I. Headlines

1. Title: Winning the Votes: A Strategic Optimization Model for Election Campaigns
2. Team Members: Erica Cheng, Michael Lai, Yen-Pu Wang, You-Yi Liu
3. Tasks Assignment:
    - Model Formulation & Exact Optimization: Michael Lai, Yen-Pu Wang
    - Heuristic Algorithm Design & Implementation: Erica Cheng, You-Yi Liu
    - Report Writing & Documentation: Erica Cheng, Michael Lai, Yen-Pu Wang, You-Yi Liu

II. Introduction

1. Motivation

The recent presidential elections sparked our interest in understanding how candidates can strategically influence election outcomes. While political biases are deeply rooted, we believe candidates and parties can still optimize their campaigns to maximize votes. Our goal is to develop a model that integrates key campaign factors to formulate an optimal strategy for securing the highest possible voter support.

2. Previous Works

To optimize election campaigns, we must consider both where and when candidates should allocate their resources. This involves selecting strategic locations and determining efficient routes, making it a combinatorial optimization problem. One well-studied class of such problems is the Orienteering Problem (OP), which involves maximizing rewards by selecting and ordering locations to visit under limited resources.

The term ”Orienteering Problem (OP)” was first introduced by Golden et al. (1987). It is a combination of node selection and determining the shortest path between the selected nodes. The objective is to maximize the total score collected from visited (selected) nodes. In this problem, not all available nodes can be visited due to the limited time budget. Therefore, the OP can be seen as a combination between two classical combinatorial problems, the Knapsack Problem and the Travelling Salesman Problem (TSP) (Vansteenwegen et al., 2011). Since then, several variants of the OP have been introduced, such as the Team OP (TOP), the (Team) OP with Time Windows ((T)OPTW) and the (Team) OP with multiple visits.

Several methods have been proposed to solve the OP. Chao et al. (1996) introduced an effective heuristic algorithm that optimizes route selection while ensuring feasibility under time constraints. More recently, Kobeaga et al. (2024) developed an exact branch-and-cut algorithm, significantly improving the ability to solve large-scale OP instances. Both approaches assume that each location can only be visited once.

While standard OP assumes each location is visited at most once, real-world problems often require multiple visits to key locations. Some Team Orienteering Problem (TOP) studies allow different team members to revisit locations, which is similar to our setting where a single campaign team may repeatedly visit influential regions. For example, Jung et al. (2024) introduced a variant called Team Orienteering Problem with Possible Multiple Visits (TOP-PMV), where each point can be revisited multiple times with diminishing rewards. Their model was applied to military reconnaissance operations and election campaigns, making it particularly relevant to our study. Unlike traditional TOP, which assigns routes to multiple agents with each location visited no more than once, their model introduces a new decision variable, the number of times a location should be visited. This allows for strategic revisits while considering diminishing returns, optimizing both the routing and the frequency of visits.

Building on these works and real-world considerations, we model our problem as an extension of OP that allows strategic multiple visits, while keeping the formulation simple and focused. Unlike team-based extensions or models with diminishing returns, our approach prioritizes clarity and tractability, making it more suitable for our election campaign setting.

3. Intended Contributions

- Strategic Model Formulation: Develop a model that quantifies key factors (e.g., campaign budget, travel time, state visits, and voter conversion rates) to maximize votes from the Electoral College through the amount of votes gained in each state.
- Algorithmic Comparisons: Develop and evaluate different optimization algorithms to determine the most effective approach for campaign strategy planning.
- Practical Usability & Interpretability: Design the model and algorithms to be easy to use, ensuring practical applicability for campaign strategists without deep technical expertise.
- Computational Trade-off Analysis: Compare exact solvers and heuristic methods in terms of runtime and solution quality, identifying their strengths and limitations.

4. Organization of the Paper

The remainder of this report is structured as follows. Section III presents the problem statement, including the primal and dual formulations and the KKT conditions for optimality. Section IV outlines the solution approaches, comparing Gurobi (exact methods) with heuristics. Section V concludes with conjectured results, key insights, and future directions. Section VI lists references, highlighting key publications relevant to our problem formulation.

III. Statement of the Problem

1. Primal Formulation

The goal is to maximize electoral votes by strategically planning campaign efforts within specific constraints. The total cost of flights and speeches must not exceed the campaign budget. While the speech duration is fixed, the flight times vary depending on the distance between cities. The challenge lies in optimizing the schedule and resources to achieve the most possible electoral votes.

The formulation of our problem is shown below:

$$
\begin{cases}
\mathcal{S}: & \text{Set of all states in the USA} \\
n: & \text{Number of states that is going to be visited} \\
x_{i,s}: & \text{The decision of visiting state } s \text{ at time } i \text{ or not} \\
s \in \mathcal{S}: & \text{State in the USA} \\
P_s^+: & \text{Positive Population in the State } s \text{ that support the Party} \\
P_s^-: & \text{Negative Population in the State } s \text{ that support the Party} \\
E_s: & \text{The electoral votes of the State } s \\
c_s: & \text{Conversion rate of the negative population to the positive population} \\
 & \text{of each state after each visit} \\
T_{s1,s2}^f: & \text{The time of traveling from state } s1 \text{ to state } s2 \\
C_{s1,s2}^f: & \text{The cost of traveling from state } s1 \text{ to state } s2 \\
T^{sp}: & \text{The time for each speech} \\
C^{sp}: & \text{The cost for each speech} \\
T: & \text{The total time before the election} \\
B: & \text{The budget}
\end{cases}
$$

$$
\begin{aligned}
\max_{x_1 \dots x_n} & \sum_{s \in \mathcal{S}} E_s \times \mathbb{1}(P_s^+ - P_s^-(1 - 2c_s \sum_{i=1}^{n} x_{i,s})) \\
\text{s.t. } & x_{i,s} \in \{0, 1\}, \forall i, s \\
& \sum_{s \in \mathcal{S}} x_{i,s} \leq 1, \forall i \\
& \sum_{i=1}^{n} \sum_{s \in \mathcal{S}} \sum_{s' \in \mathcal{S}} x_{i-1,s} x_{i,s'} T_{s,s'}^f + \sum_{i=1}^{n} \sum_{s \in \mathcal{S}} x_{i,s} T^{sp} \leq T \\
& \sum_{i=1}^{n} \sum_{s \in \mathcal{S}} \sum_{s' \in \mathcal{S}} x_{i-1,s} x_{i,s'} C_{s,s'}^f + \sum_{i=1}^{n} \sum_{s \in \mathcal{S}} x_{i,s} C^{sp} \leq B
\end{aligned}
$$

2. Dual Formulation

First, we get the Lagrangian of the objective function:

$$
\begin{aligned}
\mathcal{L}(x, \lambda, \mu, \nu, \gamma_T, \gamma_B) = & \sum_{s \in \mathcal{S}} E_s \, f_s\Big(\sum_{i=1}^{n} x_{i,s}\Big) \\
& + \sum_{i=1}^{n} \sum_{s \in \mathcal{S}} \lambda_{i,s} x_{i,s} \\
& + \sum_{i=1}^{n} \sum_{s \in \mathcal{S}} \mu_{i,s} (1 - x_{i,s}) \\
& + \sum_{i=1}^{n} \nu_i \Big(1 - \sum_{s \in \mathcal{S}} x_{i,s}\Big) \\
& + \gamma_T (T - g_T(x)) + \gamma_B (B - g_B(x)) \\
= & \sum_{s \in \mathcal{S}} \Big\{ \frac{E_s}{1 + \exp(-\alpha(P_s^+ - P_s^-(1 - 2c_s \sum_{i=1}^{n} x_{i,s})))} \\
& + \sum_{i=1}^{n} x_{i,s} \Big[\lambda_{i,s} - \mu_{i,s} - \nu_i - \gamma_T T^{sp} - \gamma_B C^{sp} - \sum_{s' \in \mathcal{S}} x_{i-1,s'} (\gamma_T T_{s,s'}^f + \gamma_B C_{s,s'}^f)\Big] \Big\} \\
& + \sum_{i=1}^{n} \Big(\sum_{s \in \mathcal{S}} \mu_{i,s} + \nu_i\Big) + \gamma_T T + \gamma_B B
\end{aligned}
$$

To get the local maximum, we get the partial derivative:

$$\frac{\partial \mathcal{L}}{\partial x_{i,s}} = E_s \frac{\partial f_s}{\partial x_{i,s}} + \lambda_{i,s} - \mu_{i,s} - \nu_i - \gamma_T \frac{\partial g_T}{\partial x_{i,s}} - \gamma_B \frac{\partial g_B}{\partial x_{i,s}} = 0, \quad \forall i = 1 \ldots n, s \in \mathcal{S}$$

$$\frac{\partial f_s}{\partial x_{i,s}} = \frac{\partial f_s}{\partial z}\frac{\partial z}{\partial x_{i,s}} = \frac{\partial f_s}{\partial z} = \frac{2\alpha P_s^- c_s \exp(-\alpha(P_s^+ - P_s^-(1-2c_s\sum_{i=1}^n x_{i,s})))}{(1+\exp(-\alpha(P_s^+ - P_s^-(1-2c_s\sum_{i=1}^n x_{i,s})))^2}$$

$$\frac{\partial g_T}{\partial x_{i,s}} = \begin{cases} T^{sp} + \sum_{s'\in\mathcal{S}}(x_{i-1,s'} + x_{i+1,s'})T^f_{s',s} & ,i \neq n \\ T^{sp} + \sum_{s'\in\mathcal{S}} x_{i-1,s'}T^f_{s',s} & ,i = n \end{cases}$$

$$\frac{\partial g_B}{\partial x_{i,s}} = \begin{cases} C^{sp} + \sum_{s'\in\mathcal{S}}(x_{i-1,s'} + x_{i+1,s'})C^f_{s',s} & ,i \neq n \\ C^{sp} + \sum_{s'\in\mathcal{S}} x_{i-1,s'}C^f_{s',s} & ,i = n \end{cases}$$

Finally, we get:

If $i \neq n$:

$$\frac{2E_s\alpha P_s^- c_s \exp(-\alpha(P_s^+ - P_s^-(1-2c_s\sum_{j=1}^n x_{j,s})))}{(1+\exp(-\alpha(P_s^+ - P_s^-(1-2c_s\sum_{j=1}^n x_{j,s})))^2} + \lambda_{i,s} - \mu_{i,s} - \nu_i$$
$$-\gamma_T(T^{sp} + \sum_{s'\in\mathcal{S}}(x_{i-1,s'} + x_{i+1,s'})T^f_{s',s})$$
$$-\gamma_B(C^{sp} + \sum_{s'\in\mathcal{S}}(x_{i-1,s'} + x_{i+1,s'})C^f_{s',s}) = 0$$

If $i = n$:

$$\frac{2E_s\alpha P_s^- c_s \exp(-\alpha(P_s^+ - P_s^-(1-2c_s\sum_{j=1}^n x_{j,s})))}{(1+\exp(-\alpha(P_s^+ - P_s^-(1-2c_s\sum_{j=1}^n x_{j,s})))^2} + \lambda_{i,s} - \mu_{i,s} - \nu_i$$
$$-\gamma_T(T^{sp} + \sum_{s'\in\mathcal{S}} x_{i-1,s'}T^f_{s',s}) - \gamma_B(C^{sp} + \sum_{s'\in\mathcal{S}} x_{i-1,s'}C^f_{s',s}) = 0$$

We can simplify the Lagrangian formulation:

$$\mathcal{L}(x,\lambda,\mu,\nu,\gamma_T,\gamma_B) = \sum_{s\in\mathcal{S}}\Bigg\{\frac{E_s(1 - \frac{2\alpha P_s^- c_s \exp(-\alpha(P_s^+ - P_s^-(1-2c_s\sum_{j=1}^n x_{j,s})))}{1+\exp(-\alpha(P_s^+ - P_s^-(1-2c_s\sum_{j=1}^n x_{j,s})))}\sum_{i=1}^n x_{i,s})}{1+\exp(-\alpha(P_s^+ - P_s^-(1-2c_s\sum_{j=1}^n x_{j,s})))}$$
$$-\sum_{i=1}^{n-1}\sum_{s'\in\mathcal{S}} x_{i+1,s'}(\gamma_T T^f_{s,s'} + \gamma_B C^f_{s,s'})\Bigg\} + \sum_{i=1}^n\Big(\sum_{s\in\mathcal{S}}\mu_{i,s} + \nu_i\Big) + \gamma_T T + \gamma_B B$$

The final dual problem:

$$\begin{aligned} \min_{\lambda,\mu,\nu,\gamma_T,\gamma_B} & \sum_{i=1}^n\Big(\sum_{s\in\mathcal{S}}\mu_{i,s} + \nu_i\Big) + \gamma_T T + \gamma_B B + \phi(\lambda,\mu,\nu,\gamma_T,\gamma_B) \\ \text{s.t. } & \lambda_{i,s} \geq 0, \quad \forall i, s, \\ & \mu_{i,s} \geq 0, \quad \forall i, s, \\ & \nu_i \geq 0, \quad \forall i, \\ & \gamma_T \geq 0, \\ & \gamma_B \geq 0. \end{aligned}$$

where

$$\phi_s(\lambda, \mu, \nu, \gamma_T, \gamma_B) \triangleq \sup_{\{x_{1,s},\ldots,x_{n,s}\}\in[0,1]^n} \frac{E_s\left(1 - \frac{2\alpha P_s^- c_s \exp(-\alpha(P_s^+ - P_s^-(1-2c_s\sum_{j=1}^n x_{j,s})))}{1+\exp(-\alpha(P_s^+ - P_s^-(1-2c_s\sum_{j=1}^n x_{j,s})))}\sum_{i=1}^n x_{i,s}\right)}{1+\exp(-\alpha(P_s^+ - P_s^-(1-2c_s\sum_{j=1}^n x_{j,s})))} - \sum_{i=1}^{n-1}\sum_{s'\in\mathcal{S}} x_{i+1,s'}(\gamma_T T_{s,s'}^f + \gamma_B C_{s,s'}^f)$$

3. KKT Conditions

- Primal Feasibility:

$$\begin{aligned} 0 \le x_{i,s} &\le 1, \quad \forall i, s, \\ \sum_{s\in\mathcal{S}} x_{i,s} &\le 1, \quad \forall i, \\ g_T(x) &\le T, \\ g_B(x) &\le B. \end{aligned}$$

- Dual Feasibility:

$$\begin{aligned} \lambda_{i,s} &\ge 0, \quad \forall i, s, \\ \mu_{i,s} &\ge 0, \quad \forall i, s, \\ \nu_i &\ge 0, \quad \forall i, \\ \gamma_T &\ge 0, \\ \gamma_B &\ge 0. \end{aligned}$$

- Complementary Slackness:

$$\begin{aligned} \lambda_{i,s} x_{i,s} &= 0, \quad \forall i, s, \\ \mu_{i,s}(1 - x_{i,s}) &= 0, \quad \forall i, s, \\ \nu_i\Big(1 - \sum_{s\in\mathcal{S}} x_{i,s}\Big) &= 0, \quad \forall i, \\ \gamma_T(T - g_T(x)) &= 0, \\ \gamma_B(B - g_B(x)) &= 0. \end{aligned}$$

- Stationarity:

If $i \neq n$:

$$\frac{2E_s\alpha P_s^- c_s \exp(-\alpha(P_s^+ - P_s^-(1-2c_s\sum_{j=1}^n x_{j,s})))}{(1+\exp(-\alpha(P_s^+ - P_s^-(1-2c_s\sum_{j=1}^n x_{j,s})))^2} + \lambda_{i,s} - \mu_{i,s} - \nu_i$$
$$-\gamma_T(T^{sp} + \sum_{s'\in\mathcal{S}}(x_{i-1,s'} + x_{i+1,s'})T^f_{s',s})$$
$$-\gamma_B(C^{sp} + \sum_{s'\in\mathcal{S}}(x_{i-1,s'} + x_{i+1,s'})C^f_{s',s}) = 0$$

If $i = n$:

$$\frac{2E_s\alpha P_s^- c_s \exp(-\alpha(P_s^+ - P_s^-(1-2c_s\sum_{j=1}^n x_{j,s})))}{(1+\exp(-\alpha(P_s^+ - P_s^-(1-2c_s\sum_{j=1}^n x_{j,s})))^2} + \lambda_{i,s} - \mu_{i,s} - \nu_i$$
$$-\gamma_T(T^{sp} + \sum_{s'\in\mathcal{S}} x_{i-1,s'}T^f_{s',s}) - \gamma_B(C^{sp} + \sum_{s'\in\mathcal{S}} x_{i-1,s'}C^f_{s',s}) = 0$$

IV. Intended Approaches

1. Sample Dataset

Our analysis utilizes a comprehensive dataset encompassing all 50 U.S. states and the District of Columbia. For each jurisdiction, we collected:

- **Electoral College Votes**: The constitutionally allocated number of electoral votes per state (and D.C.).
- **Popular Vote Distribution**: Vote totals for candidates from the two major political parties based on the 2024 U.S. Presidential Election results.
- **Inter-State Travel Times**: Estimated flight durations between all pairs of states. These values were calculated by:
  - Selecting each state's most populous city as its representative point.
  - Computing pairwise distances using geographic coordinates (latitude and longitude).
  - Converting distances to travel times based on a standard private jet cruising speed.

This dataset serves as the foundation for our campaign resource optimization model, which aims to maximize electoral impact by strategically allocating time and budget across states.

Additional Variables

To support the optimization model, we define the following parameters:

- **speech_num**: The total number of campaign speeches allowed within the campaign period.
- **speech_time**: The duration (in hours) of each speech.
- **speech_cost**: The financial cost of organizing each speech.
- **days**: The number of remaining days in the campaign.
- **total_time**: The maximum time available for the campaign.
- **total_budget**: The maximum budget available for the campaign.
- **flight_hour_cost**: The cost per hour of flight time between cities.

- **conversion_rate**: The percentage of negative voters converted to positive voters after each speech.
- **start_state**: The initial state where the campaign begins.

Sample Case

For the example scenario below, we use the following parameter values:

- **speech_num** = 10
- **speech_time** = 3 hours
- **speech_cost** = $300,000
- **days** = 14
- **total_time** = 16 * days (16 operational hours per day)
- **total_budget** = $5,000,000
- **flight_hour_cost** = $15,000
- **conversion_rate** = 0.02
- **start_state** = "DC"

2. Solver Approach

With nonlinearity in the original objective (all or nothing electoral votes based on whether you get more positive than negative votes) and products of binary decision variables in constraints, our problem formulation is a **Mixed-Integer Nonlinear Programming (MINLP) problem.** MILP problems are computationally challenging to solve even with solvers, as they have nonconvexity and an exponential search space. However, we are able to transform the problem into a **Mixed-Integer Programming (MILP) problem** using conventional linearization techniques, specifically multiplication of binary variables linearization and Big-M method.

The transformed problem is then solved using a commercial solver, Gurobi.

3. Heuristic Algorithm Approach

To efficiently determine the optimal campaign route, we implemented a heuristic algorithm called **Greedy Feasible Route Selection**. This method aims to maximize the electoral votes gained while ensuring feasibility under time, budget, and city visit constraints. The algorithm greedily selects the next city that provides the highest reward-to-cost ratio, meaning that each unit of cost yields the most electoral votes.

Computing the Reward-to-Cost Ratio

Each city is evaluated based on the reward-to-cost ratio, defined as:

$$\text{Reward-to-Cost Ratio} = \frac{\text{Reward}}{\text{Cost}}$$

Reward is the expected electoral votes gained by visiting a city, defined by:

$$\text{Reward} = \text{Probability of Winning} \times \text{Electoral Votes of the State}$$

- The probability of winning a state's electoral votes is based on the vote difference after visiting a city:

$$\text{Probability of Winning} = \begin{cases} 1 & \text{if Vote Difference} > 0 \\ \frac{1}{1+e^{-0.0001\times\text{Vote Difference}}} & \text{if Vote Difference} \leq 0 \end{cases}$$

  - If the vote difference is positive, the probability of winning is 100%.
  - If the vote difference is negative, the probability is estimated using a sigmoid function.
- The vote difference is defined as:

$$\text{Vote Difference} = \text{Positive Votes} - \text{Negative Votes} \times (1 - 2 \times \text{Conversion Rate})$$

Cost is the weighted sum of time and budget costs.

- Time Cost

$$\text{Time Cost} = \frac{\text{Travel Time from Previous City} + \text{Speech Time}}{\text{Total Time Limit}}$$

- Budget Cost

$$\text{Budget Cost} = \frac{\text{Travel Cost from Previous City} + \text{Speech Cost}}{\text{Total Budget Limit}}$$

We weighted the time and budget costs because their scales differ: time is measured in hours, while the budget is in monetary units. Weighting them ensures both components contribute appropriately to the total cost without one dominating the other due to its larger scale.

Algorithm Logic

1. Select the next city iteratively until no feasible options remain:
   - Consider only states where our popular votes are currently lower than the opponent's.
   - Choose the next city with the highest reward-to-cost ratio, ensuring it does not violate any constraints (time, budget, or number of city visits).
   - Update the campaign state.
     - Deduct the used time, budget, and number of visits from the respective constraints.
     - Update the current electoral votes (objective value).
     - Increase positive votes and decrease negative votes due to the speech, influenced by the conversion rate.
     - Add the newly visited city to the campaign route.
2. Include additional electoral votes.
   - After the route is finalized, electoral votes are also counted from states we did not visit but where our positive votes still exceed the opponent's.

Complexity Analysis

The time complexity of this algorithm is $O(n \cdot k)$, where:

- n is the larger value between the number of cities and the number of states.
- k is the number of visited cities in the final route.

V. Results & Conclusion

1. Conjectured Results

The results from the solver and our heuristic algorithm are shown in the table below:

Results \ Method	Solver	Heuristic
Route	[PA, AZ, AZ, NC, GA, GA, NC, AZ, MI, WI]	[PA, MI, WI, GA, GA, NV, NV, AK, AK, AK]
Electoral Votes Obtained	313	292
Time Spent by Solution (s)	0.97	0.0007

Solver Results

- The solver's resulting route visits only swing states, which makes sense as swing states are crucial for an election campaign. This proves that using the solver can yield a realistic and effective campaign strategy.
- The solver yields 313 electoral votes, which is a strong result, indicating a winning strategy.
- The time spent by the solver to find the solution is 0.97 seconds, which is fast and acceptable for solving problems.

Heuristic Algorithm Results

- The heuristic algorithm, while not providing the exact same route, also visits a similar set of states, focusing on swing states like the solver's solution.
- With 292 electoral votes, the heuristic solution is greater than the 226 electoral votes (which represents the number of electoral votes Democrats have in 2024) and would be sufficient to win in a two-party system, as: $538 - 292 = 246 < 292$. Therefore, the heuristic algorithm proves to be a helpful strategic planning tool for election campaigns.

Results Comparison

- The optimality gap between the solver's result and the heuristic solution is calculated as:

$$\text{Optimality Gap} = \frac{313 - 292}{313} = 0.067$$

  - This small optimality gap of 6.7% suggests that the heuristic algorithm performs quite well and provides a solution close to the optimal one.
- The time spent by the heuristic algorithm to obtain the solution is 0.0007 seconds, which is significantly faster than the solver's 0.97 seconds. This indicates that the heuristic method can be more time-efficient in real-world scenarios, especially when dealing with larger datasets or more complex problems.

2. Conclusion

In this report, we demonstrate that a carefully optimized campaign strategy can yield significant electoral votes gains, even when balancing between budget and time constraints. Our analysis using both an exact solver and a heuristic algorithm reveal that strategic visits to swing states can produce a winning outcome, with the solver achieving slightly higher electoral votes and the heuristic method offering computational efficiency. These results show not only our optimization model as a valuable decision-support tool for campaign planning, but the logic behind real-world campaign organizing strategies.

Looking forward, the model's framework provides a solid foundation for further refinement, such as incorporating multiple cities per state, opponent dynamics, and additional real-world logistical constraints, which will be elaborated in the next section. By extending the model in these ways, future research can offer even more realistic and adaptive strategies for election campaigns to maximize voter conversion and secure electoral victory.

3. Possible Future Works

First, the current model assumes one city per state, simplifying planning but limiting precision. Expanding to multiple cities per state would provide a more realistic representation of campaign influence. However, solving for more locations increases computational complexity. Future work could address this using heuristics or by relaxing decision variables (e.g., allowing continuous x values) to identify high-priority cities without requiring exact itineraries, providing strategic guidance for campaign planning.

Second, the current model only optimizes a single party's strategy, ignoring potential opponents' influence and competitive states. A more realistic approach would incorporate game-theoretic elements, treating elections as dynamic competitions where both parties adjust their strategies in response to each other. This would ensure the model not only maximizes direct voter gains but also counters the opponent's campaign efforts.

Additionally, future models should incorporate practical constraints to better reflect real-world campaign logistics. These could include numbers of visited cities in a single day, workday time constraints, and campaign costs such as rally expenses and media coverage, ensuring feasibility and resource-efficient planning.

Voter behavior modeling can also be improved to account for regional differences in conversion rates and diminishing returns from repeated visits. Future work could also introduce dynamic strategy adjustments, allowing the campaign to respond in real time to voter sentiment shifts and opponent activity, making the model more adaptive to evolving election conditions.

Lastly, since our heuristic algorithm is greedy and considers the problem locally, it might not perform well as the number of cities increases, or if the problem becomes more complicated. One possible improvement would be to integrate techniques like simulated annealing to allow the algorithm to escape from local optima and explore a broader solution space, potentially improving the results in more complex scenarios.

VI. References

Golden, B. L., Levy, L., & Vohra, R. (1987). The orienteering problem. Naval Research Logistics, 34(3), 307–318.
https://doi.org/10.1002/1520-6750(198706)34:3%3C307::aid-nav3220340302%3E3.0.co;2-d

Vansteenwegen, P., Souffriau, W., & Oudheusden, D. V. (2011). The orienteering problem: A survey. European Journal of Operational Research, 209(1), 1–10. https://doi.org/10.1016/j.ejor.2010.03.045

Chao, I-Ming., Golden, B. L., & Wasil, E. A. (1996). A fast and effective heuristic for the orienteering problem. European Journal of Operational Research, 88(3), 475–489. https://doi.org/10.1016/0377-2217(95)00035-6

Kobeaga, G., Rojas-Delgado, J., Merino, M., & Lozano, J. A. (2024). A revisited branch-and-cut algorithm for large-scale orienteering problems. European Journal of Operational Research, 313(1), 44–68. https://doi.org/10.1016/j.ejor.2023.07.034

Jung, H.-B., Kim, H.-I., & Lee, D.-H. (2024). Team orienteering with possible multiple visits: Mathematical model and solution algorithms. Computers & Industrial Engineering, 190, 110097. https://doi.org/10.1016/j.cie.2024.110097