
After the nodes were labeled, the priority setup (i.e., the frequency of tasks to be done over a week) was established. The University of Tennessee's Facility Services' maintenance schedule and frequency of maintenance were adopted to formulate the priority setup. A weekly schedule was developed with four priorities and four tasks. The priorities and setups presented in the following sections have been assumed for the remaining weeks, and the results have been calculated for the remainder of the year. The four tasks are

- Trash empty (T).
- Vacuum (V).
- Window clean (W).
- Floor clean (F).

Figure 4.3 shows the priority formulation for different days of the week. The matrices [T,V,W,F] below each day of the week correspond to the task status for the tasks described earlier. Number 1 represents a task that should be done, and 0 represents a task that should not be done for a particular room that day. For example, a matrix [T,V,W,F] = [1,1,1,1] indicates that all four tasks must be performed on that day for that room. The case study follows the same weekly schedule for the entire year.

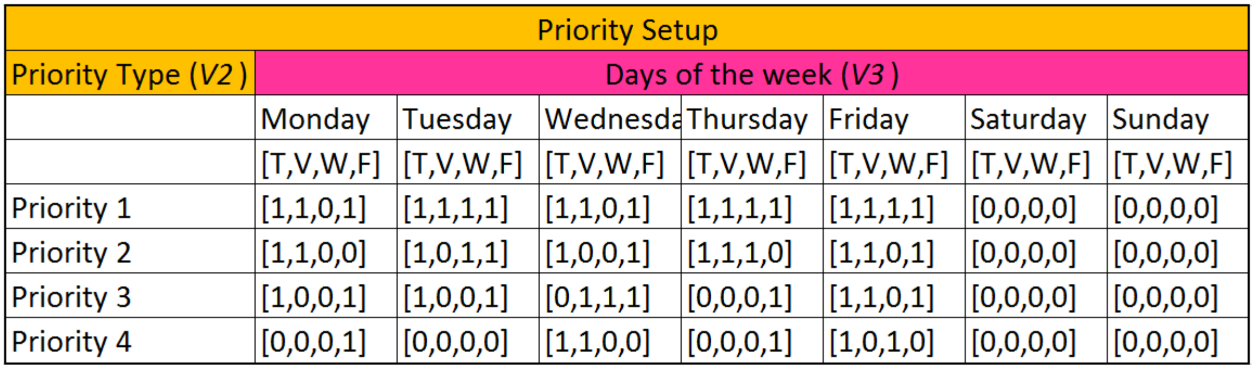
Assignment of priorities to rooms

After the priorities were established through the administrator system, they were assigned to every room. The University of Tennessee's Facilities Services' priorities were adopted for both public and non-public rooms. As mentioned earlier, public rooms are routinely used, and non-public rooms are less routinely used. Based on the rooms identified in the Min H. Kao Building, the priorities were allocated. Table 4.1 shows the priorities allocated to public and non-public rooms. All the public rooms frequently used were allocated Priorities 1 and 2 depending on the room type, and all the non-public rooms were designated Priorities 3 and 4. Four types of rooms have been identified in an educational entity: office, class, conference room, and lab. The offices that are not used by the public and that are used by only 1 or 2 occupants were given priority 4.



Building No. 100					
Floor No. 5					
Room No. (*V1*)	Room Type (*V6*)	Priority (*V2*)	Trash Bins (number)	Windows area (sq.ft.)	Floor clean Area (sq.ft)
500	Depot	-	-	-	-
501	Office	1	2	25	200
502	Office	1	1	20	150
503	Office	1	1	20	150
504	Office	1	1	20	150
505	Conference	2	3	0	600
506	Office	3	2	20	175
507	Office	3	2	20	175
508	Office	3	2	0	175
509	Class	1	2	0	150
510	Class	1	2	0	175
511	Class	1	2	0	200
512	Class	1	2	0	200
513	Office	2	1	25	125
514	Office	2	1	25	125
515	Office	2	1	25	125
516	Conference	2	4	0	550
517	Class	1	2	30	175
518	Office	3	1	20	125
519	Class	2	2	25	175
520	Class	2	2	25	175
521	Class	2	2	25	175
522	Class	2	2	30	175
523	Class	2	2	35	200
524	Class	2	2	40	200
525	Office	4	1	0	125
526	Office	4	1	0	150
527	Class	2	2	45	200
528	Conference	2	4	0	575
529	Lab	2	3	30	120
530	Lab	3	3	0	100
531	Lab	3	3	0	100
532	Class	1	2	35	125
533	Lab	1	3	0	100
534	Class	1	2	25	250
535	Conference	2	4	50	250
536	Lab	1	2	0	100

Figure 4.2: Screen shot of room types, priorities and attributes setup



Priority Setup							
Priority Type (*V2*)	Days of the week (*V3*)						
	Monday	Tuesday	Wednesda	Thursday	Friday	Saturday	Sunday
	[T,V,W,F]	[T,V,W,F]	[T,V,W,F]	[T,V,W,F]	[T,V,W,F]	[T,V,W,F]	[T,V,W,F]
Priority 1	[1,1,0,1]	[1,1,1,1]	[1,1,0,1]	[1,1,1,1]	[1,1,1,1]	[0,0,0,0]	[0,0,0,0]
Priority 2	[1,1,0,0]	[1,0,1,1]	[1,0,0,1]	[1,1,1,0]	[1,1,0,1]	[0,0,0,0]	[0,0,0,0]
Priority 3	[1,0,0,1]	[1,0,0,1]	[0,1,1,1]	[0,0,0,1]	[1,1,0,1]	[0,0,0,0]	[0,0,0,0]
Priority 4	[0,0,0,1]	[0,0,0,0]	[1,1,0,0]	[0,0,0,1]	[1,0,1,0]	[0,0,0,0]	[0,0,0,0]

Figure 4.3: Screen shot of priority setup

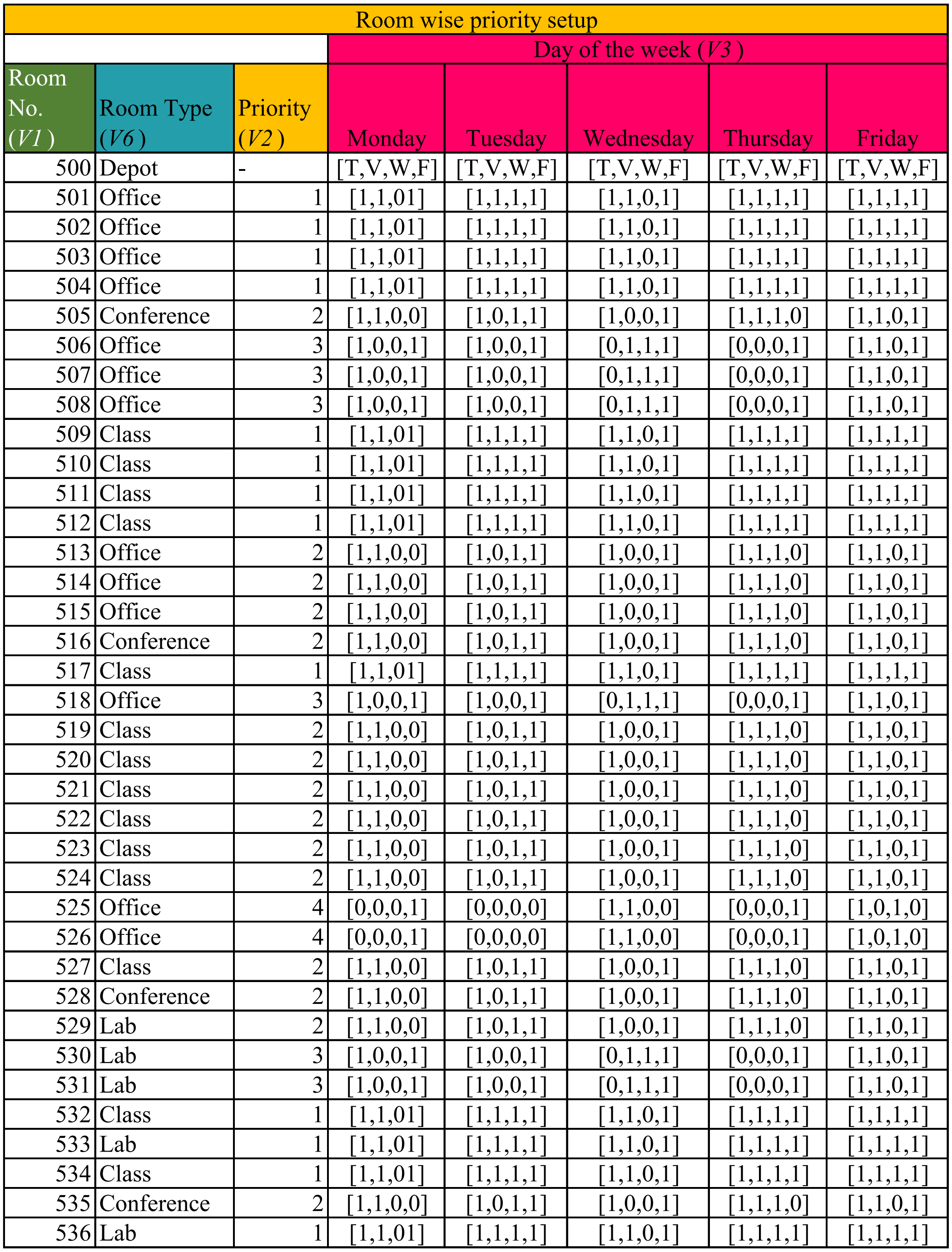
The classrooms are the most frequented rooms in the institution; therefore, they were assigned Priorities 1 and 2. The conference rooms are less frequently used and by fewer people in a day and may not be used during a few days of the week; hence, they were allocated Priority 2. The labs are allocated Priorities 1 to 3, depending on their accessibility and frequency of use. Figure 4.4 shows the room types, priority numbers, and the priorities allocated for the week of January 19, 2015. The same priorities are assumed for the remaining weeks until January 19, 2016, for routing the custodians.

Table 4.1: Priority setup for public and non-public rooms

Room Type	Priority
Public	Priority 1
	Priority 2
Non-public	Priority 3
	Priority 4

Task time setting

The task times were taken from ISSA standard times (Frank D, 2010) for emptying trash, vacuum, floor cleaning and window cleaning. Table 4.2 shows the assumed speed of tasks. For the case study, a custodian's walking speed between the rooms is set constant at 1 foot per second. The task speeds are kept constant regardless of a custodian's skill level. The trash emptying speed is set at 30 seconds per bin.



Room wise priority setup							
			Day of the week (*V3*)				
Room No. (*V1*)	Room Type (*V6*)	Priority (*V2*)	Monday	Tuesday	Wednesday	Thursday	Friday
500	Depot	-	[T,V,W,F]	[T,V,W,F]	[T,V,W,F]	[T,V,W,F]	[T,V,W,F]
501	Office	1	[1,1,01]	[1,1,1,1]	[1,1,0,1]	[1,1,1,1]	[1,1,1,1]
502	Office	1	[1,1,01]	[1,1,1,1]	[1,1,0,1]	[1,1,1,1]	[1,1,1,1]
503	Office	1	[1,1,01]	[1,1,1,1]	[1,1,0,1]	[1,1,1,1]	[1,1,1,1]
504	Office	1	[1,1,01]	[1,1,1,1]	[1,1,0,1]	[1,1,1,1]	[1,1,1,1]
505	Conference	2	[1,1,0,0]	[1,0,1,1]	[1,0,0,1]	[1,1,1,0]	[1,1,0,1]
506	Office	3	[1,0,0,1]	[1,0,0,1]	[0,1,1,1]	[0,0,0,1]	[1,1,0,1]
507	Office	3	[1,0,0,1]	[1,0,0,1]	[0,1,1,1]	[0,0,0,1]	[1,1,0,1]
508	Office	3	[1,0,0,1]	[1,0,0,1]	[0,1,1,1]	[0,0,0,1]	[1,1,0,1]
509	Class	1	[1,1,01]	[1,1,1,1]	[1,1,0,1]	[1,1,1,1]	[1,1,1,1]
510	Class	1	[1,1,01]	[1,1,1,1]	[1,1,0,1]	[1,1,1,1]	[1,1,1,1]
511	Class	1	[1,1,01]	[1,1,1,1]	[1,1,0,1]	[1,1,1,1]	[1,1,1,1]
512	Class	1	[1,1,01]	[1,1,1,1]	[1,1,0,1]	[1,1,1,1]	[1,1,1,1]
513	Office	2	[1,1,0,0]	[1,0,1,1]	[1,0,0,1]	[1,1,1,0]	[1,1,0,1]
514	Office	2	[1,1,0,0]	[1,0,1,1]	[1,0,0,1]	[1,1,1,0]	[1,1,0,1]
515	Office	2	[1,1,0,0]	[1,0,1,1]	[1,0,0,1]	[1,1,1,0]	[1,1,0,1]
516	Conference	2	[1,1,0,0]	[1,0,1,1]	[1,0,0,1]	[1,1,1,0]	[1,1,0,1]
517	Class	1	[1,1,01]	[1,1,1,1]	[1,1,0,1]	[1,1,1,1]	[1,1,1,1]
518	Office	3	[1,0,0,1]	[1,0,0,1]	[0,1,1,1]	[0,0,0,1]	[1,1,0,1]
519	Class	2	[1,1,0,0]	[1,0,1,1]	[1,0,0,1]	[1,1,1,0]	[1,1,0,1]
520	Class	2	[1,1,0,0]	[1,0,1,1]	[1,0,0,1]	[1,1,1,0]	[1,1,0,1]
521	Class	2	[1,1,0,0]	[1,0,1,1]	[1,0,0,1]	[1,1,1,0]	[1,1,0,1]
522	Class	2	[1,1,0,0]	[1,0,1,1]	[1,0,0,1]	[1,1,1,0]	[1,1,0,1]
523	Class	2	[1,1,0,0]	[1,0,1,1]	[1,0,0,1]	[1,1,1,0]	[1,1,0,1]
524	Class	2	[1,1,0,0]	[1,0,1,1]	[1,0,0,1]	[1,1,1,0]	[1,1,0,1]
525	Office	4	[0,0,0,1]	[0,0,0,0]	[1,1,0,0]	[0,0,0,1]	[1,0,1,0]
526	Office	4	[0,0,0,1]	[0,0,0,0]	[1,1,0,0]	[0,0,0,1]	[1,0,1,0]
527	Class	2	[1,1,0,0]	[1,0,1,1]	[1,0,0,1]	[1,1,1,0]	[1,1,0,1]
528	Conference	2	[1,1,0,0]	[1,0,1,1]	[1,0,0,1]	[1,1,1,0]	[1,1,0,1]
529	Lab	2	[1,1,0,0]	[1,0,1,1]	[1,0,0,1]	[1,1,1,0]	[1,1,0,1]
530	Lab	3	[1,0,0,1]	[1,0,0,1]	[0,1,1,1]	[0,0,0,1]	[1,1,0,1]
531	Lab	3	[1,0,0,1]	[1,0,0,1]	[0,1,1,1]	[0,0,0,1]	[1,1,0,1]
532	Class	1	[1,1,01]	[1,1,1,1]	[1,1,0,1]	[1,1,1,1]	[1,1,1,1]
533	Lab	1	[1,1,01]	[1,1,1,1]	[1,1,0,1]	[1,1,1,1]	[1,1,1,1]
534	Class	1	[1,1,01]	[1,1,1,1]	[1,1,0,1]	[1,1,1,1]	[1,1,1,1]
535	Conference	2	[1,1,0,0]	[1,0,1,1]	[1,0,0,1]	[1,1,1,0]	[1,1,0,1]
536	Lab	1	[1,1,01]	[1,1,1,1]	[1,1,0,1]	[1,1,1,1]	[1,1,1,1]

Figure 4.4: Screen shot of room wise priority setup with task schedules for a week

The vacuuming speed is set at 0.025 feet per second regardless of the type of floor. The floor clean speed is set at 0.1 square feet per second. The window cleaning speed is set at 0.2 square feet per second.

Table 4.2: Task time speed setup

Task	**Task Speed Setup**
Walking speed	1 feet per second
Vacuuming speed	0.025 sq. ft. per sec
Floor cleaning speed	0.1 sq. ft. per sec
Window clean	0.2 sq. ft. per sec
Trash empty	30 seconds per bin

4.3 Evaluation modeling

This section presents the metrics identified to evaluate and compare the current and the new systems. The results were analyzed based on the following metrics: total daily time, task completion time, total annual labor cost, weekly utilization. The results have been calculated for m custodians.

4.3.1 Total daily time

The total daily time is the sum of the time required for all the custodians to complete the tasks on a particular day of the week. The total daily time is the sum of regular time and overtime recorded. The regular time of the shift is 8 hours; any time beyond the regular 8 hour shift is considered overtime as expressed below:

$$T_t = \sum_{i=1}^{m} (T_{r_i} + T_{o_i}) \tag{4.1}$$

where, T_{r_i} is the regular time in hours and T_{o_i} is the overtime in hours of i^{th}th custodian.

4.3.2 Task completion time

The task completion time is the maximum of the total daily times recorded for all custodians on a given day. In other words, the task completion time is defined as the time taken for a floor or a facility to be cleaned when all the custodians start together. Therefore, the total task completion time is the total time of the custodian who takes the maximum time.

$$T_{t_{max}} = Max(T_{t_i}) \tag{4.2}$$

where T_{t_i} is the total time recorded for i^{th}th custodian in a particular day, for all $i = 1, 2, ..m$.

4.3.3 Total annual labor cost

The total annual labor cost is the sum of annual hourly labor cost and the annual fixed cost per custodian. The hourly cost is a sum of regular hours' cost, overtime hours' cost and burden cost. As mentioned in the methodology chapter, burden costs are typically 25% of the total hourly cost. The fixed cost is assumed to be \$10,000 per year for the case study.

$$C_{t_y} = \sum_{i=1}^{m}(C_{ry_i} + C_{oy_i}) + (C_{by_i}) + (C_{fy_i}) \tag{4.3}$$

where, C_{t_y} is the total annual labor cost of one custodian, C_{ry} is the total regular time recorded for custodian i in a year, C_{oy_i} is the total overtime recorded for custodian i in a year. C_{by_i} is the total burden cost for the custodian i which is $0.25(C_{ry_i} + C_{oy_i})$. C_{fy_i} is the fixed cost per year for ith custodian which is assumed \$10,000 per year. The regular time wage is \$8 per hour per custodian and the overtime wage is \$10 per hour per custodian.

4.3.4 Weekly utilization

The weekly utilization in this case study is the total time recorded over the total shift time which is 8 hours in a day of the week per one custodian. Utilization is expressed in percentage, and if the percentage is above 100 that means the custodian has worked overtime. It is used to determine whether a custodian is working above or below the shift time and by how much.

4.4 Results and analysis of case study

This section presents the results obtained by executing the optimization system and its analysis using the metrics defined in the previous section. The custodian routing optimization was implemented for 1 custodian, 2 custodians, 3 custodians, and 4 custodians for the floor plan. The cost and time calculations were tabulated and analyzed in this section for all custodian models except the single-custodian model because the total custodian time per day for one custodian model for this floor plan exceeds the number of hours in a day and no feasible solutions are obtained in such a case. However, the single-custodian model's results are shown in Appendix A.

4.4.1 Two Custodian route

The 2-custodian route for the current system was calculated using the University of Tennessee Facilities Services' assignment method. The custodians were assigned a section of the floor to be cleaned and the rooms belonging to section assigned to the custodian could be cleaned in any order depending on the custodian. The new system involves its own optimized route for the custodians, and the route is shown in the optimization system to the custodian. The routes for each of these cases with different number of custodians are presented in Appendix A. The travel and task times are taken for each room in the current system to derive the total labor time per day. Table 4.3 shows the total times recorded using the current system for each custodian over 5 days of the week of January 19, 2015.

The current system's results table indicates that the total time recorded for custodians on each day throughout the week exceeds the 8 hour shift limit which requires the custodians to work overtime. The overtime recorded is higher than the normal regular hours in most instances which adds to extra cost thereby increasing the total labor cost. In some cases the overtime has been recorded up to 16 hours which is twice as the regular working hours. Table 4.4 shows the results obtained after the implementation of the new system. It is observed there is a significant reduction of labor time over few days of the week.

During the test week, the new routing implementation showed highest improvement in total time on Monday, with a 16.3% improvement compared to the current system. The lowest improvement in total labor time (2.49%) was seen on Tuesday, Wednesday, Thursday, and Friday had 9.9%, 4.9% and 12% improvement in total labor time, respectively. The overtime in the current system is as high as 64 hours per week, which has been reduced to 50 hours in the new system. Figure 4.5 shows the comparison of total labor times for 2-custodian routing for the new and current systems. The x-axis in the graph represents the days of the week with 1 signifying Monday, 2 signifying Tuesday, and so on to Friday. Though the decrease in total daily time is marginal, total task-time completion is significantly improved. This means if two custodians start at the same time on a given day, the time taken to complete all the tasks and return to the depot is significantly decreased up to 5 hours.

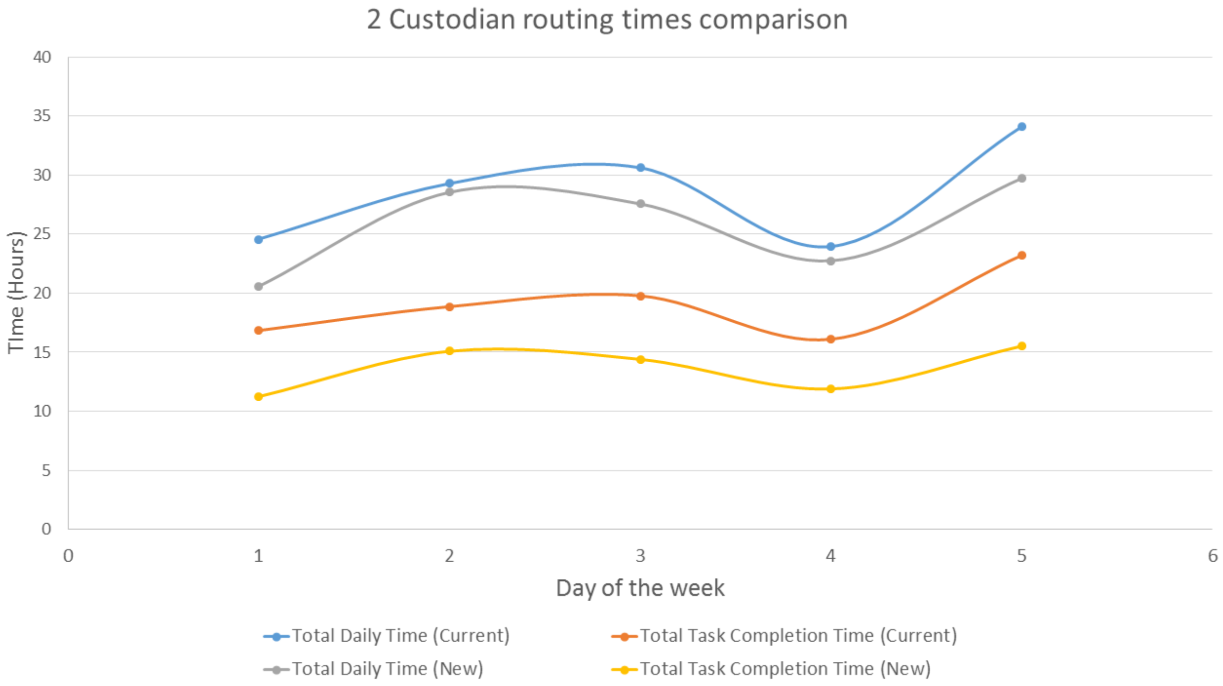


Figure 4.5: Labor time comparison for 2 custodians

Table 4.3: 2 Custodian results tabulated for current system

Current System										
	Monday		Tuesday		Wednesday		Thursday		Friday	
	M 1	M 2	M 1	M 2	M 1	M 2	M 1	M 2	M 1	M 2
Total Time Recorded (Hrs)	16.85	7.71	18.85	10.44	19.76	10.87	16.11	7.82	23.17	10.97
Regular Time Recorded (Hrs)	8	8	8	8	8	8	8	8	8	8
Overtime Recorded (Hrs)	8.85	0	10.85	2.44	11.76	2.87	8.11	0	15.17	2.97
Overtime Rounded (Hrs)	9	0	11	3	12	3	9	0	16	3
Cost of regular time ($)	64	64	64	64	64	64	64	64	64	64
Cost of overtime ($)	90	0	110	30	120	30	90	0	160	30
Total cost ($)	154	64	174	94	184	94	154	64	224	94
Burden cost ($)	38.5	16	43.5	23.5	46	23.5	38.5	16	56	23.5
Total cost per day ($)	192.5	80	217.5	117.5	230	117.5	192.5	80	280	117.5
Total cost per year on this day ($)	10010	4160	11310	6110	11960	6110	10010	4160	14560	6110
Costs/year	Hourly Cost	Fixed Cost ($)	Total ($)			Utilization per week	Max Utilization	Min Utilization	Difference (%)	
Total Custodian 1	57850	10000	67850			Time (hrs)	23.17156	7.716991		
Total Custodian 2	26650	10000	36650			Percentage	289.6445	96.46239	193.1821	
Total all custodians			104500							

Table 4.4: 2 Custodian results tabulated for new system

New System										
	Monday		Tuesday		Wednesday		Thursday		Friday	
	M 1	M 2	M 1	M 2	M 1	M 2	M 1	M 2	M 1	M 2
Total Time Recorded (Hrs)	11.221	9.337	15.109	13.454	14.412	13.168	11.884	10.863	15.553	14.203
Regular Time Recorded (Hrs)	8	8	8	8	8	8	8	8	8	8
Overtime Recorded (Hrs)	3.221	1.337	7.109	5.454	6.412	5.168	3.884	2.863	7.553	6.203
Overtime Rounded (Hrs)	4	2	8	6	7	6	4	3	8	7
Cost of regular time ($)	64	64	64	64	64	64	64	64	64	64
Cost of overtime ($)	40	20	80	60	70	60	40	30	80	70
Total cost ($)	104	84	144	124	134	124	104	94	144	134
Burden cost ($)	26	21	36	31	33.5	31	26	23.5	36	33.5
Total cost per day ($)	130	105	180	155	167.5	155	130	117.5	180	167.5
Total cost per year on this day ($)	6760	5460	9360	8060	8710	8060	6760	6110	9360	8710
Costs/year	Hourly Cost	Fixed Cost ($)	Total ($)			Utilization per week	Max Utilization	Min Utilization	Difference (%)	
Total Custodian 1	40950	10000	50950			Time (hrs)	15.553	9.337		
Total Custodian 2	36400	10000	46400			Percentage	194.4125	116.7125	77.7	
Total all custodians			97350							

Although the labor time is substantially minimized, the overtime can be further reduced by adding more personnel. An additional custodian was added to the current and the new systems, and the analysis is presented in the next section.

4.4.2 Three Custodian route

An additional custodian has been added to the current system, and the results are tabulated in Table 4.5 and 4.6. The floor plan is now divided into three parts for allocating areas to the three custodians. The overtimes have been reduced in comparison to the current 2-custodian system with an additional resource. Yet, the overtime for the entire week for all custodians still remains on the higher side at 50 hours per week, or 2600 hours of overtime per year, representing significant costs in labor over time and adding to total labor time.

An additional custodian is added in the optimization system. The route of the custodians' route is presented in Appendix A. Tables 4.7 and 4.8 show the total labor time and over time recorded with the new system's implementation, resulting in significant improvements in total labor time. The highest improvement was on Friday with 28.6% followed by Monday (15.6%), Wednesday (10.2%), Thursday (4.7%), and Tuesday (0.59%). Figure 4.6 shows the improvements in total labor time from the current system to the new system. Furthermore, overtime has been minimized from 50 hours in the current system to 16 hours per week in the new system.

Table 4.5: 3 Custodian results (Custodian 1 and 2) tabulated for current system

Current system										
	Monday		Tuesday		Wednesday		Thursday		Friday	
	M1	M2	M1	M2	M1	M2	M1	M2	M1	M2
Total Time Recorded (Hrs)	8.98	12.136	10.80	15.01	11.05	15.16	8.98	12.14	14.10	15.82
Regular Time recorded (Hrs)	8	8	8	8	8	8	8	8	8	8
Overtime Recorded (Hrs)	0.98	4.1361	2.80	7.01	3.05	7.16	0.98	4.13	6.10	7.82
Overtime Rounded (Hrs)	1	5	3	8	4	8	1	5	7	8
Cost of regular time ($)	64	64	64	64	64	64	64	64	64	64
Cost of overtime ($)	10	50	30	80	40	80	10	50	70	80
Total cost ($)	74	114	94	144	104	144	74	114	134	144
Burden cost ($)	18.5	28.5	23.5	36	26	36	18.5	28.5	33.5	36
Total cost per day($)	92.5	142.5	117.5	180	130	180	92.5	142.5	167.5	180
Total cost per year on this day ($)	4810	7410	6110	9360	6760	9360	4810	7410	8710	9360

Table 4.6: 3 Custodian results (Custodian 3) tabulated for current system

Current system					
	Monday	Tuesday	Wednesday	Thursday	Friday
	M3	M3	M3	M3	M3
Total Time Recorded (hrs)	7.786	10.175	10.08	8.565	9.161
Regular Time recorded (hrs)	8	8	8	8	8
Overtime Recorded (Hrs)	0	2.175	2.08	0.565	1.161
Overtime Rounded (hrs)	0	3	3	1	2
Cost of regular time ($)	64	64	64	64	64
Cost of overtime ($)	0	30	30	10	20
Total cost ($)	64	94	94	74	84
Burden cost ($)	16	23.5	23.5	18.5	21
Total cost per day($)	80	117.5	117.5	92.5	105
Total cost per year on this day ($)	4160	6110	6110	4810	5460
Yearly Cost (3 Custodian)					
Cost per year	Hourly Cost	Fixed Cost	Total		
Custodian M1	31200	10000	41200		
Custodian M2	42900	10000	52900		
Custodian M3	12480	10000	22480		
Total			116580		
Utilization (3 Custodian)					
	Maximum Utilization	Minimum Utilization	Utilization Difference		
Time	15.82	3.05			
Percentage	197.8	38.22	159.6		

Table 4.7: 3 Custodian results (Custodian 1 and 2) tabulated for new system

New system										
	Monday		Tuesday		Wednesday		Thursday		Friday	
	M1	M2	M1	M2	M1	M2	M1	M2	M1	M2
Total Time Recorded (Hrs)	7.786	7.884	10.175	9.559	10.08	9.746	8.565	7.69	9.161	7.699
Regular Time recorded (Hrs)	8	8	8	8	8	8	8	8	8	8
Overtime Recorded (Hrs)	0	0	2.175	1.559	2.08	1.746	0.565	0	1.161	0
Overtime Rounded (Hrs)	0	0	3	2	3	2	1	0	2	0
Cost of regular time ($)	64	64	64	64	64	64	64	64	64	64
Cost of overtime ($)	0	0	30	20	30	20	10	0	20	0
Total cost ($)	64	64	94	84	94	84	74	64	84	64
Burden cost ($)	16	16	23.5	21	23.5	21	18.5	16	21	16
Total cost per day($)	80	80	117.5	105	117.5	105	92.5	80	105	80
Total cost per year on this day ($)	4160	4160	6110	5460	6110	5460	4810	4160	5460	4160

Table 4.8: 3 Custodian results (Custodian 3) tabulated for new system

New system					
	Monday	Tuesday	Wednesday	Thursday	Friday
	M3	M3	M3	M3	M3
Total Time Recorded (hrs)	4.72	8.996	8.266	6.877	8.678
Regular Time recorded (hrs)	5	8	8	8	8
Overtime Recorded (Hrs)	0	0.996	0.266	0	0.678
Overtime Rounded (hrs)	0	1	1	0	1
Cost of regular time ($)	40	64	64	56	64
Cost of overtime ($)	0	10	10	0	10
Total cost ($)	40	74	74	56	74
Burden cost ($)	10	18.5	18.5	14	18.5
Total cost per day($)	50	92.5	92.5	70	92.5
Total cost per year on this day ($)	2600	4810	4810	3640	4810
Yearly Cost (3 Custodian)					
Cost per year	Hourly Cost	Fixed Cost	Total		
Custodian M1	26650	10000	36650		
Custodian M2	23400	10000	33400		
Custodian M3	20670	10000	30670		
Total			100720		
Utilization (3 Custodian)					
	Maximum Utilization	Minimum Utilization	Utilization Difference		
Time	10.175	4.72			
Percentage	127.18	59	68.18		

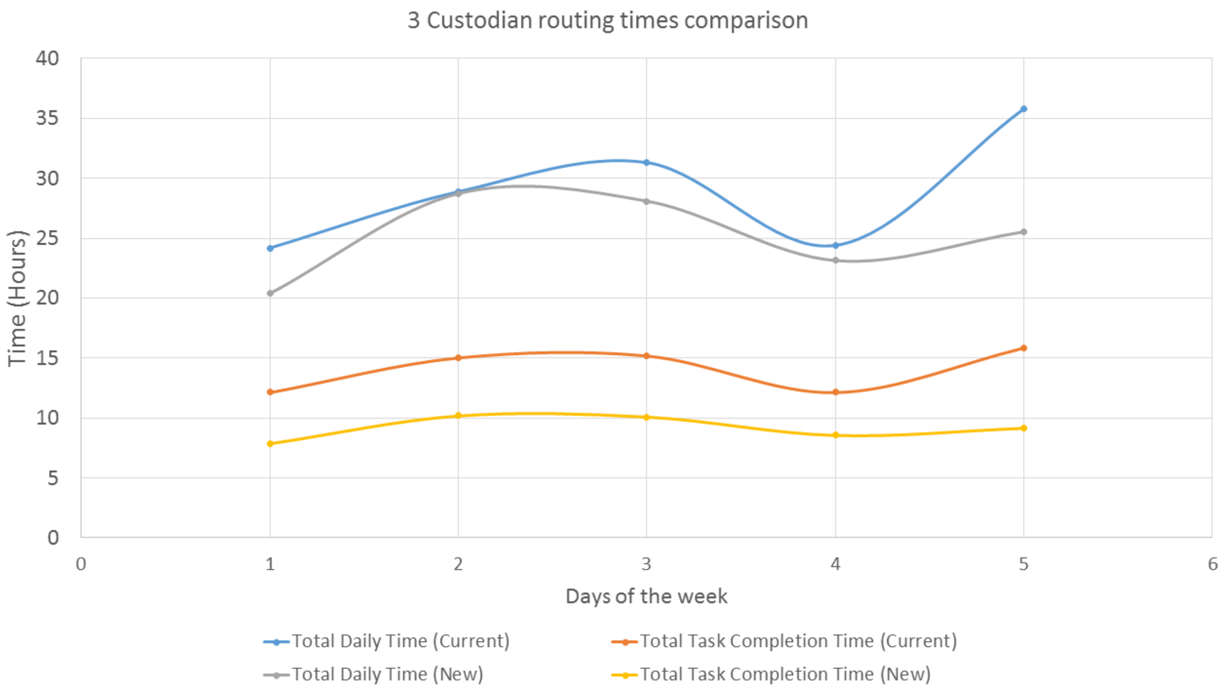


Figure 4.6: Labor time comparison for 3 custodians

4.4.3 Four Custodian route

A fourth custodian has been added to further reduce the custodians' overtime per week. The 4-custodian model's routes and calculations are presented in Appendix A. The current system's total labor time is shown in Tables 4.9 and 4.10. Table 4.11 and 4.12 shows the results for 4 custodian routing for the new system. When the two systems' results are compared, the total labor time has been minimized in the new system by 8.56%, 3.8%, 4.6%, 2.8% and 13.4% over the five days of implementation.

Furthermore, the overtime has been reduced from 17 hours per week in the current system to zero hours or no overtime over the entire week. At first glance, the 4 custodians might seem to be a good fit for the current setup of priorities. However, for some days, few custodians have been used for less than 6 hours of the total time and the other custodians have been used for more than 8 hours. Figure 4.7 shows the comparison between new system and the current system. Total task completion time is still better in the new system.

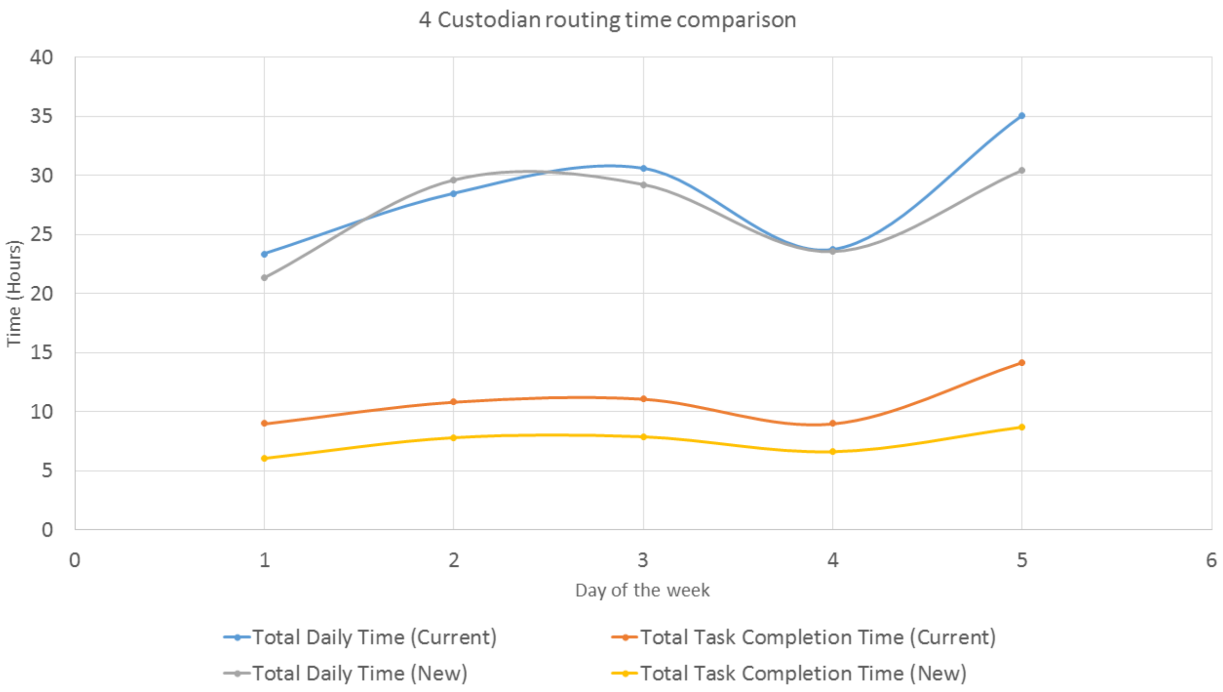


Figure 4.7: Labor time comparison for 4 custodians

4.4.4 Sensitivity analysis

All of the new system's results enable the user to perform a sensitivity analysis, in which the number of custodians needed per day can be chosen instead of having the same number of custodians over the entire week. In this case study, the following changes in the number of custodians have been implemented over the 5 days of the week:

1. Monday - 3 Custodians
2. Tuesday - 4 Custodians
3. Wednesday - 4 Custodians
4. Thursday - 3 Custodians
5. Friday - 4 Custodians

Table 4.13 shows the results of the sensitivity analysis.

Table 4.9: 4 Custodian results (Custodian 1 and 2) tabulated for current system

Current System										
	Monday		Tuesday		Wednesday		Thursday		Friday	
	M1	M2	M1	M2	M1	M2	M1	M2	M1	M2
Total Time Recorded (Hrs)	8.9	5.55	10.8	6.87	11.05	7.08	8.98	5.50	14.1	6.8
Regular Time recorded (Hrs)	8	6	8	7	8	8	8	6	8	7
Overtime Recorded (Hrs)	0.9	0	2.8	0	3.05	0	0.98	0	6.1	0
Overtime Rounded (Hrs)	1	0	3	0	4	0	1	0	7	0
Cost of regular time ($)	64	48	64	56	64	64	64	48	64	56
Cost of overtime ($)	10	0	30	0	40	0	10	0	70	0
Total cost ($)	74	48	94	56	104	64	74	48	134	56
Burden cost ($)	18.5	12	23.5	14	26	16	19	12	33.5	14
Total cost per day($)	92.5	60	118	70	130	80	93	60	167	70
Total cost per year on this day ($)	4810	3120	6110	3640	6760	4160	4810	3120	8710	3640

Table 4.10: 4 Custodian results (Custodian 3 and 4) tabulated for current system

Current System										
	Monday		Tuesday		Wednesday		Thursday		Friday	
	M3	M4	M3	M4	M3	M4	M3	M4	M3	M4
Total Time Recorded (Hrs)	5.80	3.0	7.62	3.17	6.55	5.91	5.85	3.34	8.8	5.29
Regular Time recorded (Hrs)	6	4	8	4	7	6	6	4	8	6
Overtime Recorded (Hrs)	0	0	0	0	0	0	0	0	0.8	0
Overtime Rounded (Hrs)	0	0	0	0	0	0	0	0	1	0
Cost of regular time ($)	48	32	64	32	56	48	48	32	64	48
Cost of overtime ($)	0	0	0	0	0	0	0	0	10	0
Total cost ($)	48	32	64	32	56	48	48	32	74	48
Burden cost ($)	12	8	16	8	14	12	12	8	19	12
Total cost per day($)	60	40	80	40	70	60	60	40	93	60
Total cost per year on this day ($)	3120	2080	4160	2080	3640	3120	3120	2080	4810	3120

Table 4.11: 4 Custodian results (Custodian 1 and 2) tabulated for new system

New system										
	Monday		Tuesday		Wednesday		Thursday		Friday	
	M1	M2	M1	M2	M1	M2	M1	M2	M1	M2
Total Time Recorded (Hrs)	6.03	6.04	7.6	7.2	7.3	7.1	6.6	5.8	7.6	8.6
Regular Time recorded (Hrs)	7	7	8	8	8	8	7	6	8	8
Overtime Recorded (Hrs)	0	0	0	0	0	0	0	0	0	0
Overtime Rounded (Hrs)	0	0	0	0	0	0	0	0	0	0
Cost of regular time ($)	56	56	64	64	64	64	56	48	64	64
Cost of overtime ($)	0	0	0	0	0	0	0	0	0	0
Total cost ($)	56	56	64	64	64	64	56	48	64	64
Burden cost ($)	14	14	16	16	16	16	14	12	16	16
Total cost per day($)	70	70	80	80	80	80	70	60	80	80
Total cost per year on this day ($)	3640	3640	4160	4160	4160	4160	3640	3120	4160	3120

Table 4.12: 4 Custodian results (Custodian 3 and 4) tabulated for new system

New system										
	Monday		Tuesday		Wednesday		Thursday		Friday	
	M3	M4	M3	M4	M3	M4	M3	M4	M3	M4
Total Time Recorded (Hrs)	5.9	3.3	7.7	6.8	7.8	6.9	5.9	5.09	7.9	6.1
Regular Time recorded (Hrs)	6	4	8	7	8	7	6	6	8	7
Overtime Recorded (Hrs)	0	0	0	0	0	0	0	0	0	0
Overtime Rounded (Hrs)	0	0	0	0	0	0	0	0	0	0
Cost of regular time ($)	48	32	64	56	64	56	48	48	64	56
Cost of overtime ($)	0	0	0	0	0	0	0	0	0	0
Total cost ($)	48	56	64	64	64	64	56	48	64	64
Burden cost ($)	12	8	16	14	16	14	12	12	16	14
Total cost per day($)	60	40	80	70	80	70	60	60	80	70
Total cost per year on this day ($)	3120	2080	4160	3640	4160	3640	3120	3120	4160	4160

Table 4.13: Sensitivity analysis results

New System																		
	Monday			Tuesday				Wednesday				Thursday			Friday			
	M 1	M 2	M 3	M 1	M 2	M 3	M 4	M1	M 2	M 3	M4	M 1	M 2	M 3	M 1	M 2	M 3	M4
Total Time (Hrs)	7.78	7.88	4.72	7.67	7.29	7.78	6.82	7.32	7.10	7.86	6.90	8.56	7.69	6.87	7.69	8.67	7.90	6.10
Regular Time (Hrs)	8	8	5	8	8	8	7	8	8	8	7	8	8	7	8	8	8	7
Overtime (Hrs)	0	0	0	0	0	0	0	0	0	0	0	0.56	0	0	0	0	0	0
Overtime Rounded	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Cost of regular time	64	64	40	64	64	64	56	64	64	64	56	64	64	56	64	64	64	56
Cost of overtime	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0
Total cost	64	64	40	64	64	64	56	64	64	64	56	74	64	56	64	64	64	56
Burden cost ($)	16	16	10	16	16	16	14	16	16	16	14	18.5	16	14	16	16	16	14
Total cost per day ($)	80	80	50	80	80	80	70	80	80	80	70	92.5	80	70	80	80	80	70
Total cost per year on this day ($)	4160	4160	2600	4160	4160	4160	3640	4160	4160	4160	3640	4810	4160	3640	4160	4160	4160	3640

The total labor time per week in 4 custodian route for the new system is 142 hours per week, whereas the total labor time has been reduced to 135 hours per week. Therefore, the number of paid working hours is further reduced even with the decreased number of custodians on 2 days. Thus, the new system can provide results for the administrator who can control the variables, in this case number of custodians, to suit the requirements of the entity (time) to be reduced. In addition, the results provide data for efficiently using personnel.

Table 4.14: Total yearly labor cost and utilization rates

Yearly costs				Utilization			
Cost per year	Hourly Cost ($)	Fixed Cost ($)	Total ($)		Max	Min	Diff.
M1	20930	10000	30930	Time	8.67	4.72	
M2	20800	10000	30800	Percentage	108	59	49.4
M3	18720	10000	28720				
M4	10920	10000	20920				

Utilization analysis

As mentioned earlier, apart from the cost and labor time, utilization difference is a major metric used to compare the new and current systems. Utilization is the total number of hours worked per day over the total shift time, which is 8 hours. Figure 4.8 shows the utilization percentages in all the cases above for both the new and current systems. The difference in utilization of the custodian who works the maximum time during a week and a custodian who works the minimum time in the same week is very high in the current system. It goes as high as 190% in a 2-custodian routing. This creates a difference between the custodian working times, leading to overtime in some cases and under utilization of some custodians. The difference in utilization is far less in the new system compared to the old system by more than half in all cases. However, in the new system, the gap between the maximum utilized custodians and the minimum utilized custodians is still apparent and must be minimized. In a 4-custodian routing, the difference in custodian utilization is reduced from 66% to 49%.

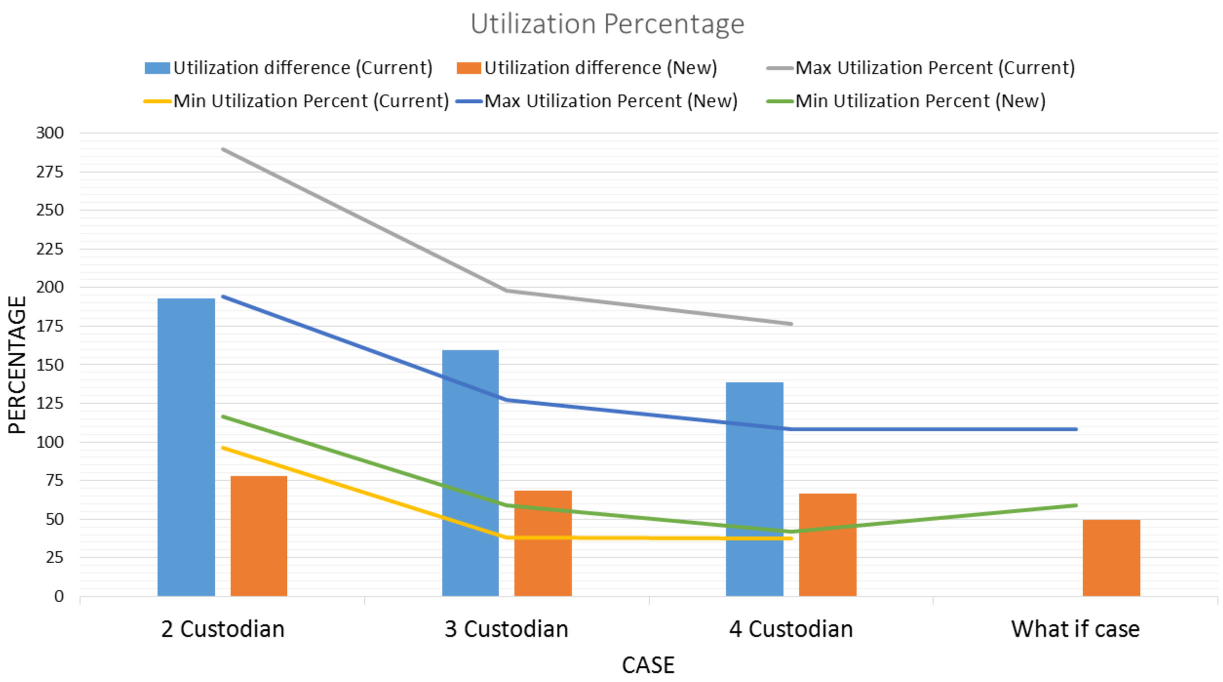


Figure 4.8: Utilization analysis of all the cases

4.4.5 Impact of new system on labor costs

Minimized labor time reduces labor costs in the new system. Figure 4.9 shows the comparison of total labor costs over a year for all the cases presented. The new system can reduce cost with improved routing involving a fixed number of custodians. However, the labor costs in the new system tend to increase with the addition of one extra custodian for the same tasks. The fixed costs play a major role in increasing these costs with an increased number of custodians. Though the labor time is significantly reduced in all the cases compared to the current system, the addition of a custodian results in an additional $10,000 per year per custodian. Labor cost calculations are presented in the earlier section for each case. The best case shows the least number of custodians necessary with least cost over the 4 custodian method.

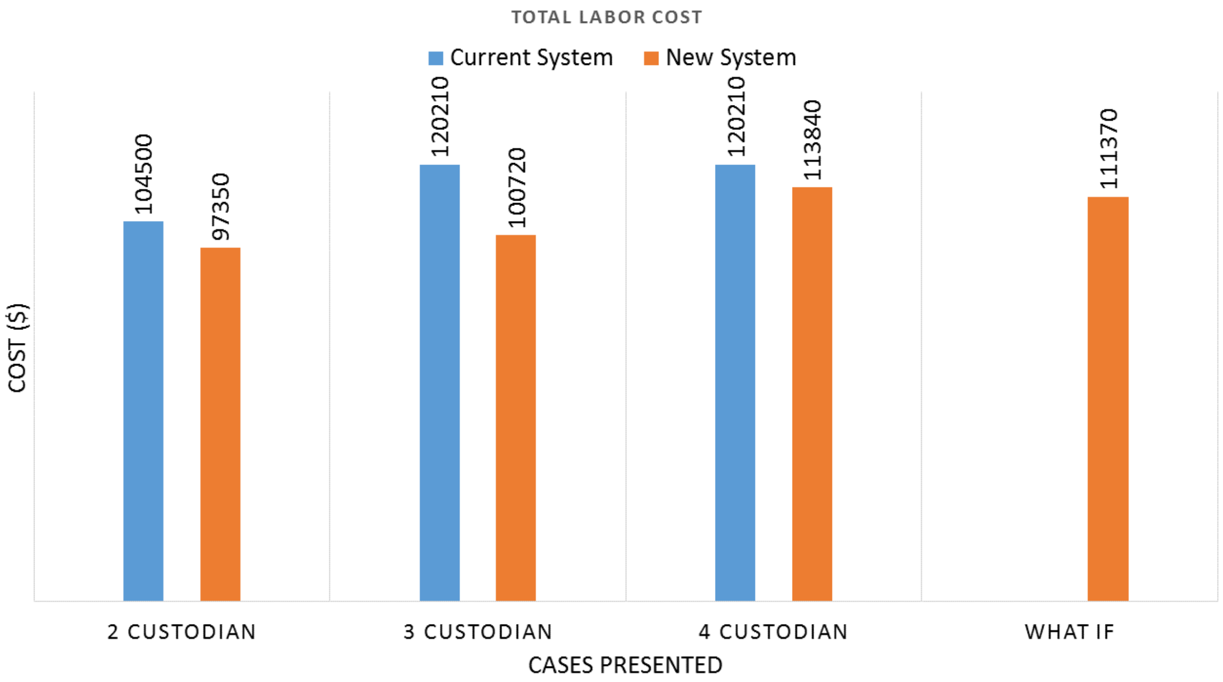


Figure 4.9: Total labor cost analysis

4.5 Summary of analysis

Based on the analysis, the total labor time over a week is significantly reduced in the new system in all the routing cases. Task-completion time is much lower in the new system compared to the current system. The new system provides data that an administrator can use to further change variables according to the requirements. Custodian's overtime is gradually decreased in each case, and the sensitivity analysis yielded zero overtime per week. The utilization differences have gradually been brought closer in all cases and up to 50% in the sensitivity analysis case, thus ensuring that the work is uniformly distributed among the custodians according to needs. Fixed cost significantly affects the total labor costs. The fixed costs tend to increase when the number of custodians is increased. Labor costs are increased with increased number of custodians; however, the labor costs are less in the new system compared to the current system in all the cases presented.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

A custodian management system focused on reduction in labor time using an optimization model has been presented. The implemented system consists of user friendly interfaces for the facility manager and the custodian and a secure database to manage and store schedules. The model and its implementation result in significant reduction in maintenance costs. A case study based on facility management data from the University of Tennessee showcases the results of the implementation. Benefits of the system are outlined as follows:

Budget planning

The maintenance budget for a facility can be effectively managed by an administrator. The developed optimization framework uses clustering and VRP to route custodians, resulting in savings in total daily time and cost per custodian. These cost savings allow an administrator to operate a facility within budgetary constraints.

Utilization rates

The optimization model yields a significant improvement in utilization rates of custodians. Overtime of each custodian is significantly reduced, indicative of balanced utilization of custodians.

Plan for new buildings

The custodian management system can be used to design schedules and to make custodian hiring decisions for planned facilities. The presented model is capable of determining the number of required custodians and their routes for each day using only the building floor plan. Various scheduling combinations can be tested, allowing the facility manager to pick the combination best suited to the new facility.

Usability

The designed system is user friendly and comprehensive. A facility manager can define room setup, room cleaning priority, and required room cleanliness levels with minimum training. A custodian has access to a graphical interface showing the room sequence, tasks in each room, and estimated time available for cleaning a room.

5.2 Future work

Model and System

The presented optimization model based on labor costs may be extended to include material costs. This will minimize the total maintenance cost in an organization. The model could be extended to include multiple buildings and floors using a multiple depot variation of VRP. The current model has been tested on a building operated by an educational institution. Variation in costs in other types of facilities may be factored into a future iteration of the model.

Software

The database system could be set up using a dedicated server rather than using a host database server. A dedicated server results in greater database security. An integrated automated data flow system could be designed to automate the creation of spreadsheets used by the Matlab and VB applications. Calibrated images for floor plans, e.g. CAD drawings, could be used in future versions of the software to standardize floor plan calculations.

Mobile device application

The scheduling software can be implemented as an app for mobile devices. This will enable custodians to use the scheduling interface while working and will allow the administrator to get real time updates on the maintenance status of each building.

Bibliography

Al-Zubaidi, H. (1997). Assessing the demand for building maintenance in a major hospital complex. *Property Management*, 15(3):173–183. 1

Al-Zubaidi, H. and Christer, A. (1997). Maintenance manpower modelling for a hospital building complex. *European Journal of Operational Research*, 99(3):603–618. 1, 18

Bjoberg, S., Larsen, A., and Oiseth, H. (2007). *Livssykluskonstader for bygninger*. someone. 1

Brown, G. G., Ellis, C. J., Graves, G. W., and Ronen, D. (1987). Real-time, wide area dispatch of mobil tank trucks. *Interfaces*, 17(1):107–120. 22

Christer, A. (1990). The contribution of operational research modelling to the management of building maintenance. *Building Maintenance and Modernization World Wide*, 1(na):na. 2

Christofides, N., Mingozzi, A., and Toth, P. (1981). Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical programming*, 20(1):255–282. 23

Clarke, G. u. and Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568–581. 25

Dantzig, G., Fulkerson, R., and Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2(4):393–410. 37

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1):80–91. 23

Dekker, R. (1996). Applications of maintenance optimization models: a review and analysis. *Reliability and System Safety*, 51(1):229–140. 19

Duffuaa, S. and Raouf, A. (1992). Simulation model for determining maintenance staffing in an industrial environment. *Simulation*, 59(2):93–99. cited By (since 1996)0. 18

Eilon, S., Watson-Gandy, C., and Christofides, N. (1971). *Distribution management*. Griffin London. 23

Evans, S. R. and Norback, J. P. (1985). The impact of a decision-support system for vehicle routeing in a foodservice supply situation. *Journal of the Operational Research Society*, pages 467–472. 22

Fisher, M. L. and Jaikumar, R. (1978). *A decomposition algorithm for large-scale vehicle routing*. Department of Decision Sciences, Wharton School, University of Pennsylvania. 25

Fisher, M. L. and Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. *Networks*, 11(2):109–124. 25

Frank D, P. J. (2010). *Infoclean smart staffing: Bidding & estimating book*. na. 2, 40, 72

Gillett, B. E. and Miller, L. R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations research*, 22(2):340–349. 25, 27

Golden, B. L. and Wasil, E. A. (1987). Or practicecomputerized vehicle routing in the soft drink industry. *Operations research*, 35(1):6–17. 22

Kovàcs, A. (2008). Solving the vehicle routing problem with genetic algorithm and simulated annealing. 27

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358. 23, 27

Laporte, G., Mercure, H., and Nobert, Y. (1992). A branch and bound algorithm for a class of asymmetrical vehicle routeing problems. *Journal of the Operational Research Society*, pages 469–481. 23

Laporte, G. and Osman, I. H. (1995). Routing problems: A bibliography. *Annals of Operations Research*, 61(1):227–262. 20

Matai, R., Mittal, M. L., and Singh, S. (2010). *Traveling salesman problem: An overview of applications, formulations, and solution approaches*. INTECH Open Access Publisher. 37

Mathews, B. and Waters, C. (1986). Computerized routeing for community nurses-a pilot study. *Journal of the Operational Research Society*, pages 677–683. 22

Muppaneni, H. (2014). A lean information management model for efficient operations of an educational entity at the university of tennessee. 49

Renaud, J. and Boctor, F. F. (2002). A sweep-based algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research*, 140(3):618–628. 27

Rommert Dekker, P. A. S. (1998). On the impact of optimization models in maintenance decision making: the state of the art. *Reliability and System Safety*, 60(1):111–119. 19

Surekha, P. and Sumathi, S. (2011). Solution to multi-depot vehicle routing problem using genetic algorithms. *World Applied Programming*, 1(3):118–131. 25

Tay, L. and Ooi, J. T. (2001). Facilities management: a jack of all trades? *Facilities*, 19(10):357–363. 6

Teodorovic, D. (1986). *Transportation networks: a quantitative treatment*, volume 6. Taylor & Francis. 30

Wauters, B. (2005). The added value of facilities management: benchmarking work processes. *Facilities*, 23(3/4):142–151. 1

Williams, B. (1996). Cost-effective facilities management: a practical approach. *Facilities*, 14(5/6):26–38. 1

Wren, A. and Holliday, A. (1972). Computer scheduling of vehicles from one or more depots to a number of delivery points. *Operational Research Quarterly*, pages 333–344. 25

Appendix

Appendix A

Custodian routes and results

A.1 Single custodian model results

A.1.1 Current system results for single custodian

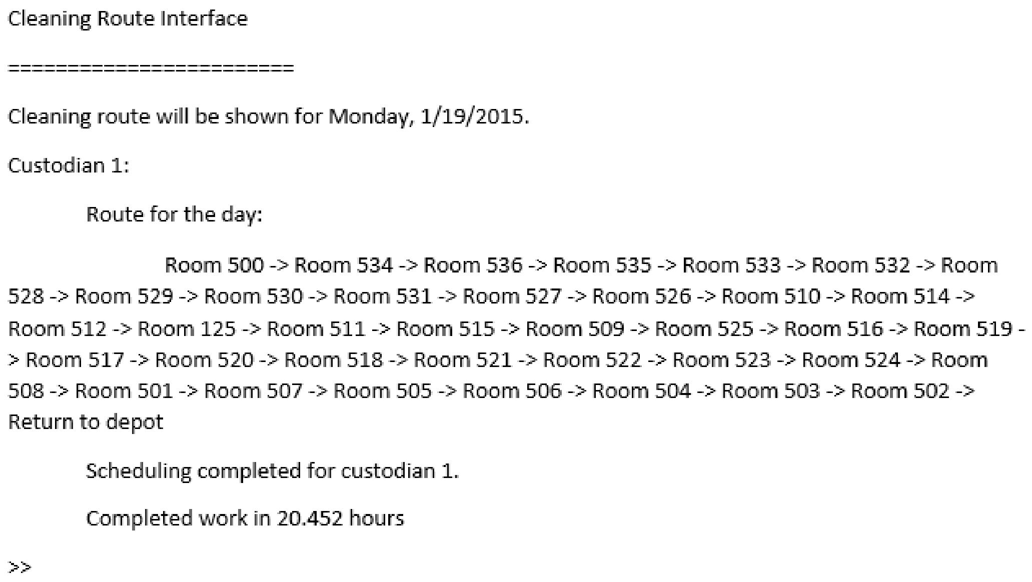
	Current System				
	Monday	Tuesday	Wednesday	Thursday	Friday
Total Time (Hrs)	20.91119	25.98572	27.55513	21.52918	32.65658
Regular Time (Hrs)	8	8	8	8	8
Overtime (Hrs)	12.91119	17.98572	19.55513	13.52918	24.65658
Overtime rounded	13	18	20	14	25
Cost of regular time	64	64	64	64	64
Cost of overtime	130	180	200	140	246.5658
Total Cost ($)	194	244	264	204	310.5658
Burden Cost ($)	48.5	61	66	51	77.64145
Total Cost per day ($)	242.5	305	330	255	388.20725
Total Cost per year on this day ($)	12610	15860	17160	13515	20186.777
	Hourly cost	Fixed Cost	Total		
Total Labor cost per year($)	79331.777	10000	**89331.78**		

A.1.2 New system results for single custodian

	New System				
	Monday	Tuesday	Wednesday	Thursday	Friday
Total Time (Hrs)	20.452	28.145	27.36	22.552	29.537
Regular Time (Hrs)	8	8	8	8	8
Overtime (Hrs)	12.452	20.145	19.36	14.552	21.537
Overtime rounded	13	21	20	15	22
Cost of regular time	64	64	64	64	64
Cost of overtime	130	210	200	150	220
Total Cost ($)	194	274	264	214	284
Burden Cost ($)	48.5	68.5	66	53.5	71
Total Cost per day ($)	242.5	342.5	330	267.5	355
Total Cost per year on this day ($)	12610	17810	17160	14177.5	18460
	Hourly cost	Fixed Cost	Total		
Total Labor cost per year($)	80217.5	10000	**90217.5**		

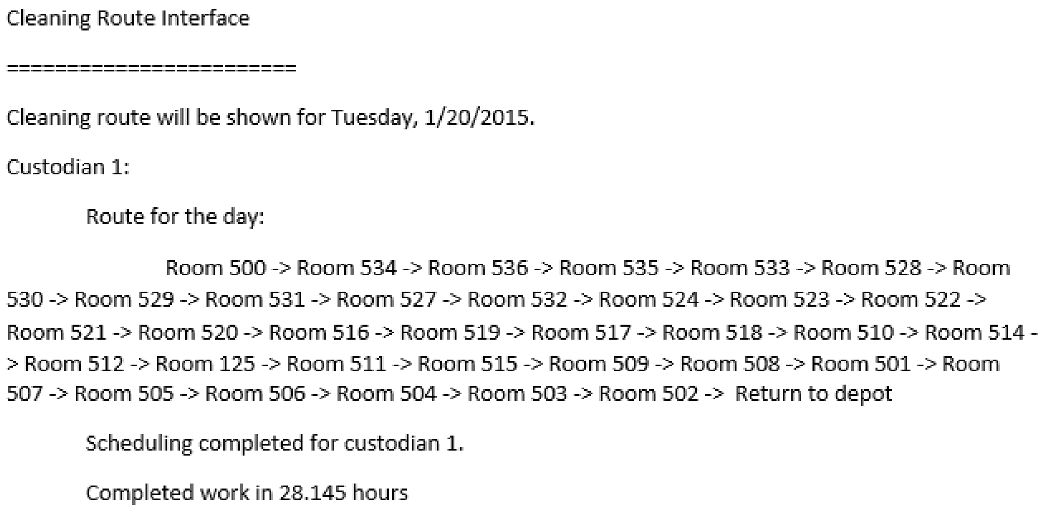
A.1.3 Routes for new system in the software

Single custodian routes for new system



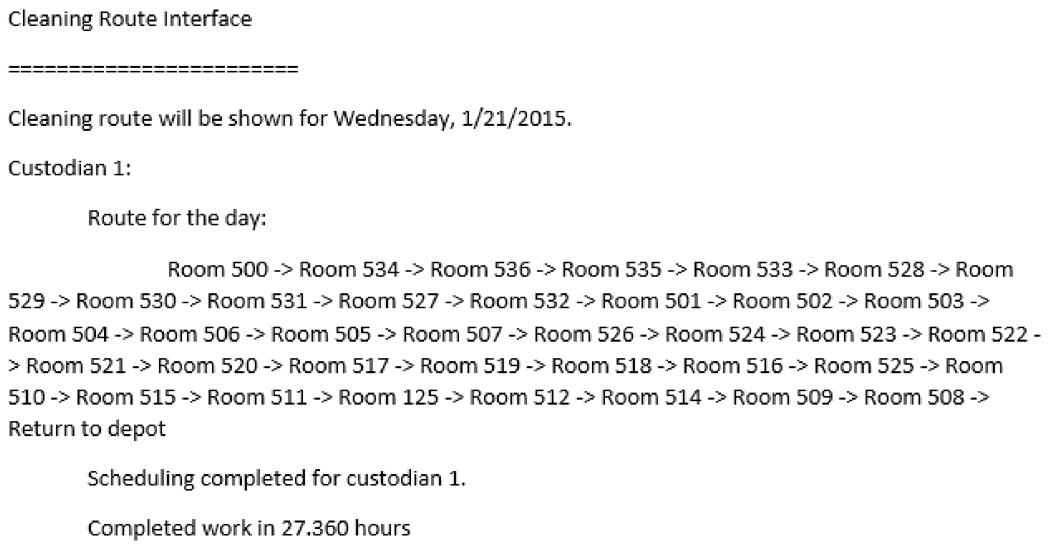
```
Cleaning Route Interface

========================

Cleaning route will be shown for Monday, 1/19/2015.

Custodian 1:

          Route for the day:

                    Room 500 -> Room 534 -> Room 536 -> Room 535 -> Room 533 -> Room 532 -> Room
528 -> Room 529 -> Room 530 -> Room 531 -> Room 527 -> Room 526 -> Room 510 -> Room 514 ->
Room 512 -> Room 125 -> Room 511 -> Room 515 -> Room 509 -> Room 525 -> Room 516 -> Room 519 -
> Room 517 -> Room 520 -> Room 518 -> Room 521 -> Room 522 -> Room 523 -> Room 524 -> Room
508 -> Room 501 -> Room 507 -> Room 505 -> Room 506 -> Room 504 -> Room 503 -> Room 502 ->
Return to depot

          Scheduling completed for custodian 1.

          Completed work in 20.452 hours

>>
```

Figure A.1: Single custodian route for Monday



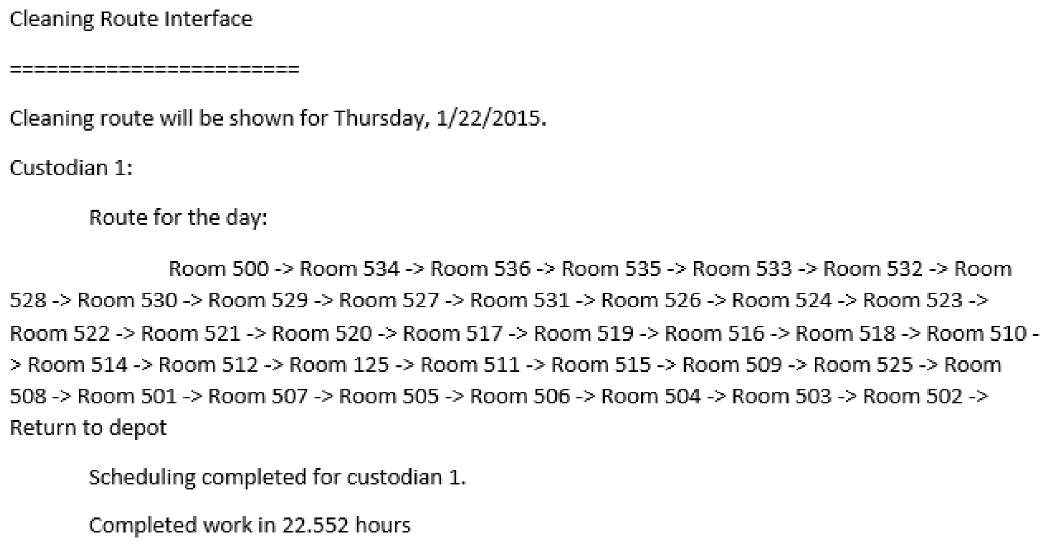
```
Cleaning Route Interface

========================

Cleaning route will be shown for Tuesday, 1/20/2015.

Custodian 1:

          Route for the day:

                    Room 500 -> Room 534 -> Room 536 -> Room 535 -> Room 533 -> Room 528 -> Room
530 -> Room 529 -> Room 531 -> Room 527 -> Room 532 -> Room 524 -> Room 523 -> Room 522 ->
Room 521 -> Room 520 -> Room 516 -> Room 519 -> Room 517 -> Room 518 -> Room 510 -> Room 514 -
> Room 512 -> Room 125 -> Room 511 -> Room 515 -> Room 509 -> Room 508 -> Room 501 -> Room
507 -> Room 505 -> Room 506 -> Room 504 -> Room 503 -> Room 502 ->  Return to depot

          Scheduling completed for custodian 1.

          Completed work in 28.145 hours
```

Figure A.2: Single custodian routes for Tuesday



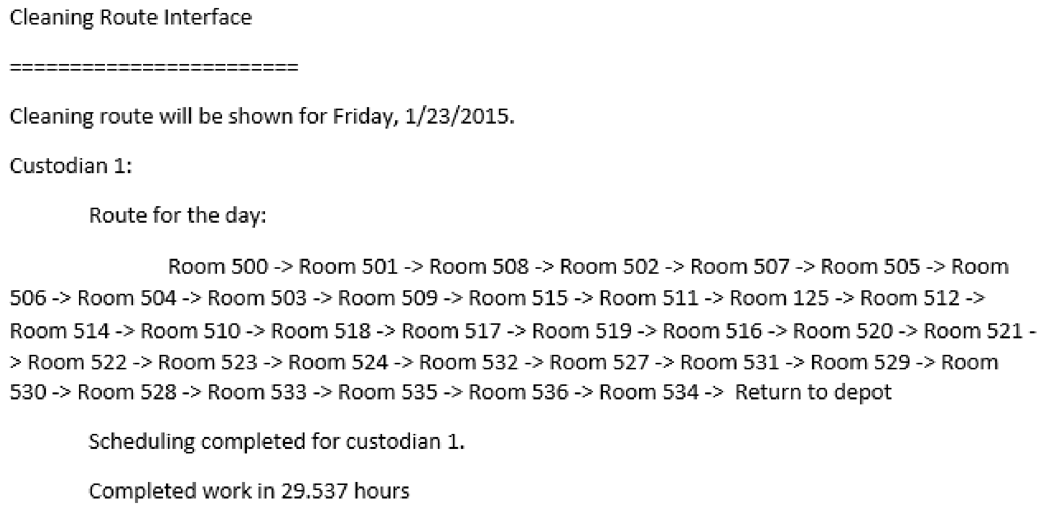
```
Cleaning Route Interface

========================

Cleaning route will be shown for Wednesday, 1/21/2015.

Custodian 1:

        Route for the day:

                Room 500 -> Room 534 -> Room 536 -> Room 535 -> Room 533 -> Room 528 -> Room
529 -> Room 530 -> Room 531 -> Room 527 -> Room 532 -> Room 501 -> Room 502 -> Room 503 ->
Room 504 -> Room 506 -> Room 505 -> Room 507 -> Room 526 -> Room 524 -> Room 523 -> Room 522 -
> Room 521 -> Room 520 -> Room 517 -> Room 519 -> Room 518 -> Room 516 -> Room 525 -> Room
510 -> Room 515 -> Room 511 -> Room 125 -> Room 512 -> Room 514 -> Room 509 -> Room 508 ->
Return to depot

        Scheduling completed for custodian 1.

        Completed work in 27.360 hours
```

Figure A.3: Single custodian routes for Wednesday



```
Cleaning Route Interface

========================

Cleaning route will be shown for Thursday, 1/22/2015.

Custodian 1:

        Route for the day:

                Room 500 -> Room 534 -> Room 536 -> Room 535 -> Room 533 -> Room 532 -> Room
528 -> Room 530 -> Room 529 -> Room 527 -> Room 531 -> Room 526 -> Room 524 -> Room 523 ->
Room 522 -> Room 521 -> Room 520 -> Room 517 -> Room 519 -> Room 516 -> Room 518 -> Room 510 -
> Room 514 -> Room 512 -> Room 125 -> Room 511 -> Room 515 -> Room 509 -> Room 525 -> Room
508 -> Room 501 -> Room 507 -> Room 505 -> Room 506 -> Room 504 -> Room 503 -> Room 502 ->
Return to depot

        Scheduling completed for custodian 1.

        Completed work in 22.552 hours
```

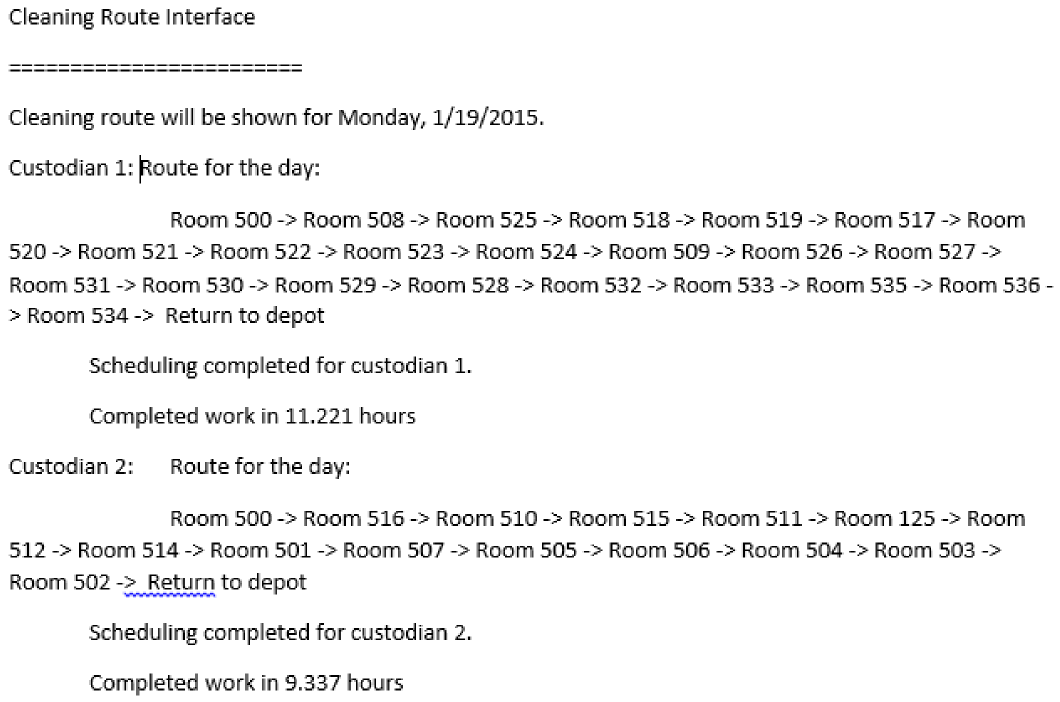
Figure A.4: Single custodian routes for Thursday



```
Cleaning Route Interface

========================

Cleaning route will be shown for Friday, 1/23/2015.

Custodian 1:

          Route for the day:

                    Room 500 -> Room 501 -> Room 508 -> Room 502 -> Room 507 -> Room 505 -> Room
506 -> Room 504 -> Room 503 -> Room 509 -> Room 515 -> Room 511 -> Room 125 -> Room 512 ->
Room 514 -> Room 510 -> Room 518 -> Room 517 -> Room 519 -> Room 516 -> Room 520 -> Room 521 -
> Room 522 -> Room 523 -> Room 524 -> Room 532 -> Room 527 -> Room 531 -> Room 529 -> Room
530 -> Room 528 -> Room 533 -> Room 535 -> Room 536 -> Room 534 ->  Return to depot

          Scheduling completed for custodian 1.

          Completed work in 29.537 hours
```

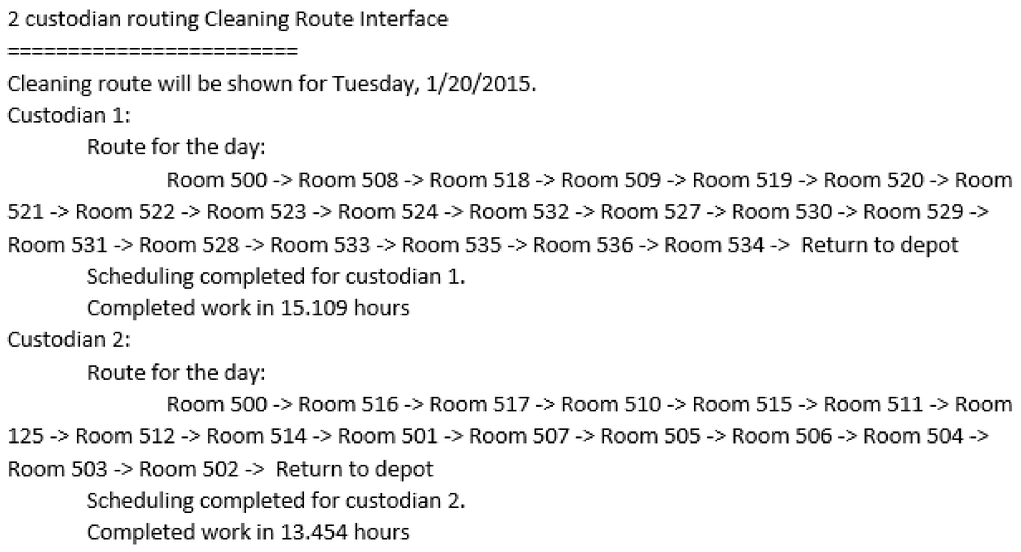
Figure A.5: Single custodian routes for Friday

Two custodian routing for new system



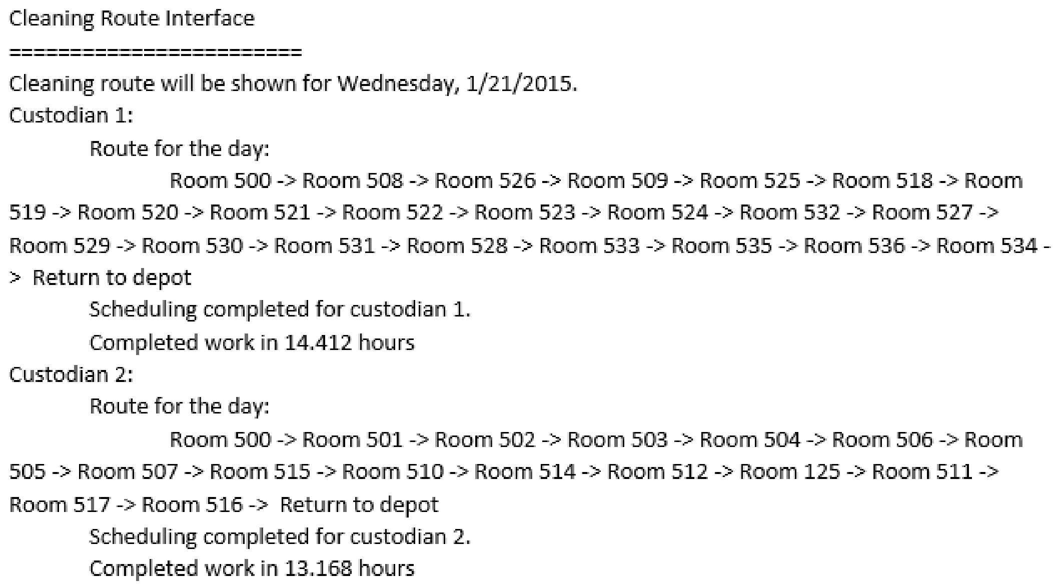
```
Cleaning Route Interface

========================

Cleaning route will be shown for Monday, 1/19/2015.

Custodian 1: Route for the day:

                    Room 500 -> Room 508 -> Room 525 -> Room 518 -> Room 519 -> Room 517 -> Room
520 -> Room 521 -> Room 522 -> Room 523 -> Room 524 -> Room 509 -> Room 526 -> Room 527 ->
Room 531 -> Room 530 -> Room 529 -> Room 528 -> Room 532 -> Room 533 -> Room 535 -> Room 536 -
> Room 534 ->  Return to depot

          Scheduling completed for custodian 1.

          Completed work in 11.221 hours

Custodian 2:    Route for the day:

                    Room 500 -> Room 516 -> Room 510 -> Room 515 -> Room 511 -> Room 125 -> Room
512 -> Room 514 -> Room 501 -> Room 507 -> Room 505 -> Room 506 -> Room 504 -> Room 503 ->
Room 502 ->  Return to depot

          Scheduling completed for custodian 2.

          Completed work in 9.337 hours
```

Figure A.6: Two custodian routes for Monday



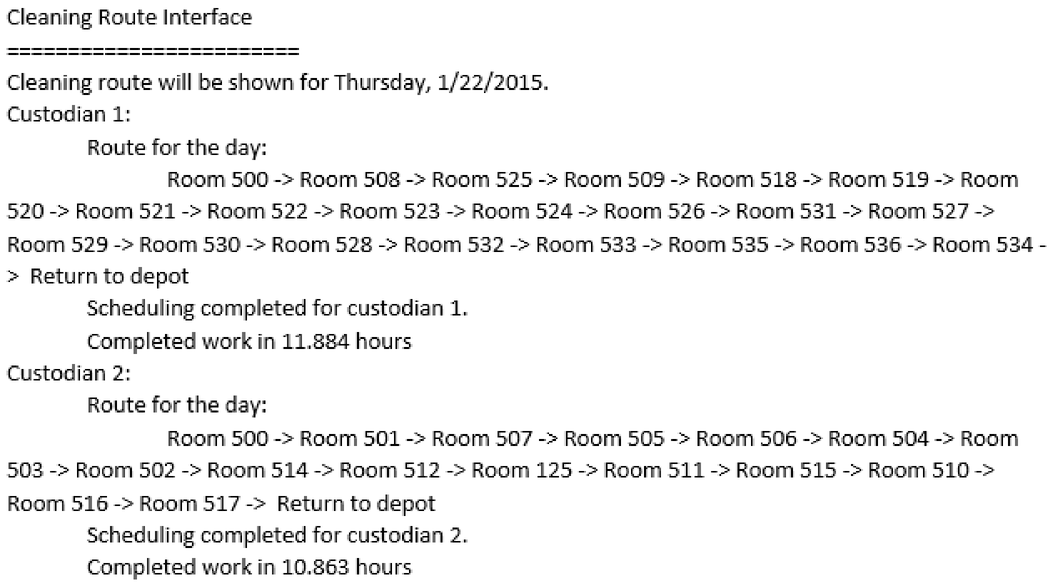
```
2 custodian routing Cleaning Route Interface
========================
Cleaning route will be shown for Tuesday, 1/20/2015.
Custodian 1:
          Route for the day:
                    Room 500 -> Room 508 -> Room 518 -> Room 509 -> Room 519 -> Room 520 -> Room
521 -> Room 522 -> Room 523 -> Room 524 -> Room 532 -> Room 527 -> Room 530 -> Room 529 ->
Room 531 -> Room 528 -> Room 533 -> Room 535 -> Room 536 -> Room 534 ->  Return to depot
          Scheduling completed for custodian 1.
          Completed work in 15.109 hours
Custodian 2:
          Route for the day:
                    Room 500 -> Room 516 -> Room 517 -> Room 510 -> Room 515 -> Room 511 -> Room
125 -> Room 512 -> Room 514 -> Room 501 -> Room 507 -> Room 505 -> Room 506 -> Room 504 ->
Room 503 -> Room 502 ->  Return to depot
          Scheduling completed for custodian 2.
          Completed work in 13.454 hours
```

Figure A.7: Two custodian routes for Tuesday



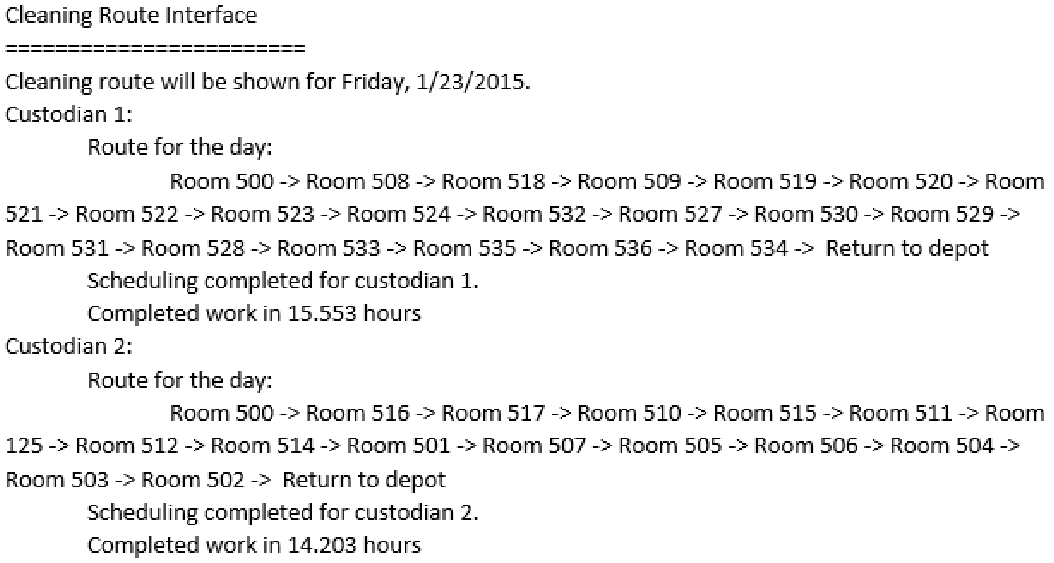
```
Cleaning Route Interface
========================
Cleaning route will be shown for Wednesday, 1/21/2015.
Custodian 1:
          Route for the day:
                    Room 500 -> Room 508 -> Room 526 -> Room 509 -> Room 525 -> Room 518 -> Room
519 -> Room 520 -> Room 521 -> Room 522 -> Room 523 -> Room 524 -> Room 532 -> Room 527 ->
Room 529 -> Room 530 -> Room 531 -> Room 528 -> Room 533 -> Room 535 -> Room 536 -> Room 534 -
>  Return to depot
          Scheduling completed for custodian 1.
          Completed work in 14.412 hours
Custodian 2:
          Route for the day:
                    Room 500 -> Room 501 -> Room 502 -> Room 503 -> Room 504 -> Room 506 -> Room
505 -> Room 507 -> Room 515 -> Room 510 -> Room 514 -> Room 512 -> Room 125 -> Room 511 ->
Room 517 -> Room 516 ->  Return to depot
          Scheduling completed for custodian 2.
          Completed work in 13.168 hours
```

Figure A.8: Two custodian routes for Wednesday



```
Cleaning Route Interface
=======================
Cleaning route will be shown for Thursday, 1/22/2015.
Custodian 1:
        Route for the day:
                Room 500 -> Room 508 -> Room 525 -> Room 509 -> Room 518 -> Room 519 -> Room
520 -> Room 521 -> Room 522 -> Room 523 -> Room 524 -> Room 526 -> Room 531 -> Room 527 ->
Room 529 -> Room 530 -> Room 528 -> Room 532 -> Room 533 -> Room 535 -> Room 536 -> Room 534 -
>  Return to depot
        Scheduling completed for custodian 1.
        Completed work in 11.884 hours
Custodian 2:
        Route for the day:
                Room 500 -> Room 501 -> Room 507 -> Room 505 -> Room 506 -> Room 504 -> Room
503 -> Room 502 -> Room 514 -> Room 512 -> Room 125 -> Room 511 -> Room 515 -> Room 510 ->
Room 516 -> Room 517 ->  Return to depot
        Scheduling completed for custodian 2.
        Completed work in 10.863 hours
```

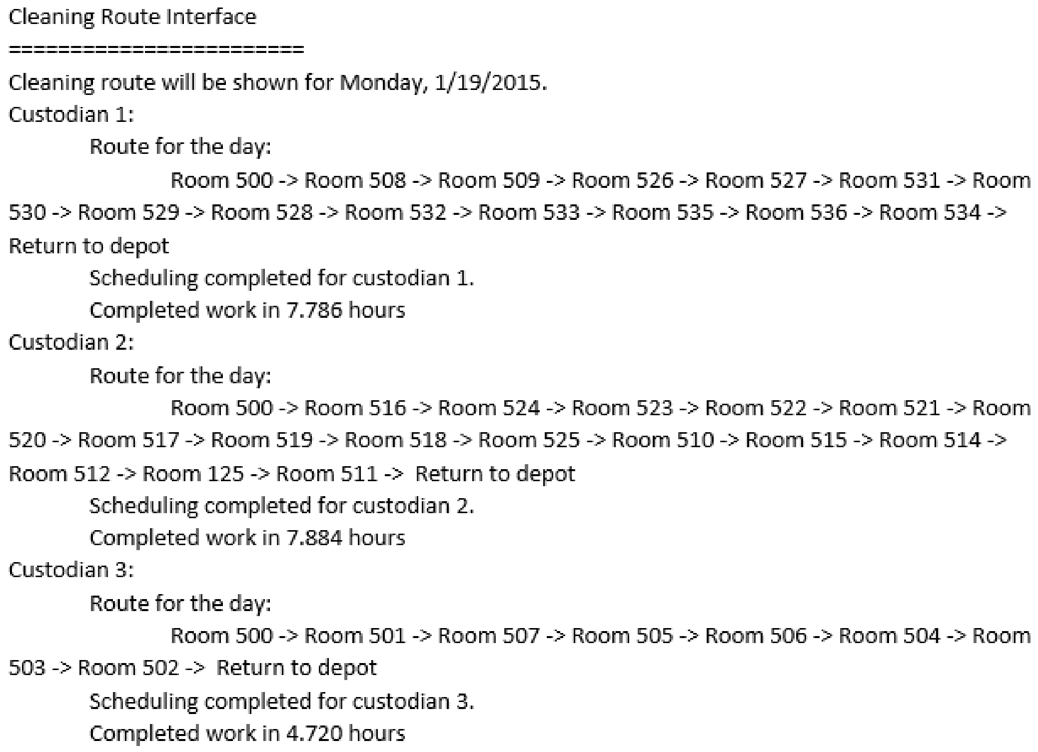
Figure A.9: Two custodian routes for Thursday



```
Cleaning Route Interface
=======================
Cleaning route will be shown for Friday, 1/23/2015.
Custodian 1:
        Route for the day:
                Room 500 -> Room 508 -> Room 518 -> Room 509 -> Room 519 -> Room 520 -> Room
521 -> Room 522 -> Room 523 -> Room 524 -> Room 532 -> Room 527 -> Room 530 -> Room 529 ->
Room 531 -> Room 528 -> Room 533 -> Room 535 -> Room 536 -> Room 534 ->  Return to depot
        Scheduling completed for custodian 1.
        Completed work in 15.553 hours
Custodian 2:
        Route for the day:
                Room 500 -> Room 516 -> Room 517 -> Room 510 -> Room 515 -> Room 511 -> Room
125 -> Room 512 -> Room 514 -> Room 501 -> Room 507 -> Room 505 -> Room 506 -> Room 504 ->
Room 503 -> Room 502 ->  Return to depot
        Scheduling completed for custodian 2.
        Completed work in 14.203 hours
```

Figure A.10: Two custodian routes for Friday

Three custodian routes for new system



```
Cleaning Route Interface
========================
Cleaning route will be shown for Monday, 1/19/2015.
Custodian 1:
        Route for the day:
                Room 500 -> Room 508 -> Room 509 -> Room 526 -> Room 527 -> Room 531 -> Room
530 -> Room 529 -> Room 528 -> Room 532 -> Room 533 -> Room 535 -> Room 536 -> Room 534 ->
Return to depot
        Scheduling completed for custodian 1.
        Completed work in 7.786 hours
Custodian 2:
        Route for the day:
                Room 500 -> Room 516 -> Room 524 -> Room 523 -> Room 522 -> Room 521 -> Room
520 -> Room 517 -> Room 519 -> Room 518 -> Room 525 -> Room 510 -> Room 515 -> Room 514 ->
Room 512 -> Room 125 -> Room 511 ->  Return to depot
        Scheduling completed for custodian 2.
        Completed work in 7.884 hours
Custodian 3:
        Route for the day:
                Room 500 -> Room 501 -> Room 507 -> Room 505 -> Room 506 -> Room 504 -> Room
503 -> Room 502 ->  Return to depot
        Scheduling completed for custodian 3.
        Completed work in 4.720 hours
```

Figure A.11: Three custodian routes for Monday

```
3 custodian routing
Cleaning Route Interface
=======================
Cleaning route will be shown for Tuesday, 1/20/2015.
Custodian 1:
        Route for the day:
                Room 500 -> Room 508 -> Room 509 -> Room 532 -> Room 527 -> Room 531 -> Room
529 -> Room 530 -> Room 528 -> Room 533 -> Room 535 -> Room 536 -> Room 534 ->Return to depot
        Scheduling completed for custodian 1.
        Completed work in 10.175 hours
Custodian 2:
        Route for the day:
                Room 500 -> Room 511 -> Room 125 -> Room 512 -> Room 514 -> Room 515 -> Room
520 -> Room 522 -> Room 521 -> Room 519 -> Room 517 -> Room 518 -> Room 516 ->Return to depot
        Scheduling completed for custodian 2.
        Completed work in 9.559 hours
Custodian 3:
        Route for the day:
                Room 500 -> Room 524 -> Room 523 -> Room 510 -> Room 501 -> Room 507 -> Room
505 -> Room 506 -> Room 504 -> Room 503 -> Room 502 ->Return to depot
        Scheduling completed for custodian 3.
        Completed work in 8.996 hours
```

Figure A.12: Three custodian routes for Tuesday

```
Cleaning Route Interface
=======================
Cleaning route will be shown for Wednesday, 1/21/2015.
Custodian 1:
        Route for the day:
                Room 500 -> Room 508 -> Room 526 -> Room 509 -> Room 532 -> Room 527 -> Room
531 -> Room 530 -> Room 529 -> Room 528 -> Room 533 -> Room 535 -> Room 536 -> Room 534 ->
Return to depot
        Scheduling completed for custodian 1.
        Completed work in 10.080 hours
Custodian 2:
        Route for the day:
                Room 500 -> Room 511 -> Room 125 -> Room 512 -> Room 514 -> Room 515 -> Room
519 -> Room 517 -> Room 518 -> Room 520 -> Room 521 -> Room 522 -> Room 524 -> Room 525 ->
Room 516 ->  Return to depot
        Scheduling completed for custodian 2.
        Completed work in 9.746 hours
Custodian 3:
        Route for the day:
                Room 500 -> Room 501 -> Room 502 -> Room 503 -> Room 504 -> Room 506 -> Room
505 -> Room 507 -> Room 523 -> Room 510 ->  Return to depot
        Scheduling completed for custodian 3.
        Completed work in 8.266 hours
```

Figure A.13: Three custodian routes for Wednesday

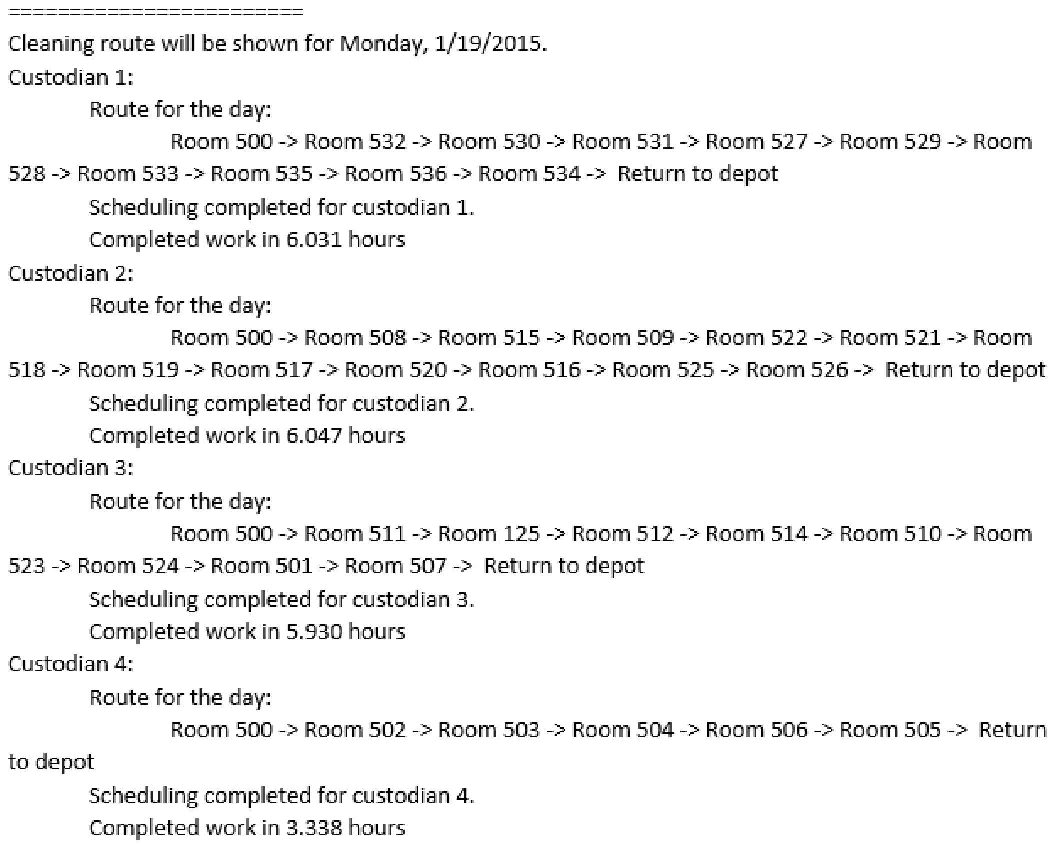
```
Cleaning Route Interface
========================
Cleaning route will be shown for Thursday, 1/22/2015.
Custodian 1:
        Route for the day:
                Room 500 -> Room 508 -> Room 509 -> Room 526 -> Room 531 -> Room 527 -> Room
530 -> Room 529 -> Room 528 -> Room 532 -> Room 533 -> Room 535 -> Room 536 -> Room 534 ->
Return to depot
        Scheduling completed for custodian 1.
        Completed work in 8.565 hours
Custodian 2:
        Route for the day:
                Room 500 -> Room 511 -> Room 125 -> Room 512 -> Room 514 -> Room 515 -> Room
518 -> Room 517 -> Room 519 -> Room 520 -> Room 521 -> Room 522 -> Room 524 -> Room 525 ->
Room 516 ->  Return to depot
        Scheduling completed for custodian 2.
        Completed work in 7.690 hours
Custodian 3:
        Route for the day:
                Room 500 -> Room 510 -> Room 523 -> Room 501 -> Room 507 -> Room 505 -> Room
506 -> Room 504 -> Room 503 -> Room 502 ->  Return to depot
        Scheduling completed for custodian 3.
        Completed work in 6.877 hours
```

Figure A.14: Three custodian routes for Thursday

```
Cleaning Route Interface
========================
Cleaning route will be shown for Friday, 1/23/2015.
Custodian 1:
        Route for the day:
                Room 500 -> Room 508 -> Room 509 -> Room 532 -> Room 527 -> Room 530 -> Room
529 -> Room 531 -> Room 528 -> Room 533 -> Room 535 -> Room 536 -> Room 534 ->  Return to depot
        Scheduling completed for custodian 1.
        Completed work in 10.368 hours
Custodian 2:
        Route for the day:
                Room 500 -> Room 511 -> Room 125 -> Room 512 -> Room 514 -> Room 515 -> Room
520 -> Room 522 -> Room 521 -> Room 519 -> Room 517 -> Room 518 -> Room 516 ->  Return to depot
        Scheduling completed for custodian 2.
        Completed work in 10.394 hours
Custodian 3:
        Route for the day:
                Room 500 -> Room 501 -> Room 507 -> Room 505 -> Room 506 -> Room 504 -> Room
503 -> Room 502 -> Room 510 -> Room 523 -> Room 524 ->  Return to depot
        Scheduling completed for custodian 3.
        Completed work in 9.161 hours
```

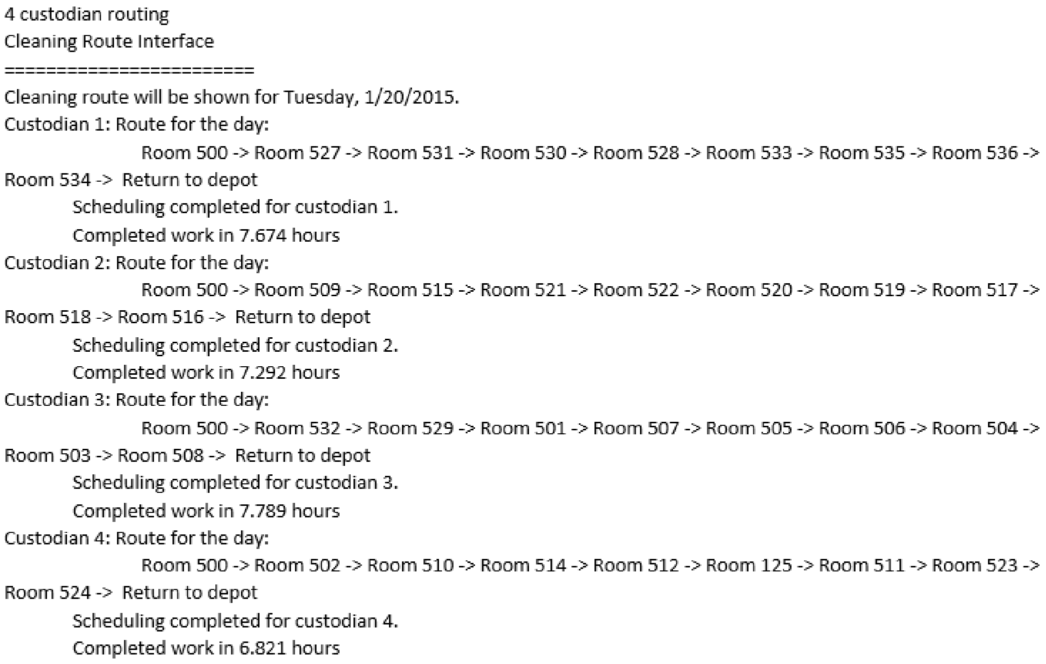
Figure A.15: Three custodian routes for Friday

Four custodian routes for new system



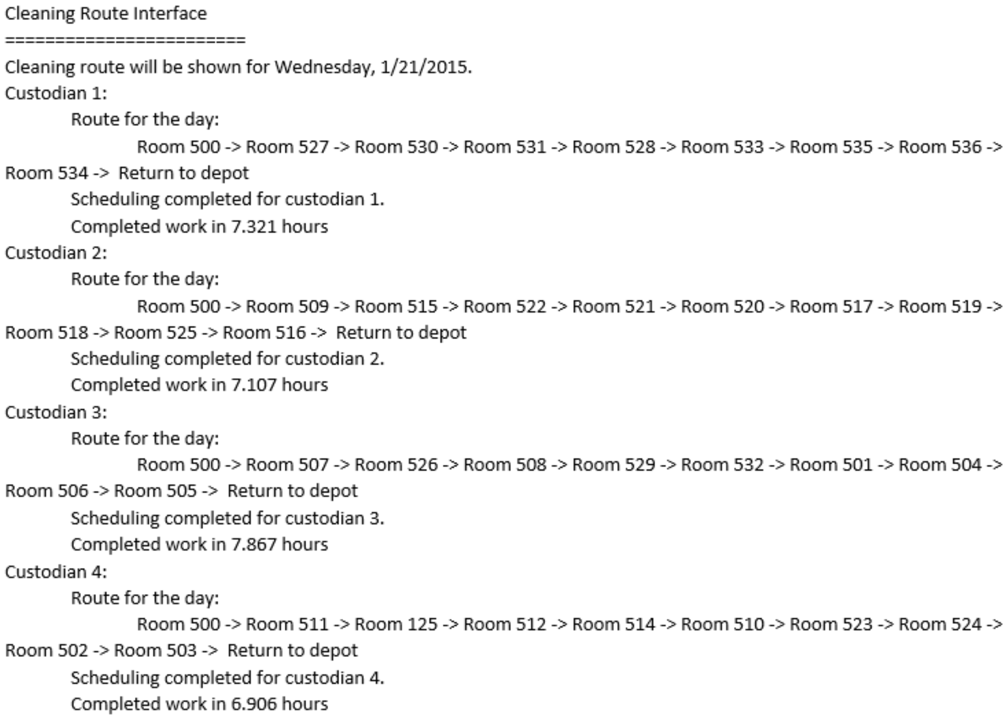
```
=======================
Cleaning route will be shown for Monday, 1/19/2015.
Custodian 1:
        Route for the day:
                Room 500 -> Room 532 -> Room 530 -> Room 531 -> Room 527 -> Room 529 -> Room
528 -> Room 533 -> Room 535 -> Room 536 -> Room 534 ->  Return to depot
        Scheduling completed for custodian 1.
        Completed work in 6.031 hours
Custodian 2:
        Route for the day:
                Room 500 -> Room 508 -> Room 515 -> Room 509 -> Room 522 -> Room 521 -> Room
518 -> Room 519 -> Room 517 -> Room 520 -> Room 516 -> Room 525 -> Room 526 ->  Return to depot
        Scheduling completed for custodian 2.
        Completed work in 6.047 hours
Custodian 3:
        Route for the day:
                Room 500 -> Room 511 -> Room 125 -> Room 512 -> Room 514 -> Room 510 -> Room
523 -> Room 524 -> Room 501 -> Room 507 ->  Return to depot
        Scheduling completed for custodian 3.
        Completed work in 5.930 hours
Custodian 4:
        Route for the day:
                Room 500 -> Room 502 -> Room 503 -> Room 504 -> Room 506 -> Room 505 ->  Return
to depot
        Scheduling completed for custodian 4.
        Completed work in 3.338 hours
```

Figure A.16: Four custodian routes for Monday



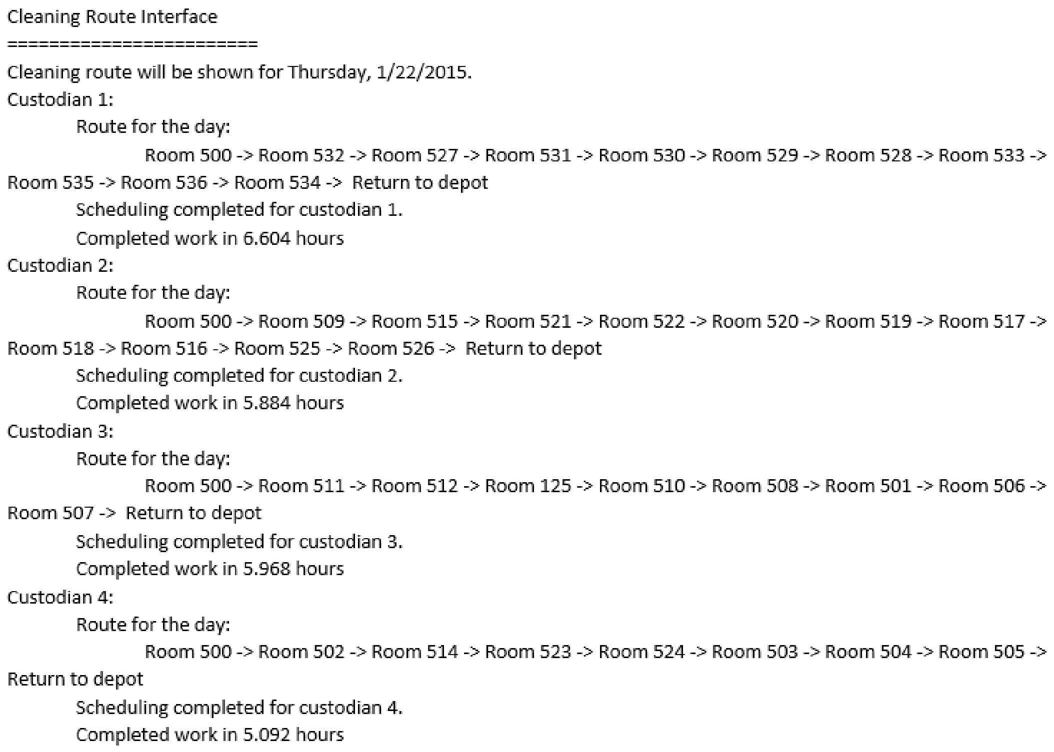
```
4 custodian routing
Cleaning Route Interface
========================
Cleaning route will be shown for Tuesday, 1/20/2015.
Custodian 1: Route for the day:
                    Room 500 -> Room 527 -> Room 531 -> Room 530 -> Room 528 -> Room 533 -> Room 535 -> Room 536 ->
Room 534 ->  Return to depot
          Scheduling completed for custodian 1.
          Completed work in 7.674 hours
Custodian 2: Route for the day:
                    Room 500 -> Room 509 -> Room 515 -> Room 521 -> Room 522 -> Room 520 -> Room 519 -> Room 517 ->
Room 518 -> Room 516 ->  Return to depot
          Scheduling completed for custodian 2.
          Completed work in 7.292 hours
Custodian 3: Route for the day:
                    Room 500 -> Room 532 -> Room 529 -> Room 501 -> Room 507 -> Room 505 -> Room 506 -> Room 504 ->
Room 503 -> Room 508 ->  Return to depot
          Scheduling completed for custodian 3.
          Completed work in 7.789 hours
Custodian 4: Route for the day:
                    Room 500 -> Room 502 -> Room 510 -> Room 514 -> Room 512 -> Room 125 -> Room 511 -> Room 523 ->
Room 524 ->  Return to depot
          Scheduling completed for custodian 4.
          Completed work in 6.821 hours
```

Figure A.17: Four custodian routes for Tuesday



```
Cleaning Route Interface
========================
Cleaning route will be shown for Wednesday, 1/21/2015.
Custodian 1:
          Route for the day:
                    Room 500 -> Room 527 -> Room 530 -> Room 531 -> Room 528 -> Room 533 -> Room 535 -> Room 536 ->
Room 534 ->  Return to depot
          Scheduling completed for custodian 1.
          Completed work in 7.321 hours
Custodian 2:
          Route for the day:
                    Room 500 -> Room 509 -> Room 515 -> Room 522 -> Room 521 -> Room 520 -> Room 517 -> Room 519 ->
Room 518 -> Room 525 -> Room 516 ->  Return to depot
          Scheduling completed for custodian 2.
          Completed work in 7.107 hours
Custodian 3:
          Route for the day:
                    Room 500 -> Room 507 -> Room 526 -> Room 508 -> Room 529 -> Room 532 -> Room 501 -> Room 504 ->
Room 506 -> Room 505 ->  Return to depot
          Scheduling completed for custodian 3.
          Completed work in 7.867 hours
Custodian 4:
          Route for the day:
                    Room 500 -> Room 511 -> Room 125 -> Room 512 -> Room 514 -> Room 510 -> Room 523 -> Room 524 ->
Room 502 -> Room 503 ->  Return to depot
          Scheduling completed for custodian 4.
          Completed work in 6.906 hours
```

Figure A.18: Four custodian routes for Wednesday



```
Cleaning Route Interface
=======================
Cleaning route will be shown for Thursday, 1/22/2015.
Custodian 1:
        Route for the day:
                Room 500 -> Room 532 -> Room 527 -> Room 531 -> Room 530 -> Room 529 -> Room 528 -> Room 533 ->
Room 535 -> Room 536 -> Room 534 ->  Return to depot
        Scheduling completed for custodian 1.
        Completed work in 6.604 hours
Custodian 2:
        Route for the day:
                Room 500 -> Room 509 -> Room 515 -> Room 521 -> Room 522 -> Room 520 -> Room 519 -> Room 517 ->
Room 518 -> Room 516 -> Room 525 -> Room 526 ->  Return to depot
        Scheduling completed for custodian 2.
        Completed work in 5.884 hours
Custodian 3:
        Route for the day:
                Room 500 -> Room 511 -> Room 512 -> Room 125 -> Room 510 -> Room 508 -> Room 501 -> Room 506 ->
Room 507 ->  Return to depot
        Scheduling completed for custodian 3.
        Completed work in 5.968 hours
Custodian 4:
        Route for the day:
                Room 500 -> Room 502 -> Room 514 -> Room 523 -> Room 524 -> Room 503 -> Room 504 -> Room 505 ->
Return to depot
        Scheduling completed for custodian 4.
        Completed work in 5.092 hours
```

Figure A.19: Four custodian routes for Thursday

```
Cleaning Route Interface
=======================
Cleaning route will be shown for Friday, 1/23/2015.
Custodian 1:
        Route for the day:
                Room 500 -> Room 527 -> Room 536 -> Room 534 -> Room 535 -> Room 533 -> Room 528 ->  Return to depot
        Scheduling completed for custodian 1.
        Completed work in 7.699 hours
Custodian 2:
  Route for the day:
                Room 500 -> Room 508 -> Room 518 -> Room 509 -> Room 531 -> Room 529 -> Room 530 -> Room 519 ->
Room 520 -> Room 521 -> Room 522 -> Room 523 -> Room 524 -> Room 532 ->  Return to depot
        Scheduling completed for custodian 2.
        Completed work in 8.678 hours
Custodian 3:
        Route for the day:
                Room 500 -> Room 510 -> Room 515 -> Room 514 -> Room 512 -> Room 125 -> Room 511 -> Room 517 ->
Room 516 ->  Return to depot
        Scheduling completed for custodian 3.
        Completed work in 7.905 hours
Custodian 4:
        Route for the day:
                Room 500 -> Room 501 -> Room 507 -> Room 505 -> Room 506 -> Room 504 -> Room 503 -> Room 502 ->
Return to depot
        Scheduling completed for custodian 4.
        Completed work in 6.108 hours
```

Figure A.20: Four custodian routes for Friday

Vita

Dinesh Reddy Patlolla was born on 23rd of April 1991 in Shankarpally, India. He spent his entire childhood and adolescent life in Hyderabad, India. He began his undergraduate degree at TKR College of Engineering and Technology affiliated to Jawaharlal Nehru Technological University where he graduated with distinction in 2008. Following the completion of his undergraduate degree, Dinesh started his Masters degree at University of Tennessee, Knoxville. During his Masters he has worked with various industries on projects. In 2015 he graduated with Master of Science in Industrial Engineering.