MASTERARBEIT | MASTER'S THESIS

Titel | Title

P-Median Model Under Stochastic Conditions for Optimal Medical Facility Placement: A Case Study on the Vienna Marathon

verfasst von | submitted by

Stefan Kovacevic B.Sc.

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 915
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Betriebswirtschaft
Betreut von | Supervisor:	Univ.-Prof. Mag. Dr. Karl Franz Dörner Privatdoz.

Abstract

Medical coverage is a critical component of sports logistics and ensures robust service for all participants. Yet existing work rarely optimizes station siting at route events under multiple uncertainties. Therefore, this study determines the optimal locations of medical stations along the Vienna City Marathon route by minimizing the total weighted distance to runners with each route point weighted by its estimated probability of medical need. Using official route coordinates, Haversine distances, a km level injury risk profile, historical ranges for weather and participants and two spatial resolutions (1,000 m and 100 m), this research implements a program that identifies optimal placements. Specifically, a weighted p-median mixed-integer model is solved on 1,000 m and 100 m grids with Gurobi across multiple scenarios, while a genetic algorithm approximates the 100 m case. Key findings show that a finish area cluster *(km 38–42)* plus one mid-course node *(km 17–18)* consistently minimizes the weighted distance. The zones remain stable across different scenarios, although exact metres shift on the 100 m grid and the genetic algorithm reproduces the exact pattern near the optimum. The current real world plan is evenly spaced and yields a worse objective than the optimized layouts. If resources permit, seven stations emerge as an effective expansion point. Finally, the study provides actionable zones rather than single points and supports scenario triggered plans. Consequently, the workflow is transferable to other route events, which offers practical siting guidance for marathon organizers.

Keywords: sports logistics, marathon, medical stations, optimal placement, p-median, weighted distance, uncertainty, scenario analysis

Abstract (Deutsch)

Die medizinische Versorgung ist ein zentraler Bestandteil der Sportlogistik und gewährleistet einen robusten Service für alle Teilnehmenden. Gleichwohl optimiert die bestehende Forschung die Standortwahl von Stationen bei Routenveranstaltungen nur selten unter mehreren Unsicherheiten. Deshalb ermittelt diese Studie die optimalen Standorte medizinischer Stationen entlang der Route des Vienna City Marathon, indem die gesamte gewichtete Distanz zu den Läuferinnen und Läufern minimiert wird, wobei jeder Routenpunkt mit seiner geschätzten Wahrscheinlichkeit eines medizinischen Bedarfs gewichtet ist. Unter Verwendung offizieller Routenkoordinaten, Haversine Distanzen, eines kilometerbasierten Verletzungsrisikoprofils, historischer Bandbreiten für Wetter und Teilnehmendenzahl sowie zweier räumlicher Auflösungen (1.000 m und 100 m) entwickelt diese Studie ein Programm, das optimale Platzierungen identifiziert. Konkret wird ein gewichtetes p-Median-gemischt-ganzzahliges Modell auf 1.000 m und 100 m Rastern mit Gurobi über mehrere Szenarien gelöst, während ein genetischer Algorithmus den 100 m Fall approximiert. Zentrale Ergebnisse zeigen, dass ein Cluster im Zielbereich (*km 38–42*) plus ein Knoten im Streckenmittelteil (*km 17–18*) die gewichtete Distanz konsistent minimiert. Die Zonen bleiben szenarioübergreifend stabil, obwohl sich die exakten Meter auf dem 100 m Raster verschieben während der genetische Algorithmus das Muster nahezu optimal reproduziert. Im derzeitigen realen Plan sind die Standorte gleichmäßig verteilt und weisen einen schlechteren Zielfunktionswert auf als die optimierten Layouts. Wenn es die Ressourcen erlauben, erweisen sich sieben Stationen als wirksamer Ausbaupunkt. Letztendlich liefert die Studie umsetzbare Zonen statt einzelner Punkte und unterstützt durch Szenarien ausgelöste Pläne. Folglich ist der Prozess auf andere Routenveranstaltungen übertragbar und bietet praktische Standortleitlinien für Marathonorganisatorinnen und -organisatoren.

Schlagwörter: Sportlogistik, Marathon, medizinische Stationen, optimale Platzierung, P-Median, gewichtete Distanz, Unsicherheit, Szenarioanalyse

Table of Contents

List of Figures

List of Tables

List of Abbreviations

GA..Genetic algorithm
km ....Kilometre
MILP.. Mixed-integer linear program
m .... Meter
s .... Second
V1.. Variant 1
V2.. Variant 2
V3.. Variant 3

1. Introduction

Sports logistics management refers to the planning, implementation and control of goods, capacity, services and information during a sports event (Herold et al., 2020a). It is one of the most under researched areas in sport management (Pott et al., 2023) and sits at the intersection of logistics and sports management. Recent studies have begun to conceptualize its scope and adopt value chain thinking and cross functional perspectives to integrate logistics into sport event management (Herold et al., 2020a; Pott et al., 2023). In this view, sports logistics ensures that resources are designed and allocated in an optimal way for safe and efficient mass events. In essence, it connects two fields and provides a framework to deliver the right resources to the right place at the right time. The importance of sports logistics is evident in large scale events. For example, organizing committees invested approximately 1.2 billion dollars in logistics and transportation for the 2012 London Olympics and at the 2008 Beijing Olympics they prepared 104 hospitals and deployed 400,000 volunteers to support spectators and athletes (Pott et al., 2023). Such efforts illustrate why effective logistics planning is crucial because even minor failures can have major consequences for event outcomes. As one report notes, “the consequences of operations failure have huge implications for the outcome of the on-field performance”, which underscores that precise logistics operations are required to orchestrate sporting events successfully (Pott et al., 2023). Operational failures, such as poor transportation and failing technology, were visible during the 2010 Commonwealth Games and deemed predictable and preventable for the $6.8 billion event (Kauppi et al., 2013). Mass participation events like city marathons likewise rely on meticulous coordination. Emergency logistics research emphasizes optimal positioning of responders to ensure low response times and effective service coverage (Huizing et al., 2020). This highlights a clear research gap and more scholarly work is needed on the optimization of logistics for sporting events.

Marathons present a distinctive logistics problem because the event unfolds along a route with dispersed spectators and athletes. The global popularity of these events is immense with more than 800 marathons organized globally annually and often draws both world champions and recreational athletes to the same starting line (Burke et al., 2025). Consequently, participants are competing in an entirely different race in terms of absolute and relative physiological demands, highlighted by the fact that the average recreational runner is typically on the course for 80 to 85 minutes longer than elite competitors (Burke et al., 2025). The courses pass through different neighbourhoods and road types. This means that access for service vehicles changes and bottlenecks may occur at the start, on narrow sections and at the finish. Unlike enclosed venue sports, services and support must be distributed over a wide area rather than concentrated in one hub. As a result, the route layout, city setting and crowd movement create uneven demand for assistance. Medical incidents often cluster in certain places and times as groups form and as fatigue or heat stress increase along the course. (Nguyen et al., 2008; Tan et al., 2014; Chiampas & Jaworski, 2009). Crowd and security management add further complexity because open city events are harder to secure than single site events, which reinforces the need for clear roles, communication and coordinated response among stakeholders (Hall et al., 2019).

Aid stations and medical coverage must therefore be positioned across the route, typically with higher capacity near late race hotspots and the finish where demand surges. Organizers deploy fixed posts at intervals and mobile responders who move through crowds or narrow streets so that response remains fast even when roads are closed or congested. Evidence from large races shows that a small share of participants requires care, where a disproportionate number of cases accumulates late in the race and at the finish area. This supports concentrating resources in the final part while a baseline remains elsewhere (Nguyen et al., 2008; Tan et al., 2014; Chiampas & Jaworski, 2009; Brock et al., 2025). The Vienna City Marathon illustrates these challenges for example. It has a multi-district urban route with varying access constraints and a demand peak toward the end. This calls for deliberate station placement and mobile capability to ensure rapid coverage despite the event’s scale and dispersion (Brock et al., 2025).

Medical services are a core component of sports logistics because they safeguard athlete welfare, sustain public confidence and satisfy regulatory expectations. In endurance road races, timely access to care is primarily. Planners therefore aim to keep the distance from any point on the course to the nearest aid resource as small as possible. This aligns with facility location principles and with the p-median objective of minimizing total distance between demand points and stations (Hakimi, 1964; ReVelle & Swain, 1970; Jia et al., 2005). Marathons amplify this challenge since the event unfolds along an open route with dispersed participants and spectators. Demand for care is not uniform but incidents cluster in space and time as fatigue, dehydration and thermal stress accumulate with pronounced surges late in the race and at the finish area (Nguyen et al., 2008; Tan et al., 2014; Chiampas & Jaworski, 2009; Brock et al., 2025). Effective governance and coordination mechanisms such as the Incident Command System, which is a standardized structure that integrates medical, safety and transport units under unified command, are essential to orchestrate distributed teams and to dispatch resources swiftly on crowded or partially closed roads (Yao et al., 2017). At the same time, strict rescue time aims and reliability requirements make distance reduction more than a convenience. Exceeding time thresholds can degrade outcomes. This motivates layouts that remain robust across plausible race day conditions (Shen et al., 2021). In practice, organizers combine fixed posts at regular intervals with mobile responders who penetrate dense sections and they scale capacity toward late race hotspots, while preserving baseline coverage elsewhere. Risk aware and scenario based planning is therefore critical to decide where stations should be located and how they should be staffed on marathon day (Nguyen et al., 2008; Tan et al., 2014; Hall et al., 2019; Shen et al., 2021).

Open road marathons create a logistics environment that differs from enclosed venues. Athletes and spectators are dispersed along a 42.195 km course, access for support vehicles is uneven and demand for assistance varies across space and time (Nguyen et al., 2008; Tan et al., 2014; Chiampas & Jaworski, 2009). Although all participants cover the same distance on the same day, elite and recreational competitors experience vastly different physiological and psychological demands (Burke et al., 2025). Within sports logistics, medical services are central because timely care underpins athlete safety, public confidence and regulatory compliance. Hence organizers strive to minimize response distance through deliberate placement of fixed posts and mobile teams, which matches the p-median objective (Hakimi, 1964; ReVelle & Swain, 1970; Jia et al., 2005). Uncertainty factors magnify risk and are decisive for medical safety. Hotter race day conditions raise casualty rates and shift demand toward late race hotspots (Tan et al., 2014; Nguyen et al., 2008). Larger starting fields increase the absolute number of encounters. Injury risk typically rises in the final third so incidents cluster near the finish (Nguyen et al., 2008; Brock et al., 2025). These drivers also interact and a hot and high participation edition can overwhelm layouts that fit an average day. Governance via the mentioned Incident Command System supports coordinated dispatch on crowded or partially closed roads (Yao et al., 2017). Yet strict rescue time aims mean spatial design must remain reliable when conditions get worse. Therefore, the exceeding of thresholds degrades outcomes (Shen et al., 2021). The Vienna City Marathon exemplifies these dynamics. A multi-district route with varied access and a late peak in medical demand makes decentralized and scenario aware station placement essential (Brock et al., 2025).

Sports logistics is relevant but yet under researched, which creates a need for operational methods that support real decisions at events (Herold et al., 2020a; Pott et al., 2023). Marathons add specific complexity because the race unfolds on city streets with dispersed athletes and spectators, uneven access and crowd movement that shifts demand across space and time (Nguyen et al., 2008; Tan et al., 2014). Medical service is central to safety and credibility, while fast access to care requires short response distance. This aligns with a facility location view and with the p-median objective to minimize total distance between demand points and stations (Hakimi, 1964; ReVelle & Swain, 1970; Jia et al., 2005). Uncertainty raises the stakes. Weather, field size and route specific injury risk interact and push medical demand toward late race hotspots. Strict rescue time goals further reinforce the need for robust layouts under variable conditions (Shen et al., 2021).

Against this backdrop, the study targets a concrete case. The aim is to place medical stations along the Vienna City Marathon route so that average response distance stays low across plausible race day scenarios while inputs remain transparent for organizers. Hence the research question is: *Where should medical stations be located along the Vienna City Marathon route under uncertain weather, participant numbers and injury risk probability?* A secondary aim is to assess how sensitive optimal locations are to the number of stations and to the scenario mix.

2. Literature review

This chapter reviews theoretical and empirical research related to logistics optimization in sport events and focuses on models that support efficient and resilient planning. It begins by outlining the main logistical challenges in sports event management before introducing quantitative optimization frameworks.

2.1. Sports Logistic Challenges

Sport events are complex projects with definitive end dates (Cieśliński et al., 2016). In event management, time priority frequently outweighs quality and cost, as delays can endanger the event (Pott et al., 2023; Shen et al., 2021). This foundational view motivates a domain specific structure. In the case of this research, such structured logistical understanding forms the basis for analysing how medical resources can be optimally placed along the Vienna Marathon route.

Although logistics is central to the organization of sport events, the academic field of sports logistics management remains under researched compared with areas such as marketing or legacy topics within sport management (Herold et al., 2020a; Pott et al., 2023). There is a lack of a specific definition and a suitable theoretical or conceptual framework to unify the various logistical activities (Herold et al., 2020a). Traditional sport management models often insufficiently cover inbound and outbound logistics (Herold et al., 2020a). Logistic capabilities should be regarded as a strategic resource that can contribute to competitive advantage (Herold et al., 2020a). This study builds on that perspective by treating medical service logistics as a strategic element that directly affects participant safety and overall event quality.

Against this backdrop, a four pillar framework structures the domain into venue logistics, sports equipment logistics, athlete logistics and fan or spectator logistics (Herold et al., 2020a), which also guides this research. In line with the research question, where to place emergency stations during a sport event, this study focuses on venue logistics. Additionally, logistics design depends on the event type such as local or regional versus major or mega, because infrastructure readiness, staff mix, audience scale and event duration differ substantially across categories (Herold et al., 2020a). Case applications from Super League Football demonstrate how activities, organizational structures and resources can be mapped to systematically manage logistics behind the scenes of elite events (Herold et al., 2020b). In the context of the Vienna Marathon, this corresponds to the placement of logistical and medical processes across the 42 km course to ensure efficient emergency coverage and coordinated response.

Within the four pillars, first typical challenges include managing high traffic volumes and sudden flow increases during sports mega events (Shen et al., 2021) and jointly optimizing the location and capacity of temporary parking lots or park and ride facilities (Ruan et al., 2016). The objective is to increase the number of visitors who can complete their planned trips within the available time budget (Ruan et al., 2016), while avoiding parking shortfalls or surpluses that diminish the spectator experience (Herold et al., 2020a; Herold et al., 2020b) in fan logistic perspective. Second, the venue logistics with the coordination of complex internal operations such as security, VIP suites and crowd management (Herold et al., 2020a; Herold et al., 2020b). Then athlete logistics in optimizing travel planning (e.g., across time zones) to safeguard readiness and performance (Herold et al., 2020a; Herold et al., 2020b). The last is the sports equipment logistics by handling significant volumes and sizes of specialized equipment, where costs are typically of lower concern than ensuring utility for players (Herold et al., 2020a). Among these four categories, the present work directly contributes to venue logistics by optimizing emergency medical station locations along the marathon route, where time critical access is essential.

Mega events such as the Olympic Games and the FIFA World Cup are among the largest non-military logistical operations worldwide and despite their transient nature require long planning phases up to ten years in some cases (Herold et al., 2020a; Shen et al., 2021). In sport management, logistics covers the planning, control and execution of receiving, tracking, storage, transport, distribution, installation, return of all materials and equipment for external clients (e.g., media) as well as the functional areas of the organizing committee (Minis et al., 2006; Minis et al., 2009). Key difficulties include time and resource constraints with strict non-negotiable deadlines, fixed schedules and limited budgets and personnel (Minis et al., 2009; Kauppi et al., 2013). They also include demand uncertainty and diversity with peaks around event times and a broad spectrum of items from sailboats to medical goods and technology with late requests by internal and external clients (Minis et al., 2006; Minis et al., 2009). Limited data arises because there is little transferable knowledge from previous Olympic games due to rapid Organizing Committee of the Olympic Games dissolution (Minis et al., 2006). Operational complexity leaves virtually no margin for error, since visible and preventable failures in transport or technology are globally exposed (Minis et al., 2009; Kauppi et al., 2013). Coordination and stakeholder management require multiorganization collaboration and interorganizational relations, yet vertical and horizontal coordination across many stakeholders remains challenging (Mosadeghi et al., 2020).

Parallels from the broader logistics context reinforce these pressures. The problem in logistics as described by Takashima (2023), which highlights systemic challenges in the transportation sector due to labour shortages, stricter working hour regulations and delayed cargo delivery. The report emphasizes that maintaining operational efficiency under such restrictions requires coordinated efforts among shippers, receivers and logistics companies, as well as initiatives to shorten waiting and handling times. These dynamics parallel the complexity of sports event logistics, where personnel coordination, efficient routing and time sensitive responses are critical to ensure unproblematic operations and medical service readiness (Takashima, 2023). In the Vienna Marathon setting, similar operational dependencies exist between organizers, medical teams and emergency responders, which underlines the need for clear coordination structures and optimized resource allocation. In response to ongoing supply chain volatility, excess inventory and rising capital costs, sporting goods companies are increasingly adopting advanced integrated business planning and data analytics to enhance demand forecasting and align supply more closely with sales (McKinsey & Company, 2024).

Similarly, mass participation events such as city marathons have to be managed as projects with phased planning, implementation and control. This again requires coordinated structures for administration, communications, broadcasting, security, medical services, accommodation, catering, budgeting, procurement and post event reporting (Witkowski et al., 2016). As indicated, there are many facets to organizing sporting events, while this research concentrates on where to place medical services. Security and safety arrangements form an integral part of logistics planning and include medical services within the overall organisational and operational structure of the event (Witkowski et al., 2016). In the context of mega events, authorities often need to integrate temporary emergency facilities alongside fixed ones because of the sudden nature of incidents (Shen et al., 2021). Rescue time targets are frequently given strict priority over monetary cost minimization, which can necessitate options such as air rescue (Shen et al., 2021).

In addition, large road races should adopt the previous mentioned Incident Command System to integrate public safety, hospitals and race medical teams (Yao et al., 2017). This approach reduces the risk of mass casualty overload and enables rapid coordinated decisions (Yao et al., 2017). Moreover, contextual factors such as weather, altitude, concurrent regional events, course layout, terrain, participant volumes, fitness levels and acclimatization requirements affect injury rates, pacing, dropouts and the demand placed on logistics and medical systems (Yao et al., 2017). Accordingly, these drivers should be quantified and incorporated into planning (Yao et al., 2017). Therefore, this study considers selected factors that influence the placement of emergency stations. Kristiansen et al. (2016) demonstrated that co-hosted international youth events require robust stakeholder networks and clear communication structures to overcome organizational complexity and coordination challenges. Their findings underline the significance of local partnerships and adaptive management for successful event execution (Kristiansen et al., 2016). Likewise, this research integrates stochastic factors such as weather, participant volume and injury risk to mirror real world uncertainty in planning medical station placement along the Vienna Marathon course.

Finally, technology plays an active role in supporting effective event management. Predictive and integrated tools, such as dashboard systems with intelligent controls, communication networks and augmented reality, can help to anticipate problems and enable quick responses (Cieśliński et al., 2016). A process oriented management model with a clear map of main and support processes helps marathon organizers make better strategic decisions and improve sustainability (García-Vallejo et al., 2020). To achieve excellence in running events, sustainability should be part of all logistics processes and stakeholder coordination (García-Vallejo et al., 2020). Including sustainability goals together with response time and capacity targets ensures that logistics systems stay reliable under real conditions and continue to meet stakeholder expectations (Herold et al., 2020a).

2.2. P-Median Model in Sports Logistic and Event Planning

Building on the identified logistical complexities, the next section examines the p-median model, which is one of the most widely applied quantitative frameworks for facility placement. This model provides the mathematical foundation for addressing location allocation challenges in large scale sports events such as marathons.

The p-median model is a classic discrete facility location problem that selects *p* facility locations and assigns each demand point to one facility so that the sum of distance from all nodes to their nearest open facility is minimized (Huizing et al., 2020; Hakimi, 1964). In the classic formulation each demand is served by one and only one centre while the goal is to minimize the average distance (ReVelle & Swain, 1970). The p-median problem is a fundamental spatial optimization problem central to facility location analysis in urban decision making (Liang et al., 2024). The core concept is to select *p* facility sites from candidate locations to minimize the total weighted distance between all demand points and their assigned closest facility. It is a discrete location problem (Liang et al., 2024; ReVelle & Swain, 1970). The objective in the Integer Linear Programming formulation is to minimize $\sum d_{ij}x_{ij}$ and constraints that ensure each demand point is assigned to exactly one facility and exactly *p* facilities are opened (Liang et al., 2024). In this study, the same optimization logic is applied to identify medical service points along the Vienna Marathon route by minimizing the total weighted distance between runners and their nearest emergency station.

The search for facilities that minimize the sum of distance can be traced to the 1600s (Gwalani et al., 2021). Historical foundations trace back to Hakimi and his graph medians (Hakimi, 1964) and to ReVelle and Swain (1970), who presented a network p-median linear programming (ReVelle & Swain, 1970). The discrete p-median problem that determines optimum locations for switching centres was first introduced by S. L. Hakimi in 1964 (Jia et al., 2005; Karatas & Yakici, 2018). ReVelle and Swain in 1970 formulated the p-median as the first mathematical model using linear integer programming (Jia et al., 2005; Güden & Süral, 2019; ReVelle & Swain, 1970). In emergency services planning, the total average distance should be minimized to increase accessibility and effectiveness. The p-median problem aims to determine facility locations that minimize total weighted distances between demand points and centres (ReVelle & Swain, 1970; Gwalani et al., 2021). By extending these historical principles to the marathon context, this work adapts a well-established mathematical foundation to a dynamic sporting environment with human safety as the central objective.

Beyond the deterministic p-median problem, stochastic versions exist because location decisions are costly and hard to reverse and parameters such as costs demand and distances fluctuate over time. The stochastic p-median problem aims to minimize expected costs and can be embedded in multi commodity or multi objective frameworks (Snyder, 2006). Liang et al. (2025) developed a deep reinforcement learning framework that expands the traditional p-median model using an adaptive encoder–decoder structure. Their Attention Integrated Adaptive Model includes attention mechanisms to better capture the relationships between demand points and facilities. This approach improves optimization performance compared to heuristic and classical solvers. When applied to hospital location planning in Beijing, it successfully reduced total travel distances and showed strong potential for real applications in medical facility placement (Liang et al., 2025).

From a computational perspective, the p-median problem is classified as a nondeterministic polynomial-time hard problem on a general network for an arbitrary number of facilities (Gwalani et al., 2021; Jia et al., 2005; Güden & Süral, 2019; Jánošíková et al., 2021). Exact solutions can be obtained by using linear programming with integrality constraints and they often require branch and bound or branch and cut techniques (Güden & Süral, 2019; Gwalani et al., 2021). Because of exponential complexity these exact methods become infeasible for larger problems (Gwalani et al., 2021). Heuristics are commonly used in operations research to find a good solution fast for large instances (Gwalani et al., 2021). Interchange algorithms and their variants are also widely used and can produce solutions close to optimal while studies evaluate their performance by objective value runtime and stability across spatial demand patterns (Gwalani et al., 2021). Construction algorithms build a solution iteratively by adding one facility per iteration and include greedy addition (Gwalani et al., 2021). Metaheuristics such as genetic algorithms (GA), tabu search and simulated annealing are also used although gains over classic exchange heuristics are often not substantial and they require more complex input parameters (Gwalani et al., 2021). Consequently, this research applies a metaheuristic method and compares its performance with other solution approaches.

Dynamic extensions broaden the modeling scope. Güden and Süral (2019) developed a dynamic p-median formulation that allows facilities to open relocate or close across a planned horizon. This decides the routes for mobile facilities and allocates the demand to open facilities to minimize total cost with the movements (Güden & Süral, 2019). The solution approach introduces a branch and price algorithm and a dynamic radius formulation based on discretized distances to improve tractability for larger instances (Güden & Süral, 2019). Dynamic p-median models plan the location of mobile and immobile facilities over a planning horizon and minimize transportation and movement costs (Güden & Süral, 2019). Although the Vienna Marathon is a one day event, the idea of dynamic adjustment is conceptually relevant, since medical resource needs may shift over time as temperature and runner density change along the route.

Taken together, these models show two complementary capabilities that are useful for endurance events. First, they support static or periodically resolved facility locations to place posts efficiently. Second, they enable operational replanning during the event to maintain low expected response times when conditions change (Huizing et al., 2020; Güden & Süral, 2019). The p-median problem is widely applied in private and public sectors (Jia et al., 2005; Karatas & Yakici, 2018). In the private sector, it supports facility distribution plans, which includes warehouse location (Jia et al., 2005; Gwalani et al., 2021; ReVelle & Swain, 1970). In the public sector, it is relevant for siting facilities where proximity matters such as hospitals post offices and fire stations (Jia et al., 2005; ReVelle & Swain, 1970; Karatas & Yakici, 2018).

2.3. Sport Event Optimization with P-Median Model

After introducing the theoretical background of the p-median model, the following section turns to its practical applications in sports logistics and event planning. These studies demonstrate how the model can guide real world facility placement and resource distribution decisions.

The p-median model is a key tool for the facility location problems (Liang et al., 2024; ReVelle & Swain, 1970). It is crucial for sustainable urban development and emergency management planning (Liang et al., 2024). In mass participation events, the optimization aims to maximize accessibility to medical resources and reduce response times. Location strategy chooses optimal medians to minimize total distance (Liang et al., 2024). Solution methods must address rapid growth in complexity with network size. Approximations heuristics and metaheuristics such as simulated annealing, variable neighbourhood search and GA's provide practical solutions for large real world problems (Liang et al., 2024).

The p-median problem and its variants are crucial for planning emergency services, which act as critical support processes for mass sports events such as marathons. Therefore, it is an essential tool for locating emergency service centres, where proximity and quick response time are highly desirable (Jia et al., 2005). It has been used to find dynamic ambulance positioning strategies by minimizing average response time to service calls and by using scenarios to represent different demand conditions (Jia et al., 2005). The p-median model can incorporate uncertainty by considering stochastic travel characteristics and demand patterns (Jia et al., 2005). Empirical studies indicate that minimizing average response time produces consistently better system performance than maximizing coverage alone (Jánošíková et al., 2021). This principle is directly relevant for the Vienna Marathon model, where minimizing expected weighted distance ensures both efficiency and equitable access to emergency aid across the entire race.

Guidance for athlete health and safety at large scale sports events reinforces the need for structured medical planning and risk management. Mountjoy et al. (2020) emphasize the importance of heat illness prevention, coordinated emergency response, environmental monitoring and effective communication among medical teams. The paper stresses the need for a standardized medical infrastructure with defined responsibilities and pre event preparedness to reduce response time and improve treatment outcomes (Mountjoy et al., 2020). These recommendations support the methodological foundation of this work, where temperature and participant data are incorporated as stochastic parameters in optimizing the Vienna Marathon's medical network.

To address real world complexities in event and emergency logistics, the p-median problem has been extended in several directions. Hierarchical p-median applies the median objective at two levels, such as Advanced Life Support (ALS) and Basic Life Support (BLS) and can reduce the overall average response time (Jánošíková et al., 2021). The previously mentioned dynamic p-median with mobile facilities allows the number and placement of facilities to change over a planning horizon through opening, relocation or closing. It considers mobile and immobile facilities at the same time which is relevant for temporary event infrastructure (Güden & Süral, 2019). A backup p-median is designed for congested environments, where the closest medical team may be busy. It minimizes distance to the primary facility and to multiple backup services and performance can be evaluated with discrete event simulation under stochastic conditions and different assignment policies (Karatas & Yakici, 2018).

Metaheuristics such as Tabu Search and Simulated Annealing are recognized as state of the art for facility location problems (Gwalani et al., 2021). Classic improvement heuristics such as Exchange and the Greedy Randomized Iterative Approach often reach solutions close to the optimum across various demand distributions, particularly clustered ones (Gwalani et al., 2021). McCarthy et al. (2011) showed how the Chicago Marathon implemented different systems to coordinate medical emergency and municipal agencies. The event improved their real time communication and response efficiency through a unified command (McCarthy et al., 2011). The Vienna Marathon model takes inspiration from these practical implementations by translating emergency coordination principles into a quantitative optimization framework using the p-median approach.

After the explanation of the p-median model the next part describes applications in sport events. Shen et al. (2021) designed a mixed-integer linear program (MILP) for emergency resource and allocation that fits sports mega events. It distinguishes temporary or fixed facilities and material or human resources and it validates the model on the 2022 Beijing Winter Olympics with sensitivity analysis. The framework imposes time reliability constraints, uses a response time threshold T and routes air rescue if ground response cannot meet T. It also enforces material reliability via a service level α. Demand weights on the road section combine a crash rate factor with a priority tier for VIP and operations traffic. Rescue time matrices follow from the shortest path, while stock levels use a bisection procedure. The matrix real coded GA solves a complex MILP model for emergency resource location and allocation during sports mega events and improves convergence speed (Shen et al., 2021). Shen et al. (2021) compared their framework with classic location models such as the p-centre and p-median formulations. Their discussion outlines the contexts in which each model is most appropriate for mega event traffic emergencies. This aligns with the p-median problem role in emergency logistics and with hybrid strategies where local staging centres or temporary facilities minimize total distance while they meet response time needs (Jia et al., 2005).

Further research focuses on city marathons. GA's are a leading metaheuristic used with Geographic Information Systems for facility location. The Event Geo Scout framework, a geographic decision support system that integrates real time spatial data, applies a GA to optimize ambulance locations during the Málaga City Marathon (Montenegro & Muñoz, 2024). In this case, the system provides maps that assign clients to facilities and supports smartphone based computation. It identifies real time crowd density peaks to improve emergency response, which shows how facility location optimization can be combined with live geospatial data in marathon settings. This example highlights the value of metaheuristics in event safety planning (Montenegro & Muñoz, 2024). The present research builds on this direction by integrating a stochastic p-median formulation instead of a purely GA based approach and by focusing on optimization before the event rather than during real time operations.

There is also related evidence outside sports that informs event planning. Huizing et al. (2020) study emergency logistics with median logic and introduce the Median Routing Problem. The goal is to minimize expected emergency response time while the system schedules non-emergency jobs. A heuristic stays on average 3.4 percent from optimal on real instances. This highlights the central role of solving the underlying p-median subproblem for operational routing (Huizing et al., 2020). Other emergency medical service models address large scale emergencies. Jia et al. (2005) propose a general model that can act as covering p-median or p-centre and provide Los Angeles case examples. The approach shows benefits in reduced life loss and economic loss compared to traditional methods (Jia et al., 2005). Therefore the p-median model has been formulated to emergency location siting problems and response time minimization is a central system objective (Jánošíková et al., 2021). These examples collectively support the methodological validity of using a p-median structure for optimizing emergency medical stations in a marathon scenario with stochastic influences.

The p-median problem is primarily used in location planning for facilities that require optimal accessibility. This includes permanent sports facilities and the optimization of emergency and hybrid logistics systems for major sports events. It also supports the configuration of public culture and sports facilities in villages and towns of different sizes (Shi et al., 2013). There is additional literature on event management planning that applies quantitative models. Shi et al. (2013) analyse residents needs for cultural and sports facilities in Chinese villages and towns. They propose a median location model to place facilities under limited public resources. The p-median model minimizes the total weighted distance from each demand point to its assigned facility with constraints to ensure that each demand point is served by exactly one facility and that exactly *p* facilities are opened. Local data of the study shows that weighted solutions tend to locate facilities near settlements with higher needs. After computing the optimal sites, the authors develop a configuration matrix that specifies which types of cultural and sports facilities each settlement level should include according to Chinese planning standards. In summary, the paper concludes that rural sports facilities are often unevenly distributed, underprovided and that quantitative models like the p-median model can help to achieve fairer coverage when public resources are limited. This application of the p-median model to permanent facilities complements the emergency focused studies and demonstrates the model's broad use across sport contexts (Shi et al., 2013). Similarly, the present study applies the p-median framework to ensure that medical resources are distributed evenly and efficiently along the Vienna Marathon route despite limited capacity.

A final example examines stadium location and attendance in Major League Baseball. Lim et al. (2019) analysed how stadium placement influenced attendance between 2006 and 2017. Key factors included population, income and distance from the county population centre. The model found that attendance decreases with the squared distance, which shows that shorter distances to population centres are linked to higher attendance. Model tests confirmed that this squared distance specification provided the best fit. In the full attendance model, other significant predictors included team quality, payroll, number of star players, stadium capacity, ticket price and seasonal trends. These findings connect to the p-median problem logic, where distance plays a key role in explaining access related demand patterns for sports venues (Lim et al., 2019). In a similar way, the present study applies weighted demand to represent varying risk factors such as weather, participant numbers and injury likelihood, which influence how emergency service access is optimized along the marathon route.

2.4. Sport Event Optimization with Other Models

While the p-median model forms the core of many optimization frameworks, other models also contribute valuable insights into location planning and emergency management. The next section therefore compares alternative approaches such as covering and multi criteria models and highlights their strengths and limitations in sports contexts.

Optimization models other than the p-median model are widely applied in sports event logistics and emergency service planning. Covering models aim to ensure that demand is covered in a specified distance or time limit while maximizing the number of people who are covered. The Maximal Covering Location Problem seeks to maximize the covered demand for a fixed number of facilities under a maximum distance constraint (Jia et al., 2005; Kwon et al., 2019). Recent studies have combined multi criteria decision making methods with covering models to plan sports facilities. For example, Arabnarmi et al. (2024) examined the optimal locations of sports complexes for traditional games with a hybrid approach that integrates spatial analysis. They applied the Best–Worst Method to assign weights to inter-city and intra-city criteria and found that social factors such as public acceptance and demographics were more influential than economic or environmental factors. A Geographic Information System was then used to overlay the weighted criteria and identify the most suitable construction areas in Iran (Arabnarmi et al., 2024).

Kwon et al. (2019) applied the Maximal Covering Location Problem to optimize rooftop sports facility locations in metropolitan areas. Stochastic extensions such as the Maximum Expected Covering Location Problem address facility unavailability by maximizing the expected population coverage (Jia et al., 2005; Jánošíková et al., 2021). The Location Set Covering Problem minimizes the number of facilities needed to ensure that all demand points are covered (Jia et al., 2005), while backup coverage models add redundancy so that a secondary facility is available when the primary one is occupied (Jia et al., 2005). Bi-criteria models, such as the Maximum Expected Covering Location Problem with p-Median Performance, combine coverage and efficiency by maximizing expected coverage for high priority patients while minimizing the average response time for all patients. However, simulation studies show that pure coverage objectives can produce poorer overall results than models that focus directly on minimizing response times (Jánošíková et al., 2021). A spatial optimization study of Emergency Medical Service facilities in Chengdu, China, found that 55 new facilities were required to meet the policy goal of covering 90% of the population within 15 minutes (Deng et al., 2021).

The Capacitated Traffic Assignment Problem describes how traffic distributes itself across a network when each road link has a limited capacity. Nie et al. (2004) showed that even large scale can be solved efficiently by combining mathematical optimization methods. Specifically, two techniques are used. The Inner Penalty Function method, which adds a penalty to prevent exceeding road capacity and the Augmented Lagrangian Multiplier method, which balances penalties and dual variables to improve convergence. Both methods use the Gradient Projection algorithm to solve subproblems efficiently, because it converges quickly and can handle large networks without excessive computation. Together, these approaches make the problem computationally tractable even for complex transportation systems (Nie et al., 2004). However, as Wu et al. (2021) pointed out, heuristic methods such as the Ant Colony Algorithm, although effective for certain optimization problems, can become very slow when applied to large spatial datasets. The algorithm's random search process depends heavily on the pheromone from previous iterations, which often leads to local optima instead of the true best solution and increases computational time (Wu et al., 2021).

During the Athens 2004 Olympic Games, detailed resource and planning models were developed to forecast the number of vehicles and personnel needed for VIP transport. These models relied on several input variables, like event schedules, the attractiveness of each sport, session types and passenger density functions, which were calculated in 15 minute intervals to capture the dynamic flow of spectators and delegates (Minis et al., 2009). The model also integrated time dependent demand fluctuations and operational constraints such as venue accessibility, parking availability and the synchronization of arrivals and departures. By combining these variables, planners were able to anticipate transport demand throughout the day and allocate the required number of vehicles, drivers and staff. This accordingly improved efficiency and avoided congestion (Minis et al., 2009).

For the placement of automated external defibrillators, multi criteria decision making methods were applied to ensure optimal accessibility and coverage. Du et al. (2024) used the Fuzzy Analytic Network Process together with Grey Relational Analysis to assign relative weights to key factors like demand, accessibility, visibility, staff training and the social environment. These criteria reflect both logistical and human aspects and emphasized not only where devices should be placed but also how effectively they can be used during emergencies. Among all factors, staff training was identified as the most important criterion with a relative weight of 0.4434, which highlights that human readiness and operational competence play a more decisive role in survival outcomes than location factors alone (Du et al., 2024).

Beyond the p-median model, various optimization and planning approaches are used in sports logistics and event management. This includes related location models and specialized forecasting frameworks (Liang et al., 2024; Minis et al., 2009). Alternative location models such as the p-centre problem aim to minimize the maximum distance between any demand point and its assigned facility (Liang et al., 2024; Snyder, 2006). They are particularly useful in situations where urgency and reliability are critical, such as in the placement of hospitals or fire stations (Kwon et al., 2019; Jia et al., 2005). For large scale emergencies, p-centre models help identify facility locations that minimize the maximum weighted distance so that areas with weak coverage receive sufficient protection, which reduces potential loss of life and economic damage (Jia et al., 2005).

In emergency service planning, modelling uncertainty and sustainability has become increasingly important. Two stage robust optimization approaches minimize the combined cost of preparedness, which includes fixed input costs and worst-case response, which includes allocation costs (Li et al., 2025). Sustainability extensions further add a deprivation cost for unmet demand and an environmental impact cost for transport emissions (Li et al., 2025). The Generalized Budget Uncertainty Set captures correlated demand uncertainty and incorporates environmental cost, which helps to reduce carbon emissions without significantly changing the optimal site locations (Li et al., 2025). Overall, there are many existing models that optimize the organization of sports events. The present study focuses on robustness and sustainability with the Vienna Marathon framework, which considers uncertain race day conditions and aims to maintain efficient under different stochastic scenarios.

2.5. Emergency Services for Sports Mega Events

The reviewed optimization methods form the theoretical foundation for practical emergency service planning at large scale sports events. The following section applies these concepts to the medical and emergency context of sports mega events.

Emergency services are a key component of sports mega events to ensure reliable medical support for all participants and spectators. Such large scale events require strict traffic and emergency resource plans as well as stable operations, which makes quantitative models essential for locating and allocating emergency resources (Shen et al., 2021). Studies show that runners in both the fastest and slowest performance groups are more likely to require medical care than those in the middle of the field (Breslow et al., 2019). In practice, emergency planning integrates both temporary and fixed facilities (García-Vallejo et al., 2020), prioritizes strict rescue time targets over cost minimization and weights demand by crash risk and priority level (Shen et al., 2021). Predictive technologies such as Information and Communication Technology and Augmented Reality dashboards help anticipate disruptions and accelerate emergency responses (Cieśliński et al., 2016).

Optimization logistics for marathon and mass participation events focus on sustainable management, strategic medical positioning and data driven resource allocation. As endurance events continue to grow globally (Chiampas & Jaworski, 2009; García-Vallejo et al., 2020; Yao et al., 2017), planning complexity increases. Organizers must apply a structured process management approach (García-Vallejo et al., 2020) that treats the marathon as a project with clear planning, implementation and control phases (Witkowski et al., 2016). A process map helps to classify activities into management, core and support processes (García-Vallejo et al., 2020). This structure parallels the Vienna Marathon model, where medical logistics are treated as an integrated process rather than an isolated optimization task.

Within the technical and athletic core of the event, close coordination between all operational areas is essential (García-Vallejo et al., 2020). Tasks include course design and drink station layout, while logistics and security ensure that all elements are positioned correctly and safely. Planning for infrastructure, security and venue occupancy usually begins six to eleven months before the event and medical services must be fully integrated, familiar with the course and safety aspects, which cannot be managed externally (García-Vallejo et al., 2020). Preparedness must also anticipate demand surges due to extreme weather or catastrophic incidents (Chiampas & Jaworski, 2009; Yao et al., 2017). The Wroclaw Marathon illustrates this through six ambulances strategically placed at the start, along the route and at the finish (Cieśliński et al., 2016; Witkowski et al., 2016).

Medical resource allocation should reflect expected casualty patterns and injury severity (Nguyen et al., 2008; Tan et al., 2014). Most incidents occur near the finish line and during peak completion times. The average patient presentation rate is 0.992 per 1,000 attendees and the transport to hospital rate is 0.027 per 1,000 persons (Arbon et al., 2001). Runners typically seek medical attention three to five hours after starting and the halfway station is often the second busiest (Nguyen et al., 2008). Staff and supplies should therefore be concentrated at halfway and finish points with early tents reallocated later in the event (Nguyen et al., 2008; Tan et al., 2014). Minor ailments such as headaches, blisters and sunburn make up about 80% of all cases (Arbon et al., 2001). Other common reasons include medication requests, musculoskeletal injuries, dehydration and skin irritations. Life threatening emergencies, particularly cardiac events, require immediate treatment (Nguyen et al., 2008; Tan et al., 2014). These empirically observed patterns inform the weighting system in the Vienna Marathon model, where high risk sections receive higher injury risk factors.

Forecasting medical demand relies on biomedical, environmental and psychosocial variables. Models use multivariable regression and indicators such as the patient presentation rate and transfer to hospital rate (Van Remoortel et al., 2020; Arbon et al., 2001). The patient presentation rate typically ranges between 0.5 and 2.0 per 1,000 participants, which emphasizes the need for context specific models (Van Remoortel et al., 2020). At the summer Olympic race walk and marathon events, 96% of athletes admitted to medical stations were diagnosed with exertional heat illness, which shows notably higher rates than in comparable races (Sugawara et al., 2022).

Emergency and medical services depend on comprehensive planning, forecasting models and coordinated strategies (Başdere et al., 2014; Mountjoy et al., 2015; Van Remoortel et al., 2020). The aim is to provide efficient on site care and prevent local hospitals from overload, which requires readiness, situational awareness, coordinated decision making and clear communication (Chan et al., 2019; Başdere et al., 2014; McCarthy et al., 2011; Vasquez et al., 2015). For example, the Chicago Model integrates structure, information and communication to improve emergency response (Başdere et al., 2014; McCarthy et al., 2011). At the Youth Olympic Games, a hospital based model proved financially and logistically efficient (Mountjoy et al., 2015). Effective planning demands venue specific risk assessments, contingency plans and language coordination (Vasquez et al., 2015; Hardcastle et al., 2012). High risk zones can also be mapped with used data. In the Gothenburg Half Marathon, the identified most exhausting points for runners strongly matched ambulance requiring cases, which confirmed the method’s usefulness for identifying dangerous sections (Nilson et al., 2018).

Technology also enhances event safety through data driven resource management. Water station optimization, for example, is often based on experience but can be improved with machine learning models such as gradient boosting trees and random forests (Simons, 2023). These models account for temperature and humidity, which strongly influence pace and hydration needs. Recommendations include six to nine evenly spaced stations in hot conditions and three to five in cooler weather to ensure fluid intake of 400–800 ml per hour (Simons, 2023). For situational awareness, GPS tracking systems monitor runner movement and detect emergencies like fainting or collisions, which enables fast response and dispatch (Cieśliński et al., 2016). The Vienna Marathon framework complements such systems by applying pre event stochastic optimization instead of real time monitoring to achieve balanced spatial coverage and robust medical preparedness.

In summary, the reviewed literature demonstrates that effective emergency planning for sports mega events depends on robust optimization models and detailed empirical understanding of medical demand patterns. These insights directly inform the methodological framework and data design presented in the next chapter.

3. Data Collection

After the theoretical and empirical insights above, the following chapter describes the data collection process and explains how uncertainty factors are quantified to model the real world marathon conditions.

3.1. Goal & Problem Context

This p-median model is formulated on the fixed race route of the Vienna City Marathon, where exactly *p* stations are selected and placed along the course. The goal is to determine the optimal placement of *p* medical stations that minimizes the weighted sum of distances from each route point to its nearest station.

In this study, two spatial resolutions will be analysed. Variant 1 (V1) uses a 1-kilometre (km) grid, which yields 44 data points with one per every km plus start and finish. These points are real GPS coordinates and represent the full Vienna Marathon course. Variant 2 (V2) uses a 100-meter (m) grid with 425 data points. This produces a finer and more precise representation of the route, while V1 is also exact but faster to compute. *Figure 1* shows the actual marathon route for comparison. All maps and route graphics in this thesis were produced with the online tool GPS Visualizer – Draw on a map (GPS Visualizer, n.d.).

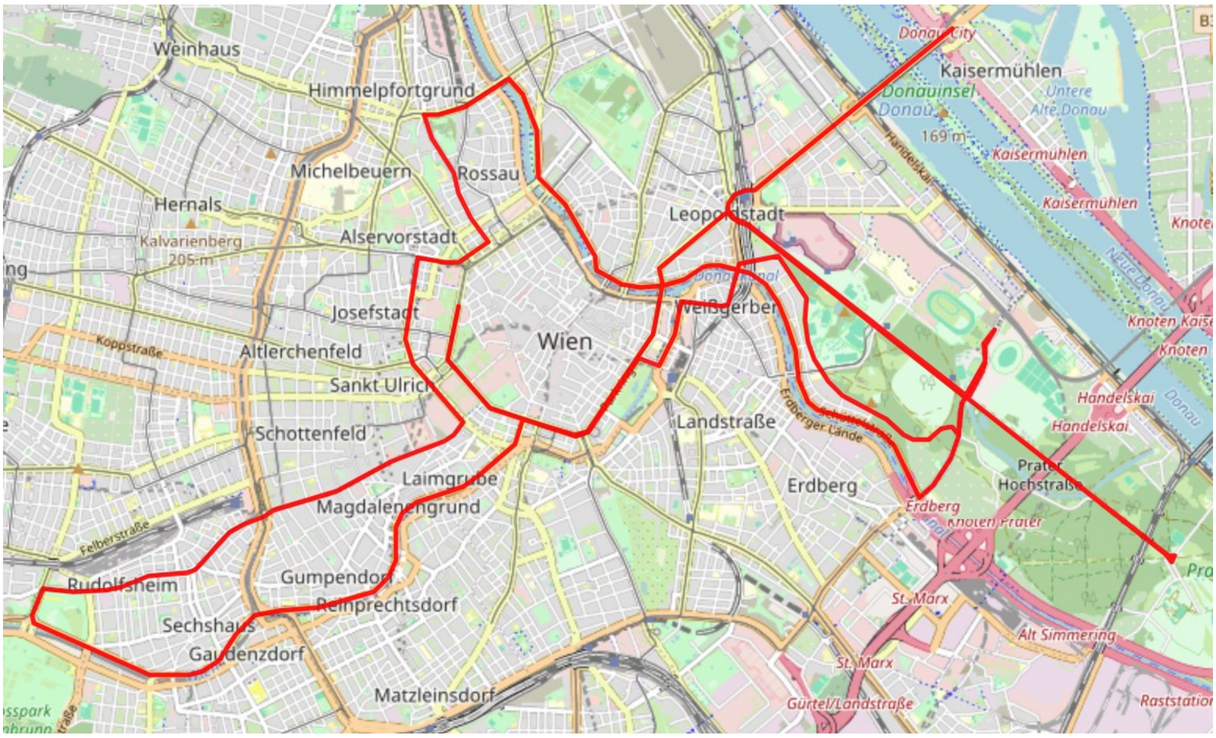

Figure 1: Original marathon route

For better orientation, *figure 2* illustrates the runners path. Participants start at (1), proceed to (2) and continue to (3). From (3) the course follows a long out and back spur to (8). After returning, the field passes (2) again and continues out to (9). The course then turns back, passes (10) in the late km’s and heads straight to the finish in the centre of the city.

Figure 2: Runners route at the Vienna Marathon

Figures 3 and *4* provide V1 and V2 for a comparison to the original route. These figures clearly show that V2 (100 m) traces the original marathon course more precisely than V1 (1000 m). Because distances between support points are computed as straight line, the denser grid in V2 yields to a more faithful representation of the route geometry.

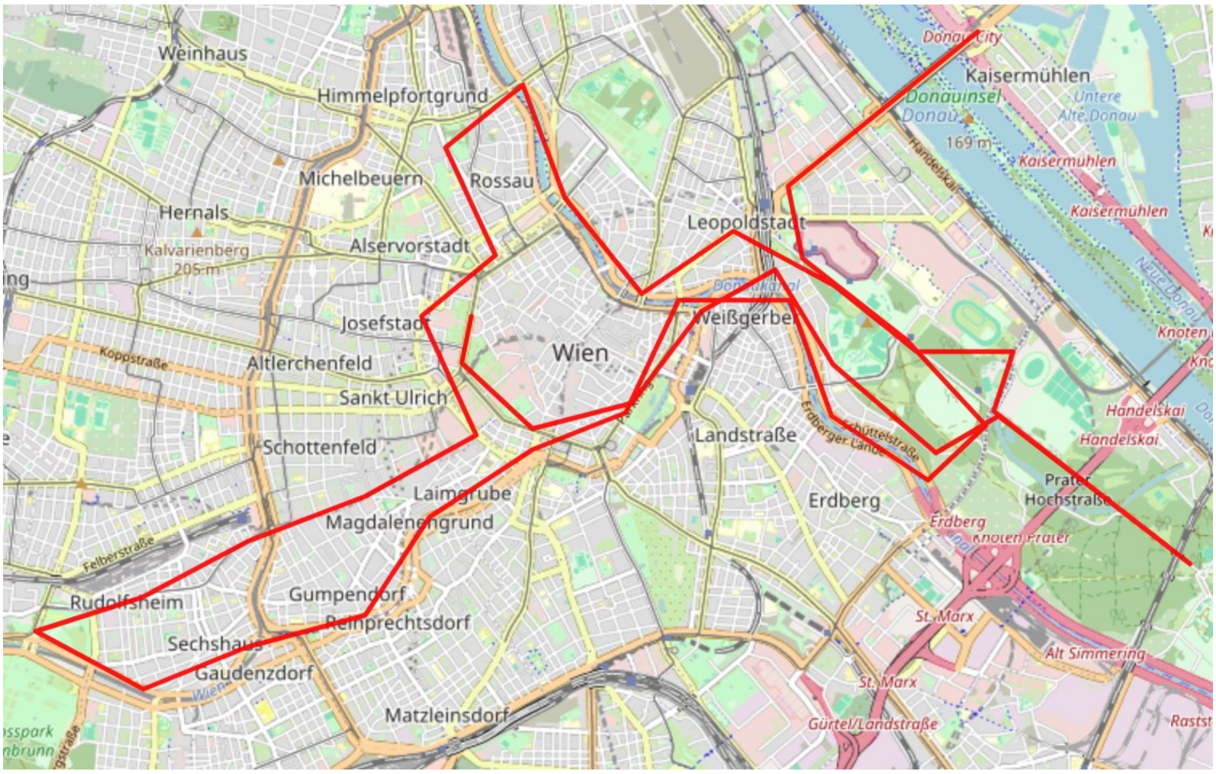

Figure 3: Route for variant 1 with 1000 meter distances

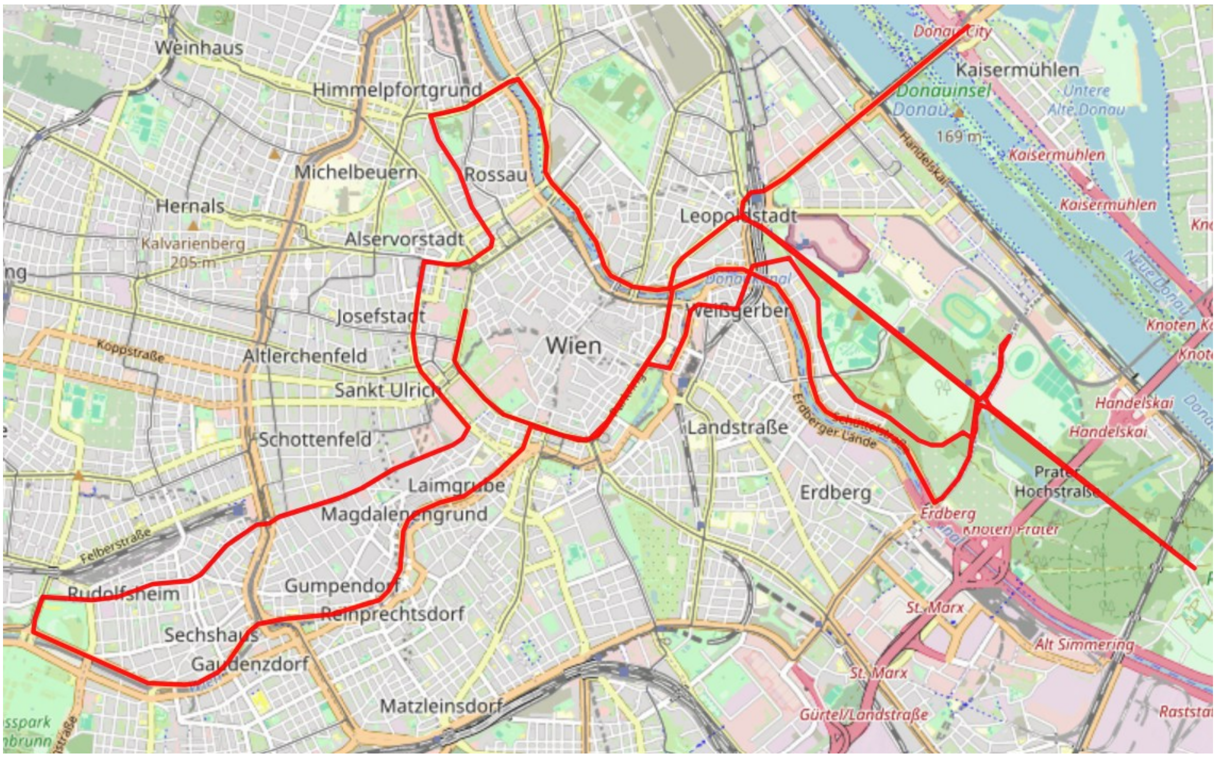

Figure 4: Route for variant 2 with 100 meter distances

3.2. Distance Metric & Distance Matrix

The distances will be computed by the Haversine formula, which returns the great-circle distance between pairs of latitude and longitude coordinates in m. This choice ensures geodesic consistency across both spatial resolutions and makes results directly comparable. With this formula the following analysis maintains high data integrity because of the use of real GPS coordinates.

$$\boldsymbol{d = 2r * arcsin\left(\sqrt{sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\lambda_1) * \cos(\lambda_2) * sin^2\left(\frac{\Delta\lambda}{2}\right)}\right)}$$

Notation: λ denotes longitude (in radians), φ denotes latitude, r is the Earth's radius (≈ 6,371,000 m)

Furthermore the Haversine formula computes the linear distance between every pair of route points in both variants, which creates two different distance matrices. However the real marathon route is not a straight *42.195 km* line and has curves or circles. Therefore successive route points do not have the same distance to each other. Haversine captures this geometry consistently, although a planar approximation would be fast and sufficiently accurate at very short ranges (≈100 m), we use one uniform formula for both variants to preserve model consistency and comparability. Numerically the rounding effects at extremely small angles like under 1 m are negligible at the 100 m scale. Finally our program will create distance matrices for both variants to highlight the difference between 44 datapoints and 425 datapoints.

3.3. Uncertainty Factors

In order to model a realistic real world scenario, the program has to consider several sources of uncertainty. To that end, this study incorporates three data driven factors with weather, participants and injury risk based on historical records. Together, they illustrate the real setting and influence the optimal placement of medical stations.

Importantly, weather affects emergency demand, especially during endurance events such as a marathon. Higher temperatures substantially raise the risk of severe medical incidents during events like marathons, where each additional rise in the heat stress index is associated with a *22%* increase in serious cases (Poh & Anantharaman, 2025). Roberts (2000) found out that higher heat and humidity levels are strongly associated with increased rates of exercise. These associates collapse and other medical incidents during marathons, which indicates that heat stress substantially elevates medical risk (Roberts, 2000).

Accordingly, this research employs temperature observations from the past ten years to provide a realistic input. *Table 1* documents the day temperatures during the main event over that period. The main race window is between 6 a.m. and 6 p.m. For the simulation, the program simulates temperatures within the observed range from the ten years minimum to the ten years maximum. Historical weather observations for Vienna were obtained from the Time and Date archive (Time and Date AS, 2025).

Date	***Year***	***Low 6 a.m. -12 p.m.***	***High 6 a.m. -12 p.m.***	***Low 12 p.m. -18 p.m.***	***High 12 p.m. -18 p.m.***
12. Apr 15	2015	13	18	17	28
10. Apr 16	2016	8	9	9	12
23. Apr 17	2017	5	10	9	11
22. Apr 18	2018	14	23	23	26
07. Apr 19	2019	7	14	15	19
cancelled	2020	-	-	-	-
12. Sep 21	2021	13	24	24	26
24. Apr 22	2022	9	17	16	21
23. Apr 23	2023	10	19	20	22
21. Apr 24	2024	2	7	7	9
06. Apr 25	2025	-1	3	3	4

Table 1: Historical data of temperature in C° during the Vienna Marathon main event

Similarly, for the participants factor the same ten years period is used. *Table 2* reports the number of attendees over the last ten years. This uncertainty influences the likelihood of emergencies because a higher number of participants generally increases the probability that someone will require medical treatment. For 2015 till 2023, participant numbers were compiled from the official Vienna City Marathon history pages (Vienna City Marathon, n.d.). For 2024, the figure was taken from the German Road Races coverage “Wien Marathon 2024: Sportliche Highlights & sportliches Miteinander” (German Road Races, 2024). For 2025, the count comes from the APA-OTS press release “Erste positive Bilanz für den 42. Vienna City Marathon” (VCM Group, 2025).

Date	*Year*	*Number of participants*
12. Apr 15	2015	42.742
10. Apr 16	2016	42.511
23. Apr 17	2017	42.766
22. Apr 18	2018	41.919
07. Apr 19	2019	40.590
cancelled	2020	-
12. Sep 21	2021	26.016
24. Apr 22	2022	31.933
23. Apr 23	2023	39.871
21. Apr 24	2024	42.625
06. Apr 25	2025	46.083

Table 2: Number of participants during the Vienna Marathon main event

Thirdly, the injury risk at each route point is the final uncertainty factor. *Table 3* shows the constructed Vienna Marathon injury probability dataset from a literature review. This dataset lists the estimated probability that a participant requires medical treatment at route point *i*. Notably, values are very low at the start, rise gradually through mid-race, increase sharply in the final third and peak near the finish line.

In marathon races, the likelihood of medical need is minimal in the early km but rises sharply in the later stages. Studies consistently show that the risk of medical emergencies at the beginning of a marathon is very low, whereas the second half, especially beyond *km 30*, shows a pronounced increase in medical encounters (Kim et al., 2012; Nguyen et al., 2008). For instance, an event medicine study found that roughly two-thirds of runners who required medical help needed it during the final portion of the race (Nguyen et al., 2008).

Further evidence supports this progressive increase in medical risk throughout a marathon. Poh and Anantharaman (2025) reported that higher environmental heat and humidity substantially increase the likelihood of medical encounters during endurance events, as thermal strain and dehydration collectively raise the risk of serious incidents (Poh & Anantharaman, 2025). Similarly, Day and Thompson (2010) observed that cardiac arrests clustered in the final km's of the race, typically beyond *km 30*, which highlights the need for increased medical coverage near the finish area (Day & Thompson, 2010). Both studies emphasize that runners physiological reserves deteriorate significantly after the halfway mark and heat stress increases this effect near the finish. These consistent findings further validate the shape of the injury risk table, which assigns low values for early km's, moderate weights around mid-course and a pronounced spike in the closing km.

Evidence of the weights for the injury risk table establishes a clear risk escalation pattern. For instance, a German newspaper article reports that "from about *km 20* it becomes dangerous," and that serious incidents mostly occur after *km 30* (Bartens, 2017). This observation aligns with marathon data, which shows that the majority of medical interventions and sudden collapses take place in the final stretch of the race (Webner et al., 2012; Nguyen et al., 2008). Thereby, it anchors the station siting logic in empirically documented incident dynamics.

This late race increase corresponds with evidence that cumulative fatigue, dehydration, heat stress and glycogen depletion contribute to a higher incidence of collapse and medical complications in the final part of the marathon (Nguyen et al., 2008; Webner et al., 2012). Consequently, the later sections of the course are prioritized in the model. Webner et al. (2012) found that most serious medical emergencies, including sudden cardiac arrest, occur in the final km of the marathon course, which highlights the importance of allocating medical resources toward the closing sections (Webner et al., 2012). Like previously mentioned Roberts (2000) similarly reported that most medical encounters arise at the finish area, where exhaustion and heat stress peak (Roberts, 2000). These findings validate the injury risk table's structure with very low risk per km at the start and a steep rise in predicted medical need toward the race's end, which ensures that medical station planning aligns with the known escalation of risk in the late stages of a marathon.

i (km)	*estimated probability (in %) for emergency need at point i*	*comments*
0	0,1	almost 0, no real probability
1	0,5	very low
2	0,5	
3	1	
4	1	
5	1	
6	1	
7	1	
8	1,5	
9	2	
10	2	still very rare
11	2,5	
12	3	
13	3	
14	4	
15	5	small increase
16	5	
17	6	
18	7	
19	8	
20	10	start of critical point
21	12	
22	14	
23	16	
24	18	
25	20	
26	22	
27	25	
28	28	
29	30	

30	35	strong increase
31	40	
32	45	
33	50	double increase
34	55	
35	60	
36	65	
37	70	
38	75	
39	80	
40	85	
41	90	
42	95	
43	95	almost sure emergency case

Table 3: Injury probability dataset for every route point i

3.4. Weighting logic

In order to make the uncertainty factors comparable, this research applies min–max normalization to the observed event day ranges. Consequently, the temperature based weather factor is scaled to $r_{norm} \in \{0,1\}$ and the participants factor to $t_{norm} \in \{0,1\}$ with the following formulas.

$$r_{i\_norm} = \frac{\boldsymbol{weather - (-1)}}{\boldsymbol{28 - (-1)}}$$

$$t_{i\_norm} = \frac{\boldsymbol{participants - 26,016}}{\boldsymbol{46,083 - 26,016}}$$

Here *−1 °C* and *28 °C* are the observed minimum and maximum temperatures on the event day, while *26,016* and *46,083* are the corresponding minima and maxima of the total participants like in the previous shown datasets.

Importantly, the injury risk dataset is defined at the km level. Thus, we need to handle the variants different as follows:

- Variant 1 (1000 m): use scaled km risk v_i directly.
- Variant 2 (100 m): assign each 100 m point with the risk of its enclosing km bin (*km_idx = ⌊id/10⌋*).

Notably, this research does not divide the risk by ten because the values remain normalized $v_i \in \{0,1\}$. In other words, a risk of a certain km applies to all ten 100 m points within that km, which preserves the calibration and enables the finer grid.

Significantly, weather and participant counts do not act uniformly. They rather amplify existing risk hotspots. Therefore, we define the scenario sharpness with s = $\frac{1}{2}(r_{norm}+t_{norm})$, which leads to an equal influence for both factors, while the injury risk remains the primary factor in the point weights. Overall, every point is directly influenced by the injury risk table v_i, while weather and participant levels determine how this table affects each location.

Additionally, for scenarios in which the scaled weather and participant values are close to 0, the model introduces a risk floor ε. Specifically, ε = *0.10* ensures that at least 10% of the injury risk table still count at historical minima. In other words, the weights of the route points never collapse to be equally important across the route. Hence low values of sharpness s yield to more uniform weights, whereas large values of s allow the hotspots in the risk table to dominate. In this case, we set ε = *0.10* to keep later km's represented but less strongly than under high scaled factor levels.

Now after all introductions, everything is prepared to set the total weight of every point *i* with the three factors in both variants. P_{mean} denotes the average injury risk over the entire course, while the scenario sharpness *s* from weather and participants ensures both factors contribute equally to the weight. With the risk floor ε, we preserve a minimum share of local risk in mild scenarios. Importantly, for mild scenarios (small s) the weights are pulled toward the course average with at least ε of the local signal remaining. By contrast, for severe scenarios (large s), the model amplifies hotspots and lets local risk dominate.

$$\boldsymbol{\omega_i = (1 - s)(\varepsilon v_i + (1 - \varepsilon)\, P_{mean}) + s v_i}$$

Furthermore, if $s = 0$ then $\omega_i = (\varepsilon v_i + (1 - \varepsilon)\, P_{mean}$,which is protected by the risk floor ε. If $s = 1$ then $\omega_i = v_i$ counts, which means it is completely driven by the injury risk table. In other words, s governs how strongly the scenario affect local risk, while the risk floor ε guarantees that the signal never gets vanishes.

3.5. Scenario simulation

Firstly, the program simulates a specified number of scenarios within the ten years observed ranges for temperature and participants. Thereafter, all other quantities are computed automatically. Each scenario instance is then solved with Gurobi for both spatial resolutions. During this process, the program records the objective values for every single scenario as well as the most frequently selected stations across all scenarios. To ensure sufficient replication and stable frequency patterns, 100 scenarios are simulated and solved. Finally, the program displays the number of selections for every location and computes the average objective function value across all 100 scenarios.

3.6. Coordinate Data Processing

The official course coordinates were exported as a GPX file from the main website of the event. Subsequently, the GPX was uploaded to the GPS Visualizer website, which allowed to remove auxiliary points (e.g., half-marathon markers, water stations) and retain one point per km plus start and finish. Consequently, this process yielded 44 official points, which were saved to a CSV file for use in V1 of the program.

By contrast, for V2 the program parses the same GPX file and interpolates points at approximately 100 m intervals with geodesic distances. It traverses each segment of the polyline and inserts intermediate points whenever the accumulated path length exceeds 100 m. This strategy leads to the 425 route points. In other words, the only difference between the variants is the coordinate source. While V1 reads the cleaned GPS Visualizer output, V2 reads directly from the GPX and generates an interpolated 100 m grid.

All coordinates for both variants are listed in the appendix. Minor discrepancies may occur due to the different extraction methods. For instance, route point $i = 1$ in V1 may not exactly align with route point $i = 10$ in V2. Nonetheless, because both variants precisely represent the main route, these small differences are immaterial for our purpose because the goal is to identify the optimal locations for medical stations along the route.

4. Methodology

Based on the collected and processed data, the next chapter presents the methodological design of the study and formalizes the p-median optimization model used to determine optimal medical station placement.

4.1. Modelling

This study computes the optimal placement of medical stations on the Vienna City Marathon route with a weighted p-median model under uncertainty. The objective is to minimize the total weighted distance from each route point to its nearest open station. The primary driver of a point's weight is the injury risk table v_i, whose values are normalized and determine the importance of route point i. Two scenario factors with weather and participant numbers slightly adjust how strongly v_i influences the weight, so that hotter or larger events place more emphasis on high risk segments.

4.2. Sets, Parameters and Variables

The index i denotes the set of route points. Accordingly, $i = 44$ in V1 and $i = 425$ in V2. Index j represents all candidate locations, where a station may be placed. In this study, a station can be located at any route point, as the objective is to determine the optimal placement along the course. Whether a given point is practically feasible remains a separate decision for the organizers. For this reason, we set $i = j$.

Distances between GPS coordinates are computed with the Haversine formula to form a distance matrix $D = [d_{ij}]$ and are measured in m. The number of stations is p in the real world baseline is $p = 4$. However, this research also examines how increasing p affects the objective value.

For each route point, a weight w_i is assigned to indicate its importance relative to other points along the course. This weight is composed of the three external and uncertain factors. These factors are weather, participants and injury risk. Weather is represented by temperature, participants by the number of attendees and the injury risk table captures the probability that at least one participant requires medical treatment at point i.

Finally, the decision variables are $x_{ij} \in \{0,1\}$, which indicate whether point *i* is assigned to candidate station *j* or not and $y_j \in \{0,1\}$, which decides whether a station is opened at location *j*. All model parameters are summarized in the following table.

Parameter	***Description***
i	Route point index along the course
j	Candidate station site index
d_{ij}	Distance between point *i* and site *j*
p	Number of stations to open
w_i	Weight of route point *i*
x_{ij}	Assignment variable (binary) 1 if point *i* is served by station *j*, else 0
y_j	Open-station variable (binary) 1 if a station is opened at site *j*, else 0

Table 4: Overview of parameters

4.3. Mathematical Formulation

Now we turn to the mathematical formulation of the p-median model on the Vienna City Marathon. The baseline objective is following.

$$\boldsymbol{min} \sum_{i \in I} \sum_{j \in J} w_i * d_{ij} * x_{ij}$$

Overall, the model minimizes the total distance between all route points and their assigned stations. The constraints follow the standard p-median structure:

(1) Unique assignment: each route point is assigned to exactly one station,

$$\sum_{j \in J} x_{ij} = 1 \quad \forall_i \in I.$$

(2) Open-facility consistency: assignments are only allowed to open stations,

$$x_{ij} \leq y_j \qquad \forall_i \in I,\ j \in J.$$

(3) Cardinality of open stations: the number of stations to open equals *p*. In the real world baseline we set *p=4*,

$$\sum_{j \in J} y_j = p \qquad p = 4.$$

(4) Integrality:

$$x_{ij}, y_j \in \{0, 1\}.$$

5. Model Implementation

In the following chapter, the Python implementation of the p-median model with Gurobi will be presented. An overview of the code architecture and the flow from the input data and scenario factors to the solver and the evaluation is seen in *figure 5*. Furthermore, it will be shown how the model selects station locations that minimizes the weighted distance. Additionally, short pseudocodes cover the core steps and the full code is in the appendix.

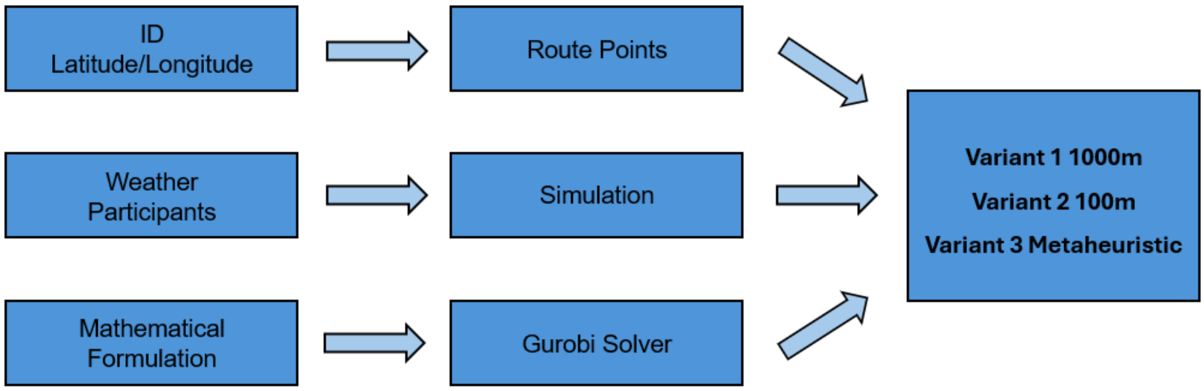

Figure 5: Overview of code architecture

5.1. Route Point

First, this file defines a lightweight data container for route points along the course. It is used by all variants to hold geometry and scenario dependent fields. The field *id* is the route index (e.g., 44 indices in V1; 425 indices in V2). Each route point stores a latitude and longitude from the official coordinates. Moreover, the three uncertainty factors weather, participants, and injury risk are included as fields to be simulated later. Finally, the float field weight w_i represents the final point weight used in the objective.

In the appendix, the code overview shows how a list of route points is constructed. Notably, the file also provides a compact string representation for logging/debugging in scenario loops, with no impact on optimization. In other words, each scenario operates on a fresh copy of the route to avoid side effects.

5.2. Gurobi Solver

The second file builds the MILP p-median model described in chapter 3 by using Gurobi and returns the model together with the decision variables. Inputs are the route with the weights w_i, the distance matrix d_{ij}, and the number of stations *p*.

Nevertheless, the decision variable $\mathrm{x_{ij}} \in \{0,1\}$ assigns point *i* to site *j* and $\mathrm{y_j} \in \{0,1\}$ indicates whether a station is opened for candidate *j*. These variables are stored in Python dicts *x[(i,j)]* and *y[j]* for convenient access. The objective minimizes the weighted distance from each point to its assigned station. Moreover, the standard p-median constraints for unique assignment, open-facility consistency and cardinality are implemented. Each constraint is added explicitly with a descriptive name like seen in the appendix to ensure traceability in the code.

5.3. Simulation

Thirdly, the Simulation file provides scenario inputs and normalization for the model. It samples the factors weather and participants within the ten year historical data ranges and returns min–max normalized values, together with the per km injury risk table, which was referenced in the chapter about the data collection. Furthermore, the module uses an internal random number generator to draw factor values uniformly over the observed ranges and has no fixed seed set here.

For instance, the weather factor samples temperature uniformly from *−1 °C* to *28 °C* and the participants factor samples total attendance uniformly from *26,016* to *46,083*. Each draw is then min–max normalized to [0,1] and leads to r_{norm} and t_{norm}. Moreover, the injury risk factor is indexed by km bins (0–43) and is retrieved from a tabulated profile, where values are scaled to [0,1] by the construction.

The functions are stateless and produce scalar outputs from given inputs return. Moreover, the class deliberately does not compute composite quantities (e.g., scenario sharpness s or point weights w_i). Instead, it provides the building blocks used for weighting and scenario assembly. Lastly, units are consistent with weather in °C, participants as counts and injury risk as unitless normalized scores in [0,1]. The appendix presents the complete Simulation code.

5.4. Coordinates 425 datapoints

Finally, the last file before the main variants is Coordinates_425_datapoints. It exists because the coordinate acquisition differs between the variants. V1 reads a cleaned CSV from GPS Visualizer, whereas this file parses the official GPX directly and constructs an interpolated 100 m grid. Consequently, it produces route points with id, latitude and longitude analogous to V1.

In order to generate the 100 m grid, the script extracts all track points from the GPX and sets a step size of 100 m. It then traverses each GPX segment, computes the geodesic distance between consecutive points and inserts intermediate points whenever the accumulated path length exceeds 100 m. These steps are seen in the program in the appendix below. *Figure 6* provides a simple pseudocode of this important step before the main part of the second variant.

```
Build Interpolated Route (gpx_file, step_m = 100):
   points  ← read gpx lat & lon in order (gpx_file)       // original GPX track
   route ← [ pts[0] ]                      // interpolated points
   current point  ← pts[0]                 // first point
   accumulated distance ← 0

   for next in pts[1:]:
      segment distance ← geodesic distance(current point, next point)
      while accumulated distance + segment distance ≥ step_m:
         r ← (step_m - accumulated distance) / segment distance
         lat ← current lat + r * (next.lat – current lat)
         lon ← current lon + r * (next.lon – current lon)
         route append ( (lat, lon) )              // fixed 100 m step
         current point ← (lat, lon)
         segment distance ← geodesic_distance(current point, next point)
         accumulated distance ← 0
      accumulated distance ← accumulated distance + segment distance
      current point ← next point
   return route                            // ≈425 points for V2
```

Figure 6: Pseudocode for Coordinates_425_datapoints

While the accumulated path length exceeds 100m, the program computes a linear interpolation ratio to estimate latitude/longitude and appends the new point. It then resets the accumulator and continues until the remaining segment falls below 100m. By contrast, V1 consumes a CSV exported from GPS Visualizer with one point per km, whereas this script derives a finer 100 m grid directly from the GPX via geodesic interpolation. With this file, all preparations for the main program are complete and the variants are presented in the next part.

5.5. Variant 1

After preparing the data and model, the implementation of the first variant to solve the p-median and identify the optimal locations for the medical stations along the Vienna Marathon route is shown. The purpose of V1 is to obtain an exact MILP solution with Gurobi on the course of 44 route points. To that end, the file also simulates a scenario loop to tally the most frequently selected locations after a specified number of random scenarios. The appendix shows the whole code. The header with all imports, parameters and constants. Importantly, to ensure comparability across variants, we measure the total runtime for each.

As next step, the route input method reads the 44 official points from the CSV and creates a list of route point objects with the defined fields (id, latitude, longitude). In other words, this anchors the data basis and is consistent with the earlier route & coordinates description. Then the program employs the Haversine formula to compute great-circle distances in m between every pair of route points. The core function of the calculation builds a symmetric distance matrix $D = [d_{ij}]$, which serves as the basis for the solver. In other words, the matrix captures all pairwise separations required by the p-median optimization.

After constructing the distance matrix, the program calls the Simulation module to generate scenarios. It draws weather and participant values uniformly within the historical ranges for each scenario and then returns min–max normalized factors for the specific run. Moreover, this block sets the total number of scenarios with 100.

This step bridges the data basis for scaling and computing scenario sharpness s = $\frac{1}{2}(r_{norm}+t_{norm})$, after which the weights $\omega_i = (1 - s)(\varepsilon v_i + (1 - \varepsilon)\, \mathrm{P}_{mean}) + s v_i$ are computed for each route point in the given scenario. The risk floor ε is fixed at 10%. In other words again, when *s* is small, the influence of the injury risk from *table 3* is dampened but never falls below 10%. When *s* is large, the model amplifies hotspots, so risk dense sections dominate particularly on hot days with many participants.

The final step inside the scenario loop is the handoff to the Gurobi backend. The program passes the binaries $\mathrm{x_{ij}}$ and $\mathrm{y_j}$, the weighted objective *min* $\sum_{i\in I} \sum_{j\in J} \omega_i * d_{ij} * x_{ij}$ and the standard p-median constraints to Gurobi_solver. Consequently, Gurobi_solver returns the model together with the decision variables x_{ij} and y_j. In other words, the call delegates optimization without duplicating the code, which demonstrates the pipeline transition.

Finally, V1 concludes with the prints of the solutions. The program extracts open stations from the decision variable y_j, increments a station frequency counter and accumulates the objective value. Consequently, this block prepares the later result figures, where the interpretation is deferred to subsequent chapters. Moreover, to enable cross variant comparison, the runtime measurement ends only after all scenarios are simulated. The script then computes the average objective and reports the most-frequently selected stations across the 100 scenarios. This means that a uniform output format is enforced across variants, which supports reproducibility.

These were the principal building blocks of V1 with the 1000 m resolution. By contrast, the next section introduces V2 (100 m) with the same weighted p-median formulation on a finer grid.

Consequently, we outline its pipeline and code structure in parallel to V1 and note some differences in coordinate ingestion and especially in distance matrix size. For an overview a pseudocode is provided in *figure 7*.

```
RunVariant1 (route_csv_file, scenarios = 100):
1. Load route points (id, latitude, longitude) from the CSV file → route_points
2. Build a symmetric distance matrix using the Haversine formula → distance_matrix
3. Create counter for chosen station → station_frequency
   Create empty list for objective values → objectives
   Start time
4. For each scenario from 1 to 100 scenarios:
    a) Draw weather and participant count from historical ranges
       Normalize both to interval [0, 1]
       sharpness ← average of the two normalized values
    b) For each route point:
        Look up injury risk for its kilometre from table
        Compute point weight with epsilon = 0.1:
          Use weight formula
    c) Build a p-median mixed-integer linear program:
        - Decision: open station at point j (yes/no) and assign each point i to a station j
        - Constraints: each point is assigned to exactly one open station;
                       exactly "number_of_stations" stations are open
        - Objective: minimize sum over all points
      Solve with Gurobi
    d) If optimal solution found:
        Add objective value to objectives
        Increase station_frequency for every point selected as a station
5. After all scenarios:
    Report mean objective value across scenarios
    Sort points by station_frequency to identify most frequent stations
    Report total runtime
6. Return the ranked station zones and the summary statistics
```

Figure 7: Pseudocode for variants

5.6. Variant 2

By contrast, V2 largely mirrors V1 with a few targeted differences. It solves the same weighted p-median in Gurobi on a finer 100 m grid with 425 route points. Moreover, we retain the scenario loop, runtime tracking and a standardized output format to ensure comparability with V1. The code header and ingestion phase are identical, since the coordinates are read from the CSV exported by the Coordinates_425_datapoints tool. Consequently, V2 yields a heavier distance matrix and a larger MILP instance, while the solver logic and formulation remain unchanged. Importantly, the index set now comprises 425 100 m IDs rather than 44 1000m IDs. In other words, the ingestion is identical, but the grid is finer.

V2 employs the Haversine formula and constructs the symmetric distance matrix $D = [d_{ij}]$, exactly as in V1. However, the instance size expands to 425×425 compared to 44×44 in V1. Consequently, the matrix construction and the downstream MILP become substantially heavier. Notably, the memory and solve time increase, even though the solver logic remains unchanged. The scenario sampling and min–max normalization are identical to V1. For each scenario, the program draws weather and participants within the historical bounds and derives r_{norm} and t_{norm}.

V2 operates on 100 m IDs (0, … ,425), whereas the injury risk table is defined at the km level (0, … ,43). Consequently, each 100 m point must inherit the risk of its enclosing km. Importantly, this is done via integer division and capping with km_idx = *min(*$43,$ *id) // 10).* The program replicates the km level value for the ten 100 m points within that km. It does not divide the risk by 10. Moreover, it preserves the scale of v_i. A numeric division would attenuate the risk incorrectly. Similarly, when the positions for readability are printed, the 100 m IDs are converted to km's with floating division (id / 10.0). *Figure 8* highlights this part with a pseudocode.

```
For each scenario:
  1) Draw weather and participants; normalize to [0, 1]
    sharpness = (weather_norm + participants_norm) / 2
    epsilon = 0.1
  2) Compute route-average injury risk:
    P_mean = average over all points of risk_table
  3) For each route point:
      kilometre_index = min(43, floor(point.id / 10))   // map 100 m id → km bin (0–43)
      risk_norm = risk_table[ kilometre_index ]
      compute weight
      point.weight = weight
      point.injury_risk = risk_norm
      point.weather = weather
      point.participants = participants
```

Figure 8: Adapting injury risk table on variant 2

The MILP call remains unchanged, although the problem instance becomes larger. By contrast, the final print block differs because of the earlier mapping, where positions are stored as 100 m step indices and later converted to km by dividing by 10. This produces a human readable summary, as the output reports km positions while retaining the underlying 100 m resolution. The appendix illustrates this step for V2.

These were the key differences in the building blocks of V2 relative to V1. Because the 100 m grid implies a much larger distance matrix and far more combinations, the solving time can increase substantially under the exact MILP. Consequently, this study introduces a third variant with a metaheuristic approach that aims to solve the large instance more efficiently while preserving the same weighted p-median objective. In other words, Variant 3 (V3) applies a GA to the V2 setting and then compares solution quality and runtime against the exact methods. Moreover, we retain the scenario loop and standardized outputs to ensure fair and reproducible comparisons across all three variants.

5.7. Variant 3

To reduce the runtime on the 100 m grid of V2, this study adopts a metaheuristic built on the same data and weighting structure. Consequently, V3 applies the GA to search the optimal location in the 425 route point space. Because the exact MILP scales sharply with the problem size, the GA aims to approximate the same weighted p-median objective. The total code is seen in the appendix.

The main structure mirrors V2, because reading coordinates, computing Haversine distances, and building the distance matrix are identical. The first difference arises in the solution process, whereas V2 calls the Gurobi solver and V3 enters the start of the GA pipeline with the population initialization. Importantly, because we do not involve Gurobi_solver, this file computes the weights locally. It simulates weather and participants, normalizes them to r_{norm} and t_{norm} and then constructs point weights. By contrast to the 100 m grid, the injury risk table is defined at km resolution (0–43). Consequently, each 100 m point inherits the risk of its enclosing km via km_idx = *min(*$43,$ *id) // 10)* again in V3.

V3 replicates the km level value for all ten 100 m points in that km but does not divide the risk by 10. Thereby it preserves the scale of v_i. Next, V3 computes scenario sharpness s = $\frac{1}{2}(r_{norm}+t_{norm})$ and applies the weighting rule $\mathrm{w_i}$ = (1 – s)($\varepsilon \mathrm{v_i} + (1 - \varepsilon)\ \mathrm{P_{mean}}) + s\mathrm{v_i}$. As a summarize, V3 computes the weights locally and passes them to the GA instead of Gurobi.

Firstly, it initializes a diverse population of 100 completely random solutions, where each is a size *p* set with unique indices. Subsequently, each individual encodes *p* open stations as a sorted tuple of indices from {0,…,425}. Importantly, the fitness is the same objective as in the previous variants:

Fitness(S) **=** $\sum_i \boldsymbol{\omega_i} * \boldsymbol{min_{j \in S}}\ \mathbf{d_{ij}}$**.**

This variant still measures the weighted distance from each point to its nearest chosen site. Consequently, V3 replaces the MILP's objective evaluation with a direct evaluator on candidate sets.

As next step, the algorithm selects parents by ranking individuals by their minimization fitness and retain an elite fraction of 20 %. Moreover, it records the current best individual as elitism. In other words, the procedure promotes exploitation of high quality solutions while it preserves the top performer for the next generation. Although selection identifies strong parents, the algorithm still needs to generate new solutions. To that end, two parents are combined by taking the union of their station indices. If the union is smaller than *p*, the program repairs the pool by adding random unused indices. It uses the sample of p unique indices from this pool to form the child in a feasible construction. In other words, crossover mixes strong partial structures from different parents while maintaining solution validity.

Finally, the GA applies mutation with probability rate. Specifically, the algorithm replaces one gene with a new index which is not already present in the solution. Crucially, this operation preserves feasibility as the child still contains exactly *p* unique stations while maintaining diversity and helping the search escape local minima. Consequently, mutation injects controlled randomness without disrupting the solution's structure.

Ultimately, the algorithm iterates over a fixed number of generations. After each evaluation, it updates the global best and proceeds to the next generation. Crucially, V3 adopts an early stopping criterion, where the run terminates after 100 consecutive generations without improvement. Elitism retains the top parents and offspring produced via crossover/mutation constitute the new population. Additionally, a helper method tracks improvements within the loop and triggers termination when no gains are observed over the patience window. Consequently, V3 replaces the Gurobi solver with a metaheuristic search that has a clear and reproducible termination rule. The final printout mirrors the other variants, where the selected stations are tallied and reported at the end and the average GA objective across all scenarios is computed. Additionally, to ensure a comparison, the total runtime is measured and reported.

In sum, the three core variants have now been presented. To deepen the analysis of this real world setting, this study will examine targeted cases like specifically, the actual real world station layout at the Vienna Marathon and a set of extreme scenarios. Additionally, to assess potential improvements in practice, this research will vary the number of stations *(e.g., p= 5,…,12)* and evaluate how this change affects the objective and the structure of optimal placements. Consequently, the next section compares these configurations to the baseline variants and quantifies any gains or trade-offs. For an overview, *figure 9* presents a pseudocode of the third variant.

```
Run Genetic Algorithm (route_points, distance_matrix, scenario_weights):
1) Build initial population of random candidate solutions
   Each candidate = set of number of stations unique route indices
2) For every candidate, compute fitness:
    fitness = sum over all route points of scenario_weight point × distance nearest open station
3) Repeat until no improvement for a fixed number of 100 generations:
    a) Select parents with ranking by fitness and keep best fraction
    b) Create children:
         Pick two parents
         Combine their station sets
         Repair: keep unique indices
         With small probability, mutate: replace one index with a new unused index
         Evaluate fitness of the child
    c) Form the next population from the best parents and the best children
4) Return the best candidate and its fitness
```

Figure 9: Pseudocode of genetic algorithm

5.8. Extreme Points

In this chapter, this study increases the Monte Carlo runs with three deterministic extreme point scenarios to probe the boundaries and a typical case. These scenarios are summarized with their fixed values in *table 5.*

Cold & few participants	Weather: -1°C; Participants: 26.016
Average	Weather: 13,5°C; Participants: 36.049
Hot & many participants	Weather: 28°C; Participants: 46.083

Table 5: Overview extreme scenarios

Consequently, we can systematically contrast station placements and objectives under controlled conditions. Thereby this improves the interpretability for later analysis and replaces the randomness with fixed inputs. The structure stays the same with the weighted p-median objective. The only difference is that the inputs are fixed. The appendix provides the code for the extreme scenarios.

The first variant takes 44 route points as input and constructs a 44×44 distance matrix with the Haversine formula. Specifically, it bypasses the simulation loop and sets weather and participant counts to the selected extreme. The code in the appendix illustrates the fixed input pathway with no simulation and the hardcoded values for the three extreme scenarios.

V2 takes 425 route points as input and otherwise employs exactly the same extreme point search code. Specifically, the injury risk table is mapped from km bins to 100 m points by dividing it with 10 like before. This means that each 100 m point inherits its km bin risk value. Similarly, the point IDs are expanded to 425 entries.

The GA uses the same input with 425 route points as V2. Consequently, the optimization problem is identical to V2. Therefore, the solution should coincide with V2's once the early stopping rule is met after 100 iterations without improvement. Hence, a separate extreme point calculation is not required here.

5.9. Number of Stations

This study aims to extend the real case setting with four stations by systematically varying the number of open facilities *p* from 4 to 12. It examines how additional stations affect the weighted p-median objective. Accordingly, it moves one step beyond the baseline Vienna City Marathon setup and test $p \in \{4,5,\ldots,12\}$ under an otherwise unchanged modelling pipeline. For each *p*, we compute the average objective across 100 Monte Carlo scenarios while keeping the model structure fixed. In other words, only *p* varies and the rest of the workflow remains identical. As *p* increases, the runtime naturally rises because the script solves the problem for nine different values of *p* across 100 scenarios for each selected variant. This subsection therefore focuses on the setup and code, while results are analysed later.

At the runtime start, the user selects the specific variant. Consequently, the script resolves the corresponding CSV path and loads the route data. This driver mirrors the variant specific main scripts. However, all three variants can be run within a single program. The selection logic is shown in the appendix. For the analysis, the program executes each variant across 100 scenarios to compare solutions and runtimes. As a result, runtime increases substantially due to the higher computational demand.

Importantly, the main parts like route loading from the CSV, the Haversine computation, distance matrix construction, weight calculation and the GA steps remain the same as before. The only difference is that everything now resides in a single file. The first substantive change is a loop that iterates *p* from 4 to 12, which is running the program for each value of *p*. After each run, the script computes the average objective across all scenarios and identifies the most frequently selected stations. V1 and V2 build a Gurobi model and solve it exactly, as in the main program.

By contrast, V3 with the GA uses a population of 100, an elite fraction of 0.2, a mutation rate of 0.1 and an early stopping criterion of 100 generations without improvement. Additionally, the GA has a maximum of 1,000 generations. This configuration ensures a fair runtime comparison against the exact MILP runs. To ensure reproducible GA results for each p, this part initializes a dedicated random number generator with a fixed seed *random.Random(42)* immediately before the GA run.

The console prints both, the average objective value and the list of frequently selected sites. For each scenario, the script clones the base route, instantiates a fresh simulation and computes the weights per point from simulated weather and participant counts, together with the injury risk table. Additionally, V2 and V3 map 100 m points to km bins, as before. Beyond the average objective and the list of frequently selected stations, the script records total runtime per *p* and per variant to facilitate later efficiency comparisons.

In the Results section, this study will assess whether an optimal number of stations exists and quantify the marginal improvements in the objective as *p* increases. Ultimately, this study will compare all three variants side by side with their including runtimes.

5.10. Real World Stations

Finally, this chapter computes the weighted p-median objective for the real Vienna setup with the four stations as a fixed layout baseline. Accordingly, there is no optimization and only an evaluation of the current locations. The four real world station coordinates are hard coded and *p* is fixed at four. We reload all the route points from the CSV, construct the distance matrix and build the weights w_i exactly as in the main pipeline. Each real station coordinate is snapped to its nearest route node and sets $y_j = 1$ for those indices and $y_j = 0$ otherwise. This research then computes and reports the objective value for the real world stations. Additionally, for a fair comparison, this study computes the objective function for both variants. Because V2 includes more route points, its objective value will be higher in absolute terms. Consequently, the real world layout will have two objective values, one for each variant.

6. Results

With all model variants executed, the next section reports the empirical findings, compares the optimized layouts and evaluates their performance under varying scenarios.

6.1. Real World Stations

As a point of comparison, this chapter first examines the real world case and evaluates the station layout. The aim is to establish a fixed layout by using the official coordinates for the emergency stations, which are evenly spaced along the course. As shown in *figure 10*, the route is reconstructed faithfully and the stations are positioned adjacent to the course.

Figure 10: Original Emergency Stations during the Vienna Marathon

Thus, the pipeline remains the same but there is no optimization and only an evaluation of the objective over 100 scenarios. Because the grids differ (1000 m vs. 100 m), raw objectives are not directly comparable across variants. Nevertheless, each variant can be compared with the real case baseline. Accordingly, this study computes the objective for the real world stations under both grid resolutions. The averages are *13,924.38* for V1 and *129,323.20* for V2.

Furthermore, the real and evenly spaced layout provides as reference level against which the optimized variants will be assessed. Subsequent sections examine whether model driven placements reduce the objective relative to their evenly spaced locations. Given the earlier hotspots indications we anticipate that the optimized placements will reallocate capacity towards those zones which also cover a lot of beginning sections as well. Thereby, the variants will improve this baseline with the 1000 m resolution and then a more precisely one on the 100 m grid.

6.2. Solution Variant1_1000m

First, the results for the 1000 m setup with 44 route points and an exact MILP over 100 scenarios will be reported. Accordingly, the mean weighted p-median objective is *9,796.77*. Most scenarios lie between *~9,300* and *~10,500*, while the overall range spans *~7,800* (best) to *~14,100* (worst). The appendix shows the values of every single scenario. Subsequently, this research compares these values with the other variants.

The key outcome is the placement of the four emergency stations, as shown in *Table 6.* Notably, the most frequently selected indices are *i = {42, 17, 35, 38},* which forms a stable quartet with only low frequency swaps at nearby locations. While the selection of most frequently chosen stations across 100 scenarios could be misleading, if spatially close alternatives were selected with similar frequencies across different runs, the results in this case are sufficiently distinct to disregard that concern. Moreover, the total runtime over 100 scenarios is *11.33 seconds (s)*, which demonstrates that the 1000 m setup for 44 given route points is solved very quickly.

i	***Number of Selections***
42	100
17	83
35	83
38	81
39	19
31	17

Table 6: Most selected stations for Variant1 with 1000m

Notably, the solution reveals a robust late race hotspot as *km 42* is a must have and was selected in every scenario. This indicates consistently high weighted demand near the finish. Together with *km 35* and *km 38*, it forms a tight anchor zone that covers the final ~7–10 km, where risk is concentrated. As shown in *figure 11*, coverage also extends to early segments of the course, because the route loops and certain sections are passed more than once, where stations in the close finish area incidentally serve parts of the opening km’s as well. Crucially, this effect is handled correctly by the distance matrix, which captures the true point distances along the looped geometry. Moreover, *km 17* acts as a mid-course support and appears in

83% of the runs, which suggests a secondary demand node in the middle third. Occasional swaps around *km 31/34/39* reflect modest sensitivity to the scenario variation rather than structural change.

Figure 11: Solution for Variant1_1000m with four locations

Accordingly, this spread is informative when benchmarking against the 100 m grid or testing larger *p*. Overall, the 1000 m setup provides a clean baseline with the mean of *9,796.77*, that already is an improvement on the current real world layout. Operationally, if resources are tight, the organizers should prioritize late race coverage (*km 38–42*) and maintain at least one mid-course station (around *km 17*). The mid-course node near *km 17* sits on the western part and is essential to cover this isolated section. Without it, response distances in the west would increase dramatically. The two posts in the east are justified because the course passes there twice in the outbound and late return, so they jointly cover early km's and the late race zones where demand peaks.

6.3. Solution Variant2_100m

This section reports results for V2 with the 100 m grid and 425 route points. The mean weighted p-median objective across 100 scenarios is *88,405.11*. Because V2 includes around 10 times more demand points than V1, the objective naturally scales upward. Therefore, raw magnitudes are not directly comparable to V1. Scenario wise values range from around *73,700* to *96,000*, with most runs clustering at *~85,000–92,000*. All 100 scenario values are listed in in the appendix.

The placement of the four emergency stations is reported in *table 7.* Notably, the most frequently selected sites are *i = {17.5, 41.6, 38.6, 35.1}.* However, compared with V1 the pattern is less clear, because the 100 m grid disperses more selections across adjacent nodes. Accordingly, low frequency neighbours such as *38.5, 35.2*, and *41.7* also appear as credible alternatives for effective placements. Moreover, the total runtime is substantially higher than in V1 with *1,165.57 s* (~*19.4 minutes*) and reflects the larger 425×425 instance.

i	*Number of Selections*
17.5	81
41.6	78
38.6	60
35.1	54
38.5	29
35.2	24
41.7	22
4.9	19
32.5	19
8.3	6
38.7	3
5.0	2
38.4	2
35.0	1

Table 7: Most selected stations for Variant2 with 100m

These results mirror V1, where the solution again concentrates on a late race cluster (*km 35–42*) together with a mid-course node (*km 17.5*). Moreover, because the route includes loops and repeated segments, some selected locations also provide incidental coverage of early km's, as previously discussed. Accordingly, the spatial logic matches V1 and *figure 11* shows placements that are closely aligned with *figure 12*. Furthermore, the 100 m discretization captures the route geometry more faithfully, where selected sites appear at finer km decimals and station frequencies spread slightly across adjacent nodes. Hence, a more granular yet qualitatively similar placement.

Figure 12: Solution for Variant2_100m with four locations

Because V2 discretizes demand at 100 m, the absolute objective is naturally higher. Therefore, cross variant comparisons should emphasize patterns and relative improvements, not raw magnitudes. Operationally, prioritize finish area coverage and retain one mid-course station around *km 17–18*. In sum, the 100 m grid refines rather than changes the same hotspots identified in V1.

6.4. Solution Variant3_Metaheuritsic

In order to reduce the total runtime of V2, this study tests a GA metaheuristic to obtain solutions more quickly while the quality maintains the same. The setup remains exactly like as V2 with 425 route points across 100 scenarios. The population size is 100 with 20 % elite selection, the mutation rate with 10 % and early stopping is applied after 100 generations. The mean objective value is *88,443.64* and effectively the same as V2 with *88,405.11*, which leads to a very small difference. The distribution ranges from ~*75,000* to ~*96,000* and, similarly to V2, clusters around *85,000–92,000* like seen in the appendix.

The most frequently selected stations in *table 8* are *i = {17.5, 41.6, 38.6, 35.1}* with additional low frequency neighbours *(e.g., 41.7, 38.5, 35.2, 4.9, …).* Overall, the GA yields a longer tail of rare picks due to the solution procedure. Nevertheless, the total runtime was *20.67 s* and is about 56× faster than V2 with *1,165.57 s* but with exactly the same four solutions.

i	*Number of Selections*
17.5	61
41.6	57
38.6	48
35.1	45
41.7	32
38.5	29
35.2	18
17.4	17
4.9	11
31.3	11
41.8	9
8.3	9
35.5	8
38.7	7
34.0	4
35.4	3
38.7	3
31.2	3
8.4	3
35.0	3
17.6	2
17.3	2
34.9	2
41.5	1
9.4	1
37.7	1
9.6	1
37.6	1
41.9	1
38.8	1
38.4	1
33.8	1

Table 8: Most selected stations for Variant3 with Metaheuristic

The graphical solution is identical to *figure 12*, since the setup is the same. Because both variants optimize the same objective with the same weights, the mean outcomes coincide. The GA consistently finds near optimal solutions across scenarios. Thus, the metaheuristic delivers quality that is good enough compared with the detailed and exact method of V2, while it achieves a significantly lower runtime. Additionally, to obtain a clearer result this variant could increase to 1,000 scenarios. This would not increase the runtime significantly for the GA, whereas for V2 the approximately *19.4 minute* runtime would increase substantially.

The GA uses stochastic operators like random initialization, crossover, mutation and stops after a finite number of generations. Combined with many near equivalent adjacent locations at the 100 m resolution, this yields to more unique station picks. Overall, the aggregate pattern with a late race cluster, which also covers early route sections plus one mid-course node matches V2. Finally, the practical takeaway is that the MILP should be used when exactness is required, whereas the GA should be used for speed with larger grids, higher *p* or quick sensitivity sweeps, since it preserves the same spatial logic at a fraction of the runtime.

6.5. Solution Extreme Scenarios

After reviewing all optimal station placements, this study already provides a solid solution for the real world event. To provide a clearer overview of key factors and scenario variability, the next section also analyses the extreme scenarios, which examines the optimal placement when those factors exert maximum or minimal influence. Importantly, the risk floor ε ensures that even when weather and participant counts have no effect the injury risk table still contributes at least 10%. Otherwise, the search for emergency placements would become more only based on distance, as every route point would carry equal importance.

6.5.1. Extreme Scenario Variant1_1000m

The results for the two extreme scenarios and for the average scenario are provided in *table 9* below.

Scenario	***Objective***	***Stations***	***Factors***
Cold & few participants	10,821.22	[17, 35, 38, 42]	Weather: -1°C; Participants: 26.016
Average	9,880.97	[17, 35, 38, 42]	Weather: 13,5°C; Participants: 36.049
Hot & many participants	7,696.93	[31, 34, 39, 42]	Weather: 28°C; Participants: 46.083

Table 9: Results of extreme scenarios for variant 1

Notably, the pattern remains stable under cold/few and average conditions, where the quartet *{17, 35, 38, 42}* persists and reinforces a late race hotspot plus one mid-course site. Under cold/few conditions the mid-course node reappears, because early and mid-sections gain relative weight. The epsilon floor keeps mild local risk visible, so early km's never go to zero even in cool scenarios. By contrast, under hot/many conditions the set shifts upstream to *{31, 34, 39, 42}*, what effectively extends the finish area cluster backward along the course. Consequently, the mid-course station drops out, which reflects the greater weight on late segments under more demanding conditions.

These three scenarios span the stochastic mean, which is useful for framing later comparisons to the 100 m grid and to higher numbers of stations *p*. Moreover, the weighting scheme with risk floor *ε* and severity means that more severe settings further concentrate weight on the already risky late segments. Consequently, pulling stations closer around the finish reduces the weighted distance, which explains the lower objective in that scenario. In addition, the cold/few participants case yields the same solution as the 100 random scenarios in V1 and *figure 11* already illustrates that placement. By contrast, for the hot/many participants extreme, *figure 13* shows the corresponding emergency station layout.

Figure 13: Extreme scenario with hot and many participants condition for Variant1_1000m

6.5.2. Extreme Scenario Variant2_100m

The three deterministic scenarios with fixed inputs reveal clear differences on the 100 m grid compared with the 1000 m grid. Specifically, under colder conditions with fewer participants, V2 yields an objective of *95,254.57* with stations *{5.0, 17.5, 38.4, 41.7}*. Under average conditions, it returns *80,999.68* with *{17.5, 35.1, 38.6, 41.6}*. Finally, under hot weather with many participants, this variant gives *74,281.05* with *{4.9, 32.5, 38.7, 41.6}*, which illustrates where the patterns diverge in comparison to the coarser grid. These results are presented in *table 10*.

Scenario	***Objective***	***Stations***	***Factors***
Cold & few participants	95,254.57	[5.0, 17.5, 38.4, 41.7]	Weather: -1°C; Participants: 26.016
Average	89,099.68	[17.5, 35.1, 38.6, 41.7]	Weather: 13,5°C; Participants: 36.049
Hot & many participants	74,281.05	[4.9, 32.5, 38.7, 41.6]	Weather: 28°C; Participants: 46.083

Table 10: Results of extreme scenarios for variant 2

Notably, V2 exhibits a late race cluster around *km 38–42* in all scenarios. However, the early node at *km 4.9–5.0* emerges when the model must cap long assignments in the opening km's with only four stations available. In particular, under cold and few participant conditions, where severity is lower, the weights are more uniform and early coverage gains importance. Because every route point becomes more evenly weighted and only the risk floor ε nudges the weights, the stations tend to be more evenly spaced along the course, which yields to *{5.0, 17.5, 38.4, 41.7}*. *Figure 14* illustrates this evenly spaced configuration.

Figure 14: Extreme scenario with cold and few participants for Variant2_100m

Under hot weather with many participants, the late race weights become dominant. Nevertheless, a single early site around *km 4.9* still reduces total distance, while the other three stations are placed in the late segments in line with the finish area logic. Consequently, the key difference between the two extreme scenarios is the mid-course station. When conditions are cold with few participants, late segments carry less weight, so the model has more latitude to position sites at the start and mid-course. By contrast, under hot and crowded conditions the late segments gain importance, the layout becomes less evenly spread and the model concentrates three of four stations in the final part of the course. Even so, the chosen locations still incidentally cover early km's, consistent with the main setup. This pattern is illustrated in *figure 15*.

Figure 15: Extreme scenario with hot and many participants for Variant2_100m

The raw objectives are not comparable across grids, but the patterns can be compared. Both variants identify a finish area hotspot near *km 41–42* and indicate the need for one additional support site. Importantly, the 100 m discretization captures geometry and loops more faithfully, so a distinct early station near *km 5* becomes worthwhile, whereas in V1 the 1000 m grid is coarser and that early coverage is effectively absorbed by the mid-course node at *km 17*. Under hot conditions, V1 shifts stations upstream within the late segment, for example *31, 34, 39,* and *42*, while V2 still keeps an early site at *km 4.9*. This occurs because the finer grid allows the model to balance early coverage and late concentration more precisely, which makes the early site effective. With V1 and the 1000 m grid, it is harder for the program to place stations effectively in this extreme scenario. But after the comparison of *figures 13* and *15*, it underlines that both solutions are very similar, since *km 4.9-5.0* from V2 and *km 34* from V1 are very close to each other. The only difference is the higher flexibility of V2 to place the station more precisely.

6.5.3. Extreme Scenario Variant3_Metaheuritsic

Now this study examines the extreme scenarios for V3, which uses exactly the same setup as V2. For completeness, the GA settings remain unchanged from earlier experiments and the results from a single run are reported in *Table 11.*

Scenario	***Objective***	***Stations***	***Factors***
Cold & few participants	95,299.92	[17.5, 35.5, 38.4, 41.6]	Weather: -1°C; Participants: 26.016
Average	89,143.39	[17.5, 31.3, 38.6, 41.6]	Weather: 13,5°C; Participants: 36.049
Hot & many participants	74,281.05	[4.9, 32.5, 38.7, 41.6]	Weather: 28°C; Participants: 46.083

Table 11: Results of extreme scenarios for variant 3

These results again reveal late race hotspots with one early or mid-support, which shifts between approximately *km 4.9 to 5.0* and *km 17 to 18* depending on the scenario. Importantly, the GA shows dispersion because it relies on stochastic initialization, crossover, mutation and stops after a finite number of generations. This creates many near equivalent nodes at 100 m, so the algorithm can return different but similarly good station sets. The objectives are very close to those of V2 and the small gaps reflect stochastic search rather than a different spatial logic.

Additionally, this study does not overinterpret a single GA realization, since one run is seed dependent. Therefore, this variant requires repeated runs and may select neighbouring sites with similar quality. By contrast, V2 serves as the authoritative reference because the exact MILP is deterministic, while V3 primarily demonstrates speed with comparable patterns. The strength of V3 is not decisive here because only one run is available. Consequently, V2 does not involve a substantially larger runtime in this setting and it guarantees the exact solution. For that reason, this study does not include a graphical illustration for V3, since the solutions are qualitatively good but not as definitive as those from V2. Since both variants share the same setup, the discussion and figures can focus on the solutions already reported for V2.

6.6. Number of Stations

In the following chapter, this study looks beyond the real world case and examines how increasing the number of stations improves the objective. Because, in a p-median model, a larger *p* necessarily lowers the objective, we focus on how much it improves and where the number of stations becomes efficient. Accordingly, this study reports both cumulative improvement relative to the real world baseline $p = 4$ and marginal improvement from $p-1$, up to $p = 12$. Each value of *p* is evaluated over 100 scenarios and the most frequently selected stations will be listed. Moreover, to compare the different variants, the program records runtime per *p* and total runtime. For clarity, the summary tables show only the most selected stations across the 100 scenarios. Finally, at the end of each variant, we provide a recommendation on a more suitable number of stations.

6.6.1. Number of Stations Variant1_1000m

V1 runs quickly with about *4 s* per *p* and *36.40 s* in total. The mean objective drops from *9,866.76* at $p=4$ to *3,710.65* at $p=12$. Cumulatively, the improvement rises from *15 %* at $p = 5$ to *62 %* at $p = 12$, while the marginal gain per added station ranges between *8 %* and *16 %.* Consistently, a core cluster near *km 42–43* and *km 38–40* remains, while the first additional sites fill the mid-course gap around *km 31–34* and *km 24*. As *p* increases, coverage extends toward *km 37* and the early sections between *km 1* and *9*. An overview is provided in *table 12*. The selected stations follow the same pattern up to $p = 8$, then shift slightly at $p = 10$ with the inclusion of *km 24*. At $p = 12$, new early nodes at *km 1* and *km 9* appear, which strengths the late race cluster. Notably, $i = 43$ represents the finish line rather than *km 43*.

p	*Selected Stations*	*Runtime*	*Objective*	*Improvement cumulative*	*Improvement marginal*
4	{42; 17; 35; 38}	4.43 s	9,866.76	-	-
5	{38; 17; 41; 43; 35}	4.29 s	8,373.93	15%	15%
6	{17; 31; 34; 38; 41; 43}	4.00 s	7,027.71	29%	16%
7	{17; 31; 34; 38, 41; 43; 40}	3.99 s	6,370.69	35%	9%
8	{17; 31; 34; 41; 43; 38; 37; 40}	4.03 s	5,732.15	42%	10%
9	{17; 31; 34; 41; 43; 37; 38; 40; 24}	3.94 s	5,166.70	48%	10%
10	{17; 24; 34; 37; 41; 43; 31; 38; 39; 40}	3.93 s	4,521.63	54%	12%
11	{17; 24; 34; 37; 41; 43; 38; 31; 39; 19; 40}	3.94 s	4,168.15	58%	8%
12	{17; 24; 31; 34; 37; 38; 40; 41; 43; 39; 19; 1}	3.85 s	3,710.65	62%	11%

Table 12: Improvement by increasing number of stations for Variant1_1000m

Figure 16 illustrates the cumulative and marginal improvements. The curve shows clear gains up to about *p = 6–7* with marginal plus of *15–16 %* and cumulative rising from *15 %* to *29 %.* Between *p = 7* and *p = 10* the gains remain solid, with *9–12 %* marginal per added station and cumulative with *35 % to 54 %.* Beyond *p=10* the returns taper with *9–11 %* marginal and cumulative with *58–62 %*. Overall, the pattern stays stable with the late race core *(41–42, 38–40)* persists, added sites first fill the mid-course gap *(31–34, 24)* and then extend to early km's *(1–9)*.

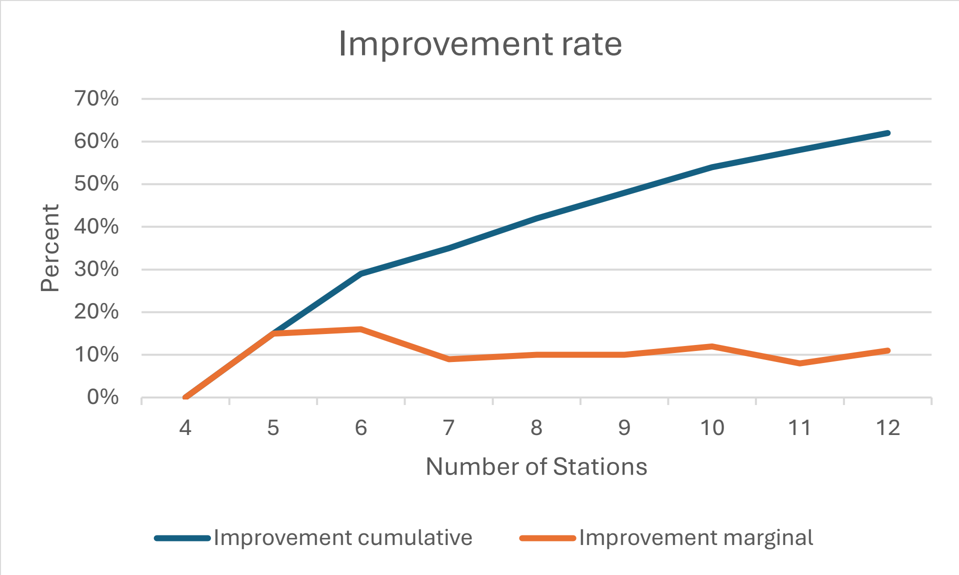

Figure 16: Improvement for number of stations in Variant1_1000m

In sum, the curve suggests *p = 8* as a sensible target, where coverage is balanced and the cumulative reduction reaches *42 %,* which offers an attractive cost benefit. If resources permit, *p = 9–10* yields an additional *6–12 %* improvement and remains efficient. Beyond *p = 10* the gains become modest, so higher counts should only be considered when staffing and logistics are readily available. As an illustration, *figure 17* depicts an even layout that concentrates on the late race corridor while also filling the mid-course gap. Some early km's are incidentally covered by those late stations as well because of the route geometry.

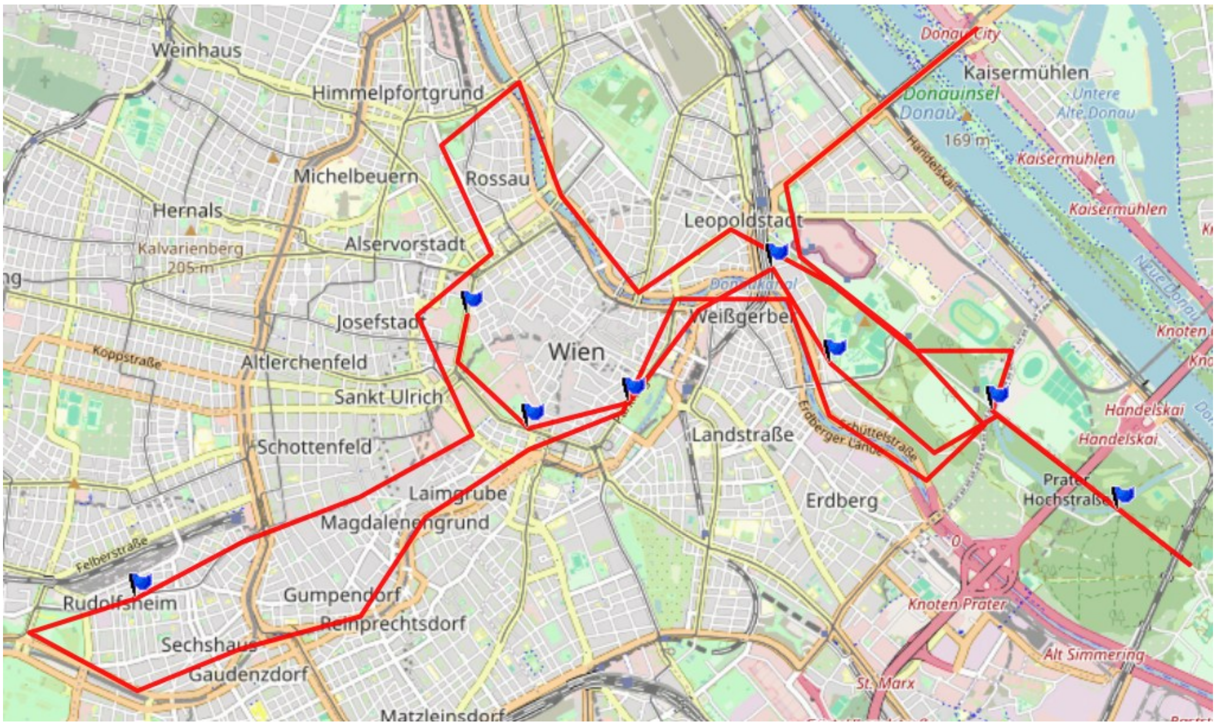

Figure 17: 8 stations for Variant1_1000m

6.6.2. Number of Stations Variant2_100m

Similarly, every solution exhibits a persistent late race cluster with numerous selections in *km 37–42*. In addition, an early anchor at *km 4.9* emerges from *p = 5* onward, while mid-course anchors recur around *km 17.0–17.5, 22.8/24.9* and *km 32.5/32.8*. As *p* increases, additional sites like *km 34–36.9* arise. Nevertheless, the late race focus remains intact and *table 13* reports the complete results for V2.

p	*Selected Stations*	*Runtime*	*Objective*	*Improvement cumulative*	*Improvement marginal*
4	{17.5; 41.6; 38.6; 35.1}	4,418.28 s	88,451.24	-	-
5	{17.5; 4.9; 32.5; 41.6; 38.6}	2,665.27 s	76,630.13	13%	13%
6	{4.9; 32.5; 17.5; 41.3; 8.2; 38.6}	2,345.32 s	66,434.51	25%	13%
7	{4.9; 17.5; 32.5; 22.8; 37.9; 9.4; 41.6}	2,233.90 s	58,454.27	34%	12%
8	{32.5; 17.5; 22.8; 4.9; 38.2; 41.6; 39.7; 37.9}	9,405.38 s	54,384.03	39%	7%
9	{32.5; 17.5; 38.1; 10.6; 22.8; 35.3; 25.0; 36.9; 39.3}	11,846.48 s	50,525.27	43%	7%
10	{38.1; 17.0; 32.5; 39.3; 12.3; 24.9; 40.9; 42.0; 37.0; 35.3}	38,260.26 s	47,137.89	47%	7%
11	{17.0; 24.9; 40.9; 42.0; 4.9; 12.3; 32.8; 34.5; 0.8; 32.5; 38.3}	5413.54 s	44,336.37	50%	6%
12	{24.9; 4.9; 17.0; 32.8; 34.5; 38.2; 37.1; 42.1; 42.0; 21.9; 40.9; 37.0}	4136.28 s	41,858.66	53%	6%

Table 13: Improvement by increasing number of stations for Variant2_100m

For V2, the objective decreases monotonically from *88,451.24* to *41,858.66* with *12* stations. The cumulative improvement increases from *13 %* to *53 %*, with the largest step between p = *4* and p = *5.* Likewise, the marginal gain is highest at the outset with around *13 %* and subsequently tapers to about *6 %*. By contrast, the runtime profile differs significantly from V1. It is already higher with about 2*,000–4,400 s* and then rises to *9,400 s* and *11,800 s* for p = *8* and p = *9*. But the main peak is at *38,360 s* for p = *10.* Thereafter, the solver still requires around *5,000 s* for p = *11–12*. Overall, the total runtime is *80,724.71 s (≈22 hours)*, which represents a dramatic increase relative to V1 (*36.40 s*).

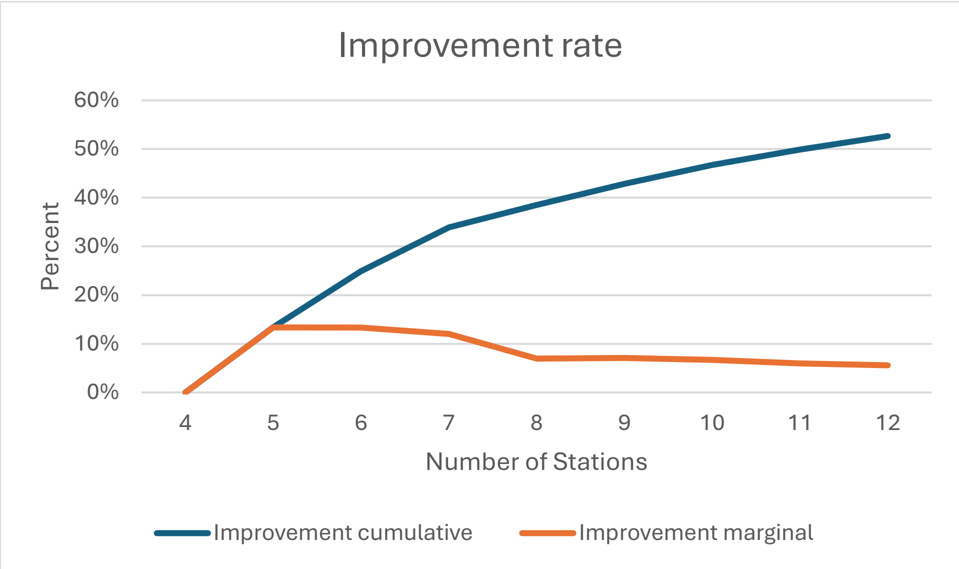

Figure 18: Improvement for number of stations in Variant2_100m

Overall, the model gains most of the objective reduction with p = *5–7*. However, beyond p = *7* the returns diminish sharply with each additional station. Additional sites mainly thicken the late race block and add redundancy rather than opening new high impact areas. The station geography is consistent with weight peaking late in the course. Early and mid-course anchors provide baseline coverage, while the final *km 5* absorb additional capacity.

Therefore, the primary recommendation is *p = 7*, which represents the efficiency sweet spot with a *34 %* cumulative gain and still reasonable marginal benefits, as illustrated in *figure 18.* If resources allow and extra robustness in the late race sector is desired, organisers could extend to *p = 8*. In contrast, any setting above *p = 8* should be considered only with abundant resources or strict coverage targets, because the additional gains are around *6–7 %* at best and runtimes may be volatile. The graphical illustration for *p = 7* stations is shown in *figure 19*.

Figure 19: 7 stations for Variant2_100m

6.6.3. Number of Stations Variant3_Metaheuritsic

Because runtime is significantly high in V2, this study also tests the quality of the solutions in V3. First of all, the total runtime drops to *670.77 s*. The station pattern remains the same, as a stable late race cluster persists in *km 37–42*. In addition, an early anchor at *km 4.9–5.0* appears frequently. Mid-course anchors recur around *km 17–18, 22–25, 32–33.9*, and *35.3–35.4*. Nevertheless, across 100 scenarios the algorithm selects many more distinct points within these zones than in V2, which indicates near ties among adjacent locations. The runtimes are consistently low at about *23–126 s*. Consequently, the GA in V3 is much faster than in V2. An overview of all selected stations and runtimes is provided in *table 14*.

p	*Selected Stations*	*Runtime*	*Objective*	*Improvement cumulative*	*Improvement marginal*
4	{17.5; 41.6; 35.1; 38.6}	23.46 s	89,200.91	-	-
5	{17.5; 4.9; 32.5; 41.6; 8.2}	32.28 s	77,572.33	13%	13%
6	{17.5; 4.9; 32.5; 8.2; 22.6; 41.3}	125.94 s	67,484.94	24%	13%
7	{32.5; 4.9; 17.5; 37.9; 22.8; 41.6; 9.5}	114.83 s	58,911.54	34%	13%
8	{32.5; 17.5; 4.9; 22.8; 33.9; 37.9; 35.3; 41.6}	56.35 s	54,249.88	39%	8%
9	{32.5; 17.5; 35.3; 33.9; 22.8; 39.3; 41.9; 37.0; 38.7}	75.08 s	50,395.92	44%	7%
10	{32.5; 38.1; 12.3; 4.9; 22.8; 35.3; 17.0; 37.0; 39.3; 41.9}	74.04 s	47,373.25	47%	6%
11	{32.5; 38.2; 42.0; 17.0; 12.3; 24.9; 0.8; 37.0; 35.3; 25.4; 42.1}	80.66 s	44,503.21	50%	6%
12	{17.0; 0.8; 24.9; 35.4; 38.2; 18.9; 37.0; 42.0; 38.1; 4.9; 12.3; 42.1}	88.14 s	42,079.29	53%	5%

Table 14: Improvement by increasing number of stations for Variant3_Metaheuristic

Importantly, the cumulative improvement rises from *13 %* at *p = 5* to *53 %* at *p = 12*. The growth is steep up to *p = 7* and then levels off with a further gentle flattening after *p = 8*. By contrast, the marginal improvement is *13 %* when adding the *5th–7th* station and then decreases steadily, till it reaches *5 %* at *p = 12*. Thus, marginal gains decline noticeably after *p = 7*. Both improvement curves are shown in *figure 20*.

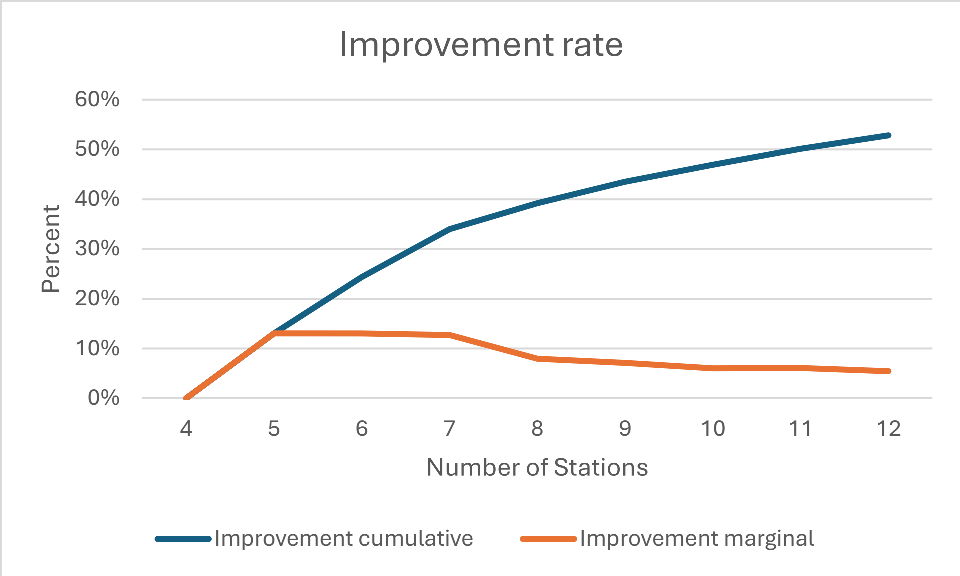

Figure 20: Improvement for number of stations in Variant3_Metaheuristic

The geographic zones are consistent with V2 and indicate that the solutions remain qualitatively strong. By contrast, V3 is dramatically faster with an overall runtime of around 11 minutes compared to approximately 22 hours in V2. At *p = 7*, V3 delivers comparably good solutions in around *115 s*, whereas V2 required *2,234 s*. Because the candidate grid is dense, the metaheuristic often selects different points within the same zone across scenarios. Therefore, after 100 scenarios the exact station location is not always clear, even though the zones themselves are robust. As in V2, benefits are front loaded are the marginal gains, which drop after *p = 7*, while cumulative gains level off after *p = 7* and again after *p = 8*.

In summary, *p = 7* remains the efficiency sweet spot, which offers strong cumulative gains with a very fast runtime. If additional robustness in the late race sector is desired, *p = 8* is a reasonable extension with modest extra benefit. Importantly, the *p = 7* station set in V3 differs from that of V2. For a more precise solution choice, refer to *figure 19* (V2), which is more definitive. Nevertheless, V3 finds a good *p = 7* solution in a much faster time.

7. Discussion

The following chapter interprets these results in light of the theoretical framework and assesses their practical relevance for event logistics and emergency service planning.

In sum the results answer the research question. A stable finish cluster and one mid-course node minimize total weighted distance under uncertainty. The optimized layouts show where to place stations along the route. The core result is a stable late race hotspot from *km 37* to *42* and one mid-course node near *km 17* to *18* supports the rest of the route. The real layout is close to the target, since three of four posts already lie in good zones. *Figure 21* illustrates a comparison between the real world layout and the solution of V2. Only one location in the north moves into the efficient zone, which reduces the total distance.

Moreover, the eastern part of the route is very efficient, because it lies close to the finish cluster as well as to early km's due to loops and crossings. Therefore, one medical station in this area can cover the early parts of the route, as runners pass this section twice. Since participants follow the official course and some zones are passed two times, the matrix is an appropriate choice. The comparison uses the real world case and the solution of V2, because V2 is the most precise option.

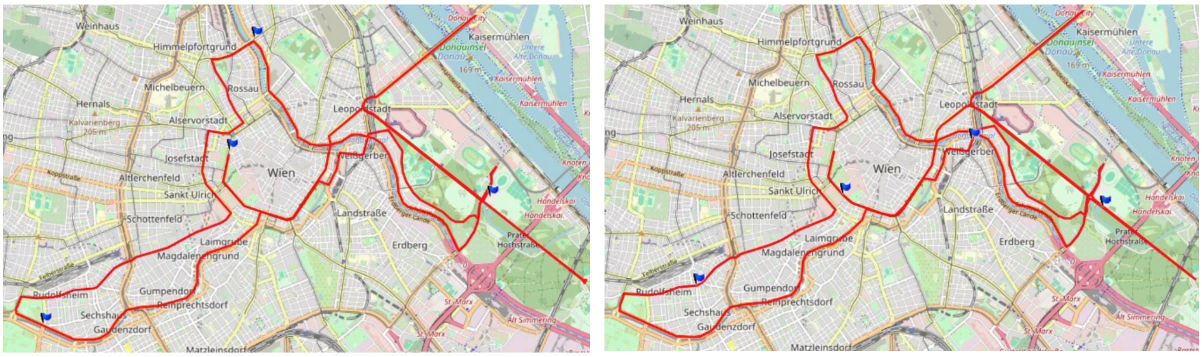

Figure 21: Comparison between real world case and variant 2 solution

The real world layout is nearly equally spaced and the objective is higher than in the optimized sets. The gap results from a missing focus on the late hotspots and from not placing a site in the efficient area in the east. Nevertheless, the real world plan shows good practice with only one post that needs to be replaced into the efficient zone. The final results show that the zones remain stable across the scenarios, while the exact km varies more on the 100 m grid. Yet the zones do not change.

In the next step this study compares the three main variants. Overall, all three variants deliver good and qualitative solutions. V1 provides a clear pattern with very short total time. V2 is the most precise on the 100 m grid, but its runtime is the longest, although still acceptable. The 100 m grid follows the route more closely while the 1000 m grid is coarse. Patterns and relative gains are comparable across both grids. In larger real world cases V2 may be too large. Therefore, V3 reproduces the pattern of V2 with a much lower runtime. Consequently, V3 suits real world cases where V2 is too large and cannot return a solution in a reasonable time. *Table 15* provides a final side by side comparison.

	Variant 1 (1000m)	***Variant 2 (100m)***	***Variant 3 (100m)***
Solver	Gurobi	Gurobi	Genetic algorithm
Runtime	Very fast	Very long	Fast
Solution	Good & clear	Exact point choice	High qualitative
Disadvantage	Coarse grid	Long runtime	More ties among nearby points

Table 15: Final comparison of the variants

Moreover, the use of a GA shows that the model is adaptable. The method applies to any route based problem and not only a marathon. Examples include Ironman, triathlons, car or bike races and other events with a fixed route. The required changes concern the coordinates and the reasonable ranges for the uncertain factors. In addition, this model can also be adjusted by the number of coordinates. Thus a grid with 1 m steps is possible if a case requires it. This was not useful in this study but it may help on short routes with many tight curves. In case of other sport events, the injury risk table also needs to be adapted but should have similar pattern with an increasing risk throughout the route.

For longer routes such as 100 km races, Ironman events or Tour de France, the GA delivers solutions close to the exact solver in much less time. Consequently, when the route problem becomes too large, V3 offers a viable high quality procedure. In such cases a metaheuristic can replace an exact solver that requires too much time.

Moreover, the model is sensitive to the scenario sharpness s and the risk floor ε. When weather and participant numbers are low, s is close to zero. A small s pulls weights toward the course mean and smooths peaks. As a result, all points become more equal in importance. To keep the finish area visible, this study used a risk floor ε that keeps a ten percent risk share in every scenario based on the injury table. By contrast, a large s on hot days with many participants raises the importance of hotspots near the finish line. Furthermore, the number of stations was also an interesting part in the analysis. Overall seven stations highlight the sweet spot with most improvement of the objective function. The organizers could consider eight stations for extra safety near the finish line if resources are available.

These findings align with the foundational facility location literature and extend it to a route event under uncertainty. The objective to minimize the sum of distances between demand nodes and selected facilities follows the median logic on networks set out by Hakimi (1964) and the interpretation of average access costs in ReVelle and Swain (1970). The model applies this objective on a discretized road route with per km weights and scenario inputs for weather, participants and injury risk. The classic formulations are static and do not include route structure or uncertainty. This study preserves the same goal and adapts it to a marathon context with late race hotspots that matter for access to care (Hakimi, 1964; ReVelle & Swain, 1970).

Relative to work on emergency service siting, the approach is consistent with the p-median perspective that Jia et al. (2005) recommend for medical access. They argue for scenario inputs when demand or travel conditions vary. Therefore, the contribution is to operationalize this view on a marathon by weighting km according to risk and by showing that a finish area cluster plus one mid-course node minimizes expected distance under uncertainty. Jia et al. (2005) discuss model families and stochastic inputs in general settings (Jia et al., 2005). Consequently, this case delivers a route level implementation with actionable zones instead of single points.

The governance assumptions match the mass event guidance in Yao et al. (2017). They emphasize an Incident Command System and a unified command across medical, safety and transport units. Both accounts require coordinated dispatch and clear communication. The difference is the scope, where Yao et al. (2017) describe the organizational frame (Yao et al., 2017). This study adds the quantitative siting logic that an operations centre can execute on race day.

The empirical medical literature explains the spatial and temporal pattern that this optimization recovers. Nguyen et al. (2008) and Tan et al. (2014) report the cluster of incidents in the late race and near the finish. Heat and fatigue are key drivers. Chiampas and Jaworski (2009) document surge effects under extreme conditions that force contingency actions. This leads to the finish cluster from *km's 37* to *42* with a supporting mid-course node. Which is therefore consistent with prior evidence. The difference is that these studies are descriptive and protocol oriented. The model of this study converts the regularities into route weights and station choices that minimize expected distance (Nguyen et al., 2008; Tan et al., 2014; Chiampas & Jaworski, 2009).

Vienna specific evidence further confirms the layout. Brock et al. (2025) show that a small share of participants required care while incidents concentrated at the finish of the Vienna City Marathon. The optimization internalizes this pattern through higher weights for late km's and a finish cluster recommendation. Brock et al. (2025) document the workload and the deployed resources (Brock et al., 2025). This study explains, where to place resources to shorten access distances across plausible race day conditions.

Furthermore, the rescue time priorities and robustness assumptions are in line with Shen et al. (2021). They argue that strict rescue time constraints dominate monetary costs at mega events and that reliable coverage is essential when traffic and conditions vary. Shen et al. (2021) propose an integrated allocation and routing approach with a modified GA (Shen et al., 2021). This study focuses on siting through a p-median, uses distance as a proxy for time and adds scenario variation in weather, participation and injury risk. Both approaches pursue fast access, yet they differ in scope and method.

Finally, the exploration of a heuristic variant connects to Montenegro and Muñoz (2024). They use a GA for an uncapacitated facility location model in a marathon context. The common ground is the use of metaheuristics for large search spaces and the event setting (Montenegro & Muñoz, 2024). The difference lies in the objective and constraints. They solve an open facility problem with a platform focus (Montenegro & Muñoz, 2024). On the other side, this study fixes the number of stations and targets average distance on a discretized route with scenario weights. This shows that the exact p-median core and a heuristic extension are complementary to GA based event tools rather than duplicative.

In sum, literature and findings converge on three pillars. First, minimizing distance is an appropriate objective for emergency access at mass events. Medical demand at marathons peaks late and clusters at the finish, which justifies a finish area cluster supported by a mid-course node. Governance should allow coordinated dispatch and the models need to account for uncertainty in weather and participant numbers. This study combines these aspects for a specific urban marathon with a scenario based p-median model along the route. The approach identifies stable zones that reduce the expected response distance and provide clear and practical siting decisions. (Hakimi, 1964; ReVelle & Swain, 1970; Jia et al., 2005; Nguyen et al., 2008; Tan et al., 2014; Chiampas & Jaworski, 2009; Yao et al., 2017; Shen et al., 2021; Brock et al., 2025; Montenegro & Muñoz, 2024).

Finally, for practical deployment this study highlights a few aspects. The model places medical stations on the route. In the field, the organizers must verify that a post can stand at the proposed point. The route can pass streets, rivers, fences or private areas that block access. Therefore, the output is a mathematical best placement and it serves as a decision aid. The real post may shift a few m but the provided zone remains the same. The same principle applies to the station positions in this study. The solutions place stations on the route, while organizers can choose a nearby feasible spot next to the route that makes the most sense. The main takeaway is to keep the finish cluster and one mid-course post. Use V2 for an exact point choice, since the plan phase for such a large sport event starts months in advance and the team has time to run the program for one day.

8. Limitations

While the findings provide strong evidence for the proposed model's effectiveness, certain methodological and contextual limitations must be acknowledged, as outlined next.

Although the results appear clear, this study has several limitations that readers should consider. First, the scope and population matter. The analysis treats all Sunday participants as one group, which forms the basis for the participants variable. This set includes marathon, half marathon and relay runners. However, the model assumes equal medical relevance across all runner types, which may not appear in every course segment. Half marathon runners leave the course after *km 21*. Nevertheless, the model still counts them in the second half. The reason is the lack of separate data for these runners and the fact that they may also require medical care. This simplification likely has only a modest effect on the results. The risk per segment remains the same, while the participant count should fall after *km 21*. Moreover, the most critical segments lie beyond *km 21*. Even so, a simple robustness check is advisable. Reduce the number of participants after *km 21* by *30* to *50 %* and then assess any change in the station plan and the objective value.

Second, the spatial representation is a limitation. The model places each medical post on a route point. In practice, a station stands a few m to the side of the main course. The study does not verify on site feasibility for every candidate point *j*. Consequently, rivers, busy streets, fences or private property may block access or create safety risks. Therefore, the maps and outputs should guide zone level decisions about where to place a station best. Whether a station can be implemented at that exact spot and how many m the organizers can shift from the model optimum, remains an operational decision for the event team. In sum, the model can be used to identify a suitable zone and then verify access, safety and available utilities before the final placement. Additionally, this study did not consider possible placements outside the main route, which should not change the solution much. Nevertheless, this could be investigated in future work.

Third, the distance metric uses great circle m rather than road travel time. Actual travel time is not known with precision during a mass event, since it depends on closures, crowd pressure, police lines and temporary one way rules. Therefore great circle distance offers a stable and comparable proxy across all candidate points. In other words, a shorter straight line distance usually implies a faster arrival. However, since inner cities are structured with bridges, water, rail lines or blocked crossings, this can weaken the link. Even so, this bias is small for a robust zone plan, because nearby points are close under both measures and a zone level decision absorbs small errors.

Fourth, the risk and weight design introduces uncertainty. The injury risk table is estimated based on historical patterns and therefore can contain errors. Values near the finish may be biased upward by fatigue and global risk share inside the weight scheme. In addition, the chosen scheme mixes medical risk and approach distance in one objective. This mix can shift the final placement. In other words, the injury risk table drives most decisions, which means any bias in that table can dominate the result. Consequently, readers should view the outputs at zone level as guidance. Moreover, a simple sensitivity check helps. Different risk tables and weight sets can be tested and the results can report how much the recommended stations move.

Furthermore, the solution procedure requires attention. A rule that picks the most frequent stations across scenarios can place two posts very close to each other when results are not decisive. In other words, the aggregation step can place facilities side by side even though the base objective prefers a separation. Such proximity is not optimal even if it appears often. In this study the pattern did not occur, therefore it remains a residual risk. Consequently, it should be acknowledged yet set aside for the present conclusions.

Moreover, discretization and grids limit placement accuracy. The model restricts all posts to predefined candidate points rather than positions in between. Finer grids reduce this discretization error yet do not remove it. In practice, a 100 m grid is precise enough to identify near optimal placements, since a shift of 100 m does not change medical access. Additionally, a more detailed resolution with closer steps is possible but is not relevant here. In other applications, the grid can be adapted to 1 m resolution if local constraints require that level of detail.

9. Conclusion

Building on these considerations, the final chapter summarizes the key contributions of this research and highlights implications for both academic and practical applications. The main goal of this study is to identify optimal locations for medical stations along the Vienna City Marathon route with a route based weighted p-median model under uncertainty. The model minimizes the total weighted distance from each route point to its nearest station. To that end, this research compares two resolutions with 1000 m and 100 m grids against the real world baseline.

First, the method recap follows. The scenario design for weather and participants uses a sharpness parameter and a risk floor. Risk weights increase toward the finish and vary across the scenarios. The exact MILP runs on the 1000 m and 100 m grids and the GA runs on the 100 m grid to compute optimal placements. Finally, the evaluation uses 100 scenarios for each variant. In summary, the exact model fits the 1000 m grid and allows rapid analysis. The GA performs well on the 100 m grid and reproduces the same pattern. However, the MILP on the 100 m grid provides the best qualitative solution. The trade-off between spatial precision and runtime remains manageable.

This study shows qualitative robustness and sensitivity. The results reveal a clear core pattern across all variants. A late race hotspot persists from *km 35* to *42* and one mid-course node appears around *km 17* to *18*. The same qualitative layout emerges on both spatial resolutions. In comparison with the real world layout of the Vienna Marathon, the evenly spaced baseline yields a worse objective than the optimized plans. The optimized plans shift capacity toward the finish cluster and mid-course node. This gap shows clear room for improvement in practice. Overall, the quantitative signals are strong. The most frequently selected points concentrate around *km 17* to *18, 35, 38* and *41* to *42*. By contrast, runtime differs significantly between MILP and GA. Even so, each approach delivers a qualitative solution.

The key drivers of the results are the uncertainty factors with the injury risk table as the primary factor. Injury risk rises in the final third of the course and peaks near the finish. Weather and participant counts amplify existing hotspots through the sharpness parameter. As a result, the sharpness parameter and the risk floor determine how strongly the injury risk table shapes the outcome of a given scenario. The risk floor also ensures that a minimum level of local risk remains visible in mild conditions.

In the following, the conclusion continues with the implications. Placements remain stable under scenario variation across both grids. As operational guidance for organizers, they should prioritize a finish area cluster plus one node near *km 17* to *18*. Organizers should treat locations as zones rather than single GPS points. Moreover, the inner loops in the east should be used to gain coverage efficiency and consider dynamic staff with higher capacity near the finish of the race. If resources allow an increase in the number of medical stations, the organizers should raise the count to the sweet spot of seven stations to provide a robust service and capture most of the potential improvement. They should verify each site on location for access, space, shade and safe exits. Importantly, the organizers should concentrate their staff, stretchers, cooling equipment and ambulances toward the last third and the finish. Capacity can be shifted throughout the day. They can start with early posts and later increase the staff, equipment, cooling and transport between *km 35* and *42*.

The workflow can be applied to other route based events such as half marathons, cycling races or triathlons. In each case, the route coordinates, scenario ranges and station budget can be adapted to local conditions. After every event, results from the review phase may inform updates to the risk table and scenario probabilities, while the zone map can serve as a standing reference for permits and partner coordination. Through this annual cycle, the model evolves into a continuous planning tool rather than a single analysis. The limitations have been discussed in the previous section. On site feasibility checks, the straight line distance proxy and spatial discretization affect the accuracy, while weight settings and the estimated risk table influences the results.

Future studies should run on site feasibility checks for every zone. Each candidate site should be verified for access, space, shade, utilities and safe exits. Ambulance entry points and barriers should be considered. Based on these checks, researchers should define an offset tolerance in m that the model must respect. The optimization should restrict solutions to these feasible areas. This step will reduce the gap between model output and what crews can deploy on race day. Additionally, future work could integrate if the placement of emergency stations outside the route would improve the objective function. This study just checked possible emergency locations along the Vienna marathon route but not outside of it.

Possible future work is a dynamic p-median model. Instead of placing stations once, locations could be adjusted over time and across race segments. The event could be divided into distinct periods such as early hours, mid-race and the finish window with demand weights, station counts and staff levels varying by period. A small number of temporary posts might operate only in the late phase, while a clear move plan for mobile teams could be defined and tested through tabletop drills. This would align resources with the late race surge without oversizing the early course. Updated weights could further test how mobile teams or temporary posts affect expected response distances. After a comparison between static and dynamic versions, it may show whether shorter distances and more stable results can be achieved. Overall, this direction offers valuable potential for future research.

The next model could replace great circle distance with estimated travel time between nodes. Separate speed profiles should be defined for foot patrols, bike teams and ambulances under road closure conditions. Travel time estimates should also consider one way segments, pinch points and spectator density. In addition, a sensitivity analysis could compare outcomes based on distance versus time to assess how the objective value and selected zones change. If travel time modeling change the finish cluster or mid-course node, the revised layout should be adopted. If not, distance can remain a suitable proxy.

The next Vienna City Marathon should serve as a pilot. Organizers should preapprove a scenario set and a station plan. During the event, teams should record incidents by km and by period together with response times. After the event, a review should compare the observed pattern with the predicted zones. The risk table and the scenario probabilities should then be updated. This learning loop will turn the model into a standing planning tool and will support transfer to other route events such as half marathons, cycling races and triathlons.

Finally, a finish cluster near the efficient eastern zone and one mid-course node form a robust and practical layout. This pattern remains consistent across scenarios and spatial resolutions and shows clear improvement over the baseline. Therefore, the study provides a solid and applicable foundation for medical planning under uncertainty.

References

Arbon, P., Bridgewater, F. H. G., & Smith, C. (2001). Mass gathering medicine: A predictive model for patient presentation and transport rates. *Prehospital and Disaster Medicine,* 16(3), 150–158.

Arabnarmi, B., Khalilian, F., Kheybari, S., & Ishizaka, A. (2024). Sports complex location selection for traditional games. *Operations Management Research,* 17, 340–362.

Başdere, M., Ross, C., Chan, J. L., Mehrotra, S., Smilowitz, K., & Chiampas, G. (2014). Acute incident rapid response at a mass-gathering event through comprehensive planning systems: A case report from the 2013 Shamrock Shuffle. *Prehospital and Disaster Medicine,* 29(3), 320–325.

Bartens, W. (2017, April 2). Marathon: Ab Kilometer 20 wird es gefährlich. *Süddeutsche Zeitung*. Retrieved October 19, 2025, from https://www.sueddeutsche.de/gesundheit/start-der-marathon-saison-ab-kilometer-20-wird-es-gefaehrlich-1.3443685

Breslow, R. G., Shrestha, S., Feroe, A. G., Katz, J. N., Troyanos, C., & Collins, J. E. (2019). Medical tent utilization at 10-km road races: Injury, illness, and influencing factors. *Medicine & Science in Sports & Exercise,* 51(12), 2451–2457.

Brock, R., Krammel, M., Kornfehl, A., Veigl, C., Schnaubelt, B., Neymayer, M., Grassmann, D., Zeiner, A., Aigner, P., Gabriel, R., Drapalik, S., & Schnaubelt, S. (2025). Emergency point-of-care blood gas analysis during mass gathering events: Experiences of the Vienna City Marathon. *Journal of Clinical Medicine,* 14(7), 2504.

Burke, L. M., Whitfield, J., & Hawley, J. A. (2025). The race within a race: Together on the marathon starting line but miles apart in the experience. *Free Radical Biology and Medicine*, 227, 367–378.

Chan, J. L., Constantinou, V., Fokas, J., Van Deusen Phillips, S., & Chiampas, G. (2019). An overview of Chicago (Illinois, USA) marathon prehospital care demographics, patient care operations, and injury patterns. *Prehospital and Disaster Medicine,* 34(3), 308–316.

Chiampas, G., & Jaworski, C. A. (2009). Preparing for the surge: Perspectives on marathon medical preparedness. *Current Sports Medicine Reports,* 8(3), 131–135.

Day, S. M., & Thompson, P. D. (2010). Cardiac risks associated with marathon running. *Sports Health: A Multidisciplinary Approach,* 2(4), 301–306.

Deng, Y., Zhang, Y., & Pan, J. (2021). Optimization for locating emergency medical service facilities: A case study for health planning from China. *Risk Management and Healthcare Policy,* 14, 1791–1802.

Du, J., Du, Y., Zhang, Y., Liu, Y., & Wei, D. (2024). Identifying key factors and strategies for optimizing automated external defibrillator deployment in China. *Mathematics,* 12, 2829.

García-Vallejo, A. M., Albahari, A., Añó-Sanz, V., & Garrido-Moreno, A. (2020). What's behind a marathon? Process management in sports running events. *Sustainability,* 12, 6000.

German Road Races. (2024). Wien Marathon 2024: Sportliche Highlights & sportliches Miteinander. *German Road Races*. Retrieved October 17, 2025, from https://news.germanroadraces.de/wien-marathon-2024-sportliche-highlights-sportliches-miteinander/

GPS Visualizer. (n.d.). *Draw on a map*. Retrieved October 17, 2025, from https://www.gpsvisualizer.com/draw/

Güden, H., & Süral, H. (2019). The dynamic p-median problem with mobile facilities. *Computers & Industrial Engineering*, 137, 615–627.

Gwalani, H., Tiwari, C., & Mikler, A. R. (2021). Evaluation of heuristics for the p-median problem: Scale and spatial demand distribution. *Computers, Environment and Urban Systems,* 88, 101656.

Hakimi, S. L. (1964). Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research,* 12(3), 450–459.

Hall, S. A., Manning, R. D., Keiper, M., Jenny, S. E., & Allen, B. (2019). Stakeholders' perception of critical risks and challenges hosting marathon events: An exploratory study. *Journal of Contemporary Athletics,* 13(1), 37–49.

Hardcastle, T. C., Samlal, R., Naidoo, R., Hendrikse, S., Gloster, A., Ramlal, M., Ngema, S., & Rowe, M. (2012). A redundant resource: A pre-planned casualty clearing station for a FIFA 2010 stadium in Durban. *Prehospital and Disaster Medicine,* 27, e1–e4.

Herold, D. M., Breitbarth, T., Schulenkorf, N., & Kummer, S. (2020a). Sport logistics research: Reviewing and line marking of a new field. *The International Journal of Logistics Management,* 31(2), 357–379.

Herold, D. M., Schulenkorf, N., Breitbarth, T., & Bongiovanni, I. (2020b). An application of the sports logistics framework: The case of the Dallas Cowboys. *Journal of Convention & Event Tourism.*

Huizing, D., Schäfer, G., Van der Mei, R. D., & Bhulai, S. (2020). The median routing problem for simultaneous planning of emergency response and non-emergency jobs. *European Journal of Operational Research,* 283(1), 224–238.

Jánošíková, Ľ., Jankovič, P., Kvet, M., & Zajacová, F. (2021). Coverage versus response time objectives in ambulance location. *International Journal of Health Geographics,* 20, 32.

Jia, H., Ordóñez, F., & Dessouky, M. (2005). A modeling framework for facility location of medical services for large-scale emergencies. *Working Paper,* Daniel J. Epstein Department of Industrial and Systems Engineering, University of Southern California.

Karatas, M., & Yakici, E. (2018). An analysis of p-median location problem: Effects of backup service level and demand assignment policy. *European Journal of Operational Research,* 272(1), 207–218.

Kauppi, K., Moxham, C., & Bamford, D. (2013). Should we try out for the major leagues? A call for research in sport operations management. *International Journal of Operations & Production Management,* 33(10), 1368–1399.

Kim, J. H., Malhotra, R., Chiampas, G., D'Hemecourt, P., Troyanos, C., Cianca, J., Smith, R. N., Wang, T. J., Roberts, W. O., Thompson, P. D., & Baggish, A. L. (2012). Cardiac arrest during long-distance running races. *The New England Journal of Medicine,* 366(2), 130–140.

Kristiansen, E., Strittmatter, A., & Skirstad, B. (2016). Stakeholders, challenges and issues at a co-hosted youth Olympic event: Lessons learned from the European Youth Olympic Festival in 2015. *The International Journal of the History of Sport,* 33(10), 1152–1168.

Kwon, Y. S., Lee, B. K., & Sohn, S. Y. (2019). Optimal location-allocation model for the installation of rooftop sports facilities in metropolitan areas. *European Sport Management Quarterly,* 20(2), 189–204.

Li, H., Yu, D., Zhang, Y., & Yuan, Y. (2025). A two-stage robust optimization model for emergency service facilities location-allocation problem under demand uncertainty and sustainable development. *Scientific Reports,* 15(1).

Liang, H., Wang, S., Li, H., Zhou, L., Chen, H., Zhang, X., & Chen, X. (2024). Sponet: Solve spatial optimization problem using deep reinforcement learning for urban spatial decision analysis. *International Journal of Digital Earth,* 17(1), 2299211.

Liang, H., Wang, S., Li, H., Pan, J., Li, X., Su, C., & Liu, B. (2025). AIAM: Adaptive interactive attention model for solving p-median problem via deep reinforcement learning. *International Journal of Applied Earth Observation and Geoinformation,* 138, 104454.

Lim, N., Choi, W., & Pedersen, P. M. (2019). Analyzing the relationship between stadium location and attendance: A location modeling of Major League Baseball (MLB). International Journal of Applied Sports Sciences, 31(1), 70–85.

McCarthy, D. M., Chiampas, G. T., Malik, S., Cole, K., Lindeman, P., & Adams, J. G. (2011). Enhancing community disaster resilience through mass sporting events. *Disaster Medicine and Public Health Preparedness,* 5(4), 310–315.

McKinsey & Company. (2024). Sporting goods 2024: Time to move. McKinsey & Company.

Minis, I., Angelopoulos, J., & Kyrioglou, G. (2009). Car fleet planning and management models for large event transport: The Athens 2004 Olympic Games. *Transportation Planning and Technology,* 32(2), 135–161.

Minis, I., Paraschi, M., & Tzimourtas, A. (2006). The design of logistics operations for the Olympic Games. *International Journal of Physical Distribution & Logistics Management,* 36(8), 621–642.

Montenegro, J. A., & Muñoz, A. (2024). EventGeoScout: Fostering citizen empowerment and augmenting data quality through collaborative geographic information governance and optimization. *ISPRS International Journal of Geo-Information,* 13(2), 46.

Mosadeghi, R., Barr, D., & Moller, R. (2020). The use of GIS in major sport events management: The host city's lessons learned from Gold Coast 2018, Commonwealth Games. *Applied Spatial Analysis and Policy.*

Mountjoy, M., Akef, N., Budgett, R., Greinig, S., Li, G., Manikavasagam, J., Soligard, T., Haiming, X., & Yang, X. (2015). A novel antidoping and medical care delivery model at the 2nd Summer Youth Olympic Games (2014), Nanjing, China. *British Journal of Sports Medicine,* 49(13), 887–892.

Mountjoy, M., Moran, J., Ahmed, H., Bermon, S., Bigard, X., Doerr, D., Lacoste, A., Miller, S., Weber, A., Foster, J., Budgett, R., Engebretsen, L., Burke, L. M., Gouttebarge, V., Grant, M., McCloskey, B., Piccininni, P., Racinais, S., Stuart, M., & Zideman, D. (2020). Athlete health and safety at large sporting events: The development of consensus-driven guidelines. *British Journal of Sports Medicine,* 55(4), 191–197.

Nguyen, R. B., Milsten, A. M., & Cushman, J. T. (2008). Injury patterns and levels of care at a marathon. *Prehospital and Disaster Medicine*, 23(5), 519–525.

Nie, Y., Zhang, H. M., & Lee, D. H. (2004). Models and algorithms for the traffic assignment problem with link capacity constraints. *Transportation Research Part B: Methodological, 38*(4), 285–312.

Nilson, F., Lindberg, F., Palm, G., Lundgren, L., Rayner, D., Börjesson, M., Thorsson, S., Khorram-Manesh, A., & Carlström, E. (2018). Can participants predict where ambulance-requiring cases occur at a half marathon? *Scandinavian Journal of Medicine & Science in Sports*, 28(12), 2760–2766.

Pott, C., Breuer, C., & Ten Hompel, M. (2023). Sport logistics: Considerations on the nexus of logistics and sport management and its unique features. *Logistics*, 7(3), 57.

Poh, J., & Anantharaman, V. (2025). Casualties during marathon events and implications for medical support. *Healthcare*, 13(17), 2249.

ReVelle, C. S., & Swain, R. W. (1970). Central facilities location. *Geographical Analysis*, 2(1), 30–42.

Roberts, O. (2000). A 12-year profile of medical injury and illness for the Twin Cities Marathon. *Medicine & Science in Sports & Exercise*, 32(9), 1549–1555.

Ruan, J., Liu, B., Wei, H., Qu, Y., Zhu, N., & Zhou, X. (2016). How many and where to locate parking lots? A space–time accessibility-maximization modeling framework for special event traffic management. *Urban Rail Transit*, 2(2), 59–70.

Shen, L., Lu, J., Deng, L., & Li, M. (2021). Emergency resource location and allocation in traffic contingency plan for sports mega-event. *Advances in Civil Engineering*, 2021(1).

Shi, M., Zhang, S., & Wu, W. (2013). The study on configuration of public culture and sports facilities for village and town. *Department of Construction and Real Estate, Harbin Institute of Technology.*

Simons, A. (2023). *Predicting water stations at the Enschede Marathon* [Master's thesis, University of Twente].

Snyder, L. V. (2006). Facility location under uncertainty: A review. *IIE Transactions*, 38, 537–554.

Sugawara, M., Manabe, Y., Yamasawa, F., & Hosokawa, Y. (2022). Athlete medical services at the marathon and race walking events during Tokyo 2020 Olympics. *Frontiers in Sports and Active Living*, 4.

Takashima, K. (2023). The 2024 problem in logistics: Efforts to avoid a logistics crisis. *Mitsui & Co. Global Strategic Studies Institute Monthly Report*, 1–2.

Tan, C. M., Tan, I. W., Kok, W. L., Lee, M. C., & Lee, V. J. (2014). Medical planning for mass-participation running events: A three-year review of a half marathon in Singapore. *BMC Public Health*, 14(1).

Time and Date AS. (2025). *Historic weather in Vienna*. Retrieved June 28, 2025, from https://www.timeanddate.com/weather/austria/vienna/historic

Van Remoortel, H., Scheers, H., De Buck, E., Haenen, W., & Vandekerckhove, P. (2020). Prediction modelling studies for medical usage rates in mass gatherings: A systematic review. *PLoS ONE*, 15(6), e0234977.

Vasquez, M. S., Fong, M. K., Patel, L. J., Kurose, B., Tierney, J., Gardner, I., Yazdani-Arazi, A., & Su, J. K. (2015). Medical planning for very large events: Special Olympics World Games Los Angeles 2015. *Current Sports Medicine Reports*, 14(4), 310–315.

Vienna City Marathon. (n.d.). *Media / Geschichte & Statistik. Vienna City Marathon*. Retrieved October 17, 2025, from https://www.vienna-marathon.com/?go=media

VCM Group. (2025, April 6). Erste positive Bilanz für den 42. Vienna City Marathon. *APA-OTS*. Retrieved October 17, 2025, from https://www.ots.at/presseaussendung/OTS_20250406_OTS0028/erste-positive-bilanz-fuer-den-42-vienna-city-marathon

Webner, D., Duprey, K. M., Drezner, J. A., Cronholm, P., & Roberts, W. O. (2012). Sudden cardiac arrest and death in United States marathons. *Medicine & Science in Sports & Exercise*, 44(10), 1843–1845.

Witkowski, K., Kosa, J., & Piepiora, P. (2016). Strategy of managing a project of organising a mass sports event. *Studies in Sport Humanities*, 53, 23–35.

Wu, J. (2021). Method for forecasting urban national sports and fitness demand based on ant colony algorithm. *Computational Intelligence and Neuroscience*, 2021(1).

Yao, K. V., Troyanos, C., D'Hemecourt, P., & Roberts, W. O. (2017). Optimizing marathon race safety using an incident command post strategy. *Current Sports Medicine Reports*, 16(3), 144–149.

Appendix

ID	Latitude	Longitude
0	48,231218	16,414954
1	48,225714	16,404998
2	48,220157	16,394639
3	48,215018	16,396295
4	48,209199	16,407418
5	48,203221	16,415772
6	48,199299	16,409629
7	48,203858	16,399363
8	48,212143	16,394865
9	48,212036	16,383061
10	48,204102	16,377300
11	48,201407	16,367335
12	48,196675	16,356434
13	48,189675	16,349615
14	48,187328	16,336829
15	48,184392	16,325912
16	48,188504	16,314344
17	48,190916	16,325713
18	48,194947	16,337012
19	48,198092	16,349517
20	48,202458	16,361318
21	48,210985	16,355634
22	48,215242	16,363369
23	48,222984	16,358128
24	48,227474	16,366449
25	48,219383	16,370907
26	48,212546	16,378994
27	48,216971	16,388837
28	48,21313	16,399643
29	48,208452	16,408505
30	48,208405	16,418706
31	48,204242	16,416693
32	48,198422	16,427818
33	48,193315	16,437607
34	48,197118	16,430112
35	48,203839	16,417225
36	48,201248	16,410510
37	48,207561	16,399582
38	48,214275	16,393402
39	48,211263	16,385508
40	48,204748	16,378083
41	48,202882	16,367383
42	48,207661	16,359819
43	48,210998	16,360825

Official 44 coordinates for Variant 1

ID	Latitude	Longitude						
0	48,231158	16,414969	46	48,2058	16,413857	93	48,20975	16,382462
			47	48,205226	16,414892	94	48,20887	16,382205
1	48,230593	16,413922	48	48,204651	16,415927	95	48,208129	16,381443
2	48,230028	16,412875	49	48,20398	16,416059	96	48,207387	16,380682
3	48,229463	16,411828	50	48,203109	16,415767	97	48,206646	16,379921
4	48,228898	16,41078	51	48,202211	16,415685	98	48,205904	16,37916
5	48,228334	16,409732	52	48,201319	16,415538	99	48,205163	16,378399
6	48,227769	16,408684	53	48,200469	16,415107	100	48,204421	16,377638
7	48,227205	16,407636	54	48,199652	16,414548	101	48,20368	16,376876
8	48,226641	16,406588	55	48,198868	16,413892	102	48,202938	16,376115
9	48,226076	16,405541	56	48,198139	16,413105	103	48,202197	16,375354
10	48,225506	16,4045	57	48,197472	16,412204	104	48,201684	16,374329
11	48,224937	16,403458	58	48,197139	16,411258	105	48,201602	16,373043
12	48,224367	16,402417	59	48,197941	16,410649	106	48,201889	16,371769
13	48,223797	16,401375	60	48,198744	16,410045	107	48,202135	16,37048
14	48,223228	16,400334	61	48,19955	16,409449	108	48,202424	16,369207
15	48,222658	16,399293	62	48,200352	16,408838	109	48,202642	16,36797
16	48,222088	16,398251	63	48,201088	16,408084	110	48,201791	16,367536
17	48,221519	16,39721	64	48,2017	16,407103	111	48,200939	16,367101
18	48,220949	16,396169	65	48,202166	16,405957	112	48,200336	16,366108
19	48,220379	16,395127	66	48,202519	16,40472	113	48,199738	16,365104
20	48,21981	16,394086	67	48,20277	16,403428	114	48,199165	16,364066
21	48,219416	16,392957	68	48,202994	16,402125	115	48,198605	16,363015
22	48,219252	16,391682	69	48,203219	16,400823	116	48,198079	16,361924
23	48,218651	16,39073	70	48,20365	16,399661	117	48,197618	16,360775
24	48,217828	16,390505	71	48,204338	16,398802	118	48,197354	16,359489
25	48,217217	16,391452	72	48,205152	16,398247	119	48,19709	16,358203
26	48,216907	16,392691	73	48,206032	16,398013	120	48,19681	16,356926
27	48,216352	16,393751	74	48,206928	16,397899	121	48,196421	16,355716
28	48,215798	16,394811	75	48,207819	16,39773	122	48,195817	16,354723
29	48,215244	16,39587	76	48,208687	16,397381	123	48,195021	16,354128
30	48,21469	16,39693	77	48,209533	16,396927	124	48,194132	16,354013
31	48,214136	16,39799	78	48,210358	16,396391	125	48,193235	16,354088
32	48,213581	16,399049	79	48,211173	16,395822	126	48,192346	16,353916
33	48,213027	16,400109	80	48,21193	16,395099	127	48,191526	16,353369
34	48,212473	16,401169	81	48,212633	16,39427	128	48,190773	16,35264
35	48,211919	16,402229	82	48,213152	16,393176	129	48,190156	16,351678
36	48,211365	16,403288	83	48,213513	16,391956	130	48,189826	16,350434
37	48,21081	16,404348	84	48,213759	16,390664	131	48,189591	16,349136
38	48,210256	16,405408	85	48,213948	16,38935	132	48,189367	16,347833
39	48,209702	16,406467	86	48,213837	16,388022	133	48,189148	16,346529
40	48,209148	16,407527	87	48,213429	16,386834	134	48,188968	16,345213
41	48,208594	16,408586	88	48,212969	16,385678	135	48,188874	16,343876
42	48,20804	16,409646	89	48,212649	16,384424	136	48,188793	16,342536
43	48,207485	16,410706	90	48,212406	16,383175	137	48,188717	16,341196
44	48,206931	16,411765	91	48,211521	16,382937	138	48,188612	16,33986
45	48,206375	16,412822	92	48,210635	16,3827	139	48,188408	16,33856

140	48,187854	16,337516	187	48,196843	16,344722	234	48,224758	16,357101
141	48,187159	16,336673	188	48,197124	16,346	235	48,224885	16,358224
142	48,186369	16,33603	189	48,197404	16,347278	236	48,225089	16,35953
143	48,185644	16,335244	190	48,197764	16,348509	237	48,225363	16,360812
144	48,184973	16,33435	191	48,198149	16,349725	238	48,225729	16,362033
145	48,18447	16,333246	192	48,198611	16,350878	239	48,226201	16,363178
146	48,184128	16,332008	193	48,199096	16,352011	240	48,226691	16,364306
147	48,184021	16,330676	194	48,199509	16,353204	241	48,22718	16,365436
148	48,184009	16,329332	195	48,199881	16,354428	242	48,227436	16,366519
149	48,184018	16,327988	196	48,200253	16,355653	243	48,226754	16,367395
150	48,184174	16,326675	197	48,200575	16,356908	244	48,226004	16,368122
151	48,184537	16,325445	198	48,200864	16,358182	245	48,225196	16,368714
152	48,184909	16,32422	199	48,20117	16,359445	246	48,224331	16,369069
153	48,18528	16,322995	200	48,201737	16,360471	247	48,223475	16,369465
154	48,185651	16,321771	201	48,202444	16,361299	248	48,222581	16,369605
155	48,185998	16,32053	202	48,203112	16,360585	249	48,221685	16,369673
156	48,186361	16,3193	203	48,20378	16,359684	250	48,22079	16,369539
157	48,186731	16,318074	204	48,204448	16,358784	251	48,220002	16,370073
158	48,187102	16,316849	205	48,205123	16,357895	252	48,219342	16,370983
159	48,187472	16,315623	206	48,2058	16,357008	253	48,218735	16,371975
160	48,187843	16,314398	207	48,206476	16,356121	254	48,218075	16,372873
161	48,188703	16,314453	208	48,207279	16,355585	255	48,217271	16,373472
162	48,189547	16,314917	209	48,208151	16,35526	256	48,216524	16,374218
163	48,19036	16,315412	210	48,209035	16,355222	257	48,215747	16,374898
164	48,190309	16,316754	211	48,209925	16,355413	258	48,214946	16,375509
165	48,190209	16,318091	212	48,210815	16,355604	259	48,214092	16,375816
166	48,190127	16,31943	213	48,211705	16,355794	260	48,213357	16,376491
167	48,19018	16,320771	214	48,212596	16,355985	261	48,212881	16,377631
168	48,190279	16,322107	215	48,213486	16,356176	262	48,212561	16,378884
169	48,190393	16,323439	216	48,214376	16,356367	263	48,212399	16,380206
170	48,190661	16,32472	217	48,214335	16,357549	264	48,212337	16,381547
171	48,190981	16,325976	218	48,214163	16,358869	265	48,212413	16,382884
172	48,191266	16,327249	219	48,214152	16,360191	266	48,213176	16,382999
173	48,191363	16,328584	220	48,214371	16,361414	267	48,21396	16,383281
174	48,191462	16,32992	221	48,214864	16,362539	268	48,214551	16,384295
175	48,191553	16,331258	222	48,215368	16,363654	269	48,215129	16,385326
176	48,191965	16,332387	223	48,215995	16,363739	270	48,215701	16,386365
177	48,192664	16,333233	224	48,216726	16,362956	271	48,216251	16,387429
178	48,193352	16,334097	225	48,21747	16,362205	272	48,2168	16,388495
179	48,193976	16,335066	226	48,218281	16,361623	273	48,217356	16,389552
180	48,194558	16,336084	227	48,219092	16,361041	274	48,217553	16,39063
181	48,195035	16,337224	228	48,219947	16,36063	275	48,217031	16,391717
182	48,195448	16,338418	229	48,22079	16,360185	276	48,216649	16,392904
183	48,195617	16,339705	230	48,221551	16,35947	277	48,216095	16,393965
184	48,196073	16,340855	231	48,222313	16,358755	278	48,215541	16,395025
185	48,196348	16,342135	232	48,223075	16,358039	279	48,214988	16,396085
186	48,196597	16,343428	233	48,223911	16,357554	280	48,214434	16,397145

281	48,21388	16,398205	328	48,194537	16,435291	375	48,209914	16,398777
282	48,213326	16,399265	329	48,193979	16,436347	376	48,21081	16,398801
283	48,212769	16,400322	330	48,193421	16,437401	377	48,211688	16,398788
284	48,212213	16,401379	331	48,192864	16,438456	378	48,212431	16,39803
285	48,211656	16,402436	332	48,192422	16,439294	379	48,213163	16,397248
286	48,2111	16,403493	333	48,192753	16,438438	380	48,213924	16,396534
287	48,210543	16,40455	334	48,193315	16,437389	381	48,214604	16,395725
288	48,209987	16,405607	335	48,193865	16,436325	382	48,214437	16,394421
289	48,20943	16,406664	336	48,194417	16,435263	383	48,214227	16,393112
290	48,208874	16,407721	337	48,194975	16,434208	384	48,214024	16,391801
291	48,208315	16,408775	338	48,195533	16,433152	385	48,213354	16,39129
292	48,207756	16,409829	339	48,196087	16,432093	386	48,212483	16,390956
293	48,207197	16,410883	340	48,196641	16,431033	387	48,211612	16,390623
294	48,206638	16,411937	341	48,197194	16,429973	388	48,210972	16,389981
295	48,206079	16,412991	342	48,197748	16,428913	389	48,211055	16,388643
296	48,20552	16,414045	343	48,198301	16,427853	390	48,211148	16,387305
297	48,204961	16,415098	344	48,198855	16,426793	391	48,211235	16,385966
298	48,204402	16,416152	345	48,199408	16,425732	392	48,210861	16,385136
299	48,204976	16,416981	346	48,199962	16,424672	393	48,20998	16,384869
300	48,20575	16,417666	347	48,200515	16,423612	394	48,209101	16,384583
301	48,20654	16,418305	348	48,201069	16,422551	395	48,208262	16,384112
302	48,207419	16,418414	349	48,201622	16,421491	396	48,207462	16,383497
303	48,208278	16,418575	350	48,202176	16,420431	397	48,206662	16,382883
304	48,209017	16,41934	351	48,202729	16,41937	398	48,20689	16,381708
305	48,208628	16,419081	352	48,203283	16,41831	399	48,207082	16,380474
306	48,207831	16,418494	353	48,203836	16,417249	400	48,206341	16,379713
307	48,206938	16,418623	354	48,204098	16,416246	401	48,205599	16,378953
308	48,206103	16,418156	355	48,203245	16,415845	402	48,204857	16,378192
309	48,205318	16,417499	356	48,202347	16,415778	403	48,204115	16,377431
310	48,20453	16,416851	357	48,201667	16,415547	404	48,203374	16,37667
311	48,203946	16,417267	358	48,202156	16,414473	405	48,202632	16,375909
312	48,203392	16,418327	359	48,201735	16,413357	406	48,20189	16,375149
313	48,202837	16,419386	360	48,201244	16,412253	407	48,20154	16,373936
314	48,202283	16,420445	361	48,201201	16,410919	408	48,201647	16,372639
315	48,201728	16,421504	362	48,201528	16,409687	409	48,201931	16,371363
316	48,201172	16,422561	363	48,202132	16,408693	410	48,202258	16,370115
317	48,200616	16,423618	364	48,202699	16,40765	411	48,202564	16,36885
318	48,200061	16,424677	365	48,203165	16,406499	412	48,20284	16,367569
319	48,19951	16,42574	366	48,203574	16,405302	413	48,203116	16,366289
320	48,198956	16,4268	367	48,203998	16,404117	414	48,203392	16,365009
321	48,198401	16,427858	368	48,20447	16,402973	415	48,20385	16,363906
322	48,197845	16,428916	369	48,205061	16,40196	416	48,204523	16,363013
323	48,19729	16,429974	370	48,205746	16,401088	417	48,205195	16,36212
324	48,196734	16,431032	371	48,206506	16,400372	418	48,205868	16,361227
325	48,196179	16,432089	372	48,207308	16,399767	419	48,206541	16,360334
326	48,19563	16,433155	373	48,208153	16,399309	420	48,207278	16,359709
327	48,195083	16,434223	374	48,209026	16,398986	421	48,208159	16,359983
422	48,209039	16,360257						
423	48,20992	16,360531						
424	48,2108	16,360805						

425 coordinates for Variant 2

Station	Latitude	Longitude
1	48,2048238	16,4183382
2	48,1862881	16,3203174
3	48,2115602	16,3611895
4	48,2278467	16,3666916

Official coordinates for real world emergency stations

```
class RoutePoint:
    def __init__(self, id, latitude, longitude, weather=0.0, participants=0, injury_risk=0.0):
        self.id = id                          # E.g. kilometer 1-42
        self.latitude = latitude              # Width
        self.longitude = longitude            # Length
        self.weather = weather                # Will be simulated later
        self.participants = participants      # Will be simulated later
        self.injury_risk = injury_risk        # From lookup table
        self.weight = 0.0                     # Computed later

    def __repr__(self):
        return (f"RoutePoint(ID={self.id}, Lat={self.latitude}, Lon={self.longitude}, "
                f"W={self.weather}, T={self.participants}, V={self.injury_risk}, "
                f"w_i={self.weight})")
```

Phyton RoutePoint File

```
from gurobipy import Model, GRB

def gurobi_model(route, distance_matrix, p):
    n = len(route)  # Number of points

    model = Model("Station_Optimization")
    model.setParam('OutputFlag', 0) #no output from Gurobi

    # desicion variables
    x = {}
    y = {}

    for i in range(n):
        for j in range(n):
            x[i, j] = model.addVar(vtype=GRB.BINARY, name=f"x_{i}_{j}") # constraint 4: binary variables
    for j in range(n):
        y[j] = model.addVar(vtype=GRB.BINARY, name=f"y_{j}")

    model.update()

    # objective function
    objective = sum(route[i].weight * distance_matrix[i][j] * x[i, j]
                    for i in range(n) for j in range(n))
    model.setObjective(objective, GRB.MINIMIZE)

    # constraint 1: Every point is assigned to exactly one station
    for i in range(n):
        model.addConstr(sum(x[i, j] for j in range(n)) == 1, name=f"assignment_{i}")

    # constraint 2: Only assigned to opend stations
    for i in range(n):
        for j in range(n):
            model.addConstr(x[i, j] <= y[j], name=f"assignment_{i}_{j}")

    # constraint 3: exactly p station open
    model.addConstr(sum(y[j] for j in range(n)) == p, name="number_of_stations")

    # return model + variables for later constraints
    return model, x, y
```

Phyton Gurobi_Solver File

```python
import random
class Simulation:
    def __init__(self):
        self.rnd = random.Random()

    def simulate_weather(self):
        # historical data: temperature from -1°C to 28°C
        return -1 + self.rnd.random() * (28 - (-1))

    def simulate_participants(self):
        # historical data: participants from 26,016 to 46,083
        return 26016 + self.rnd.random() * (46083 - 26016)

    # no simulation needed for injury probability and defined per point position

    def scale_weather(self, weather):
        # Normalize temperature from range [-1, 28]
        ri_norm = (weather - (-1)) / (28 - (-1))
        return ri_norm

    def scale_participants(self, participants):
        # Normalize participants from range [26016, 46083]
        ti_norm = (participants - 26016) / (46083 - 26016)
        return ti_norm

    def scale_injuryrisk(self, kilometer):
        probability_for_km = {
            0: 0.1, 1: 0.5, 2: 0.5, 3: 1, 4: 1, 5: 1, 6: 1, 7: 1,
            8: 1.5, 9: 2, 10: 2, 11: 2.5, 12: 3, 13: 3, 14: 4, 15: 5,
            16: 5, 17: 6, 18: 7, 19: 8, 20: 10, 21: 12, 22: 14,
            23: 16, 24: 18, 25: 20, 26: 22, 27: 25, 28: 28, 29: 30,
            30: 35, 31: 40, 32: 45, 33: 50, 34: 55, 35: 60, 36: 65,
            37: 70, 38: 75, 39: 80, 40: 85, 41: 90, 42: 95, 43: 95
    }
        max_value = 95.0
        probability = probability_for_km.get(kilometer, 0.0)
        vi_norm = probability / max_value
        return vi_norm
```
Phyton Simulation file

```python
# Extract all track points
gpx_points = []
for track in gpx.tracks:
    for segment in track.segments:
        for point in segment.points:
            gpx_points.append((point.latitude, point.longitude))

# stepwise in meters
step_size = 100.0

# Starting point: first GPX-point
interpolated_points = [gpx_points[0]]
current_position = gpx_points[0]
distance_accumulated = 0.0

# Loop over all points of the route
for i in range(1, len(gpx_points)):
    next_position = gpx_points[i]
    segment_distance = geopy.distance.geodesic(current_position, next_position).meters

    while distance_accumulated + segment_distance >= step_size:
        ratio = (step_size - distance_accumulated) / segment_distance
        lat = current_position[0] + (next_position[0] - current_position[0]) * ratio
        lon = current_position[1] + (next_position[1] - current_position[1]) * ratio
        interpolated_points.append((lat, lon))
        distance_accumulated = 0.0
        current_position = (lat, lon)
        segment_distance = geopy.distance.geodesic(current_position, next_position).meters
    else:
        distance_accumulated += segment_distance
        current_position = next_position

# save as text file
with open("interpolated_route_100m.txt", "w", encoding="utf-8") as f:
    for idx, (lat, lon) in enumerate(interpolated_points):
        f.write(f"{idx}\t{lat:.7f}\t{lon:.7f}\n")
```

Detail of Phyton file Coordinates_425_datapoints

```python
import time
start_time = time.time()
import csv
from Simulation import Simulation
from RoutePoint import RoutePoint

def load_route_from_csv(filename):
    route = []
    with open(filename, newline='', encoding='utf-8-sig') as csvfile:
        reader = csv.DictReader(csvfile, delimiter=';')
        for row in reader:
            route.append(RoutePoint(
                id=int(row['ID']),
                latitude=float(row['Latitude'].replace(',', '.')),
                longitude=float(row['Longitude'].replace(',', '.'))
            ))
    return route
```

Reading CSV file in Variant1_1000m

```python
import math
def compute_distance(p1, p2):
    """ Computes distance between two points (p1, p2) in meters with the Haversine formula."""
    R = 6371000  # mean radius of the Earth in meters

    lat1 = math.radians(p1.latitude)
    lat2 = math.radians(p2.latitude)
    lon1 = math.radians(p1.longitude)
    lon2 = math.radians(p2.longitude)

    dlat = lat2 - lat1
    dlon = lon2 - lon1

    a = math.sin(dlat / 2)**2 + math.cos(lat1) * math.cos(lat2) * math.sin(dlon / 2)**2
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))

    return R * c  # distance in meters
```

Computing distances with Haversine formula in Variant1_1000m

```python
import numpy as np
def create_distancematrix(route):
    """Creates a distance matrix from the list of route points."""
    n = len(route)
    matrix = np.zeros((n, n))  # Initialize matrix with zeros

    for i in range(n):
        for j in range(i, n):
            if i == j:
                matrix[i, j] = 0 # Distance to self is zero
            else:
                dist = compute_distance(route[i], route[j])
                matrix[i, j] = dist
                matrix[j, i] = dist
    return matrix
```

Creating distance matrix for each pair of route points in Variant1_1000m

```python
for scenario in range(100):  # Run 100 scenarios
    route = [RoutePoint(p.id, p.latitude, p.longitude) for p in route]
    sim = Simulation()

    weather = sim.simulate_weather()
    participants = sim.simulate_participants()

    r_norm = sim.scale_weather(weather)
    t_norm = sim.scale_participants(participants)

    # scenario s & average risk for all points
    P_mean = sum(sim.scale_injuryrisk(p.id) for p in route) / len(route)
    s = 0.5 * (r_norm + t_norm)

    for point in route:
        km = point.id

        epsilon = 0.1  # Using rate of injury risk in extrem case
        v_norm = sim.scale_injuryrisk(km)
        weight = (1 - s) * (epsilon * v_norm + (1 - epsilon) * P_mean) + s * v_norm

        point.weather = weather
        point.participants = int(participants)
        point.injury_risk = weight   # one factor(combined)
        point.weight = weight
```

Scenario loop of Variant 1_1000m

```python
    distance_matrix = create_distancematrix(route)
    model, x, y = gurobi_model(route, distance_matrix, number_of_stations)
    model.optimize()

    if model.status == GRB.OPTIMAL:
        print(f"Scenario {scenario + 1}: Objective function value = {model.ObjVal:.2f}")
        all_solutions.append(model.ObjVal)

        for j in range(len(route)):
            if y[j].X > 0.5:
                station_counter[j] += 1
    else:
        print(f"Scenario {scenario + 1}: No optimal solution found.")
```

Call of Gurobi_solver within the scenario loop

```python
# Summary statistics
average = np.mean(all_solutions)
print(f"\nAverage objective value over 100 scenarios with 4 stations: {average:.2f}")

sorted_stations = sorted(station_counter.items(), key=lambda x: x[1], reverse=True)
print("\nMost frequently selected stations for 4 stations:")
for idx, count in sorted_stations:
    print(f"Kilometer {route[idx].id}: selected {count} times")

end_time = time.time()
duration = end_time - start_time
print(f"\nTotal runtime: {duration:.2f} seconds")
```

Counting and printing the final results of variant1_1000m

```python
for scenario in range(100):  # Run 100 scenarios
    route = [RoutePoint(p.id, p.latitude, p.longitude) for p in route]
    sim = Simulation()

    weather = sim.simulate_weather()
    participants = sim.simulate_participants()

    r_norm = sim.scale_weather(weather)
    t_norm = sim.scale_participants(participants)

    # kilometer index (0-43) instead of raw 100m ID for injury risk
    P_mean = sum(sim.scale_injuryrisk(min(43, int(p.id // 10))) for p in route) / len(route)
    s = 0.5 * (r_norm + t_norm)

    for point in route: # Convert 100m step ID to kilometer index (0-43) for injury risk lookup
        km_idx = min(43, int(point.id // 10))  # 100 m → km-Index

        epsilon = 0.1 # Using rate of injury risk in extrem case
        v_norm = sim.scale_injuryrisk(km_idx)
        weight = (1 - s) * (epsilon * v_norm + (1 - epsilon) * P_mean) + s * v_norm

        point.weather = weather
        point.participants = int(participants)
        point.injury_risk = weight
        point.weight = weight
```

Scenario loop for variant2_100m

```python
# Summary statistics
average = np.mean(all_solutions)
print(f"\nAverage objective value over 100 scenarios with 4 stations: {average:.2f}")

sorted_stations = sorted(station_counter.items(), key=lambda x: x[1], reverse=True)
print("\nMost frequently selected stations for 4 stations:")
for idx, count in sorted_stations:
    km_position = route[idx].id / 10  # 100m-steps → /10 = km
    print(f"Kilometer {km_position:.1f}: selected {count} times")

end_time = time.time()
duration = end_time - start_time
print(f"\nTotal runtime: {duration:.2f} seconds")
```

Final print block of Variant2_100m

```python
def compute_weights(route, sim: Simulation, epsilon: float):
    # 1) Simulate scenario factors
    weather = sim.simulate_weather()
    participants = sim.simulate_participants()

    # 2) Normalize
    r_norm = sim.scale_weather(weather)
    t_norm = sim.scale_participants(participants)
    s = 0.5 * (r_norm + t_norm)

    # 3) Injury risk (P_mean over all points)
    km_idx_of = lambda pid: min(43, int(pid // 10))    # 100m-ID → km-Index
    P_mean = np.mean([sim.scale_injuryrisk(km_idx_of(p.id)) for p in route])

    # 4) weights
    weights = np.zeros(len(route), dtype=float)
    for idx, point in enumerate(route):
        v_norm = sim.scale_injuryrisk(km_idx_of(point.id))
        w = (1 - s) * (epsilon * v_norm + (1 - epsilon) * P_mean) + s * v_norm
        weights[idx] = w

        point.weather = weather
        point.participants = int(participants)
        point.injury_risk = v_norm
        point.weight = w

    return weights
```

Preparation for genetic algorithm in Variant3_Metaheuristic

```python
import random
def init_population(n: int, p: int, rnd: random.Random, pop_size: int = 100): #number of solutions
    population = []
    for _ in range(pop_size):
        individual = tuple(sorted(rnd.sample(range(n), p)))
        population.append(individual)
    return population

def fitness(weights: np.ndarray, distM: np.ndarray, individual: tuple[int, ...]) -> float:
    stations = np.fromiter(individual, dtype=int)
    nearest = distM[:, stations].min(axis=1)      # distance to nearest chosen station
    return float(np.dot(weights, nearest))        # weighted sum of nearest distances
```

Population initialization and fitness function of variant3_Metaheuristic

```python
def select_parents(population, weights, distM, elite_frac=0.2): #selects best 20% for parents
    # Fitness for all individuals
    scored = [(ind, fitness(weights, distM, ind)) for ind in population]
    scored.sort(key=lambda x: x[1])  # sort ascending (minimization)

    # choosing top-Fraction
    elite_count = max(1, int(len(population) * elite_frac))
    parents = [ind for ind, _ in scored[:elite_count]]

    # return best solution
    best_individual, best_value = scored[0]
    return parents, (best_individual, best_value)
```

Parent selection of variant3_Metaheuristic

```python
def crossover(parent1: tuple[int, ...], parent2: tuple[int, ...],
                n: int, p: int, rnd: random.Random) -> tuple[int, ...]:
    # 1) Pool of genes of both parents
    pool = list(set(parent1).union(parent2))

    # 2) Repair: if pool is smaller than p, fill with random unused Indices
    if len(pool) < p:
        missing = p - len(pool)
        unused = [i for i in range(n) if i not in pool]
        pool += rnd.sample(unused, missing)

    # 3) child
    child = tuple(sorted(rnd.sample(pool, p)))
    return child
```

Crossover of genetic algorithm in variant3_Metaheuristic

```python
def mutate(individual: tuple[int, ...], n: int, p: int,
                rate: float, rnd: random.Random) -> tuple[int, ...]:
    if rnd.random() < rate:
        pos = rnd.randrange(p)
        used = set(individual)
        available = [i for i in range(n) if i not in used]
        if available:
            new_gene = rnd.choice(available)
            new_list = list(individual)
            new_list[pos] = new_gene
            return tuple(sorted(new_list))
    return individual
```

Mutation of genetic algorithm in variant3_Metaheuristic

```python
def run_ga(weights: np.ndarray, distM: np.ndarray, p: int, rnd: random.Random,
           pop_size: int = 100, elite_frac: float = 0.2, mut_rate: float = 0.1,
           max_gens: int = 1000, patience: int = 100):
    n = distM.shape[0]
    population = init_population(n, p, rnd, pop_size)

    best_ind, best_val = None, float("inf")
    no_improve = 0

    for gen in range(max_gens):
        # Computinf fitenss for all individuals
        scored = [(ind, fitness(weights, distM, ind)) for ind in population]
        scored.sort(key=lambda x: x[1])

        # Bcheck best update
        if scored[0][1] < best_val:
            best_ind, best_val = scored[0]
            no_improve = 0
        else:
            no_improve += 1

        # Stopping at 100 iterations without Improvement
        if no_improve >= patience:
            break

        # parents selection
        elite_count = max(1, int(pop_size * elite_frac))
        parents = [ind for ind, _ in scored[:elite_count]]

        # Building new population
        new_pop = parents.copy()
        while len(new_pop) < pop_size:
            p1, p2 = rnd.choice(parents), rnd.choice(parents)
            child = crossover(p1, p2, n, p, rnd)
            child = mutate(child, n, p, mut_rate, rnd)
            new_pop.append(child)

        population = new_pop

    return best_ind, best_val
```

Termination of genetic algorithm in variant3_Metaheuristic

```python
# Summary statistics
if __name__ == "__main__":
    EPSILON = 0.1
    SCENARIOS = 1 #number of scenarios
    rnd = random.Random(42)

    for scenario in range(SCENARIOS):
        sim = Simulation()
        weights = compute_weights(route, sim, EPSILON)
        best_ind, best_val = run_ga(weights, distance_matrix, number_of_stations, rnd)
        print(f"Scenario {scenario + 1}: GA objective = {best_val:.2f}")
        all_solutions.append(best_val)
        for j in best_ind:
            station_counter[j] += 1

average = np.mean(all_solutions)
print(f"\nAverage GA objective over {SCENARIOS} scenarios with 4 stations: {average:.2f}")

sorted_stations = sorted(station_counter.items(), key=lambda x: x[1], reverse=True)
print("\nMost frequently selected stations for 4 stations:")
for idx, count in sorted_stations:
    km_position = route[idx].id / 10.0  # 100 m → km
    print(f"Kilometer {km_position:.1f}: selected {count} times")

end_time = time.time()
duration = end_time - start_time
print(f"\nTotal runtime: {duration:.2f} seconds")
```

Final print of genetic algorithm

```python
def run_three_extremes_v1(base_route, p: int, epsilon: float = 0.1) -> None:
    TEMP_MIN, TEMP_MAX = -1.0, 28.0
    PART_MIN, PART_MAX = 26016, 46083

    scenarios = [
        ("Cold & less participants", TEMP_MIN, PART_MIN),
        ("Average", (TEMP_MIN + TEMP_MAX) / 2.0, (PART_MIN + PART_MAX) / 2.0),
        ("Hot & many participants", TEMP_MAX, PART_MAX),
    ]

    sim = Simulation()

    for label, temp, parts in scenarios:
        route = [RoutePoint(pt.id, pt.latitude, pt.longitude) for pt in base_route]

        r_norm = sim.scale_weather(temp)
        t_norm = sim.scale_participants(parts)
        s = 0.5 * (r_norm + t_norm)

        risks = [sim.scale_injuryrisk(int(pt.id)) for pt in route]
        P_mean = sum(risks) / len(risks)

        for pt, v_norm in zip(route, risks):
            w = (1 - s) * (epsilon * v_norm + (1 - epsilon) * P_mean) + s * v_norm
            pt.weather = temp
            pt.participants = int(parts)
            pt.injury_risk = v_norm
            pt.weight = w

        D = create_distancematrix(route)
        model, x, y = gurobi_model(route, D, p)
        model.optimize()

        if model.status == GRB.OPTIMAL:
            chosen_km = [route[j].id for j in range(len(route)) if y[j].X > 0.5]
            print(f"[Extremepoint] {label}: Objective function= {model.ObjVal:.2f} | Stations(km): {chosen_km} | T={temp:.1f}°C, P={int(parts)}")
```

Extreme scenarios for Variant1_1000m

```python
def run_three_extremes_v2(base_route, p: int, epsilon: float = 0.1) -> None:
    TEMP_MIN, TEMP_MAX = -1.0, 28.0
    PART_MIN, PART_MAX = 26016, 46083

    scenarios = [
        ("Cold & less participants", TEMP_MIN, PART_MIN),
        ("Average", (TEMP_MIN + TEMP_MAX) / 2.0, (PART_MIN + PART_MAX) / 2.0),
        ("Hot & many participants", TEMP_MAX, PART_MAX),
    ]

    sim = Simulation()

    for label, temp, parts in scenarios:
        route = [RoutePoint(pt.id, pt.latitude, pt.longitude) for pt in base_route]

        r_norm = sim.scale_weather(temp)
        t_norm = sim.scale_participants(parts)
        s = 0.5 * (r_norm + t_norm)

        # kilometer index (0-43) instead of raw 100m ID for injury risk
        risks = [sim.scale_injuryrisk(min(43, int(pt.id // 10))) for pt in route]
        P_mean = sum(risks) / len(risks)

        for pt, v_norm in zip(route, risks):
            w = (1 - s) * (epsilon * v_norm + (1 - epsilon) * P_mean) + s * v_norm
            pt.weather = temp
            pt.participants = int(parts)
            pt.injury_risk = v_norm
            pt.weight = w

        D = create_distancematrix(route)
        model, x, y = gurobi_model(route, D, p)
        model.optimize()

        if model.status == GRB.OPTIMAL:
            chosen_idx = [j for j in range(len(route)) if y[j].X > 0.5]
            # Convert 100m step ID to kilometer index (0-43)
            pos_km = [route[j].id / 10.0 for j in chosen_idx]
            print(f"[Extremepoint] {label}: Objective function= {model.ObjVal:.2f} | "
                  f"Stations (km): {', '.join(f'{km:.1f}' for km in pos_km)} | "
                  f"T={temp:.1f}°C, P={int(parts)}")
```

Extreme scenarios for Variant2_100m

```python
def main():
    print("Select variant (1=1km, 2=100m, 3=100m with Metaheuristic)")
    while True:
        try:
            variant = int(input("Variant [1/2/3]: ").strip())
            if variant in (1, 2, 3):
                break
        except Exception:
            pass
        print("Please enter 1, 2, or 3.")

    # Resolve CSV path for the variant
    csv_path = None
    if variant == 1:
        csv_path = CSV_PATH_V1
    elif variant == 2:
        csv_path = CSV_PATH_V2
    else:
        csv_path = CSV_PATH_V3

    while not csv_path:
        csv_path = input("Path to coordinates CSV for this variant: ").strip()
        if not csv_path:
            print("CSV path is required.")

    # load route and distance matrix (no prints)
    route_base = load_route_from_csv(csv_path)
    distance_matrix = create_distancematrix(route_base)

    start = time.time()
```

User Selection of specific variant for Number_of_Stations

```python
def run_for_p(route_base: List[RoutePoint], distance_matrix, p: int, scenarios: int, variant: int, epsilon: float) -> Tuple[float, List[Tuple[int, int]]]:
    """Returns (avg_obj, [(station_index, count), ...] sorted by count desc)."""
    all_objs = []
    station_counter = defaultdict(int)

    n = len(route_base)
    rnd = random.Random(42)  # reproducible GA for each p

    # For GA path convert distance matrix once
    distM_np = np.array(distance_matrix) if variant == 3 else None

    for _ in range(scenarios):
        # fresh copy per scenario
        route = [RoutePoint(pt.id, pt.latitude, pt.longitude) for pt in route_base]

        # weights for this scenario
        sim = Simulation()
        compute_weights(route, sim, variant=variant, epsilon=epsilon)

        if variant == 3:
            # --- Metaheuristic path ---
            weights_np = np.array([pt.weight for pt in route], dtype=float)
            best_ind, best_val = run_ga(weights_np, distM_np, p, rnd)
            all_objs.append(best_val)
            for j in best_ind:
                station_counter[j] += 1
        else:
            # --- Gurobi path (variants 1 & 2) ---
            model, x, y = gurobi_model(route, distance_matrix, p)
            obj = 0.0
            for i in range(n):
                wi = route[i].weight
                row = distance_matrix[i]
                for j in range(n):
                    obj += wi * row[j] * x[i, j]
            model.setObjective(obj, GRB.MINIMIZE)

            model.optimize()
            if model.status == 2:  # GRB.OPTIMAL
                all_objs.append(model.ObjVal)
                for j in range(n):
                    if y[j].X > 0.5:
                        station_counter[j] += 1

    avg_val = sum(all_objs) / len(all_objs) if all_objs else float('nan')
    most_selected = sorted(station_counter.items(), key=lambda kv: kv[1], reverse=True)
    return avg_val, most_selected
```

Main part of Number_of_Stations

```python
    # p from 4..12
    for p in range(4, 13):
        avg_obj, most_selected = run_for_p(route_base, distance_matrix, p, SCENARIOS, variant, EPSILON)

        # ---- Required output format ----
        print(f"\nAverage Objective Function with {p} Stations: {avg_obj:.2f}")
        print(f"Most Selected Stations for {p} Stations:")
        for idx, cnt in most_selected:
            if variant in (2, 3):
                km = route_base[idx].id / 10.0
                print(f"  Kilometer {km:.1f}: selected {cnt} times")
            else:
                km = int(route_base[idx].id)
                print(f"  Kilometer {km}: selected {cnt} times")

    total = time.time() - start
    print(f"\nTotal Runtime: {total:.2f} seconds")


if __name__ == "__main__":
    main()
```

Final output print of Number_of_Stations

```python
# --- Average objective (100 scenarios) for real case stations ---
def realcase_stations(route, runs=100, epsilon=0.1, seed=42):
    from RoutePoint import RoutePoint
    from Simulation import Simulation
    import random
    stations = [RoutePoint(i, lat, lon) for i,(lat,lon) in enumerate([
        (48.2048238,16.4183382),(48.1862881,16.3203174),
        (48.2115602,16.3611895),(48.2278467,16.3666916),
    ])]

    dmin = [min(compute_distance(p,s) for s in stations) for p in route]
    sim = Simulation(); base = [sim.scale_injuryrisk(int(p.id)) for p in route]
    P_mean = sum(base)/len(base);
    if seed is not None: random.seed(seed)
    total = 0.0
    for _ in range(runs):
        s = 0.5*(sim.scale_weather(sim.simulate_weather())
                 + sim.scale_participants(sim.simulate_participants()))
        total += sum(((1-s)*(epsilon*v + (1-epsilon)*P_mean) + s*v)*d
                     for v,d in zip(base,dmin))
    avg = total/runs
    print(f"Average objective value with real stations over {runs} scenarios: {avg:.2f}")
    return avg
```

Evaluation code of real world stations

```
Scenario 1: Objective function value = 10249.62
Scenario 2: Objective function value = 10576.46
Scenario 3: Objective function value = 10497.69
Scenario 4: Objective function value = 8221.97
Scenario 5: Objective function value = 9525.03
Scenario 6: Objective function value = 9747.03
Scenario 7: Objective function value = 9874.63
Scenario 8: Objective function value = 10472.84
Scenario 9: Objective function value = 9921.54
Scenario 10: Objective function value = 9893.59
Scenario 11: Objective function value = 9972.40
Scenario 12: Objective function value = 10308.34
Scenario 13: Objective function value = 9732.36
Scenario 14: Objective function value = 9704.12
Scenario 15: Objective function value = 10050.26
Scenario 16: Objective function value = 10043.62
Scenario 17: Objective function value = 10362.60
Scenario 18: Objective function value = 9280.25
Scenario 19: Objective function value = 8020.58
Scenario 20: Objective function value = 7812.43
Scenario 21: Objective function value = 9722.58
Scenario 22: Objective function value = 8094.93
Scenario 23: Objective function value = 10398.37
Scenario 24: Objective function value = 9871.80
Scenario 25: Objective function value = 10179.82
Scenario 26: Objective function value = 10692.53
Scenario 27: Objective function value = 9363.02
Scenario 28: Objective function value = 9389.11
Scenario 29: Objective function value = 10162.83
Scenario 30: Objective function value = 10245.51
Scenario 31: Objective function value = 9911.92
Scenario 32: Objective function value = 9787.64
Scenario 33: Objective function value = 10515.47
```

```
Scenario 34: Objective function value = 10481.25
Scenario 35: Objective function value = 9703.66
Scenario 36: Objective function value = 9906.99
Scenario 37: Objective function value = 9542.51
Scenario 38: Objective function value = 10324.85
Scenario 39: Objective function value = 9774.51
Scenario 40: Objective function value = 9636.14
Scenario 41: Objective function value = 9517.44
Scenario 42: Objective function value = 10363.36
Scenario 43: Objective function value = 9605.67
Scenario 44: Objective function value = 9831.57
Scenario 45: Objective function value = 9666.41
Scenario 46: Objective function value = 10159.85
Scenario 47: Objective function value = 9813.06
Scenario 48: Objective function value = 9791.65
Scenario 49: Objective function value = 9807.85
Scenario 50: Objective function value = 9651.53
Scenario 51: Objective function value = 10413.08
Scenario 52: Objective function value = 10406.03
Scenario 53: Objective function value = 9799.27
Scenario 54: Objective function value = 10170.88
Scenario 55: Objective function value = 10757.83
Scenario 56: Objective function value = 9845.60
Scenario 57: Objective function value = 9814.97
Scenario 58: Objective function value = 9858.52
Scenario 59: Objective function value = 9547.23
Scenario 60: Objective function value = 9732.68
Scenario 61: Objective function value = 9715.43
Scenario 62: Objective function value = 9866.15
Scenario 63: Objective function value = 8005.99
Scenario 64: Objective function value = 10291.54
Scenario 65: Objective function value = 8693.18
Scenario 66: Objective function value = 9618.52
```

```
Scenario 67: Objective function value = 9855.58
Scenario 68: Objective function value = 10035.92
Scenario 69: Objective function value = 8688.15
Scenario 70: Objective function value = 9702.96
Scenario 71: Objective function value = 9878.98
Scenario 72: Objective function value = 10482.49
Scenario 73: Objective function value = 9870.76
Scenario 74: Objective function value = 9576.90
Scenario 75: Objective function value = 10143.27
Scenario 76: Objective function value = 9744.41
Scenario 77: Objective function value = 10151.68
Scenario 78: Objective function value = 10323.01
Scenario 79: Objective function value = 8811.93
Scenario 80: Objective function value = 10099.65
Scenario 81: Objective function value = 10001.10
Scenario 82: Objective function value = 9731.81
Scenario 83: Objective function value = 8773.74
Scenario 84: Objective function value = 9747.40
Scenario 85: Objective function value = 10258.86
Scenario 86: Objective function value = 10207.64
Scenario 87: Objective function value = 9516.79
Scenario 88: Objective function value = 10310.50
Scenario 89: Objective function value = 9369.49
Scenario 90: Objective function value = 9659.89
Scenario 91: Objective function value = 10557.94
Scenario 92: Objective function value = 9735.80
Scenario 93: Objective function value = 10146.95
Scenario 94: Objective function value = 8805.51
Scenario 95: Objective function value = 9217.12
Scenario 96: Objective function value = 9810.43
Scenario 97: Objective function value = 9668.32
Scenario 98: Objective function value = 10242.64
Scenario 99: Objective function value = 9868.00
Scenario 100: Objective function value = 9963.64
```

Results of every single scenario of V1

```
Average objective value over 100 scenarios with 4 stations: 9796.77

Most frequently selected stations for 4 stations:
Kilometer 42: selected 100 times
Kilometer 17: selected 83 times
Kilometer 35: selected 83 times
Kilometer 38: selected 81 times
Kilometer 39: selected 19 times
Kilometer 31: selected 17 times
Kilometer 34: selected 17 times

Total runtime: 11.33 seconds
```

Most selected stations for Variant1_1000m

```
Scenario 1: Objective function value = 81139.21
Scenario 2: Objective function value = 83131.83
Scenario 3: Objective function value = 88896.20
Scenario 4: Objective function value = 85427.85
Scenario 5: Objective function value = 89431.49
Scenario 6: Objective function value = 87481.30
Scenario 7: Objective function value = 89711.74
Scenario 8: Objective function value = 87723.37
Scenario 9: Objective function value = 88466.47
Scenario 10: Objective function value = 94537.45
Scenario 11: Objective function value = 88006.54
Scenario 12: Objective function value = 87969.65
Scenario 13: Objective function value = 92044.38
Scenario 14: Objective function value = 87884.56
Scenario 15: Objective function value = 89508.69
Scenario 16: Objective function value = 85068.85
Scenario 17: Objective function value = 87756.98
Scenario 18: Objective function value = 93621.67
Scenario 19: Objective function value = 91158.70
Scenario 20: Objective function value = 88554.68
Scenario 21: Objective function value = 91641.18
Scenario 22: Objective function value = 93533.17
Scenario 23: Objective function value = 87762.09
Scenario 24: Objective function value = 90522.39
Scenario 25: Objective function value = 93248.15
Scenario 26: Objective function value = 86920.70
Scenario 27: Objective function value = 88701.92
Scenario 28: Objective function value = 90170.93
Scenario 29: Objective function value = 92922.52
Scenario 30: Objective function value = 83224.65
Scenario 31: Objective function value = 90562.02
Scenario 32: Objective function value = 87536.24
Scenario 33: Objective function value = 92650.04
```

```
Scenario 34: Objective function value = 93642.47
Scenario 35: Objective function value = 88956.68
Scenario 36: Objective function value = 88599.88
Scenario 37: Objective function value = 91216.09
Scenario 38: Objective function value = 86232.09
Scenario 39: Objective function value = 89761.23
Scenario 40: Objective function value = 85608.63
Scenario 41: Objective function value = 86938.36
Scenario 42: Objective function value = 87636.89
Scenario 43: Objective function value = 82391.78
Scenario 44: Objective function value = 88919.97
Scenario 45: Objective function value = 90473.80
Scenario 46: Objective function value = 76972.59
Scenario 47: Objective function value = 89446.97
Scenario 48: Objective function value = 91194.36
Scenario 49: Objective function value = 91331.92
Scenario 50: Objective function value = 94309.09
Scenario 51: Objective function value = 92209.71
Scenario 52: Objective function value = 89444.98
Scenario 53: Objective function value = 89596.31
Scenario 54: Objective function value = 82317.66
Scenario 55: Objective function value = 91691.24
Scenario 56: Objective function value = 80970.58
Scenario 57: Objective function value = 91218.64
Scenario 58: Objective function value = 88092.32
Scenario 59: Objective function value = 77216.28
Scenario 60: Objective function value = 83913.80
Scenario 61: Objective function value = 92058.19
Scenario 62: Objective function value = 93371.96
Scenario 63: Objective function value = 90676.20
Scenario 64: Objective function value = 90339.78
Scenario 65: Objective function value = 93857.65
Scenario 66: Objective function value = 88320.23
```

```
Scenario 67: Objective function value = 86511.78
Scenario 68: Objective function value = 89119.31
Scenario 69: Objective function value = 89345.15
Scenario 70: Objective function value = 87605.76
Scenario 71: Objective function value = 88594.33
Scenario 72: Objective function value = 91704.02
Scenario 73: Objective function value = 83248.00
Scenario 74: Objective function value = 88000.03
Scenario 75: Objective function value = 90466.98
Scenario 76: Objective function value = 83246.37
Scenario 77: Objective function value = 88988.49
Scenario 78: Objective function value = 87418.56
Scenario 79: Objective function value = 91225.27
Scenario 80: Objective function value = 80479.28
Scenario 81: Objective function value = 89513.03
Scenario 82: Objective function value = 86060.84
Scenario 83: Objective function value = 91422.26
Scenario 84: Objective function value = 86728.70
Scenario 85: Objective function value = 83879.39
Scenario 86: Objective function value = 82857.28
Scenario 87: Objective function value = 81871.36
Scenario 88: Objective function value = 78657.75
Scenario 89: Objective function value = 93009.30
Scenario 90: Objective function value = 90356.63
Scenario 91: Objective function value = 90231.34
Scenario 92: Objective function value = 89397.55
Scenario 93: Objective function value = 86756.18
Scenario 94: Objective function value = 91994.86
Scenario 95: Objective function value = 86906.41
Scenario 96: Objective function value = 91711.25
Scenario 97: Objective function value = 90571.42
Scenario 98: Objective function value = 89516.39
Scenario 99: Objective function value = 90206.06
Scenario 100: Objective function value = 87063.95
```

Results of every single scenario of V2

```
Average objective value over 100 scenarios with 4 stations: 88405.11

Most frequently selected stations for 4 stations:
Kilometer 17.5: selected 81 times
Kilometer 41.6: selected 78 times
Kilometer 38.6: selected 60 times
Kilometer 35.1: selected 54 times
Kilometer 38.5: selected 29 times
Kilometer 35.2: selected 24 times
Kilometer 41.7: selected 22 times
Kilometer 4.9: selected 19 times
Kilometer 32.5: selected 19 times
Kilometer 8.3: selected 6 times
Kilometer 38.7: selected 3 times
Kilometer 5.0: selected 2 times
Kilometer 38.4: selected 2 times
Kilometer 35.0: selected 1 times

Total runtime: 1165.57 seconds
```

Most selected stations for Variant2_100m

```
Scenario 1: GA objective = 85407.80
Scenario 2: GA objective = 87395.30
Scenario 3: GA objective = 88257.65
Scenario 4: GA objective = 92842.15
Scenario 5: GA objective = 88596.67
Scenario 6: GA objective = 90626.95
Scenario 7: GA objective = 86672.62
Scenario 8: GA objective = 81624.69
Scenario 9: GA objective = 86760.73
Scenario 10: GA objective = 92988.20
Scenario 11: GA objective = 90914.31
Scenario 12: GA objective = 89805.02
Scenario 13: GA objective = 90893.03
Scenario 14: GA objective = 86975.94
Scenario 15: GA objective = 87575.42
Scenario 16: GA objective = 92625.41
Scenario 17: GA objective = 90050.15
Scenario 18: GA objective = 88055.81
Scenario 19: GA objective = 93139.93
Scenario 20: GA objective = 87130.63
Scenario 21: GA objective = 86815.11
Scenario 22: GA objective = 87860.55
Scenario 23: GA objective = 91718.32
Scenario 24: GA objective = 90139.19
Scenario 25: GA objective = 77133.08
Scenario 26: GA objective = 89703.10
Scenario 27: GA objective = 88999.65
Scenario 28: GA objective = 87868.74
Scenario 29: GA objective = 90660.31
Scenario 30: GA objective = 90550.10
Scenario 31: GA objective = 87131.03
Scenario 32: GA objective = 91489.88
Scenario 33: GA objective = 89844.55
Scenario 34: GA objective = 85999.28
Scenario 35: GA objective = 91897.94
Scenario 36: GA objective = 92288.39
Scenario 37: GA objective = 90062.79
Scenario 38: GA objective = 92821.95
Scenario 39: GA objective = 91964.60
Scenario 40: GA objective = 90065.69
Scenario 41: GA objective = 89395.39
Scenario 42: GA objective = 81995.30
Scenario 43: GA objective = 88516.15
Scenario 44: GA objective = 89358.07
Scenario 45: GA objective = 78720.01
Scenario 46: GA objective = 91442.73
Scenario 47: GA objective = 93754.65
Scenario 48: GA objective = 75055.92
Scenario 49: GA objective = 80035.37
Scenario 50: GA objective = 82241.59
Scenario 51: GA objective = 83329.82
Scenario 52: GA objective = 88255.51
Scenario 53: GA objective = 86193.43
Scenario 54: GA objective = 90597.95
Scenario 55: GA objective = 90137.11
Scenario 56: GA objective = 87481.85
Scenario 57: GA objective = 88137.56
Scenario 58: GA objective = 85836.44
Scenario 59: GA objective = 79510.34
Scenario 60: GA objective = 93481.77
Scenario 61: GA objective = 92087.81
Scenario 62: GA objective = 81990.14
Scenario 63: GA objective = 89326.31
Scenario 64: GA objective = 91766.57
Scenario 65: GA objective = 86819.47
Scenario 66: GA objective = 86066.16
Scenario 67: GA objective = 87332.38
Scenario 68: GA objective = 90823.82
Scenario 69: GA objective = 88965.47
Scenario 70: GA objective = 91568.60
Scenario 71: GA objective = 87594.63
Scenario 72: GA objective = 93883.28
Scenario 73: GA objective = 92307.47
Scenario 74: GA objective = 92684.75
Scenario 75: GA objective = 88461.37
Scenario 76: GA objective = 85676.61
Scenario 77: GA objective = 92311.78
Scenario 78: GA objective = 89077.07
Scenario 79: GA objective = 89485.04
Scenario 80: GA objective = 90203.95
Scenario 81: GA objective = 85713.77
Scenario 82: GA objective = 89743.15
Scenario 83: GA objective = 91069.65
Scenario 84: GA objective = 91265.41
Scenario 85: GA objective = 89360.94
Scenario 86: GA objective = 93423.82
Scenario 87: GA objective = 86559.63
Scenario 88: GA objective = 87995.44
Scenario 89: GA objective = 88323.15
Scenario 90: GA objective = 86745.96
Scenario 91: GA objective = 89458.33
Scenario 92: GA objective = 90027.55
Scenario 93: GA objective = 85359.19
Scenario 94: GA objective = 91571.96
Scenario 95: GA objective = 90290.26
Scenario 96: GA objective = 86954.35
Scenario 97: GA objective = 86279.31
Scenario 98: GA objective = 82571.73
Scenario 99: GA objective = 88163.61
Scenario 100: GA objective = 88224.92
```

Results of every single scenario of V3

```
Average GA objective over 100 scenarios with 4 stations: 88443.64

Most frequently selected stations for 4 stations:
Kilometer 17.5: selected 61 times
Kilometer 41.6: selected 57 times
Kilometer 38.6: selected 48 times
Kilometer 35.1: selected 45 times
Kilometer 41.7: selected 32 times
Kilometer 38.5: selected 29 times
Kilometer 35.2: selected 18 times
Kilometer 17.4: selected 17 times
Kilometer 4.9: selected 11 times
Kilometer 31.3: selected 11 times
Kilometer 41.8: selected 9 times
Kilometer 8.3: selected 9 times
Kilometer 32.5: selected 8 times
Kilometer 38.7: selected 7 times
Kilometer 34.0: selected 4 times
Kilometer 35.5: selected 4 times
Kilometer 33.9: selected 3 times
Kilometer 31.2: selected 3 times
Kilometer 8.4: selected 3 times
Kilometer 35.0: selected 3 times
Kilometer 35.4: selected 3 times
Kilometer 17.6: selected 2 times
Kilometer 17.3: selected 2 times
Kilometer 34.9: selected 2 times
Kilometer 41.5: selected 1 times
Kilometer 9.4: selected 1 times
Kilometer 37.7: selected 1 times
Kilometer 9.6: selected 1 times
Kilometer 37.6: selected 1 times
Kilometer 41.9: selected 1 times
Kilometer 38.8: selected 1 times
Kilometer 38.4: selected 1 times
Kilometer 33.8: selected 1 times

Total runtime: 20.67 seconds
```

Most selected stations for Variant3_Metaheuristic