# Mathematical Modelling and Optimization for Efficient Parking Space Allocation - MISG 2024

Rojohana Molotsi — Keaoleboga Thubisi — Jaden Pieterse — Siyamthande Mnisi — Gilbert Ngobeni — Thembinkosi Zulu — Nompumelelo Ndlovu
**Matthews Sejeso — Luke O. Joel**

School of Computer Science and Applied Mathematics, University of the Witwatersrand

Graduate Modelling Camp
10 - 13 January 2024

# Background

- Many cities are experiencing rapid growth, leading to an increased number of vehicles in urban areas.
- The increasing gap between the number of vehicles and available parking spaces leads to parking scarcity.

# Problem Statement

The Parking Space Allocation (PSA) problem is centred on the efficient distribution of available parking spaces among users. Models for PSA can be categorized based on user class (single or multi-class) and the type of community (open or closed) they serve. The primary objectives are:

- Minimizing parking space misuse - ensuring optimal utilization of available spaces.
- Reduce the walking distance for each user from their parking spot to their destination.

There is a significant imbalance between the high demand for parking and the limited availability of spaces.

A balanced parking policy is essential: allocating permanent spaces to some, while others share or face variability in availability.

# The Objectives

Our goal is to develop an optimization model that maximizes the efficiency of existing parking spaces in a competitive, policy-driven university environment.

- Explore different models for formulating parking space allocation policies.
- Focus on minimizing parking space misuse to ensure optimal utilization.
- Prioritize reducing the walking distance for each user from their parking spot to their destination.

# Aims

Develop and solve the parking space allocation problem for a campus environment.

- Study the existing parking space allocation model.
- Formulate a mathematical optimization model for the problem.
- Develop an appropriate solution approach to the model.

# Variables and Parameters

- $D_{ij} \longrightarrow$ Distance from parking space $i$ to $j$
- $N_{jk} \longrightarrow$ Total number of individuals from building $j$ belonging to class $k$
- $M_i \longrightarrow$ The number of parking spaces in the parking lot $i$
- $I \longrightarrow$ Total number of parking space(s), $i = 1, ..., I$
- $J \longrightarrow$ Total number of building(s), $i = 1, ..., J$
- $K \longrightarrow$ Class and/or category associated to each individual, $k = 1, ..., K$
- $X_{ijk} \longrightarrow$ Individual from building $j$ belonging to class $k$ assigned to parking space $i$

# Objective fuction

$$\min z = \sum_{i}^{I} \sum_{j}^{J} \sum_{k}^{K} D_{ij} X_{ijk} \tag{1}$$

subject to

$$\sum_{i}^{I} X_{ijk} = N_{jk}, \quad \forall j, k \tag{2}$$

$$\sum_{j}^{J} \sum_{k}^{K} X_{ijk} \geq M_i, \quad \forall i \tag{3}$$

$X_{ijk} \geq 0,$ and integer.

# Objective and Constraints

- The objective function in Equation (1) minimizes the total distance walked by all the individuals from each parking space to their respective buildings.
- The constraint (2) ensures that the total number of individuals belonging to a class type in each parking space must equal the sum of the parking allocations in each parking space.
- In constraint (3), the sum of the allocations for all individuals must be greater than or equal to the number of parking spaces available.

# Major Difference of our Model

- The author in the thesis used the number of reserved and unreserved spaces in the parking lot. He was only trying to solve the problem of assigning users to already marked parking spaces. But our model consider a case where it is the mathematical model that will assign different classes of users to the parking lots.
- The author also considered only two classes (reserved and unreserved) excluding the handicapped population. But our model consider four classes - reserved normal, reserved handicapped, unreserved normal and unreserved handicapped.

# Data 1

- Data was sourced and modified to cater for more than two classes from the thesis "Model and Solutions to Campus Parking Space Allocation Problem" LO Joel (2013)

| A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Building 1 | Building 2 | Building 3 | Building 4 | Building 5 | Building 6 | Building 7 | Building 8 | Building 9 | Building 10 | Building 11 | Building 12 |
| Parking lot 1 | 79 | 108 | 182 | 57 | 143 | 97 | 183 | 48 | 168 | 94 | 189 | 136 |
| Parking lot 2 | 160 | 86 | 100 | 170 | 128 | 187 | 55 | 116 | 15 | 53 | 45 | 95 |
| Parking lot 3 | 26 | 196 | 56 | 125 | 85 | 24 | 67 | 159 | 80 | 83 | 26 | 123 |
| Parking lot 4 | 143 | 62 | 83 | 156 | 13 | 147 | 157 | 93 | 66 | 169 | 106 | 196 |
| Parking lot 5 | 183 | 146 | 152 | 191 | 80 | 97 | 184 | 161 | 77 | 81 | 143 | 58 |
| Parking lot 6 | 93 | 170 | 15 | 15 | 48 | 151 | 145 | 85 | 126 | 58 | 63 | 58 |
| Parking lot 7 | 141 | 43 | 176 | 183 | 11 | 26 | 78 | 43 | 11 | 109 | 60 | 61 |
| Parking lot 8 | 195 | 147 | 125 | 59 | 152 | 69 | 155 | 163 | 124 | 93 | 55 | 169 |

Figure: Distance from Buildings to Parking Lots

# Data 2

| A | B | C | D | E |
|---|---|---|---|---|
| | Class 1 | Class 2 | Class 3 | Class 4 |
| Building 1 | 25 | 2 | 61 | 4 |
| Building 2 | 9 | 0 | 28 | 0 |
| Building 3 | 26 | 1 | 61 | 5 |
| Building 4 | 48 | 3 | 94 | 8 |
| Building 5 | 49 | 1 | 90 | 10 |
| Building 6 | 41 | 2 | 83 | 6 |
| Building 7 | 25 | 0 | 58 | 4 |
| Building 8 | 27 | 3 | 61 | 9 |
| Building 9 | 26 | 2 | 62 | 5 |
| Building 10 | 32 | 1 | 70 | 5 |
| Building 11 | 27 | 2 | 64 | 4 |
| Building 12 | 36 | 3 | 75 | 9 |

Figure: Classes of Users for parking

# Data 3

| A | B |
|---|---|
| | Capacity of each parkking lot |
| Parking lot 1 | 72 |
| Parking lot 2 | 186 |
| Parking lot 3 | 110 |
| Parking lot 4 | 198 |
| Parking lot 5 | 147 |
| Parking lot 6 | 104 |
| Parking lot 7 | 160 |
| Parking lot 8 | 141 |

Figure: Capacity of each parking

# Exact Method

**Background:**

- The simplex algorithm is a popular algorithm in optimization techniques for linear programming.
- The algorithm is used to determine the feasible solution space of a given linear programming problem, which helps to identify the optimal point from the solution.
- The simplex method uses a search process to find the intersections of constraint equations, called corner points, to find optimal points through the boundary of feasible space.

**Optimization Process:**

- Standardize to LP problem, and generate an initial feasible solution, called a basis.
- Test the solution for optimality, if optimal, stop the process.
- If the solution is not optimal, generate an improved solution, if optimal, stop the process, otherwise improve the solution for optimal results.

# Exact Allocation for Class 1

| Allocation for Class1 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **P1** | **P2** | **P3** | **P4** | **P5** | **P6** | **P7** | **P8** |
| **Building1** | | | 25 | | | | | |
| **Buidling2** | | | | 9 | | | | |
| **Building3** | | | | | | 26 | | |
| **Building4** | | | | | | 4 | | 44 |
| **Building5** | | | | 49 | | | | |
| **Buidling6** | | | | | | | 41 | |
| **Building7** | | 25 | | | | | | |
| **Building8** | 13 | | | | | | 26 | |
| **Building9** | | | | | | | 26 | |
| **Building10** | | 32 | | | | | | |
| **Building11** | | | | | | | | 27 |
| **Building12** | | | | | 36 | | | |

Figure: The Allocation for Class 1

# Exact Allocation for Class 2

| Allocation for Class2 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **P1** | **P2** | **P3** | **P4** | **P5** | **P6** | **P7** | **P8** |
| **Building1** | | | 2 | | | | | |
| **Buidling2** | | | | | | | | |
| **Building3** | | | | | | 1 | | |
| **Building4** | | | | | | 3 | | |
| **Building5** | | | | 1 | | | | |
| **Buidling6** | | | | | | | 2 | |
| **Building7** | | | | | | | | |
| **Building8** | | | | | | | 3 | |
| **Building9** | | | | | | | 2 | |
| **Building10** | | 1 | | | | | | |
| **Building11** | | | | | | | | 2 |
| **Building12** | | | | | 3 | | | |

Figure: The Allocation for Class 2

# Exact Allocation for Class 3

| Allocation for Class3 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **P1** | **P2** | **P3** | **P4** | **P5** | **P6** | **P7** | **P8** |
| **Building1** | | | 61 | | | | | |
| **Buidling2** | | | | 28 | | | | |
| **Building3** | | | | | | 61 | | |
| **Building4** | | | | | | 94 | | |
| **Building5** | | | | 90 | | | | |
| **Buidling6** | | | 63 | | | | 20 | |
| **Building7** | | 58 | | | | | | |
| **Building8** | 50 | | | 11 | | | | |
| **Building9** | | 10 | | | | | 52 | |
| **Building10** | | 51 | | | 19 | | | |
| **Building11** | | | | | | | | 64 |
| **Building12** | | | | | 75 | | | |

Figure: The Allocation for Class 3

# Exact Allocation for Class 4

| Allocation for Class4 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **P1** | **P2** | **P3** | **P4** | **P5** | **P6** | **P7** | **P8** |
| **Building1** | | | 4 | | | | | |
| **Buidling2** | | | | | | | | |
| **Building3** | | | | | | 5 | | |
| **Building4** | | | | | | 8 | | |
| **Building5** | | | | 10 | | | | |
| **Buidling6** | | | 6 | | | | | |
| **Building7** | | 4 | | | | | | |
| **Building8** | 9 | | | | | | | |
| **Building9** | | 5 | | | | | | |
| **Building10** | | | | | 5 | | | |
| **Building11** | | | | | | | | 4 |
| **Building12** | | | | | 9 | | | |

Figure: The Allocation for Class 4

# Simulated Annealing

**Background:**

- Simulated Annealing (SA) is a probabilistic optimization algorithm inspired by the annealing process in metallurgy.
- This technique is loosely based on metallurgical annealing, in which metals are heated beyond their critical temperature and cooled according to a specific schedule until they reach their lowest energy state.
- In this manner, the material is treated in a controlled manner to obtain unique properties that are useful in specific applications.

**Optimization Process:**

- SA searches for the global optimum by exploring the solution space through a series of random moves.
- At each step, the algorithm accepts moves that improve the objective function, but also allows for accepting worse moves with a decreasing probability over time.

# Simulated Annealing

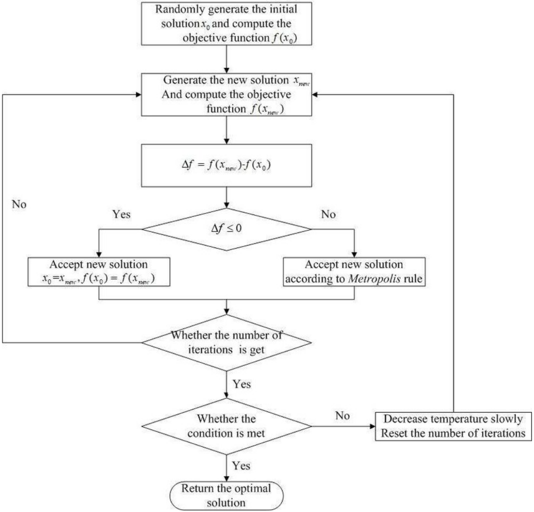


Figure: Flowchart explaining the structure of a Simulated Annealing

# Simulated Annealing

**Temperature Parameter:**

- SA uses a temperature parameter that controls the probability of accepting worse solutions.
- Initially high, the temperature decreases over time, allowing the algorithm to explore globally at the beginning and gradually refine the search locally.

**Algorithm Steps:**

- Initialization: Start with an initial solution.
- Iterative Exploration: Move to a neighboring solution based on a defined perturbation, and accept or reject the move based on probability.
- Temperature Cooling: Gradually decrease the temperature over iterations to reduce the probability of accepting worse solutions.
- Termination: Stop when a stopping criterion is met, such as a specified number of iterations or a target temperature.

# Simulated Annealing

**Advantages:**

- Versatility: Applicable to a wide range of optimization problems.
- Global Search: Can escape local optima due to the acceptance of worse solutions.
- No Derivatives: Suitable for problems where derivatives are unavailable or expensive to compute.

**Applications:**

- Traveling Salesman Problem, Job Scheduling, Network Design, and other combinatorial optimization tasks.
- Often used in machine learning for hyper-parameter tuning.

# Simulated Annealing Allocation for Class 1

| Allocation for Class1 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **P1** | **P2** | **P3** | **P4** | **P5** | **P6** | **P7** | **P8** |
| **Building1** | | | 21 | | | | | |
| **Buidling2** | | | | | | | 9 | |
| **Building3** | | | 18 | | | 16 | | |
| **Building4** | 20 | | | | | 26 | | 11 |
| **Building5** | | | | 19 | | 25 | 18 | |
| **Buidling6** | | | 14 | | | | 14 | |
| **Building7** | | 14 | | | | | | |
| **Building8** | 27 | | | | | | 21 | |
| **Building9** | | 18 | | | | | 20 | |
| **Building10** | | 16 | | | | 15 | | |
| **Building11** | | 13 | 16 | | | | | 16 |
| **Building12** | | | | | 22 | 11 | | |

Figure: Simulated Annealing Allocation for Class 1

# Simulated Annealing Allocation for Class 2

| Allocation for Class2 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **P1** | **P2** | **P3** | **P4** | **P5** | **P6** | **P7** | **P8** |
| **Building1** | | | 2 | | | | | |
| **Buidling2** | | | | | | | | |
| **Building3** | | | 1 | | | 1 | | |
| **Building4** | 3 | | | | | 3 | | 3 |
| **Building5** | | | | 1 | | 1 | 1 | |
| **Buidling6** | | | 2 | | | | 2 | |
| **Building7** | | | | | | | | |
| **Building8** | 3 | | | | | | 3 | |
| **Building9** | | 2 | | | | | 2 | |
| **Building10** | | 1 | | | | 1 | | |
| **Building11** | | 2 | 2 | | | | | 2 |
| **Building12** | | | | | 3 | 3 | | |

Figure: Simulated Annealing Allocation for Class 2

# Simulated Annealing Allocation for Class 3

| Allocation for Class3 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **P1** | **P2** | **P3** | **P4** | **P5** | **P6** | **P7** | **P8** |
| **Building1** | | | 14 | | | | | |
| **Buidling2** | | | | | | | 20 | |
| **Building3** | | | 18 | | | 19 | | |
| **Building4** | 68 | | | | | 48 | | 42 |
| **Building5** | | | | 77 | | 17 | 19 | |
| **Buidling6** | | | 20 | | | | 13 | |
| **Building7** | | 44 | | | | | | |
| **Building8** | 10 | | | | | | 14 | |
| **Building9** | | 24 | | | | | 14 | |
| **Building10** | | 26 | | | | 45 | | |
| **Building11** | | 17 | 20 | | | | | 18 |
| **Building12** | | | | | 25 | 19 | | |

Figure: Simulated Annealing Allocation for Class 3

# Simulated Annealing Allocation for Class 4

| Allocation for Class4 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **P1** | **P2** | **P3** | **P4** | **P5** | **P6** | **P7** | **P8** |
| **Building1** | | | 4 | | | | | |
| **Buidling2** | | | | | | | | |
| **Building3** | | | | | | 5 | | |
| **Building4** | | | | | | 8 | | |
| **Building5** | | | | 10 | | | | |
| **Buidling6** | | | 6 | | | | | |
| **Building7** | | 4 | | | | | | |
| **Building8** | 9 | | | | | | | |
| **Building9** | | 5 | | | | | | |
| **Building10** | | | | | 5 | | | |
| **Building11** | | | | | | | | 4 |
| **Building12** | | | | | 9 | | | |

Figure: Simulated Annealing Allocation for Class 4

# References


- **Joel, L. O. (2013)**. Model and solutions to campus parking space allocation problem (Masters Thesis).
- **Joel L. O., Adewumi A. O. and Sawyerr B. A. (2013)**. An Exact Solution for Allocating Car Parking Spaces on Campus. Proceedings of the International Science, Technology, Education, Arts, Management and Social Science (iSTEAMS) Research Nexus Conference. University of Ibadan, Ibadan, Nigeria, volume III, pp 463-472
- **Adewumi A. O., Joel L. O. and Sawyerr B. A. (2014)**. Campus Parking Space Allocation Distribution Using Pattern Search and Particle Swarm Pattern Search Algorithms. Proceedings of the Joint 7th international conference on Soft Computing and 15th International Symposium on Intelligent Systems (SCIS) and Advanced Intelligent Systems (ISIS), pp. 196-203, 2014.
- **Goyal, S. K., and Gomes, L. F. A. M. (1984)**. A model for allocating car parking spaces in universities. Transportation research Part B: methodological, 18(3), 267-269.

# Thank you!

## Questions