VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
INSTITUTE OF COMPUTER SCIENCE
DEPARTMENT OF COMPUTATIONAL AND DATA MODELING

Computer Modeling Final Master Thesis

Facility location under uncertainties in customer behaviour

Objektų vietų optimizavimas esant neapibrėžtumams klientų elgsenoje

Done by:
Rokas Mizeikis signature

Supervisor:
Doc. dr. Algirdas Lančinskas

Vilnius
2026

Contents

Abstract

Facility location problems involve strategic decisions with long term economic and operational consequences, particularly in competitive markets where customer behavior plays a central role. In competitive facility location, optimal placement depends not only on spatial demand and competitor locations, but also on assumptions regarding how customers choose among alternatives. Small changes in these assumptions can lead to significantly different solutions, making models optimized under a single behavioral specification fragile. This thesis investigates the application of deep reinforcement learning to competitive facility location under uncertainty in customer behavior. The problem is studied from the perspective of an entering firm that sequentially places new facilities in a market with existing competitors, aiming to maximize captured market share. Customer choice is modeled using Pareto-Huff filter, while demand allocation follows either binary or proportional rules, representing two behavioral assumptions. A deep Q-learning framework is developed and trained across a diverse set of simulated instances. Robustness is evaluated empirically by assessing performance under both customer behavior types and selecting a representative policy using Manhattan distance to complete market share capture. Experimental results show that the learned policies achieve competitive market share outcomes while offering substantial computational advantages over exhaustive enumeration. Although performance varies across problem settings and does not consistently reach optimality, the results indicate that reinforcement learning provides a promising and scalable foundation for competitive facility location under behavioral uncertainty.

Santrauka

Objektų vietų optimizavimas esant neapibrėžtumams klientų elgsenoje

Objektų vietų optimizavimo uždaviniai apima strateginius sprendimus, turinčius ilgalaikes ekonomines ir verslo veiklos pasekmes, ypač konkurencingose rinkose, kuriose klientų elgsena atlieka esminį vaidmenį. Konkurencingų objektų vietų parinkimo modeliuose optimalus išdėstymas priklauso ne tik nuo paklausos pasiskirstymo erdvėje ir konkurentų vietų, bet ir nuo klientų elgesnos prielaidų. Nedideli šių prielaidų pokyčiai gali lemti reikšmingai skirtingus sprendimus, todėl modeliai, optimizuoti pagal vieną elgsenos specifikaciją, dažnai yra trapūs. Šiame darbe nagrinėjamas giliojo skatinamojo mokymosi taikymas konkurenciniam objektų vietų optimizavimo uždaviniui, esant neapibrėžtai klientų elgsenai. Uždavinys analizuojamas iš į rinką įžengiančios įmonės perspektyvos, kuri nuosekliai parenka vietas naujiems objektams rinkoje su jau egzistuojančiais konkurentais, siekiant užimti maksimalią rinkos dalį. Klientų pasirinkimas modeliuojamas taikant Pareto–Huff filtrą, o paklausos paskirstymas atliekamas pagal binarinį arba proporcinį principą, atspindintį dvi skirtingas elgsenos prielaidas. Sukurtas giliojo Q-mokymosi modelis treniruojamas naudojant įvairius simuliuotus aplinkos scenarijus. Sprendimų atsparumas vertinamas empiriškai, įvertinant veikimą pagal abu klientų elgsenos tipus ir parenkant reprezentatyvią funkciją pagal Manhatano atstumą iki visiško rinkos užėmimo. Eksperimentiniai rezultatai rodo, kad išmoktos funkcijos pasiekia konkurencingus užimtos rinkos dalies rezultatus ir pasižymi reikšmingais skaičiavimo efektyvumo pranašumais, palyginti su išsamia visų sprendinių paieška. Nors skirtingose uždavinio konfigūracijose rezultatai kinta ir ne visada pasiekiamas optimalus sprendimas, gauti rezultatai leidžia teigti, kad skatinamasis mokymasis yra perspektyvi ir didelio masto problemoms spręsti tinkama kryptis konkurencingų objektų vietų optimizavimo uždaviniams, esant klientų elgsenos neapibrėžtumui.

Introduction

Facility location problems (FLPs) represent a class of long term strategic decisions faced by both private firms and public authorities. Decisions regarding the placement of production facilities, retail outlets, warehouses, hospitals, or emergency services involve substantial investment and have enduring consequences. Once established, facilities shape cost structures, accessibility, competitive dynamics and service availability over extended time horizons. As a result, facility location decisions affect not only the organization making the investment, but also competitors, customers and, in many cases, the wider regional economy.

The strategic importance of facility location extends beyond individual firms. In commercial settings, location choices influence market share, profitability and brand presence, while in public sector applications they affect societal welfare, access to essential services, safety and employment. Poorly chosen locations may lead to long term inefficiencies, whereas well designed facility networks can significantly reduce transportation costs, improve service quality and enhance regional development. Consequently, facility location has become a central topic in operations research, spatial economics and supply chain management.

Classical formulations of the facility location problem differ depending on the underlying objective. Firms may seek to minimize transportation costs, reduce response times, or maximize coverage, while public service providers may prioritize equitable access or overall socioeconomic impact. These differing goals give rise to a variety of problem formulations, each associated with specialized solution methods. However, even relatively simple extensions, such as multiple facilities, discrete candidate locations, or competition render FLPs computationally challenging due to their combinatorial nature. As instance sizes grow, exact optimization methods rapidly become infeasible, motivating extensive research into approximation algorithms and heuristic solution techniques.

Modeling real world facility location decisions introduces additional complexity. Customer demand is often uncertain, consumer behavior may vary across contexts and competitive responses can significantly alter outcomes. In particular, competitive facility location problems require explicit modeling of customer choice, as market share and profit emerges from interactions between facility locations, attractiveness and consumer preferences. Small changes in behavioral assumptions can lead to substantially different optimal solutions, highlighting the fragility of models optimized under a single, fixed specification.

This thesis focuses on competitive facility location problems faced by an entering firm that must place new facilities in a market with existing competitors. The objective is to maximize captured market share under the assumption that competitor locations, facility attractiveness and distances are known. However, the precise form of customer behavior is treated as uncertain and potentially heterogeneous across markets. Rather than optimizing for a single assumed customer choice model, the goal is to develop an approach that performs well across multiple plausible behavioral specifications.

To address this challenge, the thesis adopts a reinforcement learning perspective and proposes the use of deep Q-learning to learn facility placement strategies from interaction with simulated environments. By training across a diverse distribution of competitive configurations, demand realizations and customer behavior models, the learned policy aims to exhibit robustness in the sense of stable performance across scenarios, rather than optimality under a single assumed model. Unlike classical robust optimization, where robustness is imposed through explicit worst case constraints, the proposed approach treats robustness as a property that enables good performance on varying problem instances.

This work aims at three main objectives. First, frame competitive facility location under uncertain customer behavior as a sequential decision problem that can be addressed using reinforcement learning. Second, develop a deep Q-learning approach that can perform well across different customer behavior models without requiring explicit parameter tuning. Third, evaluate the proposed method against optimal solutions under a range of competitive and behavioral scenarios.

1 Literature overview

This section provides overview of history, significance, foundations of facility location problems, similar research work and reviews literature on the topic.

1.1 Strategic and economic importance of facility location

1.1.1 Historical evolution and theoretical origins

Facility location problems constitute one of the earliest and most enduring research streams in operations research, spatial economics and applied optimization. At their core, FLPs seek to determine the optimal placement of one or more facilities relative to spatially distributed demand to minimize cost, maximize service quality, or optimize a strategic goal. Despite the apparent simplicity of this objective, the problem class exhibits substantial analytical and computational complexity.

The origins of facility location theory can be traced back to classical geometry. The Fermat problem, later formalized by Torricelli, is attempting to find a point that minimizes the sum of Euclidean distances to three fixed points. Although initially purely mathematical curiosity, this problem introduced a fundamental principle that remains central to location theory: optimal spatial decisions emerge from balancing competing distance based trade-offs [7]. The transition from abstract geometry to economic relevance occurred with Alfred Weber's work in 1909. Weber's formulation of the industrial location problem extended Fermat's idea by incorporating weights to represent transportation costs and material flows, leading to the continuous Weber Problem [7]. Early solution methods relied on physical analogs such as the Varignon frame, which underscored both the intuition and analytical difficulty of these models [7].

A decisive shift occurred in the mid-twentieth century with the move from continuous spatial domains to discrete network representations. Hakimi's landmark contribution formalized location problems in graphs, introducing the absolute median and absolute center concepts for weighted networks [10]. His proof of the node optimality property demonstrated that for certain objective functions, such as minimizing the total weighted distance, optimal solutions exist at network vertices, justifying the discretization of space. This insight provided the theoretical foundation for modern discrete and mixed-integer formulations of FLPs, including p-median and p-center problems.

Parallel to cooperative and cost-minimizing formulations, the emergence of competitive location theory introduced strategic interaction as a defining feature. Hotelling's model of spatial competition in a linear market demonstrated how firms competing for customers tend to agglomerate toward the center, leading to the principle of minimum differentiation [11]. Subsequent literature revealed that this equilibrium is highly sensitive to modelling assumptions. When price competition or elastic demand is introduced, central clustering becomes unstable and firms instead have incentives to differentiate spatially [6, 29]. These results exposed a critical limitation of deterministic, single-objective models: optimality is often fragile and contingent on restrictive behavioral assumptions.

By the late twentieth century, it had become evident that exact analytical solutions are feasible only for quite primitive problems. Even the Euclidean p-median and p-center problems are NP-hard, including versions where facilities must be selected from the demand set. Multi facility location-allocation and stochastic extensions inherit and compound this NP-hardness.[18, 7]. As a result, research increasingly shifted toward heuristics, approximation algorithms and simulation based approaches. This historical trajectory highlights a recurring theme that motivates the present research: while exact optimal solutions may exist in theory, their practical relevance diminishes under real world conditions.

1.1.2 Economic significance and strategic implications of facility location decisions

Facility location decisions occupy a central role in economic theory and industrial organization because they shape cost structures, competitiveness and long term market dynamics. Unlike short term operational decisions, location choices are typically associated with substantial upfront investments, making them strategic commitments rather than tactical optimizations [11, 29].

From a microeconomic perspective, facility location directly influences transportation costs, market access and sometimes price. Primitive models framed location as a mechanism for minimizing production and distribution costs, while later network based formulations extended this logic to service systems such as telecommunications, emergency response and retail distribution [10]. However, cost minimization alone fails to capture the strategic dimension of competitive markets.

A crucial advancement in understanding economic relevance came from probabilistic demand modelling. Huff's gravity based model formalized customer choice as a probabilistic function of facility attractiveness and distance, replacing deterministic allocation rules with continuous market share allocation [12]. Subsequent research demonstrated that small change in behavioral parameters can significantly alter outcome [7].

These insights motivated a growing body of research emphasizing robustness rather than strict optimality [30, 9, 36, 27]. In competitive settings, robust solutions that perform consistently across uncertain scenarios may dominate fragile solutions optimized for a single behavioral specification.

1.2 Problem types and mathematical formulations

Facility location modelling encompasses a broad class of problems that differ in their spatial representation, decision scope and operational assumptions. A widely used organizing principle distinguishes between continuous, network based and discrete set models, with discrete formulations typically expressed as integer or mixed-integer programs [25]. These distinctions reflect different application contexts such as urban service placement, logistics network design and competitive retail expansion.

Notation. Throughout this section, let I denote the set of demand points (customers) and J the set of facility locations.
Distances between demand point $i \in I$ and facility $j \in J$ are denoted by d_{ij} and demand at point i by w_i.
Binary decision variables x_j indicate whether a facility is located at site j, while assignment variables y_{ij} indicate whether demand point i is served by facility j.
The parameter p denotes the prescribed number of facilities to be located.
Facility attractiveness is denoted by A_j and $\lambda > 0$ represents a distance decay or sensitivity parameter.
In competitive settings, P_{ij} denotes the probability that customer i patronizes facility j.

1.2.1 Continuous, network based and discrete location spaces

Continuous models allow facilities to be placed anywhere in a region and are often used in theoretical or geometric settings. Network based models restrict facility placement to points on a graph, such as road or communication networks, while discrete models limit feasible locations to a predefined candidate set, typically derived from demand points or zoning constraints.

In network based models $G = (V, E)$, Hakimi showed that for minisum objectives, optimal solutions exist at vertices even when facilities may be located anywhere along edges. In contrast, for minimax objectives, optimal solutions may lie in the interior of edges [10]. In Euclidean space, p-median and p-center variants allow unrestricted facility placement, leading to fundamentally different algorithmic behavior.

1.2.2 Single facility and multi facility problems

A fundamental distinction concerns the number of facilities to be located. Single facility problems often arise in public service planning, such as locating a single hospital or emergency center and can frequently be addressed using specialized geometric or network based solution methods. In contrast, multi facility problems arise in retail networks, logistics systems and telecommunications, where multiple facilities must be located simultaneously.

As the number of facilities increases, the problem rapidly becomes combinatorial due to the exponential growth of feasible facility subsets and the coupling between siting and allocation decisions [18]. This combinatorial structure determines high computational complexity in practical facility location model applications.

1.2.3 Capacity assumptions and facility count decisions

Another important modelling dimension concerns whether facilities have service limits and whether the number of facilities is fixed in advance. In uncapacitated models, facilities are assumed to have unlimited service capacity and demand is typically allocated to the best open facility under the assumed objective. Capacitated models impose explicit upper bounds on the amount of demand that can be served by each facility, introducing tighter coupling between location and allocation decisions.

Independent of capacity, models may assume that the number of facilities is fixed *a priori* (fixed p models) or determined endogenously through fixed opening costs (fixed charge models). These choices reflect different planning contexts, such as mandated service levels versus cost driven network expansion.

1.2.4 The p-median problem (minisum)

The p-median problem is a classical fixed p, uncapacitated location model that seeks to minimize the total demand weighted distance between customers and their assigned facilities. It is widely used to model systems such as distribution centers or service facilities where average travel cost is the primary concern. A standard formulation is:

$$
\begin{aligned}
\min \quad & \sum_{j\in J}\sum_{i\in I} w_i d_{ij} y_{ij}, \\
\text{s.t.} \quad & \sum_{j\in J} y_{ij} = 1, \quad \forall i \in I, \\
& y_{ij} - x_j \leq 0 \quad \forall i \in I, \forall j \in J \\
& \sum_{j\in J} x_j = p \\
& x_j \in \{0,1\} \quad \forall j \in J, \\
& y_{ij} \in \{0,1\} \quad \forall i \in I, \forall j \in J.
\end{aligned}
$$

[25]

1.2.5 The *p*-center problem (minimax)

The *p*-center problem focuses on service reach rather than efficiency by minimizing the maximum distance between any demand point and its assigned facility. This formulation is commonly used in emergency response, healthcare and public safety planning, where worst case access times are critical. The standard formulation is:

$$
\begin{aligned}
\min \quad & z \\
\text{s.t.} \quad & \sum_{j\in J} y_{ij} = 1 \quad \forall i \in I, \\
& y_{ij} - x_j \leq 0 \quad \forall i \in I,\ \forall j \in J, \\
& \sum_{j\in J} x_j = p, \\
& z - \sum_{i\in I} d_{ij} y_{ij} \geq 0 \quad \forall j \in J, \\
& x_j \in \{0,1\} \quad \forall j \in J, \\
& y_{ij} \in \{0,1\} \quad \forall i \in I,\ \forall j \in J.
\end{aligned}
$$

[25]

1.2.6 Uncapacitated and capacitated fixed charge facility location problems

When the number of facilities is not fixed *a priori*, uncapacitated facility location models minimize the sum of fixed facility location costs and transportation costs [25].

Let $f_j \geq 0$ denote the cost of locating a facility at candidate site $j \in J$. A scaling parameter α converts demand weighted distance into cost units.

The standard uncapacitated formulation is given by:

$$\begin{aligned}
\min \quad & \sum_{j \in J} f_j x_j + \alpha \sum_{j \in J} \sum_{i \in I} w_i d_{ij} y_{ij} \\
\text{s.t.} \quad & \sum_{j \in J} y_{ij} = 1 \quad \forall i \in I, \\
& y_{ij} - x_j \leq 0 \quad \forall i \in I,\ \forall j \in J, \\
& x_j \in \{0,1\} \quad \forall j \in J, \\
& y_{ij} \in \{0,1\} \quad \forall i \in I,\ \forall j \in J.
\end{aligned}$$

[25]

Capacitated variants extend this formulation by imposing an upper bound on the total demand that can be assigned to an open facility. Let C_j denote the capacity of facility j. The capacity constraint takes the form:

$$\sum_{j \in J} w_i y_{ij} - C_j x_j \leq 0, \quad \forall i \in I. \tag{1.1}$$

[7]

Introducing capacity constraints fundamentally changes the structure of the problem, substantially increasing computational complexity.

1.2.7 From classical types to competitive location

Competitive facility location models depart fundamentally from classical facility location formulations by replacing centralized allocation with decentralized customer choice. In traditional models such as the p-median, p-center, or fixed-charge facility location problems, demand is assigned deterministically to facilities according to an objective specified by the planner, typically minimizing total distance or cost. Customer behavior is therefore implicit and fully controlled by the optimization model.

In contrast, competitive facility location explicitly models the presence of multiple decision making firms and autonomous consumers. Facilities are no longer passive service points but strategic instruments whose locations influence customer choice, market share and ultimately firm profitability. Demand is not allocated by the planner but emerges from consumer preferences over competing facilities, which depend on factors such as distance, attractiveness and brand effects.

This shift has several important implications. First, the objective function changes from minimizing cost or maximizing social welfare to maximizing the captured market share of a single firm, given the fixed or anticipated locations of competitors. As a result, the optimization problem becomes asymmetric and firm specific, even though multiple firms operate in the same environment.

Second, competitive models introduce strong interdependence between facility locations. The marginal benefit of placing a facility at a given site depends not only on the spatial distribution of demand, but also on the configuration of existing competitors and on the locations of other facilities belonging to the same firm. This interdependence leads to cannibalization effects, where opening an additional facility may reduce the market share captured by previously placed facilities.

Third, customer choice models replace deterministic assignment. Unlike classical location problems, where each demand point is served by exactly one facility, competitive models can allow multiple facilities to share demand from the same customer. This results in smoother objective functions in some cases, but also introduces nonlinearity.

Finally, competitive facility location problems are inherently more sensitive to modeling assumptions. Small changes in distance sensitivity, attractiveness parameters, or the specification of customer behavior can lead to substantially different market outcomes.

1.3 Modelling customer behavior in competitive environments

Unlike classical formulations, competitive settings require explicit modelling of individual customer choice. This distinction is fundamental in retail, banking, healthcare and fuel station networks, where customers autonomously select among competing alternatives. Consequently, the validity of any competitive location model critically depend on the customers' behavioral assumptions.

The literature reflects a gradual evolution of customer choice rules, from deterministic distance based formulations toward probabilistic and dominance filtered models that better approximate observed behavior while increasing computational complexity.

1.3.1 The binary rule

The binary rule represents the earliest and most restrictive customer choice assumption in competitive location theory. Originating in Hotelling's linear city model, it assumes that each customer patronizes exactly one facility: the one offering the highest utility, typically determined solely by distance [11].

Formally, for a customer located at demand point i, the probability of choosing facility j is given by

$$P_{ij} = \begin{cases} 1 & \text{if } d_{ij} < d_{ik}, \quad \forall k \neq j, \\ 0 & \text{otherwise.} \end{cases}$$

This winner-takes-all mechanism induces sharp market boundaries and leads to minimum differentiation phenomena, where competing facilities cluster together to protect market share. The binary rule underpins the classical maximum capture (maxcap) problem, which has been extensively studied due to its tractability and compatibility with integer linear programming formulations [24].

Despite its analytical appeal, the binary rule is widely criticized for its behavioral rigidity. Small changes in facility location can cause discontinuous shifts in demand allocation and the model ignores partial patronage, brand loyalty and perception of attractiveness. Empirical evidence suggests that purely distance-minimizing behavior is rare outside of emergency or highly commoditized services [28].

1.3.2 The proportional rule

To address the limitations of deterministic assignment, Huff introduced a probabilistic gravity based model in which customers distribute their demand across facilities in proportion to perceived utility [12]. Utility is modeled as increasing in facility attractiveness and decreasing in distance, yielding the choice probability

$$P_{ij} = \frac{A_j d_{ij}^{-\lambda}}{\sum_{k\in J} A_k d_{ik}^{-\lambda}},$$

where A_j denotes the attractiveness of facility j, expressed as size of a shopping center and λ is a distance (originally travel time) sensitivity parameter.

Using empirical shopping data, Huff estimated this exponent to be approximately 2.7 for furniture shopping trips and 3.2 for clothing purchases, illustrating substantial variation across retail contexts [12]. Huff explicitly noted that this parameter should not be treated as universal, as distance sensitivity depends on the nature of the shopping trip.

The Huff rule introduces market overlap and smooths the objective function, making it attractive for gradient based optimization and simulation. However, it assumes that all facilities exert some influence on every customer, regardless of distance or dominance relationships. This leads to behavioral inconsistencies, particularly when inferior alternatives receive non-zero demand shares [12].

From a computational perspective, proportional rules substantially increase the dimensionality of the problem, as market share becomes a nonlinear function of all facility locations rather than only the nearest competitors.

1.3.3 The Pareto-Huff rule

Peeters and Plastria identified a fundamental behavioral inconsistency in the standard Huff model: it assigns strictly positive choice probabilities to facilities that are dominated by others in terms of both distance and attractiveness [23]. In particular, a facility that is farther away and no more attractive than an alternative may still receive a non-zero share of demand under the proportional rule.

To address this issue, the Pareto-Huff rule introduces a dominance based filtering step that restricts the customer's consideration set before applying the Huff probability formulation.

For a customer located at demand point $i \in I$, a facility $j \in J$ is said to be dominated if there exists another facility $k \in J \setminus \{j\}$ such that

$$d_{ik} \leq d_{ij}, \quad A_k \geq A_j,$$

with at least one of the inequalities holding strictly.

The customer specific consideration set is then defined as

$$C_i = \left\{ j \in J \;\middle|\; \nexists k \in J \setminus \{j\} \text{ such that } (d_{ik} \leq d_{ij}) \wedge (A_k \geq A_j) \right\}.$$

Only facilities belonging to this Pareto-optimal set are considered viable alternatives by customer i.

Conditional on this filtered set, choice probabilities are computed using the standard Huff formulation restricted to C_i:

$$P_{ij} = \begin{cases} \dfrac{A_j d_{ij}^{-\lambda}}{\sum_{k \in C_i} A_k d_{ik}^{-\lambda}}, & \text{if } j \in C_i, \\ 0, & \text{otherwise.} \end{cases}$$

The Pareto-Huff rule substantially improves behavioral realism by excluding dominated alternatives from consideration. However, this refinement introduces significant analytical and computational challenges. Although the problem can be discretized on a network and reduced to a finite candidate set of locations, the customer specific consideration set depends discontinuously on facility locations and attributes [23]. Infinitesimal changes in distance or attractiveness may cause a facility to enter or exit the Pareto optimal set for a given customer, leading to abrupt changes in demand allocation.

As a consequence, the resulting market share function is piecewise defined and generally non-smooth, exhibiting jump discontinuities even in discretized settings. These properties limit the effectiveness of classical gradient based optimization methods and render exact optimization difficult [28].

1.3.4 The partially binary rule

The partially binary rule represents an intermediate behavioral model that balances determinism and proportionality. It assumes that customers first select the most attractive facility from each firm and then split their demand among these representatives proportionally [28].

Under this rule, customers do not consider all facilities individually but rather evaluate firms as collections of outlets. This structure is particularly relevant for franchise systems and retail chains, where customers associate multiple outlets with a single brand identity.

The partially binary rule retains some of the computational advantages of binary assignment while accounting for variation in facility quality across locations operated by the same firm. Nevertheless, it introduces additional nonlinearity and interdependence across facilities belonging to the same firm, complicating exact optimization.

Recent work has shown that discrete competitive location problems under partially binary behavior can be reformulated as mixed-integer linear programs for small instances, but scalability remains a challenge [8].

1.4 Classical optimization methodologies

The competitive facility location problem inherits the computational difficulty of classical location models and typically adds non-linearity through customer choice rules and combinatorial coupling between multiple facilities. As a result, a toolbox has been developed that spans exact optimization, geometric and network theoretic methods, constructive heuristics and general purpose metaheuristics. The common theme across these approaches is a trade off between optimality and scalability to realistic instance sizes.

1.4.1 Exact methods: linear optimization and branch and bound

For discretized formulations where candidate facilities are restricted to a finite set of nodes, many location problems can be expressed using binary facility opening variables and assignment variables, resulting in mixed-integer linear programming formulations. Early work showed that linear programming formulations are useful not only for computing solutions, but also for evaluating heuristic solutions by providing bounds and for enforcing integrality through branch and bound when fractional solutions arise [26].

On networks, important structural results simplify exact search. In particular, for several p-center and p-median type problems, there exists optimal solutions located at graph vertices rather than on edges, allowing the continuous network problem to be reduced to a discrete one over nodes [10]. This vertex optimality insight is foundational because it justifies the discretization assumption used in many competitive location models.

However, complexity results place sharp limits on what “exact” can mean at scale. Even in purely geometric settings without competitive effects, the planar p-center and p-median problems are NP-hard under both Euclidean and rectilinear metrics and in some cases remain hard to approximate closely [18]. Related reductions show strong NP-hardness for families of planar linear facility location problems, reinforcing that exact methods are typically restricted to small instances or to problems with special structure [19].

1.4.2 Geometric and continuous space approaches

When facilities may be placed anywhere in the plane or along network arcs, geometric arguments and continuous optimization methods become relevant. In continuous facility location models, geometric arguments and optimality conditions can identify locally optimal regions, but complexity results show that these methods rarely yield efficient algorithms for finding global optima [18].

In network contexts, absolute center and absolute median formulations allow a facility to lie on an edge rather than being restricted to vertices. Hakimi's development of absolute centers and medians provides both definitions and solution procedures for these continuous on network problems and clarifies when vertex restriction is valid and when it is not [10]. This distinction matters for competitive location: discretization can be theoretically justified for some objectives, but not universally and should therefore be treated as a modelling choice rather than a free simplification.

1.4.3 Constructive heuristics

Because exact approaches rapidly become infeasible as problem size grows, a large body of work relies on constructive heuristics that build or improve solutions incrementally. These methods generate feasible solutions through a sequence of locally motivated decisions, trading optimality for scalability.

A simple class of constructive heuristics is based on greedy improvement. In this approach, facilities are added, removed, or relocated one at a time according to their marginal contribution to the objective function. At each step, the locally best move is selected and the process continues until no further local improvement is possible. Such greedy procedures are computationally efficient but can converge to suboptimal solutions because they rely solely on immediate improvements [20, 21].

A more powerful class of methods is given by interchange or vertex exchange heuristics. These methods start from a feasible solution and iteratively replace one selected facility with an unselected candidate location whenever such a substitution yields an improvement in the objective. Teitz and Bart [32] provide an early systematic analysis of vertex substitution procedures for generalized vertex median problems and formally describe an iterative exchange algorithm that terminates at a locally optimal configuration.

A different strategy is provided by aggregation heuristics, which reduce problem size by clustering demand points or candidate locations into aggregated units. The location problem is first solved on the reduced instance and the resulting solution is then refined or disaggregated. Aggregation techniques are commonly used to obtain approximate solutions for large scale discrete location problems, but they may obscure fine grained spatial structure [20, 7].

These constructive heuristics are attractive in competitive facility location settings because they rely primarily on repeated objective evaluations and can be adapted to non-linear market share objectives by substituting a capture function in place of distance based costs. However, their reliance on marginal improvements evaluated under a fixed partial configuration limits their ability to capture joint placement effects, facility cannibalization and non-linear customer choice interactions that are fundamental to competitive multi facility location problems.

1.4.4 Coverage based models

A different classical line of research replaces average distance objectives with coverage constraints or coverage maximization. The maximal covering location problem formalizes the idea of selecting a limited number of facilities to cover as much weighted demand as possible within a predefined service radius, motivated by public service and emergency planning but broadly applicable as a surrogate for accessibility and responsiveness [5].

For competitive retail settings, coverage models can serve as computationally efficient proxies for presence or reach before more detailed choice based evaluation. They also provide an interpretable axis for robustness, as solutions that maintain high coverage across different distance thresholds may be preferable when distance sensitivity is uncertain.

1.4.5 Advanced metaheuristics: Tabu search, simulated annealing and genetic algorithms

General purpose metaheuristics provide a middle ground between purely local improvement and full enumeration. Their key advantage is the ability to explore beyond local optima through controlled diversification mechanisms such as memory structures, stochastic acceptance, or population based search, while retaining the flexibility to optimize black box objectives.

Comparative empirical evidence in facility location indicates that Tabu search often performs strongly across multiple problem variants under practical computational budgets, while simulated annealing and genetic algorithms show performance that is more sensitive to problem type and experimental criteria [1]. This observation is directly relevant to competitive facility location, where objectives are frequently non-linear and discontinuous due to choice based capture and dominance effects and where metaheuristics remain viable because they rely primarily on repeated evaluation rather than gradient information or convexity assumptions.

1.5 Advanced methodologies in competitive facility location

The increasing scale, stochasticity and strategic nature of competitive facility location problems have exposed fundamental limitations of classical optimization and metaheuristic approaches. In particular, traditional methods typically assume a fixed demand distribution, a single customer behavior model and complete information regarding competitors, requiring re-optimization whenever these assumptions change. These limitations have motivated growing interest in data driven and learning based methodologies that enable generalization across problem instances rather than producing solutions tailored to a single scenario.

Recent advances in machine learning, especially in deep learning and reinforcement learning, have opened new avenues for addressing hard combinatorial optimization problems. Rather than solving each instance from scratch, learning based methods aim to extract structural regularities from distributions of problem instances and encode them into reusable decision policies [2, 34]. This perspective is particularly relevant for competitive facility location, where uncertainty in customer behavior, spatial demand and competitor actions is intrinsic.

1.5.1 Learning based approaches for combinatorial optimization

Early work on machine learning for combinatorial optimization focused on supervised learning, where neural networks were trained to imitate optimal or near optimal solutions generated by exact solvers or heuristics. While effective for small or well structured instances, supervised approaches suffer from scalability limitations and a strong dependence on labeled data, which is often costly or infeasible to obtain for large-scale facility location problems [2].

Reinforcement learning offers a more flexible alternative by framing combinatorial optimization as a sequential decision making problem. In this setting, a solution is constructed incrementally, an agent learns a policy that maximizes a cumulative reward signal aligned with the optimization objective. This formulation naturally matches facility location problems, where facilities are opened one by one and each decision influences future options and outcomes [17].

Several surveys emphasize that reinforcement learning is well suited for learning constructive heuristics for NP-hard problems, replacing hand crafted decision rules with policies learned directly through interaction with a simulated environment [17, 34]. Importantly, reinforcement learning methods can be trained over distributions of problem instances, enabling policies that generalize to previously unseen scenarios.

1.5.2 Deep Q-learning for facility location problems

Among reinforcement learning methods, value based approaches such as deep Q-networks have received particular attention. Deep Q-networks approximate the state-action value function using deep neural networks, allowing reinforcement learning to scale to high-dimensional state spaces [22]. This development has enabled reinforcement learning to be applied to problems far beyond small tabular settings.

In the context of facility location, deep Q-learning has been used to sequentially select facility locations, with the state encoding current facility placements, demand distributions, capacity constraints and the reward defined as marginal improvements in cost or service quality. Some research indicates that deep Q-learning can be used for capacitated facility location problems, improving efficiency over classic solution methods while maintaining similar performance[37].

A key advantage of deep Q-learning in this setting is its ability to learn policies that are reusable across instances, rather than producing a single solution for a specific problem realization. This property is particularly valuable in competitive facility location problems, where customer behavior, demand distributions and competitor configurations may vary across scenarios.

1.5.3 Attention mechanisms and set based representations

A major challenge in applying deep learning to facility location problems lies in representing spatial demand points and candidate locations. Classical convolutional neural networks assume grid structured inputs, which is incompatible with irregular spatial distributions and variable numbers of demand points.

Attention mechanisms and transformer architectures address this limitation by operating on sets rather than grids. Self attention allows models to learn pairwise interactions between all elements in a set, making them permutation invariant and well suited for facility location problems defined on graphs or point sets [33]. Attention based models have demonstrated strong performance in routing and location related problems by capturing long range dependencies that are difficult to encode using local heuristics [13].

In competitive facility location settings, attention mechanisms enable models to focus dynamically on strategically important demand points or competitor facilities when evaluating candidate locations. This capability is particularly relevant under probabilistic customer choice models, such as Huff or Pareto-Huff formulations, where relative attractiveness and distance jointly determine market share.

1.5.4 Graph neural networks and spatial interactions

Graph neural networks provide another powerful framework for modelling facility location problems, especially when demand points and facilities are naturally represented as nodes in a graph with weighted edges encoding distances or travel costs. Through iterative message passing, graph neural networks allow each node to aggregate information from its neighbors and capture spatial as well as competitive interactions [38].

Recent work has shown that graph neural networks can be successfully combined with reinforcement learning to solve combinatorial optimization problems on graphs, including routing, matching and facility placement [34]. These models are particularly appealing for competitive facility location problems, as they offer a principled way to represent interactions between demand points, existing competitors and candidate facility locations within a unified learning framework.

1.6 Foundations of reinforcement learning for operations research

The application of reinforcement learning to facility location problems builds on a well established theoretical framework developed in the fields of control theory, dynamic programming and artificial intelligence. Reinforcement learning provides a methodology for solving sequential decision making problems under uncertainty, where explicit modelling of future system states and outcomes is either infeasible or computationally expensive. This framework is particularly well suited to competitive facility location problems, where market share outcomes depend on complex interactions between spatial structure, customer behavior and competitor actions.

1.6.1 Markov decision process formulation

Reinforcement learning problems are commonly formalized as a Markov decision process (MDP), defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ [31].

Notation. A state is denoted by $s \in \mathcal{S}$ and an action by $a \in \mathcal{A}$.
The next state after taking action a in state s is denoted by s' and the corresponding immediate reward by r.
R_t represents the cumulative sum of rewards received after time step t.
A policy $\pi(a \mid s)$ is a conditional distribution over actions given a state.
The (discounted) return from time t is $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$.
The parameter $\gamma \in (0, 1]$ denotes the discount factor, which determines the relative importance of immediate versus future rewards by exponentially discounting rewards received at later time steps.
The state-value function under policy π is $V^\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$
and the action-value function is $Q^\pi(s, a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$.
The optimal action-value function is $Q^*(s, a) = \max_\pi Q^\pi(s, a)$.
In Bellman optimality equations, a' denotes a generic feasible action in the next state s' used in the one step lookahead maximization $\max_{a'} Q^*(s', a')$.
When actions are constrained, we denote the feasible action set in state s by $\mathcal{A}(s) \subseteq \mathcal{A}$.

At each discrete time step t, an agent observes a state $S_t \in \mathcal{S}$, selects an action $A_t \in \mathcal{A}$, receives a scalar reward $R_{t+1} \in \mathcal{R}$ and transitions to a new state according to the environment dynamics $\mathcal{P}(S_{t+1} \mid S_t, A_t)$. The objective of the agent is to learn a policy $\pi(a \mid s)$ that maximizes the expected discounted return G_t.

In the context of facility location, states encode partial placement configurations, actions correspond to selecting new facility locations and rewards represent incremental changes in market share or objective value. This sequential construction naturally satisfies the Markov property when the state representation captures all relevant placement information.

1.6.2 Dynamic programming and the Bellman equation

The theoretical foundations of reinforcement learning are rooted in dynamic programming, which characterizes optimal sequential decision making through recursive value equations [31].

The optimal action-value function $Q^*(s, a)$ satisfies the Bellman optimality equation [31, 35]:

$$Q^*(s, a) = \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} Q^*(S_{t+1}, a') \mid S_t = s, A_t = a\right]. \tag{1.2}$$

This recursive relationship forms the basis of Q-learning, a model free reinforcement learning algorithm that approximates Q^* by iteratively updating value estimates using sampled state transitions [35].

1.6.3 Tabular Q-learning and its limitations

Classical Q-learning updates the action-value function according to [35]:

$$Q_{t+1}(s, a) \leftarrow Q_t(s, a) + \alpha\left(r + \gamma \max_{a'} Q_t(s', a') - Q_t(s, a)\right), \tag{1.3}$$

where α is a learning rate. Under mild conditions, including sufficient exploration and diminishing learning rates, tabular Q-learning converges to the optimal action-value function with probability one.

However, facility location problems exhibit an exponential growth in the state-action space, as each partial placement configuration corresponds to a distinct state. This renders tabular representations infeasible even for moderately sized instances. Consequently, function approximation is required to enable generalization across states and to scale reinforcement learning methods to realistic facility location problems.

1.6.4 Deep Q-learning and function approximation

Deep Q-networks replace the tabular Q-function with a neural network parameterized by θ, yielding an approximation $Q(s, a; \theta)$. Training proceeds by minimizing the temporal difference loss [22]:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s')}\left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta)\right)^2\right], \tag{1.4}$$

where the expectation is taken over transitions sampled from experience replay and θ^- denotes the parameters of a target network that is updated periodically to improve training stability. Two mechanisms are critical for effective learning in practice: experience replay, which reduces temporal correlations between samples and the use of a target network, which mitigates non-stationarity in the learning targets [22].

The success of deep Q-learning in high-dimensional sequential decision problems has motivated its adoption in combinatorial optimization and operations research, including facility location and related network design settings, where solution construction can be naturally framed as a sequential decision process [37].

1.6.5 Experience replay and target networks

Experience replay was originally introduced to mitigate instability arising from correlated updates in sequential decision processes [16]. By storing past transitions in a replay buffer and sampling them during training, the learning procedure approximates independent and identically distributed updates. This reduces variance, improves data efficiency and stabilizes temporal difference learning when combined with function approximation.

In the context of facility location problems, experience replay serves an additional role beyond numerical stabilization. Sequential placement decisions induce long range dependencies, where early facility choices strongly constrain future actions and rewards. Replay buffers expose the agent to a diverse collection of partial configurations and demand realizations, preventing concentration to a single trajectory. This promotes more robust policy learning across heterogeneous spatial and competitive scenarios.

Target networks address a complementary source of instability in deep Q-learning. When the same network is used to both estimate current action values and construct bootstrapped targets, small parameter updates can lead to rapidly shifting training objectives. Target networks decouple these roles by maintaining a delayed copy of the Q-network whose parameters are updated only periodically. This effectively stabilizes the learning target and prevents harmful feedback loops during optimization [22].

Together, experience replay and target networks form the core stabilization mechanisms that enable deep Q-learning to scale to high dimensional, combinatorial decision problems. Their combined use has become standard in deep reinforcement learning and is essential for applying Q-learning reliably to complex facility location environments characterized by delayed rewards, non-stationary competitive interactions and large state spaces [31, 37].

1.6.6 Relevance to competitive facility location problems

When facility location is formulated as a sequential decision process, reinforcement learning provides a natural alternative to classical optimization techniques. Unlike exact solvers, reinforcement learning does not require explicit enumeration of all feasible solutions, nor does it rely on strong assumptions about convexity or linearity. By learning policies rather than single solutions, deep reinforcement learning can produce strategies that generalize across varying demand distributions, competitor placements and customer behavior models.

Recent studies demonstrate that deep Q-learning can achieve near optimal performance on capacitated facility location problems while scaling to instance sizes that are otherwise computationally unfeasible [37].

1.7 Related work

Uncertainty has long been recognized as a central challenge in facility location, as solutions optimized for a single nominal scenario often exhibit high sensitivity to changes in demand, costs, or behavioral parameters. Classical robust optimization addresses this issue by replacing probabilistic assumptions with deterministic uncertainty sets and by seeking decisions that remain feasible for all realizations within these sets. This framework emphasizes worst case protection and provides formal guarantees on solution stability, often at the expense of increased conservatism [3].

Robust optimization has been applied to facility location primarily to model uncertain demand volumes, transportation costs and service capacities. Empirical evidence shows that robust facility location models often achieve superior worst case performance and more stable performance under alternative demand realizations compared to nominal formulations. However, this improvement depends critically on the specification of the uncertainty set. In particular, overly conservative sets may lead to unnecessarily conservative solutions, while poorly structured sets may fail to capture important features of demand variability [9].

In competitive facility location, uncertainty extends beyond demand parameters to include customer behavior and competitor actions. Recent work proposes robust competitive models that explicitly account for ambiguity in demand types and behavioral responses, often resulting in two level or mixed-integer formulations that remain tractable only for moderate instance sizes. These studies demonstrate that robustness with respect to behavioral uncertainty can materially alter optimal location decisions, reinforcing the fragility of solutions optimized under a single behavioral specification [36].

Despite their theoretical appeal, robust optimization approaches face practical limitations in competitive settings. Explicit enumeration of uncertainty sets becomes difficult when uncertainty arises from high dimensional spatial distributions, heterogeneous customer preferences and strategic competitor placement. Moreover, worst case formulations may lead to overly conservative solutions that sacrifice average case performance in favor of protection against extreme scenarios.

1.7.1 Learning based approaches to robustness in combinatorial optimization

An alternative perspective on robustness has emerged from the machine learning and reinforcement learning literature, where robustness is evaluated through empirical performance rather than enforced via explicit worst case constraints. Instead of enforcing feasibility for all realizations in a predefined uncertainty set, learning based approaches train policies across a diverse distribution of problem instances, with the objective of achieving stable performance across scenarios [17].

In the context of combinatorial optimization, reinforcement learning has been increasingly used to learn constructive heuristics that generalize across instances. Surveys emphasize that such methods do not provide worst case guarantees, but can outperform classical heuristics in practice by adapting to structural regularities in the data and by amortizing computational effort across repeated problem instances [34].

Attention based and graph based neural architectures have proven particularly effective for problems defined over sets or networks, as they naturally accommodate variable size inputs and capture long range interactions. These properties are especially relevant for facility location problems, where spatial interactions and competitive effects induce global dependencies that are difficult to encode using local or rule based heuristics [33, 13].

1.7.2 Deep reinforcement learning for facility location problems

Recent studies apply deep reinforcement learning to facility location by formulating the problem as a sequential decision process in which facilities are placed one at a time and rewards correspond to marginal improvements in objective value. Zhao et al. demonstrate that deep Q-learning can successfully learn near optimal policies for capacitated facility location problems with discrete expansion sizes, achieving competitive solution quality while offering fast inference once trained [37].

A key insight from this work is that reinforcement learning enables the learning of reusable decision policies rather than single instance specific solutions. By training on a wide range of problem realizations, the learned policy implicitly encodes a trade off between competing objectives and uncertainties encountered during training. This property is particularly attractive for competitive facility location, where customer distributions, behavioral parameters and competitor configurations are inherently uncertain and difficult to model exhaustively.

2 Methodology, experimental design and results

This section presents methodology used to construct a robust solution policy for the discrete competitive facility location problem under Pareto-Huff filtered customer choice. We first describe the simulation environment and market share computation, then formalize the reinforcement learning formulation, neural architecture and training procedure. Finally, we describe the experimental benchmarking protocol and summarize the observed results.

2.1 Simulation setup and instance generation

We consider a discrete competitive facility location problem for an entering firm, in which demand is represented by a finite set of demand points $I = \{i_1, \dots, i_{n_I}\}$, preexisting facilities are located at a set $J \subset I$ and the entering firm chooses a set $X \subseteq L$ of new facilities from a candidate set $L \subset I \setminus J$ to maximize captured market share $M(X)$ [14].

2.1.1 Synthetic data

Each experimental instance is generated by sampling N demand point coordinates within a rectangular geographic bounding region. Each demand point $i \in I$ is assigned:
(i) a demand weight w_i (buying power) and
(ii) an attractiveness attribute A_i (used when the point hosts a facility).
Competitor facilities are generated by sampling J. Candidate locations L are sampled as a subset of remaining points.

2.1.2 Distance calculation

Distances between points are computed using the Haversine formula, generating a dense matrix $D \in \mathbb{R}_+^{n_I \times n_I}$. For numerical stability, a small constant is added to distances to prevent division by zero when a facility coincides with a demand point.

2.1.3 Market share capture under Pareto-Huff filtering

Customer choice and market share capture are evaluated conditional on the current set of active facilities $F = J \cup X$, where J denotes preexisting competitors and X the facilities placed so far by the entering firm. For each demand point $i \in I$, utilities are computed as

$$u_{if} = \frac{A_f}{d(i,f)^\lambda}, \quad f \in F, \tag{2.1}$$

where A_f is facility attractiveness, $d(i,f)$ is the distance between customer i and facility f and $\lambda > 0$ is the distance decay exponent.

Pareto-Huff dominance filtering. Before converting utilities into choice probabilities, a dominance filter is applied. A facility $f \in F$ is removed from customer i's consideration set if there exists another facility $k \in F$ such that

$$d(i,k) \leq d(i,f), \quad A_k \geq A_f,$$

with at least one inequality strict. Let $C_i(F) \subseteq F$ denote the resulting Pareto-optimal consideration set. Facilities $f \notin C_i(F)$ are assigned zero utility prior to normalization. This implements the Pareto-Huff filtering principle and excludes facilities that are clearly inferior in both distance and attractiveness [23, 14].

Binary and proportional allocation modes. Given the filtered utilities, two alternative customer allocation rules are evaluated in order to assess robustness with respect to behavioral assumptions:

- **Binary allocation:** all demand at i is assigned to the facility (or facilities, in case of ties) with maximal filtered utility.
- **Proportional allocation:** demand at i is distributed across facilities proportionally to their filtered utilities.

Formally, the probability that customer i allocates demand to facility f is

$$p_{if} = \begin{cases} \dfrac{1}{|\arg\max_{k \in C_i(F)} u_{ik}|}, & \text{binary mode and } f \in \arg\max_k u_{ik}, \\ \dfrac{u_{if}}{\sum_{k \in C_i(F)} u_{ik}}, & \text{proportional mode and } f \in C_i(F), \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

Market share computation. The entering firm's captured demand is obtained by aggregating its allocated shares across all customers,

$$M(X) = \frac{\sum_{i \in I} w_i \sum_{f \in X} p_{if}}{\sum_{i \in I} w_i}, \tag{2.3}$$

and is reported as a percentage of total market demand. This capture function is evaluated repeatedly during training and inference after each facility placement step, directly determining the reinforcement learning reward signal.

2.2 RL formulation as a sequential decision process

Selecting $S = |X|$ facilities from a candidate set is naturally modeled as a sequential construction process: at each step the agent selects one new facility location until S facilities are placed.

2.2.1 State representation.

The environment maintains a per demand point feature vector including: normalized latitude, normalized longitude, occupancy status (empty/competitor/self), normalized demand weight, normalized attractiveness and an indicator for whether the point is a candidate location. The resulting observation is a set (sequence) of N feature vectors.

2.2.2 Action space and feasibility mask

Each action corresponds to selecting a demand point index. A feasibility mask enforces that the agent may only choose points in L that are currently empty (not in J and not already selected into X).

2.2.3 Reward shaping

After placing a facility, the agent receives reward equal to the marginal improvement in market share:

$$r_t = M(X_t) - M(X_{t-1}), \tag{2.4}$$

where X_t is the set of facilities placed up to step t. This reward encourages placements that improve capture early while remaining aligned with maximizing final capture.

2.3 Deep Q-network with attention and transformer layers

To address the high dimensionality and structural complexity of competitive facility location under uncertain customer behavior, we employ a deep Q-learning framework augmented with attention based set encoders. The objective is to learn a reusable placement policy that generalizes across different demand realizations, competitor configurations and customer allocation rules, rather than optimizing a single instance.

2.3.1 Sequential decision formulation

The placement of facilities by the entering firm is modeled as a sequential decision process. At each step, the agent selects one candidate location from the remaining feasible set until the prescribed number of facilities is placed. This naturally defines a Markov decision process with:

- **State** s_t: the current partial placement configuration, including the set of already selected facilities, remaining candidate locations and demand attributes.
- **Action** $a_t \in L_t$: selecting one feasible candidate location from the remaining set.
- **Reward** r_t: the marginal change in captured market share induced by adding facility a_t.
- **Termination**: the episode ends when the required number of facilities has been placed or no feasible actions remain.

This construction aligns with prior reinforcement learning formulations for facility location, where full solutions are assembled incrementally and rewards reflect marginal objective improvements [37].

2.3.2 Motivation for attention based representations

A central challenge in facility location learning is that demand points, candidate locations and competitors form an unordered set of spatial items. Classical neural architectures, such as fully connected or convolutional networks, impose artificial ordering or grid structures that are incompatible with irregular spatial distributions and variable instance sizes.

To overcome this limitation, we adopt a transformer based set encoder using self attention mechanisms. Self attention provides permutation invariance and enables the model to learn pairwise and higher order interactions between demand points, candidate facilities and competitors. This is particularly important in competitive location problems, where the marginal value of placing a facility depends on global spatial context and competitive proximity rather than local features alone [33].

2.3.3 State encoding and feature representation

Each demand point $i \in I$ is represented by a feature vector that includes:

- normalized spatial coordinates,
- demand weight w_i,
- attractiveness A_i,
- distance based features to existing competitors and already placed facilities,
- binary indicators encoding feasibility and selection status.

These point wise features are embedded into a shared latent space via a linear embedding layer. The resulting set of embeddings is processed by a multilayer transformer encoder, allowing the network to learn spatial and competitive interactions across the entire instance.

2.3.4 Q-value prediction and action masking

The transformer outputs a latent representation for each candidate location. A shared linear value head maps each representation to a scalar Q-value, yielding

$$Q_\theta(s_t, a), \quad a \in L.$$

Because not all actions are feasible at every step (e.g., already selected locations or filtered candidates), an explicit action mask is applied. Infeasible actions are assigned a value of $-\infty$ prior to maximization, ensuring that the policy respects placement constraints and candidate availability.

This design allows the network to evaluate all candidate locations in parallel while maintaining strict feasibility during action selection.

2.3.5 Replay memory and target network

Training follows the deep Q-learning paradigm with experience replay and a target network for stability [22]. Each transition

$$(s_t, a_t, r_t, s_{t+1}, \texttt{done}, \texttt{mask}_{t+1})$$

is stored in a replay buffer. Mini-batches are sampled uniformly to break temporal correlations induced by consecutive placement decisions.

A separate target network Q_{θ^-} is maintained and periodically synchronized with the policy network. This stabilizes training by decoupling the bootstrapping targets from rapidly changing Q-value estimates.

2.3.6 Temporal difference target with masked maximization

For each sampled transition, the temporal difference target is computed as:

$$y_t = \begin{cases} r_t, & \text{if } \texttt{done}, \\ r_t + \gamma \max_{a' \in \mathcal{A}(s_{t+1})} Q_{\theta^-}(s_{t+1}, a'), & \text{otherwise}, \end{cases} \tag{2.5}$$

where the maximization is performed only over feasible actions using the action mask. If no feasible actions remain, the bootstrap term is omitted.

2.3.7 Loss function and optimization

The network parameters θ are optimized by minimizing the mean squared temporal difference error (1.4).

Optimization is performed using the Adam optimizer with learning rate selected empirically. Gradient clipping is applied to mitigate instability caused by rare but large reward updates, which can arise from discontinuities induced by Pareto-Huff filtering.

2.3.8 Hyperparameter selection and training configuration

The performance and stability of deep Q-learning depend critically on the choice of hyperparameters governing exploration, discounting, optimization and batch learning. We'll cover the selected hyperparameters and their rationale.

Discount factor γ. The discount factor $\gamma \in (0, 1]$ controls the relative importance of future rewards. We set $\gamma = 0.95$ to capture more market share with early facility placement decisions, but rewards are still strongly tied to the final solution. Values close to one are standard in episodic decision problems where delayed rewards are meaningful [31].

Learning rate. The Adam optimizer is used with learning rate 10^{-4}. A relatively small learning rate was required to ensure stable training due to non-smooth reward dynamics induced by Pareto-Huff filtering.

Batch size. Mini batches of size 64 are sampled from the replay buffer. This value balances gradient stability and computational efficiency and is commonly used in deep Q-learning applications. Smaller batches lead to noisy updates, while larger batches slow training process.

Replay memory. Each agent maintains a replay buffer with a fixed capacity of 10,000 transitions, from which training mini batches are sampled. The buffer size was deliberately kept moderate to balance memory usage and training efficiency under practical computational constraints.

Experience replay mitigates the strong temporal correlations inherent in sequential facility placement and improves data efficiency by allowing past transitions to be reused across multiple updates [16].

Target network update frequency. The target network parameters are synchronized with the policy network every five training episodes. Frequent updates were necessary to track the rapidly changing Q-function during early training, while still providing sufficient stabilization to prevent feedback loops.

Exploration strategy and epsilon decay. Action selection follows an ϵ-greedy strategy. The exploration rate is initialized at $\epsilon = 1.0$ and decayed with factor 0.9992 after each episode, down to a minimum of 0.05. This slow decay allows substantial exploration during approximately the first 60% of training episodes before gradually moving toward exploitative behavior. Random exploration is particularly important in competitive location problems, where local optima and cannibalization effects are common.

Parallel training and agent diversity. Multiple agents are trained in parallel with independent random seeds. This serves two purposes:
(i) improved training time,
(ii) increased solution diversity, which is later exploited during Pareto front model selection.
Parallel agents to some degree mitigates sensitivity to initialization and stochasticity in the environment.

Architectural hyperparameters. The transformer encoder uses a hidden dimension of 128 and four attention heads. This configuration was sufficient to capture spatial interactions without incurring excessive computational cost.

2.3.9 Rationale for robustness under behavioral uncertainty

Crucially, the network is trained across instances evaluated under both binary and proportional customer allocation rules, while sharing a single policy. As a result, the learned Q-function does not specialize to a single behavioral assumption but instead internalizes structural regularities that perform well across heterogeneous customer responses.

This training procedure prioritizes stability across scenarios, rather than strictly optimizing for worst case outcome. The resulting policy trades some instance specific optimality for improved generalization and consistent performance across customer behavior models, aligning directly with the objectives of this thesis.

2.4 Model selection

Robustness is evaluated jointly with respect to the two customer allocation modes by treating $(M_{\text{binary}}, M_{\text{prop}})$ as a bi-objective performance vector. Rather than selecting models optimized for a single behavioral assumption, we seek a solution that performs consistently well across both extremes.

Each trained agent produces a point in the two dimensional objective space and model selection is formulated as a multi-objective decision problem. Following established practice in multi-objective optimization, we define an idealized utopia point corresponding to perfect capture under both allocation rules, $(100, 100)$ and select a single representative solution by minimizing its Manhattan (ℓ_1) distance to this point:

$$d_{\ell_1} = |100 - M_{\text{binary}}| + |100 - M_{\text{prop}}|. \tag{2.6}$$

The minimum Manhattan distance criterion is widely used as a robust knee point identification and decision making method when no explicit preference between objectives is available. It favors solutions that achieve balanced performance between the objectives [4].

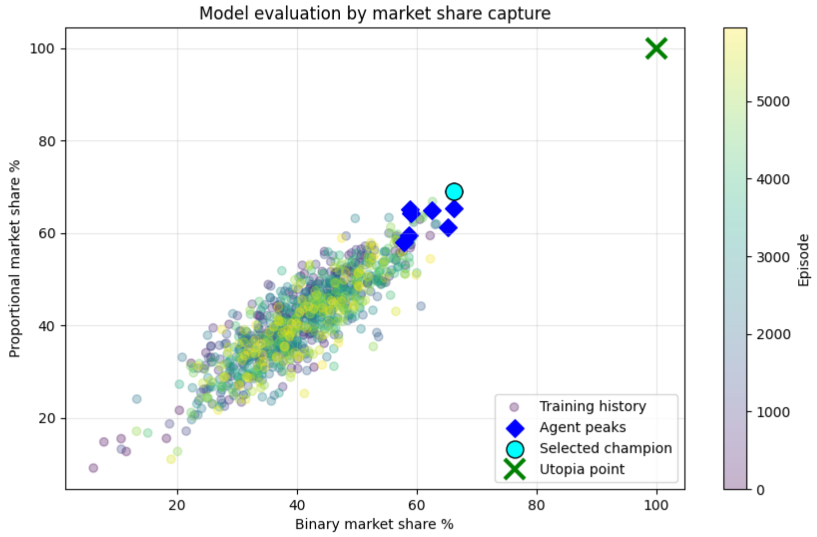


Figure 1. Model selection

In bi-objective settings, minimizing the Manhattan distance to the utopia point is equivalent to selecting a solution located near the knee region of the Pareto front, where marginal gains in one objective would incur disproportionate losses in the other. Such knee points are sometimes regarded as the most cost effective and robust choices in the absence of additional preference information [15].

This selection criterion is particularly appropriate in the present context, where customer behavior is uncertain. By selecting the agent closest to the utopia point, we obtain a single policy that exhibits stable performance across behavioral types rather than fragile optimality under a specific assumption.

Figure 1 illustrates the model selection procedure. Each point corresponds to a trained agent evaluated under both binary and proportional allocation rules. The utopia point (100, 100) is shown for reference and the selected champion agent is the one minimizing Manhattan distance to this point.

2.5 Deep Q-learning training procedure

The facility placement problem is formulated as a sequential decision process in which facilities are placed one at a time from a finite candidate set. At each step, the agent observes the current partial placement configuration and selects one feasible candidate location. Customer choice and market share capture are evaluated using Pareto-Huff filtered utilities, while the final allocation of demand follows either a binary or proportional rule.

Training is performed using deep Q-learning with function approximation. Multiple agents are trained in parallel with independent random seeds. Each agent maintains its own replay buffer and target network and learns a state-action value function by minimizing the temporal difference error using experience replay. Action feasibility is enforced by restricting decisions to the current feasible candidate set.

To encourage robustness with respect to customer behavior, policies are periodically evaluated under both binary and proportional allocation modes on a shared validation set. Model selection is treated as a bi-objective problem and a single champion policy is selected by minimizing the

Manhattan distance to the utopia point $(100, 100)$ in the $(M_{\text{binary}}, M_{\text{prop}})$ performance space.

Training configuration. Training instances are generated synthetically as described in the previous subsection. The number of demand points is fixed at $|I| = 500$, with coordinates sampled uniformly within a rectangular geographic region defined by latitude and longitude bounds $(\text{lat}_{\text{min}}, \text{lat}_{\text{max}}, \text{lon}_{\text{min}}, \text{lon}_{\text{max}}) = (54.56, 54.82, 25.02, 25.48)$. Facility attractiveness values are sampled uniformly from the interval $A_j \in [10, 80]$. The distance decay parameter in the Pareto-Huff model is fixed at $\lambda = 0.5$.

For each training episode, the number of preexisting competitor facilities is sampled uniformly from the range $|J| \in [5, 30]$and the number of candidate locations is sampled from $|L| \in [3, 50]$. The agent is allowed to place up to $S_{\text{max}} = 20$ facilities during training, while evaluation is performed for $S \in [3, 10]$ to assess generalization across different facility counts.

A total of $K = 8$ agents are trained in parallel for $E = 6000$ episodes each. Policy evaluation is performed every $\Delta = 50$ episodes.

The overall training and best model selection procedure is summarized in Algorithm 1.

Algorithm 1. Deep Q-learning training algorithm

```
Input: Episodes $E$, agents $K$, discount $\gamma$, replay size $B$, batch size $b$, target update period $T$, eval period $\Delta$.
Output: Champion policy $\theta^*$.
for $k = 1$ to $K$ (parallel) do
    Initialize Q-network $Q_{\theta_k}$, target $Q_{\bar{\theta}_k} \leftarrow Q_{\theta_k}$, replay buffer $\mathcal{M}_k$, exploration rate $\epsilon \leftarrow 1$.
    for $e = 1$ to $E$ do
        Sample an instance: demand $I$, competitors $J$, candidates $L$, facilities to place $S$.
        $X \leftarrow \emptyset$.
        for $t = 1$ to $S$ do
            Observe state $s$ and feasible action set $\mathcal{A}(s) \subseteq L$.
            Choose $a \in \mathcal{A}(s)$ via $\epsilon$-greedy w.r.t. $Q_{\theta_k}(s, a)$ and update $X \leftarrow X \cup \{a\}$.
            Reward $r \leftarrow M(X) - M(X \setminus \{a\})$; observe next state $s'$.
            Set done if $t = S$ or $\mathcal{A}(s') = \emptyset$.
            Store $(s, a, r, s', \texttt{done})$ in $\mathcal{M}_k$.
            Update $\theta_k$ from a mini-batch using TD target
                $$y = r + \gamma \max_{a' \in \mathcal{A}(s')} Q_{\bar{\theta}_k}(s', a') \quad \text{(or } y = r \text{ if done).}$$
        end for
        Decay exploration $\epsilon$; every $T$ episodes set $Q_{\bar{\theta}_k} \leftarrow Q_{\theta_k}$.
        if $e \bmod \Delta = 0$ then
            Evaluate greedy policy under binary and proportional allocation: $(M_k^{\text{bin}}, M_k^{\text{prop}})$; save best checkpoint for agent $k$.
        end if
    end for
end for
From all agents' best points $(M_k^{\text{bin}}, M_k^{\text{prop}})$, select champion by Manhattan distance to utopia $(100, 100)$:
    $$k^* = \arg\min_k \left(|100 - M_k^{\text{bin}}| + |100 - M_k^{\text{prop}}|\right).$$
return $\theta^* \leftarrow$ best checkpoint of agent $k^*$.
```

2.6 Benchmarking protocol and exact optimal computation

This subsection describes the experimental protocol used to benchmark the proposed deep Q-learning policy against exact optimal solutions under Pareto-Huff filtered customer behavior.

2.6.1 Parameter grid and instance generation

The trained policy is evaluated over a grid of problem configurations defined by the number of demand points N, the number of preexisting competitor facilities $|J|$and the selection ratio $S/|L|$, where S denotes the number of facilities to be placed and $|L|$ the number of candidate locations. Specifically, the grid consists of:

- instance sizes $N \in \{100, 200, 500, 1000\}$,
- number of competitors $|J| \in \{5, 10, 50\}$,
- selection ratios $(S, |L|) \in \{(3, 25), (5, 25), (10, 20)\}$.

All instances are generated on a square spatial domain of size 100×100 km. The distance decay parameter is fixed at $\lambda = 0.5$, facility attractiveness values are sampled uniformly from $[10, 80]$ and demand weights are sampled uniformly from $[1, 10]$.

For each configuration, two sets of experiments are conducted. First, 10 independent instances are generated to produce the scatter plots comparing deep Q-learning and optimal solution at instance level presented in Figures 2 and 3. Second, 100 independent instances are generated to compute averaged performance metrics reported in Table 1.

2.6.2 Exact baseline computation

Exact optimal solutions are computed for configurations where the number of possible facility combinations $\binom{|L|}{S}$ is sufficiently small to allow exhaustive enumeration. For these cases, all feasible subsets are evaluated using the same Pareto-Huff filter and market share computation as used during reinforcement learning.

The resulting exact solutions provide upper bounds on achievable market share under both binary and proportional allocation rules and serve as the benchmark for evaluating the learned policy.

2.7 Results and discussion

This subsection summarizes the empirical results obtained from the benchmarking procedure, covering both tabular statistics and graphical analysis.

2.7.1 Interpretation of performance metrics

Table 1 reports averaged results over 100 instances for each configuration. The column *Optimal (MS %)* denotes the market share achieved by the exact solution, computed as an equally weighted average of binary and proportional customer behavior. The column *Agent (MS %)* reports the corresponding market share achieved by the learned policy under the same weighting. The *Mean Error (%)* column shows the relative deviation of the agent's performance from the exact optimum. Computation times are reported as average calculation times per instance, where *Optimal Time (s)* corresponds to exhaustive enumeration and *Agent Time (s)* to a single forward pass of the trained policy. All timing results are averaged over 100 independent runs.

2.7.2 Graphical comparison and robustness analysis

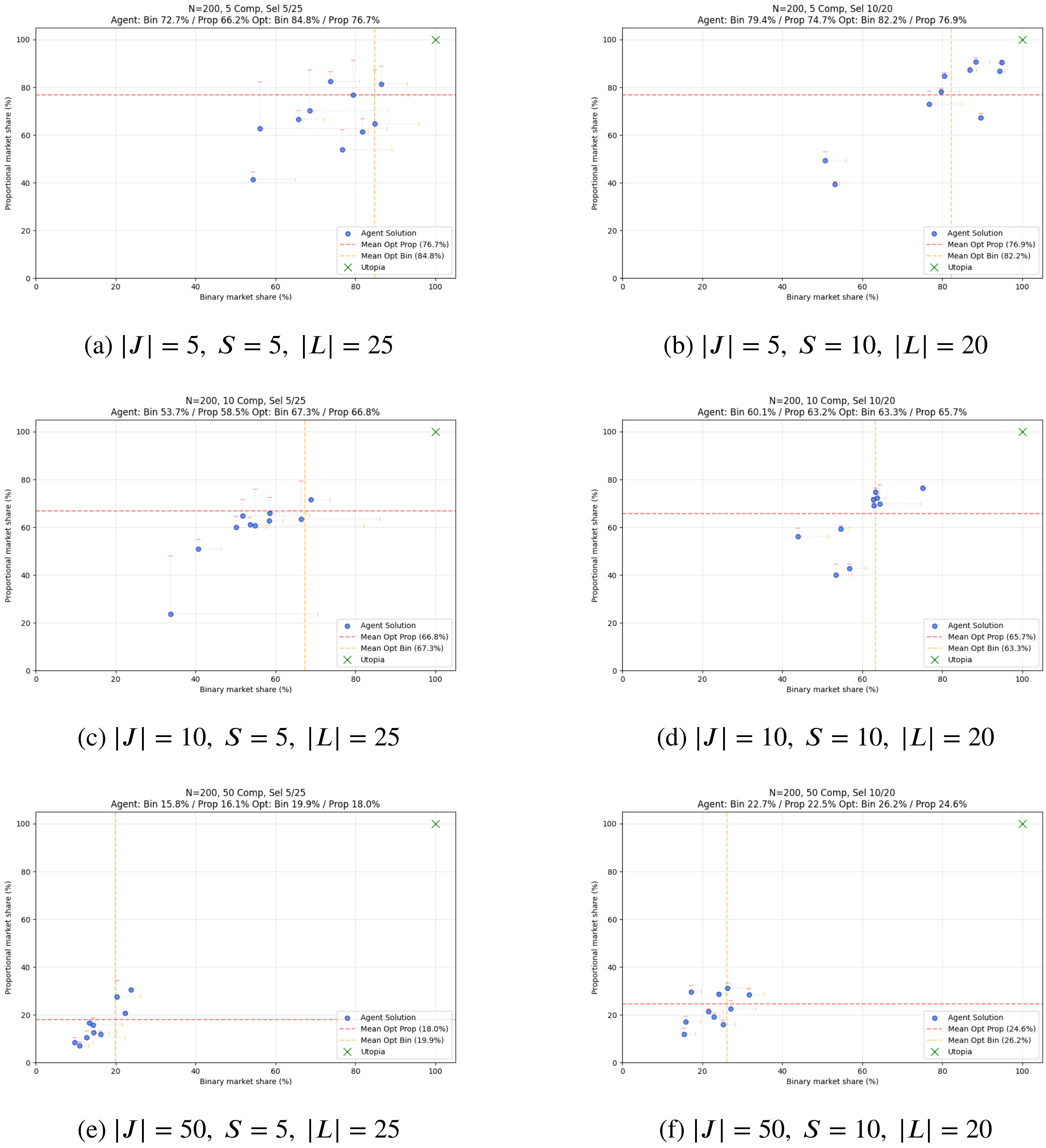
The scatter plots illustrate algorithm's effectiveness. Each blue point corresponds to the agent's solution for a single instance. For each instance, the exact optimal binary and proportional market shares are shown as small yellow and red markers, respectively, connected by a grey dashed line. The vertical yellow dashed line indicates the mean exact optimal binary market share, while the horizontal red dashed line indicates the mean exact optimal proportional market share over 10 instance. The utopia point (100, 100) is shown for reference.

Table 1. Deep Q-learning versus exact solution comparison table

N	|J|	|L|	S	Optimal (MS %)	Agent (MS %)	Mean Error (%)	Optimal Time (s)	Agent Time (s)
100	5	25	3	72.10	45.82	38.36	0.0080	0.0031
100	5	25	5	76.54	55.71	28.77	0.1998	0.0057
100	5	20	10	78.25	73.43	6.33	0.8272	0.0110
100	10	25	3	53.22	23.07	56.80	0.0083	0.0033
100	10	25	5	61.94	39.49	37.25	0.1984	0.0057
100	10	20	10	64.63	60.05	7.32	0.8477	0.0115
100	50	25	3	13.97	10.20	26.23	0.0144	0.0034
100	50	25	5	18.53	14.54	22.10	0.3431	0.0054
100	50	20	10	23.79	19.70	17.64	1.4782	0.0111
200	5	25	3	71.09	50.43	30.98	0.0081	0.0033
200	5	25	5	77.30	60.77	22.32	0.1935	0.0054
200	5	20	10	80.08	76.90	4.32	0.8426	0.0113
200	10	25	3	54.33	31.82	42.82	0.0078	0.0031
200	10	25	5	62.44	42.02	34.61	0.1864	0.0053
200	10	20	10	66.07	62.34	6.07	0.8744	0.0106
200	50	25	3	16.97	13.88	18.46	0.0236	0.0030
200	50	25	5	21.18	18.47	13.62	0.5949	0.0053
200	50	20	10	24.29	22.32	8.79	2.5388	0.0112
500	5	25	3	71.98	52.36	28.80	0.0079	0.0030
500	5	25	5	77.51	66.26	15.90	0.1951	0.0054
500	5	20	10	78.40	76.74	2.20	1.0550	0.0107
500	10	25	3	56.64	39.00	33.02	0.0097	0.0033
500	10	25	5	63.49	53.83	15.45	0.2526	0.0053
500	10	20	10	64.61	62.94	2.77	1.4236	0.0106
500	50	25	3	18.58	16.43	12.17	0.0573	0.0031
500	50	25	5	23.51	21.42	9.77	1.4269	0.0055
500	50	20	10	25.84	24.70	4.72	5.8413	0.0110
1000	5	25	3	72.37	60.49	17.36	0.0098	0.0031
1000	5	25	5	78.52	71.58	9.09	0.3040	0.0057
1000	5	20	10	77.86	76.33	2.09	1.8568	0.0112
1000	10	25	3	56.09	44.02	22.61	0.0169	0.0032
1000	10	25	5	62.86	56.42	10.60	0.4830	0.0054
1000	10	20	10	62.84	61.47	2.30	2.5253	0.0110
1000	50	25	3	19.40	16.85	14.68	0.1142	0.0034
1000	50	25	5	24.73	22.81	8.11	2.8280	0.0057
1000	50	20	10	25.92	25.17	3.26	11.4216	0.0114

Figures 2 and 3 illustrate the joint performance of the learned policy and the exact optimum under binary and proportional customer behavior. Each subfigure corresponds to a distinct combination of competitor count and facility selection size.

These plots visualize not only average performance gaps but also the variability of the agent's solutions relative to the exact Pareto optimal outcomes.



(a) $|J| = 5,\ S = 5,\ |L| = 25$

(b) $|J| = 5,\ S = 10,\ |L| = 20$

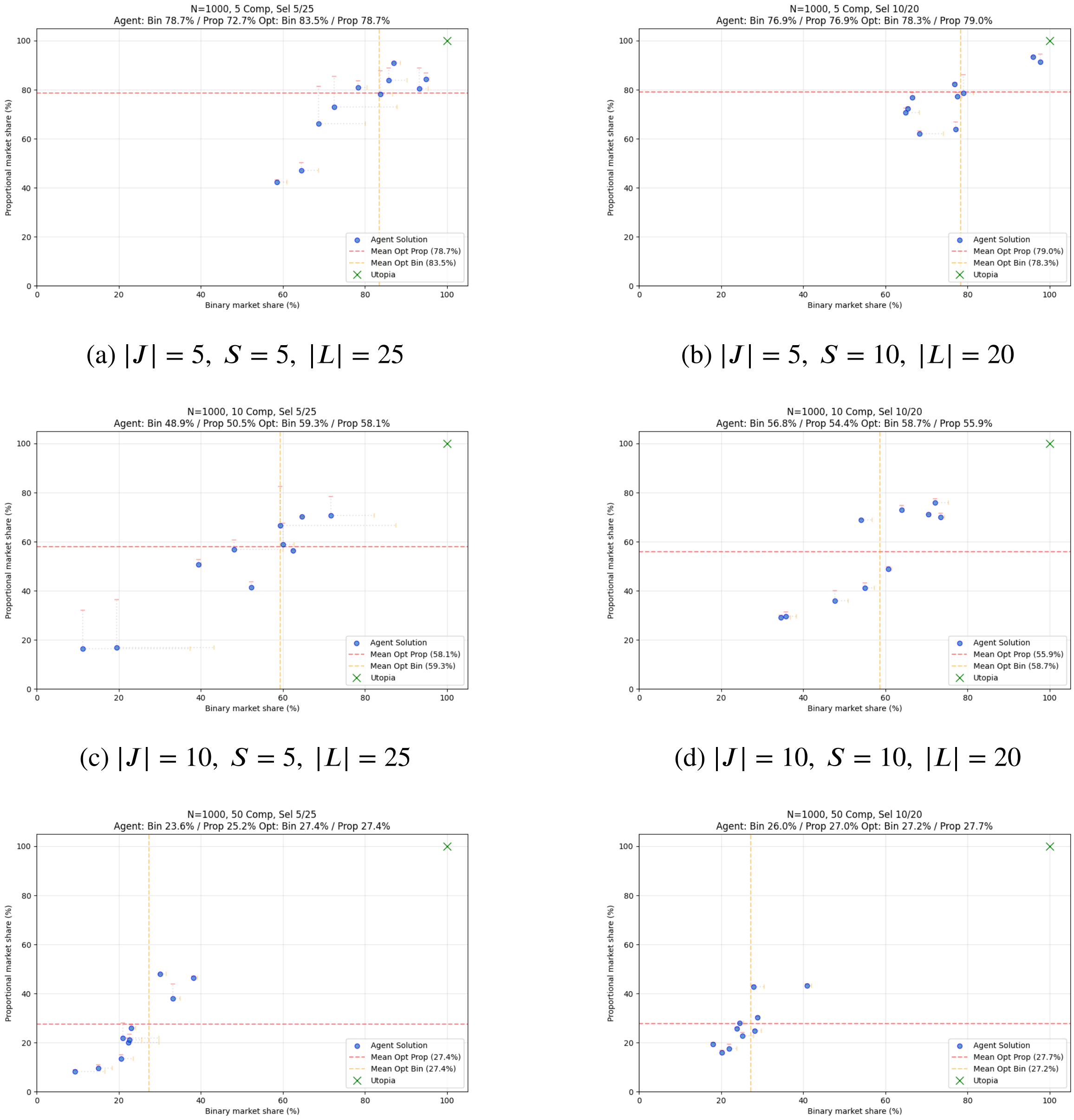
(c) $|J| = 10,\ S = 5,\ |L| = 25$

(d) $|J| = 10,\ S = 10,\ |L| = 20$

(e) $|J| = 50,\ S = 5,\ |L| = 25$

(f) $|J| = 50,\ S = 10,\ |L| = 20$

Figure 2. Binary vs proportional market share comparison for $N = 200$ under varying competition levels and selection sizes.



(a) $|J| = 5,\ S = 5,\ |L| = 25$

(b) $|J| = 5,\ S = 10,\ |L| = 20$

(c) $|J| = 10,\ S = 5,\ |L| = 25$

(d) $|J| = 10,\ S = 10,\ |L| = 20$

(e) $|J| = 50,\ S = 5,\ |L| = 25$

(f) $|J| = 50,\ S = 10,\ |L| = 20$

Figure 3. Binary vs proportional market share comparison for $N = 1000$ under varying competition levels and selection sizes.

2.7.3 Scalability and performance trends

Across all tested configurations, the learned policy consistently produces solutions in the order of milliseconds per instance, whereas exact enumeration ranges from fractions of a second to over ten seconds depending on problem size and selection ratio. This confirms that the proposed approach serves the function of optimization method, trading off optimality for substantial computational efficiency.

Performance varies across different problem settings. The agent performs best in settings where a larger number of facilities must be selected from a relatively constrained candidate set, such as $(S, |L|) = (10, 20)$. In these cases, the mean error is often below 5% and sometimes approaches the exact optimum. However, performance is less stable in sparse selection constraints, particularly when selecting 3 or 5 facilities from 25 candidates, where relative errors are higher and more variable. Performance in these cases improves with increasing instance size N, suggesting that the learned policy generalizes better as the spatial structure becomes denser.

As per original goal, selecting the champion policy by minimizing the Manhattan distance to the utopia point yields solutions that remain competitive under both binary and proportional evaluation. This confirms that the model selection strategy effectively avoids specialization to a single customer behavior assumption and promotes balanced performance across behavioral extremes.

Overall, while the learned policy does not consistently match the exact optimum, it demonstrates favorable scalability and robustness properties, supporting deep reinforcement learning as a promising research direction for competitive facility location under behavioral uncertainty.

2.8 Limitations

The experimental evaluation is based exclusively on synthetic instances generated within a fixed geographic bounding region and under simplified assumptions on demand and attractiveness distributions. Although this setup enables controlled benchmarking, additional validation on real world data would be necessary to assess practical robustness. Furthermore, the model exhibits instability and degraded performance in certain settings, indicating that further hyperparameter tuning and architectural refinement may be required. Finally, while deep Q-learning scales well computationally, it does not provide optimality guarantees and performance remains sensitive to training distribution and exploration dynamics.

Conclusions and Recommendations

This thesis investigated the applicability of deep reinforcement learning to the competitive facility location problem for entering firm under uncertain customer behavior, with particular emphasis on Pareto-Huff filtered choice models. Rather than optimizing for a single behavioral assumption, the proposed approach aimed to learn a placement policy that performs consistently across binary and proportional allocation types.

The experimental results demonstrate that deep Q-learning can successfully learn non-trivial facility placement strategies that generalize across instance sizes and competitive settings. Although the learned policies do not consistently attain almost optimal solutions, they achieve competitive market share outcomes while offering significant reductions in computation time compared to exhaustive enumeration. This confirms the suitability of reinforcement learning as optimization approach for large scale competitive location problems.

A key contribution of this work is the explicit treatment of behavioral uncertainty through multiple customer choice rules. The use of Manhattan distance as a model selection criteria proved effective in identifying policies that balance performance across competing objectives, avoiding specialization to a single behavioral assumption.

At the same time, the results highlight important limitations. Policy performance varies substantially across settings, particularly in sparse selection settings with few facilities and many candidate locations. In such cases, the learned solutions exhibit higher variance and larger gaps relative to the exact optimum. This indicates that the current model and training configuration do not yet fully capture the complex strategic interactions present in highly competitive or constrained environments.

Overall, while the proposed approach does not fully resolve robustness challenges in competitive facility location, it establishes a viable foundation for learning based solution methods and provides clear directions for further refinement.

Future Work

Several avenues for future research emerge from the findings of this thesis.

First, the robustness of learned policies could be improved through more extensive tuning of the reinforcement learning framework. This includes exploration of network architectures, hyperparameters and reward shaping strategies tailored specifically to facility location dynamics. Alternative value based or policy gradient methods may offer improved stability over deep Q-learning in non-smooth environments induced by Pareto-Huff filtering.

Second, the current study relies exclusively on synthetically generated data drawn from a fixed geographic region. Incorporating real world spatial data, such as GIS demand distributions and realistic travel networks, would enable a more rigorous assessment of practical applicability. Training on more diverse spatial and competitive scenarios may also improve generalization and reduce sensitivity to specific instance characteristics.

Third, robustness could be expanded beyond customer behavior uncertainty. Future work could introduce additional sources of stress, such as extremely volatile demand levels, aggressive competitor placement strategies, economic downturn scenarios, or highly distance sensitive customer populations. Evaluating performance under such uncertainties would provide a more comprehensive notion of robustness relevant to real world decision making.

Fourth, alternative multi-objective model selection strategies could be explored. While the Manhattan distance to the utopia point provides a simple and interpretable criterion, other approaches, such as dominance based selection, different distance metric or preference weighted objectives may better capture priorities in competitive settings.

Finally, the current formulation assumes static competitors and a one shot placement problem. Extending the framework to dynamic competitive environments, where rival firms also adapt their locations over time, represents a challenging but highly relevant direction for future research. Such extensions would move closer to realistic market dynamics and further test the limits of reinforcement learning in strategic spatial competition.

In summary, although the present approach does not yet deliver fully robust solutions across all regimes, the results suggest that reinforcement learning remains a promising research direction for competitive facility location. With improved training setups, richer data and broader uncertainty modeling, learning based methods may eventually complement or replace classical optimization techniques in complex, large scale spatial decision problems.

Declaration of Artificial Intelligence

Artificial intelligence tools (ChatGPT, Gemini) were used to find relevant references, spellchecking, suggest methodology deficiencies and format plots and tables.

References

[1] Marvin A Arostegui Jr, Sukran N Kadipasaoglu, and Basheer M Khumawala. An empirical comparison of tabu search, simulated annealing, and genetic algorithms for facilities location problems. *International Journal of Production Economics*, 103(2):742–754, 2006.

[2] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421, 2021.

[3] Dimitris Bertsimas, David B Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM review*, 53(3):464–501, 2011.

[4] Wei-Yu Chiu, Gary G Yen, and Teng-Kuei Juan. Minimum manhattan distance approach to multiple criteria decision making in multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 20(6):972–985, 2016.

[5] Richard Church and Charles ReVelle. The maximal covering location problem. In *Papers of the regional science association*, volume 32, pages 101–118. Springer-Verlag Berlin/Heidelberg, 1974.

[6] Claude d'Aspremont, J Jaskold Gabszewicz, and J-F Thisse. On hotelling's" stability in competition". *Econometrica: Journal of the Econometric Society*, pages 1145–1150, 1979.

[7] Zvi Drezner and Horst W Hamacher. *Facility location: applications and theory*. Springer Science & Business Media, 2004.

[8] Pascual Fernández, Blas Pelegrín, Algirdas Lančinskas, and Julius Žilinskas. New heuristic algorithms for discrete competitive location problems with binary and partially binary customer behavior. *Computers & Operations Research*, 79:12–18, 2017.

[9] Nalan Gülpınar, Dessislava Pachamanova, and Ethem Çanakoğlu. Robust strategies for facility location under uncertainty. *European Journal of Operational Research*, 225(1):21–35, 2013.

[10] S Louis Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations research*, 12(3):450–459, 1964.

[11] Harold Hotelling. Stability in competition. *The Economic Journal*, 39(153):41–57, 1929.

[12] David L Huff. Defining and estimating a trading area. *Journal of Marketing*, 28(3):34–38, 1964.

[13] Wouter Kool, Herke Van Hoof, and Max Welling. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*, 2018.

[14] Algirdas Lančinskas, Julius Žilinskas, Pascual Fernández, and Blas Pelegrín. Population-based algorithm for discrete facility location with ranking of candidate locations. *Journal of Computational and Applied Mathematics*, 457:116304, 2025.

[15] Wenhua Li, Guo Zhang, Tao Zhang, and Shengjun Huang. Knee point-guided multiobjective optimization algorithm for microgrid dynamic energy management. *Complexity*, 2020(1):8877008, 2020.

[16] Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3):293–321, 1992.

[17] Nina Mazyavkina, Sergey Sviridov, Sergei Ivanov, and Evgeny Burnaev. Reinforcement learning for combinatorial optimization: A survey. *Computers & Operations Research*, 134:105400, 2021.

[18] Nimrod Megiddo and Kenneth J Supowit. On the complexity of some common geometric location problems. *SIAM journal on computing*, 13(1):182–196, 1984.

[19] Nimrod Megiddo and Arie Tamir. On the complexity of locating linear facilities in the plane. *Operations Research Letters*, 1(5):194–197, 1982.

[20] Pitu B Mirchandani and Richard L Francis. *Discrete location theory*. Wiley, 1990.

[21] Nenad Mladenović, Jack Brimberg, Pierre Hansen, and José A Moreno-Pérez. The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 179(3):927–939, 2007.

[22] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[23] Peter H Peeters and Frank Plastria. Discretization results for the huff and pareto-huff competitive location models on networks. *Top*, 6(2):247–260, 1998.

[24] Charles ReVelle. The maximum capture or "sphere of influence" location problem: Hotelling revisited on a network. *Journal of Regional Science*, 26(2):343–358, 1986.

[25] Charles S Revelle, Horst A Eiselt, and Mark S Daskin. A bibliography for some fundamental problem categories in discrete location science. *European journal of operational research*, 184(3):817–848, 2008.

[26] Charles S ReVelle and Ralph W Swain. Central facilities location. *Geographical analysis*, 2(1):30–42, 1970.

[27] Francisco Saldanha-da Gama and Shuming Wang. Facility location under uncertainty. *International Series in Operations Research and Management Science*, 2024.

[28] Daniel Serra and Rosa Colomé. Consumer choice and optimal locations models: formulations and heuristics. *Papers in Regional Science*, 80(4):439–464, 2001.

[29] Arthur Smithies. Optimum location in spatial competition. *Journal of Political Economy*, 49(3):423–439, 1941.

[30] Lawrence V Snyder. Facility location under uncertainty: a review. *IIE transactions*, 38(7):547–564, 2006.

[31] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[32] Michael B Teitz and Polly Bart. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations research*, 16(5):955–961, 1968.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[34] Natalia Vesselinova, Rebecca Steinert, Daniel F Perez-Ramirez, and Magnus Boman. Learning combinatorial optimization on graphs: A survey with applications to networking. *IEEE Access*, 8:120388–120416, 2020.

[35] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

[36] Wuyang Yu. Robust competitive facility location model with uncertain demand types. *Plos one*, 17(8):e0273123, 2022.

[37] Zhonghao Zhao, Carman KM Lee, Xiaoyuan Yan, and Haonan Wang. A deep reinforcement learning framework for capacitated facility location problems with discrete expansion sizes. *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2023.

[38] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.