Learning Emergency Medical Dispatch Policies via Genetic Programming

Jordan MacLachlan
jordan.maclachlan@vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

Yi Mei
yi.mei@vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

Fangfang Zhang
fangfang.zhang@vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

Mengjie Zhang
mengjie.zhang@vuw.ac.nz
Victoria University of Wellington
Wellington, New Zealand

Jessica Signal
Wellington Free Ambulance
Wellington, New Zealand

ABSTRACT

Of great value to modern municipalities is the task of emergency medical response in the community. Resource allocation is vital to ensure minimal response times, which we may perform via human experts or automate by maximising ambulance coverage. To combat black-box modelling, we propose a modularised Genetic Programming Hyper Heuristic framework to learn the five key decisions of Emergency Medical Dispatch (EMD) within a reactive decision-making process. We minimise the representational distance between our work and reality by working with our local ambulance service to design a set of heuristics approximating their current decision-making processes and a set of synthetic datasets influenced by existing patterns in practice. Through our modularised framework, we learn each decision independently to identify those most valuable to EMD and learn all five decisions simultaneously, improving performance by 69% on the largest novel dataset. We analyse the decision-making logic behind several learned rules to further improve our understanding of EMD. For example, we find that emergency urgency is not necessarily considered when dispatching idle ambulances in favour of maximising fleet availability.

CCS CONCEPTS

• **Computing methodologies → Planning under uncertainty**.

KEYWORDS

Emergency Medical Dispatch, Genetic Programming, Hyper-Heuristic, Simulation, Dynamic Optimisation

ACM Reference Format:
Jordan MacLachlan, Yi Mei, Fangfang Zhang, Mengjie Zhang, and Jessica Signal. 2023. Learning Emergency Medical Dispatch Policies via Genetic Programming. In *Genetic and Evolutionary Computation Conference (GECCO '23), July 15–19, 2023, Lisbon, Portugal*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3583131.3590434

GECCO '23, July 15–19, 2023, Lisbon, Portugal

ACM ISBN 979-8-4007-0119-1/23/07…$15.00
https://doi.org/10.1145/3583131.3590434

1 INTRODUCTION

Emergency Medical Dispatch (EMD) is an NP-hard optimisation task concerning the prompt assignment of medical resources to emergencies in the community [40]. In EMD, a heterogeneous ambulance set must quickly respond to variant and uncertain emergencies occurring in a dynamic manner. This objective aligns with a pattern long known by the medical community; that patient mortality and ambulance response time are positively correlated [10, 16]. Research that aids EMD is undeniably valuable to society. In 2019, the U.S. Federal Communications Commission estimated that reducing the average national ambulance response time by sixty seconds would save more than 10,000 lives per year [9]. In the same year, the U.S. Department of Transportation estimated the value of *one* statistical life at US$10.9m [9].

Due to the inherently dynamic and uncertain nature of EMD, dispatching ambulances *a priori*, e.g. by traditional, mathematical optimisation techniques, is infeasible. Robust-proactive decision-making techniques are typically two-staged [13, 44]: robust solution construction and dynamic recourse. Building robust solutions to EMD is impractical, given we do not know the location or urgency of emergencies ahead of time. To address this issue, scholars have developed predictive algorithms that distribute idle ambulances to maximise *coverage* and send the nearest ambulance upon the occurrence of a new emergency [20, 30, 43]. Such a simple real-time dispatch is by no means optimal.

Real-time emergency dispatch is often the task of human decision-makers, aided by manually designed dispatch rules, while data-informed coverage criteria may aid idle ambulance relocation. The primary value of human dispatchers is audit capacity. After a fatal decision, an Emergency Services Provider can interview a dispatcher to understand their logic. This process improves dispatcher training and refines the set of manually designed dispatch rules aiding human dispatchers in line with the priorities of the Emergency Services Provider in question.

The research community has recently rekindled its focus on ambulance dispatch [15, 17]. Most notably, Liu et al. [23] designed a multiagent reinforcement learning algorithm for ambulance relocation, and Yang and Albert [51] expanded their prior stochastic programming model [49] to consider non-transport vehicles, determining when to send multiple agents, or not. However, due to several impractical assumptions, most of the existing research is less applicable to real-world dispatch. First, most problem models discretise the service region (road network) to a grid-based network, which blunts our understanding of the spatial emergency distribution [47] and oversimplifies the problem. Second, the techniques used in existing works are incomprehensible to and non-adjustable by non-expert Emergency Services Providers and are therefore incompatible with audit requirements. Third, most existing methods simplify the types of ambulances and emergencies, increasing the representational gap between the solved problems and the real world, and thus the risk of practical adoption.

The Genetic Programming Hyper Heuristic (GPHH) is an effective technique to learn online decision-making heuristics [12, 18, 24]. Based on Genetic Programming (GP), a Darwinian-inspired machine-learning algorithm, we evolve a population of heuristics and evaluate each via a simulated Decision-Making Process (DMP). We choose an EMD-specific GPHH formulation as it offers an elegant solution to the above three practical limitations. A GP-evolved heuristic can scale to larger, realistic instances without problem simplification and information loss, and is inherently *more interpretable* than alternative methods (e.g. reinforcement learning [23]). Finally, we are collaborating with our local ambulance service, Wellington Free Ambulance (WFA), to align our DMP with practical EMD, decreasing the risk of real-world uptake.

This paper's major research goal is to design a modularised DMP framework that represents all decision-making processes for EMD. The key in this modularised framework is the five types of decisions that respond to dynamic events, i.e., new emergency arrivals and ambulances becoming idle (Figure 1). For each decision, we may use a *manually* designed rule or attempt to *learn* a superior rule by GP. We aim to learn these component decisions of EMD to outperform the manually designed rules. To achieve this, we specifically investigate the following specific objectives:

(1) Develop a modular multi-tree GPHH framework that allows us to simultaneously learn GP-rules for any subset of decisions (while fixing the others to their manual rule);
(2) Explore which decisions benefit from GP, and which we must defer to the manually designed rule;
(3) Investigate whether we must learn decision rules independently or simultaneously;
(4) Identify the decision(s) that contribute most to improving EMD performance.

The following Section formally details the EMD problem and briefly introduces the GPHH algorithm for EMD. Section 3 describes the developed multi-tree GPHH for EMD. Section 4 outlines our experimental studies, presenting and discussing the achieved results before Section 5 concludes this work.

2 BACKGROUND

2.1 Problem Description

EMD is based on a fully connected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ (city road network), where $\mathcal{V}$ denotes the vertex set and $\mathcal{E}$ the edge set. Between any two vertices u, v, the matrix $\boldsymbol{\tau} = [\tau_{uv}] \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ stores the traversal time. We denote two subsets of the vertex set: one as facilities $\mathcal{F} \subset \mathcal{V}$, to which idle ambulances may return, and another as hospitals $\mathcal{P} \subset \mathcal{V}$, to which a *capacitated* ambulance must transport

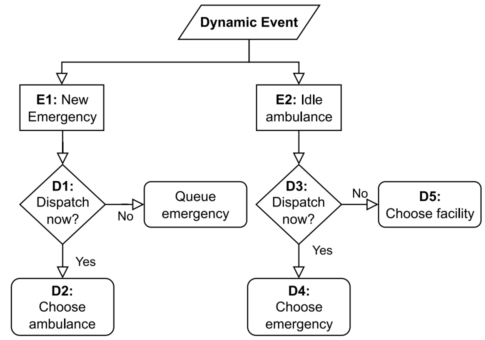

Figure 1: The decisions (D1 through D5) required in a realistic EMD environment and the dynamic events (E1 and E2) that trigger them [15].

emergencies requiring hospitalisation. The heterogeneous ambulance set $\mathcal{A}$ collaborate[1] to serve the heterogeneous emergency set $\mathcal{T}$, where $|\mathcal{A}| = m$ and $|\mathcal{T}| = n$. During the real-time dispatching process, we denote the waiting emergency set as Π.

Each emergency $t_j \in \mathcal{T}$ comprises a tuple $\langle \tau_j^r, \ell_j, \mathbf{d}_j, \gamma_j \rangle$. $\tau_j^r \in \mathbb{R}^+$ denotes the emergency receipt time; $\ell_j \doteq v_{j1} + \alpha_j \cdot (v_{j2} - v_{j1})$, the emergency's location, where $(v_{j1}, v_{j2}) \in \mathcal{E}$ and $\alpha_j \in [0, 1]$ (i.e. emergencies occur at some point along an edge); $\mathbf{d}_j = [d_{j1}, \ldots, d_{jD}] \in \mathbb{R}^D$, the demand (time required on site) of D different ambulance skills, where $d_{jD} \in \{0, 1\}$, $d_{jD} = 1$ indicates that the emergency requires hospitalisation, and $d_{jD} = 0$, the inverse; and $\gamma_j \in \{1, \ldots, U\}$, the urgency class of the emergency, where U is the number of distinct urgency classes specified by the ESP. We maintain three pieces of information for each urgency class γ: the weight (urgency) as $\omega_\gamma \in \mathbb{R}$; an ESP's target response time as $\tau_\gamma^T \in \mathbb{R}$; and the fraction of historical emergencies of type γ as $\theta_\gamma \in [0, 1]$, where $\sum_{\gamma=1}^{U} \theta_\gamma = 1$.

The emergency demand vector $\mathbf{d}_j$ and urgency class γ_j are random variables. The true demand distribution $\mathbf{d}_j \sim \mathcal{D}(\tilde{\mathbf{d}}_j)$ is conditioned on the estimated demand $\tilde{\mathbf{d}}_j$, and the true urgency class distribution $\gamma_j \sim \boldsymbol{\mathcal{U}}(\tilde{\gamma}_j)$ is conditioned on the estimated urgency class $\tilde{\gamma}_j$. An Emergency Services Provider (ESP) knows an emergency t_j's estimated values, i.e. $\tilde{\mathbf{d}}_j$ and $\tilde{\gamma}_j$, at time τ_j^r, and learn the true values, i.e. $\mathbf{d}_j$ and γ_j, when an ambulance first arrives at t_j.

Each ambulance $a_i \in \mathcal{A}$ comprises a tuple $\langle \ell_i(\tau^c), \mathbf{q}_i \rangle$. The first entry $\ell_i(\tau^c)$ denotes a_i's *location* at the current time τ^c during the dispatching process, where $\ell_i(0) \in \mathcal{F}$, and $\mathbf{q}_i = [q_{i1}, \ldots, q_{iD}] \in \mathbb{R}^D$ denotes the ambulance's D-dimensional *skill* vector to fulfill the demands. $q_{iD} = 1$ means that the ambulance a_i has the capacity to transport an emergency to a hospital, and $q_{iD} = 0$ otherwise. The actual on-site service time depends on the collaboration between ambulances. At any time, the service efficiency is proportional to the average skill of all the ambulances on site, and the service is completed when all the demands has decreased to zero.

We represent a solution to an EMD instance I as $\boldsymbol{S}_I = (\boldsymbol{X}_I, \boldsymbol{\mathcal{Y}}_I)$ where $\boldsymbol{X}_I = [\mathrm{X}_1, \ldots, \mathrm{X}_m]$ and $\boldsymbol{\mathcal{Y}}_I = [\mathrm{Y}_1, \ldots, \mathrm{Y}_m]$. Like-indexed entries correspond to a sequence of emergency path tuples for

[1] Explicitly differentiating EMD from self-interested agents (e.g. taxi dispatch) and competitive agents (e.g. competitive games).

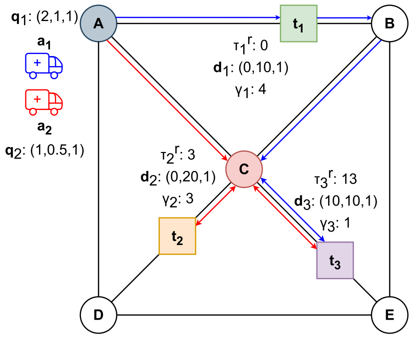

Figure 2: A simple EMD instance with one hospital (node C) and two facilities (nodes A and C). Two capacitated ambulances, a_1 and a_2 (a_2 is more skilled than a_1) begin at A. The emergencies t_1, t_2, and t_3 have receipt times (τ_j^r), demand ($\mathbf{d}_j$) and urgency (γ_j). Assume edge traversal takes one minute, hospital drop-off is instantaneous, and emergency locations are at the nearest edge quarter.

Table 1: An example solution to the instance given by Figure 2. a_1 immediately attends t_1 (service takes $1 * 10 = 10$ mins) and a_2 attends t_2 upon arrival (service takes $0.5 * 20 = 10$ mins). a_1 has one minute idle at C prior to being dispatched to t_3 at time 12, which requires a highly skilled ambulance. a_2 arrives at t_3 at time 15.75, at which point a_1 has served 2 units of demand. The remaining demand is $(10, 8)$ (or, equivalently, $(9, 10)$) which we multiply by the average skill vector $(1.5, 0.75)$ to reach a remaining demand of $(15, 6)$. Both ambulances finish at the nearest facility C at time 37.5.

Ambulance	Component	Solution
a_1	X_1	$\langle(A, 0) \to (C, 12)\rangle, \langle(C, 13) \to (C, 37.5)\rangle$
	Y_1	$\langle(0, t_1), (13, t_3)\rangle$
a_2	X_2	$\langle(A, 3) \to (C, 15)\rangle, \langle(C, 15) \to (C, 37.5)\rangle$
	Y_2	$\langle(3, t_2), (15, t_3)\rangle$

ambulance a_i:

$$\mathbf{X}_i = [x_{i1}, \ldots, x_{ip}, \ldots, x_{iP_i}, x_{iP_i+1}]$$
$$\mathbf{Y}_i = [y_{i1}, \ldots, y_{ip}, \ldots, y_{iP_i}]$$

where $P_i \leq n$ indicates the number of distinct emergencies attended by ambulance a_i. The tuple $x_{ip} \doteq \langle \ell_i(\tau_{i,p-1}), \ell_i(\tau_{i,p}) \rangle$ is a pair of locations and time stamps denoting the ambulance's start location $\ell_i(\tau_{i,p-1})$ at time $\tau_{i,p-1}$ and end location $\ell_i(\tau_{i,p})$ at time $\tau_{i,p}$. Specifically, $\ell_i(\tau_{i,0}) = \ell_i(0)$, $\tau_{i,0}$ equals the time the ambulance a_i is first dispatched to an emergency, and the final $\mathbf{X}_i$ entry denotes the path from ambulance a_i's final emergency at location $\ell_i(\tau_{i,P_i})$ to a facility at location $\ell_i(\tau_{i,P_i+1})$. The tuple $y_{ip} \doteq \langle \tau_{ip}^d, t_p \rangle$ records the time ambulance a_i is dispatched to each emergency t_p as τ_{ip}^d. The ambulance travels from one location to the next following the shortest path, which can be pre-calculated in advance. Figure 2 shows an example EMD instance with two heterogeneous ambulances and three heterogeneous emergencies, and Table 1 shows an example solution to this instance. It is not difficult to imagine a scenario where this relatively efficient dispatch scheme goes awry. Should t_3 have arrived at time = 4, for example, ambulance a_1 would not have been able to respond until time = 12.

A valid solution must adhere to the following constraints:

- the ambulance set must serve all emergency demand;
- all ambulance routes must start and end at facility nodes;
- we cannot dispatch an ambulance to an emergency prior to having received the emergency;
- an ambulance cannot depart for a new emergency prior to it having completed serving its current emergency;
- ambulance a_i's traversal time from emergency t_{p-1} to t_p must be at least as long as the shortest path between the two.

Equation (1) defines the EMD objective function to be minimised: the average weighted response time, where τ_j^a is the time the first ambulance arrives at the emergency t_j. $\vartheta_j = 1$ if the solution is valid, and some sufficiently large constant if some emergencies fail to be completed. Equation (2) defines the emergency weight ω_{γ_j} that maintains the relative contribution of each emergency urgency to the final objective value.

$$C(I) = \frac{1}{n} \sum_{j=1}^{n} (\tau_j^a - \tau_j^r) \vartheta_j \omega_{\gamma_j}, \tag{1}$$

$$\omega_{\gamma_j} = \frac{1}{(\tau_{\gamma_j}^T)^2 \cdot \theta_{\gamma_j}}, \tag{2}$$

2.2 Related Work

2.2.1 Optimising Ambulance Location. The typical approach to ambulance dispatch is to optimise the fleet's *coverage* of the service region by either optimally locating facilities [8, 38, 42] or dynamically allocating ambulances among predefined facilities [2, 14]. As most services must operate with existing structures, we are more interested in the latter. Restrepo [33], Maxwell, Henderson, and Topaloglu [28], and Maxwell et al. [29] use approximate dynamic programming to redeploy idle ambulances among facilities. Schmid [36] additionally considers ambulance dispatch (ambulance-emergency allocation). Beraldi and Bruni [4] locate ambulances among facilities in the first stage of a stochastic program, and respond to dynamic emergencies in the second stage. Alanis, Ingolfsson, and Kolfal [1] develop a two-dimensional Markov chain model to relocate idle ambulances in real time and use a Matlab emergency simulator to validate that their model is accurate. Yoon and Albert [48] merge an approximate hypercube model and a mixed-integer linear program to locate and dispatch idle ambulances, improving coverage over urgent emergencies in a dynamic environment. Liu et al. [23] develop a multiagent deep reinforcement learning algorithm for ambulance allocation, learning how many ambulances to send to each region during each epoch, given demand.

2.2.2 Optimising Ambulance Dispatch. Yoon and Albert [49] introduce *multiple-response*, preferring to send several heterogeneous ambulances, if available, to urgent emergencies. They found that multiple-response is preferred, even under high fleet utilisation, given a suitable stochastic programming (or manually designed) dispatch policy. Yoon and Albert [51] incorporate Bender's Cut [3] to their prior stochastic programming model [49] to consider real-world scale problems and non-transport ambulances, finding non-transport ambulances positively impact system performance. Guigues et al. [15] proposed a column generation algorithm using region (and time) discretisation to learn the five fundamental decisions of EMD (see Figure 1), showing improved performance over basic nearest-first dispatch rules on real-world data.

McLay and Mayorga [31] and Yoon and Albert [50] formulate EMD as a Markov Decision Process and actively learn which ambulance to send to each emergency under urgency uncertainty. The former [31] finds that it is not always optimal to send the nearest ambulance, both when queuing and passing unserved emergencies to a neighbouring service. In contrast to many predefined thresholds previously proposed [35, 41], the latter [50] designed a dynamic cutoff scheme to automatically determine when advanced ambulances should respond to emergencies under both finite and infinite horizon scenarios. In addition, Jagtenberg, Bhulai, and van der Mei [19] manually design a heuristic to solve instances with large vehicle volumes. Specifically, the dispatch heuristic balances the distance the ambulance is from the emergency and the ambulance's impact on fleet coverage. However, while the authors found that the heuristic decreased response time target violations, it increased the average overall response time.

Yang et al. [47] devised an ambulance dispatch simulation to evaluate the efficacy of a solution generated by a Gaussian-process-based search algorithm [37]. The authors advised against the common practice of blunt region discretisation [11, 15, 46, 49, 51], proposing a more realistic discretisation alternative based on a Gaussian mixture model.

2.2.3 *Genetic Programming Hyper Heuristic.*

GPHH [6] allows us to search the heuristic space to possibly outperform these expert-defined rules. GPHH generates high-level heuristics (learned rules) using low-level heuristics (primitive set elements).

Scholars have successfully applied GPHH to various vehicle dispatching problems [7], routing problems [25, 45], and scheduling problems [12, 52]. Despite this, ambulance dispatch has received relatively little attention. Runka [34] applied GPHH to the Robocup Rescue Simulation System, simultaneously coordinating an ambulance fleet, fire department, and police force in response to disasters (e.g. post-earthquake). MacLachlan et al. [26] used GPHH and an ambulance dispatch simulation to learn decisions **D1** and **D4**, conflating the two, thus observing nominal results.

2.2.4 *Summary.*

While relocating an idle ambulance (Section 2.2.1) and selecting a suitable ambulance for an emergency (Section 2.2.2) are undoubtedly important, they are not the only decisions made by real-world dispatchers. This work aims to learn all five key decisions in EMD, as shown in Figure 1.

Many existing techniques rely on black-box models (e.g. reinforcement learning), or effectively black-box models to non-experts (e.g. stochastic programming), and thus conflict with audit (interpretability) objectives. We believe tree-based GP can learn inherently *more interpretable* rules than said methods.

With regards to investigated problem models, many oversimplify the problem via region discretisation, considering too few emergency or ambulance types, or making assumptions such as passing excess emergencies to neighbouring ambulance fleets. The problem model and instances solved in this work void these issues.

3 GPHH FOR EMERGENCY MEDICAL DISPATCH

We present the general framework of the GPHH in Algorithm 1, and of the associated EMD simulation framework in Algorithm 2. In multi-tree GP, we define an individual g as a tuple $\mathcal{Z}_g = \langle \mathbf{r}_g, \mathbf{b}_g \rangle$, where $\mathbf{r}_g = [r_{g1}, \ldots, r_{gR}]$ is a list of rules and $\mathbf{b}_g = [0, 1]^R$ a binary vector that indicates whether or not we evolve the rule with GPHH ($b_{g1} = 1$) or use a manually designed rule ($b_{g1} = 0$). Thus, $\sum_{r=1}^{R} b_{gr}$ equals the number of trees simultaneously learned.

Algorithm 1 The GPHH for EMD.

Randomly *initialise* a population of multi-tree individuals.
while stopping criteria not met **do**
  Evaluate the population via Algorithm 2 and Equation (1).
  Create a new, empty population (the next generation).
  while the new population is empty **do**
    Generate one or more individuals by:
      Parent Selection and
      Crossover, *Mutation*, *Reproduction*, or *Elitism*.
    Add the new individual(s) to the new population.

Algorithm 2 The EMD simulation for fitness evaluation.

Input: EMD instance I and GP individual $\mathcal{Z}$.
Output: A solution $\mathcal{S}_I$.
$\xi, \mathcal{S}_I$ = initialise a new EMD state and solution.
Γ = a new, empty event queue (ordered by minimal time).
for i = 1 ... m **do**
  $\Gamma \leftarrow$ new facility arrival event for ambulance a_i at time 0.
for j = 1 ... n **do**
  $\Gamma \leftarrow$ new emergency event for t_j at time τ_j^r. ▷ i.e. **E1**
while Γ is not empty **do**
  $\epsilon \leftarrow$ pop most prior (earliest) event from Γ.
  trigger ϵ to update the state ξ and solution $\mathcal{S}_I$.
return S_I

To evaluate the effectiveness of an individual $\mathcal{Z}_g$, we apply it to a set of training instance samples via an EMD-specific simulated DMP (i.e. Algorithm 2). We use rule r_{g1} to make decision **D1**: *dispatch now*, r_{g2} to make decision **D2**: *choose ambulance*, and so on. Designing a DMP representative of real-world EMD is pivotal to creating a learning environment relevant to the industry.

Algorithm 3 shows the pseudocode for new emergency events (i.e. **E1**). Line 1 determines whether to immediately attend the new emergency (**D1**); Line 7, which ambulance to send (**D2**). Algorithm 4 shows the pseudocode for idle ambulance events (i.e. **E2**). Line 1 determines whether to dispatch an idle ambulance to a waiting task (**D3**); Lines 4 and 11, which facility to return to (**D5**); Line 7, which emergency to attend (**D4**). We treat r_{g1} and r_{g3} as threshold functions around zero and r_{g2}, r_{g4}, and r_{g5} as priority functions.

The purpose of this work is to simultaneously learn the important component decisions of EMD, **D1** - **D5** (Figure 1), and identify those that most impact performance. GP-Di denotes the algorithm used to learn a rule for decision **D**i (e.g. GP-D1, **D1**), and GP-DA denotes the algorithm used to learn rules for all decisions simultaneously.

3.1 Baseline Algorithm

To properly determine the effectiveness of the learned GPHH algorithms, we test our learned rules against a baseline algorithm (EMD-BL) using a set of rules manually designed alongside WFA to approximate their human decision-making logic. The first three

Algorithm 3 The new emergency event: **E1**

Input: EMD instance I, GP individual $\mathcal{Z}_g$, current state ξ, and new emergency t_j.

if $r_{g1}(t_j) \geq 0.0$ **then** ▷ i.e. **D1**
  $\Pi \leftarrow t_j$ ▷ Add the new emergency to the global queue.
  return
$\Xi \leftarrow$ idle ambulance subset $\mathcal{A}^i \subset \mathcal{A}$.
$\Lambda \leftarrow$ new priority queue of ambulances ordered by $r_{g2}(\Xi)$.
while Λ is not empty **do**
  $a_i \leftarrow$ pop most prior ambulance from Λ. ▷ i.e. **D2**
  if a_i required at t_j **then**
    send a_i to t_j.
    $\Gamma \leftarrow$ new idle a_i event: when t_j ends. ▷ i.e. **E2**
if t_j insufficiently assigned **then**
  $\Pi \leftarrow t_j$. ▷ Add the new emergency to the global queue.

Algorithm 4 The idle ambulance event: **E2**

Input: EMD instance I, GP individual $\mathcal{Z}_g$, current state ξ, and idle ambulance a_i.

if $r_{g3}(a_i) \geq 0.0$ **then** ▷ i.e. **D3**
  $\Gamma \leftarrow$ new send event: a_i to a facility by $r_{g5}(a_i)$. ▷ i.e. **D5**
  return
$\Pi \leftarrow$ global waiting queue $\Pi \subset \mathcal{T}$.
$\Lambda \leftarrow$ new priority queue of emergencies ordered by $r_{g4}(\Pi)$.
while Λ is not empty **do**
  $t_j \leftarrow$ pop most prior emergency from Λ. ▷ i.e. **D4**
  if t_j requires ambulance a_i **then**
    send a_i to t_j.
    $\Gamma \leftarrow$ new idle a_i event: when t_j ends. ▷ i.e. **E2**
    return
$\Gamma \leftarrow$ new send event: a_i to a facility by $r_{g5}(a_i)$. ▷ i.e. **D5**

of these are consistent with those previously studied. We show the manual rules below.

$$r^m_{g1}(t_j) = -1 \tag{3}$$

$$r^m_{g2}(a_i, t_j) = \texttt{travel\_time}(\ell_i(\tau^c), \ell_j) \tag{4}$$

$$r^m_{g3}(a_i) = -1 \tag{5}$$

$$r^m_{g4}(a_i, t_j) = \texttt{travel\_time}(\ell_i(\tau^c), \ell_j)/\omega_{\Upsilon_j} \tag{6}$$

$$r^m_{g5}(a_i, f_k) = g(f_k) \tag{7}$$

$r^m_{g1}(t_j)$ (Equation 3) always immediately dispatches an idle ambulance to a new emergency; $r^m_{g2}(t_j)$ (Equation 4) prioritises nearby ambulances; $r^m_{g3}(t_j)$ (Equation 5) always immediately dispatches an idle ambulance to a waiting emergency; $r^m_{g4}(t_j)$ (Equation 6) prioritises nearby emergencies, weighted by emergency urgency; and $r^m_{g5}(t_j)$ (Equation 7) dispatches an idle ambulance to a facility by $g(f_k)$: to the nearest facility with the minimal number of capacitated ambulances present.

4 EXPERIMENTS

4.1 Experiment Design

The function set is $\{+, -, \times, /, max, min, if\}$ for learning all rules. The / operator denotes protected division, returning one if the denominator is zero, and the if operator takes three variables x, y, z as input, and returns y if $x > 0$ and z otherwise. However, each of the five decisions considers a different aspect of the current EMD state. For example, **D1** examines a single new emergency, and **D2** examines a single new emergency with respect to each idle ambulance. As a result, we require a different GP terminal set for each rule $\mathbf{r}_g$. Table 2 details all terminals used (A stands for ambulance features; A:F for ambulance-facility features; A:E for ambulance-emergency features; F for facility features; G for global features; and E for emergency features), while Table 3 outlines which terminal sub-sets each rule $\mathbf{r}_g$ uses during evolution.

Table 2: The proposed terminals used in GP.

Set	Code	Description
A	AC	Ambulance transport capacity.
	SK	Relative ambulance skill, an integer.
	TTI	Approximated time until idle.
A:F	FTD	Distance: ambulance to facility.
A:E	DCA	Distance: closest alternative ambulance to emergency.
	TTD	Distance: ambulance to emergency.
F	FQ	Maximum facility capacity.
	FC	The fraction of $\mathcal{V}$ within a given RTT.
	NP	Number of ambulances present.
G	EL	The state Escalation Level (pressure).
	FB	Fraction of ambulances currently busy.
	NWA	Number of ambulances waiting for backup.
	NWT	Number of queued emergencies, i.e. $\lvert\Pi\rvert$.
	NWUT	Number of queued urgent emergencies (i.e. tier 1 or 2).
E	AD	The number of ambulances yet assigned to the emergency.
	PD	The number of ambulances yet present at the emergency.
	RTA	Binary: does this emergency require a transport ambulance?
	T	The tardiness of the current emergency.
	UR	Emergency weight: ω_{Υ_j}, where $\varphi_j = 1$.
	TR	The estimated time until we complete this emergency.

Table 3: The terminals used for learning different rules.

Rule	Terminal sub-sets	Codes
r_{g1}	Emerg and Global.	{E, G}
r_{g2}	Emerg, Ambo, Ambo:Emerg, and Global.	{E, A, A:E, G}
r_{g3}	Ambo and Global.	{A, G}
r_{g4}	Emerg, Ambo, Ambo:Emerg, and Global.	{E, A, A:E, G}
r_{g5}	Ambo, Fac, Ambo:Fac, and Global.	{A, F, A:F, G}

Table 4 shows the other GP parameters of the proposed algorithms, following the settings for other popular GPHH techniques [25, 32]. Specifically, we train each individual on five instances (five days of emergencies), rotating the instance set each generation. We test the best individual of the final generation (the ‘learned rules’) on a size-500 unseen test set (500 days of emergencies). The test performance is the average objective value of the generated solutions over the test instances. Each GP algorithm is run 30 times independently, and we compare their results by Wilcoxon rank sum test and Friedman test with a significance level of 0.05.

We test the algorithms on three synthetic datasets based on the graphs of the EGL [21, 22], EGL-G [5], and HFE [39] datasets. We collaborate with WFA to consider real-world patterns such as time-variant demand volume, approximate urgency proportions, and daily ambulance fluctuations to improve data realism. These datasets are available for public download [27].

4.2 Results and Discussions

4.2.1 Performance analysis. Table 5 shows the test performance on the `EGL`, `EGL-G`, and `HFE` datasets, including a Wilcoxon rank sum and Friedman test. The symbols after each *performance value* denote an algorithm’s *instance-level* performance relative to those *following*: a (+) indicates this method significantly outperforms the equivalently indexed other; a (−), the opposite; an (=), no significant difference. The symbols after each *average rank* denote

Table 4: The GP parameter setting.

Parameter	Value	Parameter	Value
Population size	1024	Generations	51
Tournament size	7	Crossover rate	0.8
Mutation rate	0.15	Reproduction rate	0.05
Maximal depth	8	Elitism	10

an algorithm's *dataset-level* performance relative to those *following*. The values marked in bold indicate the algorithm with the lowest test performance. From this, we make the following observations.

On average, GP-D1 performs worse than EMD-BL on all three datasets, performing significantly worse on the `EGL` *dataset.* GP-D1 suffers a high standard deviation, suggesting it contains a considerable number of outliers and overfit the training data on several instances, namely `egl-e2-b`, `egl-g1-b`, `hfe-3`, and `hfe-9`. Comparing the training and test performance in these instances confirms this hypothesis; several rules that perform well during training fail during test. We create new emergency events under two circumstances: a) when a new emergency occurs, and b) when an ambulance requests backup. Solution analysis shows that poor heuristics frequently decide not to dispatch an ambulance to such urgent events. This behaviour leads to an accumulation of incapacitated non-transport ambulances and invalid solutions which we heavily penalise. Future research will aim to capture these edge cases.

On average, GP-D2 performs the same or worse than EMD-BL on all three datasets, with no statistical significance. Intuitively, r^m_{g1} increases the likelihood that the entire fleet is busy when an urgent emergency arrives, thus offering little opportunity for GP-D2 to learn. Solution analysis over the test set supports this hypothesis: the idle ambulance set is empty on over 60% of **D2** invocations.

GP-D3 performs significantly worse than EMD-BL on the `EGL` *dataset, and significantly better than EMD-BL on the* `EGL-G` *and* `HFE` *datasets.* GP-D1 suffers a high standard deviation, suggesting it contains several outliers and overfit the training data. Poorly learned **D3** rules often fail to dispatch idle transport ambulance backup to non-transport ambulances present at urgent emergencies, deciding to return to a facility instead. Identifying or otherwise mitigating the cause of these rare events during training will inform future works. GP-D3 significantly outperforms the other individually learned rules on the largest two datasets, suggesting that an idle ambulance delay strategy for EMD is preferred over an immediate dispatch strategy in the presence of many urgent emergencies. This aligns with real-world EMD practices while contrasting with often-held assumptions in the online decision-making literature.

GP-D4 significantly outperforms EMD-BL on all three datasets. GP-D4 significantly outperforms all other individually learned rules on the `EGL` dataset and is only significantly outperformed by GP-D3 on the larger datasets. While we expected GP-D4 to perform well, as **D4** is the most explicit point at which the fleet can discriminate between emergencies, we did not expect GP-D3 to outperform it. We believe GP-D3 outperforms GP-D4 on the larger datasets as GP-D4 cannot skip Π in favour of returning to a facility. As the fleet encounters more high-urgency emergencies on the larger datasets, GP-D4 is penalised when no ambulances are idle.

On average, GP-D5 performs the same or worse than EMD-BL on all three datasets, with no statistical significance. Over the `HFE` dataset we observe that an idle ambulance only applies the **D5** rule around

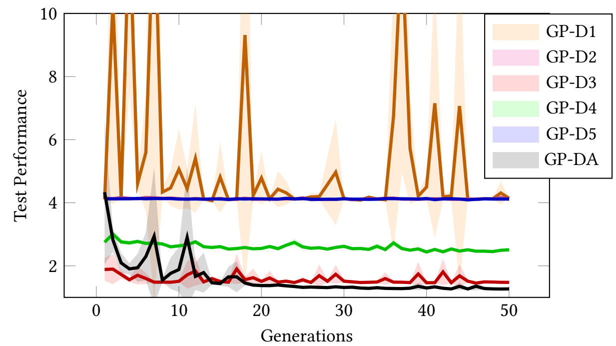

Figure 3: The test curves over generations of GP-D1, GP-D2, GP-D3, GP-D4, GP-D5, and GP-DA on the `hfe-10` instance.

20% of the time an ambulance becomes idle. The only case **D5** will apply is when Π is empty, offering few opportunities for GP to learn an appropriate **D5** rule, particularly in circumstances where the system is under significant stress.

Figure 3 suggests ***D3*** *and* ***D4*** *are the most valuable* independent *decisions of the EMD simulation.* GP-D1 is unstable and shows no learning curve. Despite this, several GP-D1 rules achieve a better test performance than GP-D3. Should we remedy the flaws of the GP-D1 learning process, **D1** may be very valuable to EMD. GP-D2 and GP-D5 are barely indistinguishable, showing no performance variability (mean or standard deviation), indicating neither algorithm was able to learn during the GPHH process. We may be able to defer to r^m_{g2} and r^m_{g5} in this case. GP-D3 and GP-D4 exhibit a good performance curve and a significantly improved default level of performance than alternate individually learned rules.

GP-DA significantly outperforms EMD-BL on all datasets and outperforms all independent decision algorithms on the `EGL-G` *and* `HFE` *datasets.* The test curve of GP-DA in Figure 3 shows that it overcomes the erratic learning path of GP-D1 and mitigates the downfalls of the manually designed rules. We wish to determine whether GP-DA performs well because it can learn more effective *individual* rules, or because, while each rule is less effective, they *combine* well. Table 6 shows the test performance of the best and mean individual decision rules (i.e. GP-D1 - GP-D5), and the test performance of the best, median (16/30), and worst performing GP-DA rule, where each *isolated decision rule* is run alongside the four other manually designed rules on the `hfe-1` instance. We observe that the individual GP-GA decision rules need not be independently high-performing to contribute to a high-performing multi-tree GP-GA rule. Merely, such individual rules need only be average. Nonetheless, a better **D4** rule tends to lead to better GP-DA test fitness.

4.2.2 Rule analysis. When using GP, it is important to understand learned rules. Via terminal frequency analysis (Figure 4), we make the following observations regarding decisions **D3** and **D4**:

GP-D3 uses `SK` 36.67% more than GP-DA (**D3**). This aligns with the industry standard of considering skill in dispatch decisions. However, as GP-D3 works in tandem with the manual rule set, learned rules are the only point at which we may incorporate skill, hence its increased use. GP-D3 uses `TTI` 36.14% less than GP-DA (**D3**). Given the high rate of use and the fact that the `TTI` terminal always equals zero for idle ambulances, this observation suggests GP-DA (**D3**) identifies the value of a zero-valued constant. GP-D3

Table 5: The test performance (weighted response times) of the proposed algorithms.

Graph	EMD-BL	GP-D1	GP-D2	GP-D3	GP-D4	GP-D5	GP-DA
egl-e1-a	0.37(0.00)(+)(-)(+)(-)(-)(-)	0.45(0.18)(-)(=)(-)(-)(-)	0.36(0.01)(+)(-)(=)(-)	0.46(0.21)(-)(-)(-)	0.30(0.10)(+)(=)	0.36(0.02)(-)	**0.27(0.03)**
egl-e1-b	0.41(0.00)(+)(=)(+)(-)(=)(-)	0.43(0.02)(-)(=)(-)(-)(-)	0.41(0.01)(=)(-)(=)(-)	0.43(0.05)(-)(-)(-)	**0.30(0.09)(+)(=)**	0.41(0.02)(-)	0.31(0.06)
egl-e1-c	0.31(0.00)(+)(-)(+)(=)(=)(=)	0.34(0.04)(-)(+)(=)(-)(=)	**0.30(0.01)(+)(=)(=)(=)**	0.66(1.05)(-)(-)(-)	0.30(0.10)(=)(=)	0.31(0.01)(=)	0.34(0.12)
egl-e2-a	0.35(0.00)(-)(-)(=)(-)(=)(-)	0.34(0.02)(=)(=)(-)(=)(-)	0.34(0.01)(+)(-)(=)(-)	0.38(0.10)(-)(-)(-)	**0.23(0.03)(+)(+)**	0.34(0.02)(-)	0.26(0.05)
egl-e2-b	0.49(0.00)(+)(=)(+)(-)(+)(-)	2.41(5.68)(=)(=)(-)(=)(-)	0.50(0.03)(+)(-)(=)(-)	0.68(0.24)(-)(-)(-)	**0.33(0.06)(+)(=)**	0.51(0.02)(-)	0.36(0.07)
egl-e2-c	0.31(0.00)(+)(=)(+)(-)(=)(-)	0.38(0.11)(-)(=)(-)(-)(-)	0.31(0.02)(+)(-)(=)(-)	0.40(0.14)(-)(-)(-)	**0.21(0.03)(+)(+)**	0.30(0.01)(-)	0.27(0.08)
egl-e3-a	0.40(0.00)(+)(-)(=)(-)(-)(-)	0.54(0.15)(-)(-)(-)(-)(-)	0.38(0.01)(=)(-)(-)(-)	0.47(0.21)(-)(=)(-)	**0.25(0.02)(+)(+)**	0.37(0.02)(-)	0.28(0.04)
egl-e3-b	0.39(0.00)(+)(+)(+)(-)(=)(-)	0.43(0.05)(-)(-)(-)(-)(-)	0.41(0.01)(=)(-)(=)(-)	0.43(0.09)(-)(=)(-)	0.29(0.06)(+)(=)	0.40(0.02)(-)	**0.28(0.04)**
egl-e3-c	0.27(0.00)(+)(+)(=)(-)(+)(-)	0.34(0.04)(-)(-)(-)(-)(-)	0.30(0.01)(=)(-)(-)(-)	0.31(0.13)(-)(=)(-)	**0.20(0.02)(+)(+)**	0.28(0.01)(-)	0.25(0.07)
egl-e4-a	0.28(0.00)(+)(-)(+)(-)(-)(-)	0.30(0.02)(-)(+)(-)(-)(-)	0.26(0.01)(+)(-)(+)(-)	0.37(0.15)(-)(-)(-)	**0.20(0.03)(+)(+)**	0.27(0.01)(-)	0.23(0.04)
egl-e4-b	0.33(0.00)(+)(+)(+)(-)(+)(-)	0.37(0.03)(-)(=)(-)(=)(-)	0.35(0.02)(=)(-)(=)(-)	0.40(0.12)(-)(=)(-)	**0.23(0.02)(+)(=)**	0.36(0.02)(-)	0.24(0.05)
egl-e4-c	0.42(0.00)(+)(-)(=)(-)(-)(-)	0.43(0.03)(-)(=)(-)(-)(-)	0.40(0.02)(+)(-)(=)(-)	0.49(0.19)(-)(-)(-)	0.35(0.19)(+)(=)	0.39(0.02)(-)	**0.35(0.18)**
egl-s1-a	0.36(0.00)(=)(=)(+)(-)(+)(-)	0.38(0.05)(=)(+)(-)(=)(-)	0.36(0.01)(+)(-)(=)(-)	0.47(0.15)(-)(-)(-)	**0.26(0.03)(+)(=)**	0.37(0.01)(-)	0.29(0.04)
egl-s1-b	0.34(0.00)(=)(-)(+)(-)(-)(-)	0.34(0.01)(-)(+)(-)(=)(-)	0.33(0.01)(+)(-)(=)(-)	0.38(0.06)(-)(-)(-)	**0.25(0.03)(+)(=)**	0.33(0.01)(-)	0.26(0.03)
egl-s1-c	**0.38(0.00)(+)(=)(+)(=)(=)(=)**	0.40(0.02)(-)(=)(=)(-)(=)	0.38(0.01)(+)(=)(=)(=)	0.46(0.16)(=)(-)(=)	0.43(0.17)(=)(=)	0.38(0.01)(=)	0.45(0.19)
egl-s2-a	0.40(0.00)(+)(+)(+)(-)(+)(-)	0.44(0.03)(=)(=)(-)(=)(-)	0.43(0.01)(=)(-)(=)(-)	0.51(0.26)(-)(=)(-)	**0.35(0.08)(+)(=)**	0.42(0.01)(-)	0.35(0.09)
egl-s2-b	0.46(0.00)(+)(+)(+)(-)(+)(-)	0.50(0.03)(-)(+)(-)(-)(-)	0.47(0.01)(+)(-)(=)(-)	0.62(0.22)(-)(-)(-)	**0.33(0.16)(+)(=)**	0.48(0.01)(-)	0.34(0.07)
egl-s2-c	0.34(0.00)(+)(=)(+)(-)(=)(-)	0.36(0.02)(-)(=)(-)(-)(-)	0.34(0.01)(+)(-)(=)(-)	0.39(0.13)(-)(-)(-)	**0.24(0.03)(+)(+)**	0.33(0.01)(-)	0.27(0.04)
egl-s3-a	0.36(0.00)(=)(-)(=)(-)(-)(-)	0.37(0.02)(-)(=)(-)(-)(-)	0.35(0.01)(=)(-)(=)(-)	0.41(0.17)(-)(-)(-)	**0.24(0.03)(+)(+)**	0.34(0.02)(-)	0.26(0.03)
egl-s3-b	0.36(0.00)(+)(+)(+)(-)(+)(-)	0.42(0.06)(-)(-)(-)(-)(-)	0.38(0.02)(=)(-)(=)(-)	0.38(0.05)(-)(=)(-)	**0.23(0.04)(+)(+)**	0.37(0.01)(-)	0.34(0.21)
egl-s3-c	0.35(0.00)(+)(+)(+)(-)(+)(-)	0.41(0.13)(-)(=)(-)(-)(-)	0.37(0.01)(+)(-)(=)(-)	0.47(0.21)(-)(-)(-)	**0.24(0.02)(+)(+)**	0.37(0.01)(-)	0.28(0.05)
egl-s4-a	0.32(0.00)(=)(+)(+)(-)(+)(-)	0.33(0.05)(=)(+)(-)(=)(-)	0.33(0.01)(+)(-)(=)(-)	1.06(1.01)(-)(-)(-)	**0.24(0.04)(+)(+)**	0.33(0.01)(-)	0.27(0.06)
egl-s4-b	0.34(0.00)(=)(-)(+)(-)(-)(-)	0.35(0.03)(=)(+)(-)(-)(-)	0.33(0.01)(+)(-)(-)(-)	0.39(0.05)(-)(-)(-)	**0.25(0.03)(+)(+)**	0.33(0.01)(-)	0.28(0.02)
egl-s4-c	0.37(0.00)(+)(=)(+)(-)(=)(-)	0.39(0.03)(-)(=)(-)(-)(-)	0.36(0.01)(+)(-)(=)(-)	0.51(0.26)(-)(-)(-)	**0.26(0.02)(+)(+)**	0.36(0.01)(-)	0.32(0.07)
Average	0.36	0.48	0.36	0.48	**0.27**	0.36	0.30
Average rank	4.24(+)(=)(+)(-)(=)(-)	5.59(-)(=)(-)(-)(-)	4.35(+)(-)(=)(-)	5.72(-)(-)(-)	**1.67**(+)(+)	4.23(-)	2.20
egl-g1-a	3.95(0.00)(+)(+)(-)(-)(+)(-)	4.44(1.77)(=)(-)(-)(=)(-)	3.98(0.05)(-)(-)(=)(-)	1.85(0.24)(+)(+)(-)	2.73(0.07)(+)(-)	3.99(0.06)(-)	**1.61(0.08)**
egl-g1-b	3.67(0.00)(+)(+)(-)(-)(=)(-)	5.80(9.97)(-)(-)(-)(-)(-)	3.71(0.04)(-)(-)(=)(-)	1.67(0.10)(+)(+)(-)	2.48(0.09)(+)(-)	3.71(0.09)(-)	**1.53(0.17)**
egl-g1-c	3.77(0.00)(+)(=)(-)(-)(-)(-)	3.78(0.13)(=)(-)(-)(=)(-)	3.77(0.06)(-)(-)(=)(-)	2.24(0.90)(+)(+)(-)	2.64(0.07)(+)(-)	3.73(0.04)(-)	**1.61(0.23)**
egl-g1-d	3.45(0.00)(+)(+)(-)(-)(+)(-)	3.51(0.06)(=)(-)(-)(=)(-)	3.51(0.05)(-)(-)(=)(-)	1.68(0.09)(+)(+)(-)	2.63(0.22)(+)(-)	3.51(0.05)(-)	**1.57(0.12)**
egl-g1-e	4.50(0.00)(=)(=)(-)(-)(=)(-)	4.64(0.53)(=)(-)(-)(=)(-)	4.49(0.05)(-)(-)(=)(-)	**1.90(0.13)(+)(+)(+)**	3.12(0.11)(+)(-)	4.49(0.06)(-)	2.19(2.72)
egl-g2-a	4.55(0.00)(+)(+)(-)(-)(=)(-)	4.79(0.22)(-)(-)(-)(-)(-)	4.61(0.06)(-)(-)(-)(-)	1.86(0.08)(+)(+)(-)	3.12(0.06)(+)(-)	4.54(0.07)(-)	**1.71(0.14)**
egl-g2-b	4.65(0.00)(=)(=)(-)(-)(=)(-)	4.68(0.07)(=)(-)(-)(=)(-)	4.64(0.06)(-)(-)(=)(-)	2.07(0.19)(+)(+)(-)	3.51(0.10)(+)(-)	4.66(0.06)(-)	**1.82(0.08)**
egl-g2-c	3.45(0.00)(+)(+)(-)(-)(=)(-)	3.73(1.50)(=)(-)(-)(=)(-)	3.49(0.07)(-)(-)(=)(-)	1.78(0.22)(+)(+)(-)	2.42(0.18)(+)(-)	3.46(0.05)(-)	**1.53(0.21)**
egl-g2-d	4.11(0.00)(+)(+)(-)(-)(+)(-)	4.20(0.06)(=)(-)(-)(=)(-)	4.21(0.06)(-)(-)(=)(-)	1.86(0.28)(+)(+)(-)	2.82(0.06)(+)(-)	4.21(0.05)(-)	**1.65(0.24)**
egl-g2-e	3.54(0.00)(=)(=)(-)(-)(=)(-)	3.59(0.10)(=)(-)(-)(=)(-)	3.60(0.07)(-)(-)(=)(-)	1.84(0.31)(+)(+)(-)	2.45(0.08)(+)(-)	3.58(0.05)(-)	**1.53(0.15)**
Average	3.96	4.31	4.00	1.88	2.79	3.99	**1.67**
Average rank	4.93(=)(=)(-)(-)(=)(-)	5.87(=)(-)(-)(=)(-)	5.70(-)(-)(=)(-)	1.97(+)(+)(-)	2.95(+)(-)	5.46(-)	**1.11**
hfe-1	3.97(0.00)(=)(=)(-)(-)(=)(-)	3.98(0.04)(=)(-)(-)(=)(-)	3.98(0.04)(-)(-)(=)(-)	1.39(0.05)(+)(+)(-)	2.38(0.09)(+)(-)	3.98(0.04)(-)	**1.21(0.11)**
hfe-2	3.77(0.00)(+)(=)(-)(-)(=)(-)	3.82(0.06)(=)(-)(-)(-)(-)	3.80(0.06)(-)(-)(=)(-)	1.44(0.12)(+)(+)(-)	2.29(0.10)(+)(-)	3.78(0.03)(-)	**1.19(0.09)**
hfe-3	4.16(0.00)(=)(+)(-)(-)(+)(-)	7.44(12.17)(=)(-)(-)(=)(-)	4.17(0.04)(-)(-)(=)(-)	1.59(0.83)(+)(+)(-)	2.36(0.08)(+)(-)	4.18(0.03)(-)	**1.19(0.07)**
hfe-4	3.69(0.00)(-)(=)(-)(-)(=)(-)	3.67(0.07)(=)(-)(-)(=)(-)	3.69(0.03)(-)(-)(=)(-)	1.61(0.63)(+)(+)(-)	2.18(0.09)(+)(-)	3.68(0.04)(-)	**1.21(0.24)**
hfe-5	3.70(0.00)(+)(+)(-)(-)(+)(-)	3.78(0.12)(+)(-)(-)(=)(-)	3.78(0.03)(-)(-)(=)(-)	1.38(0.08)(+)(+)(-)	2.29(0.15)(+)(-)	3.77(0.03)(-)	**1.14(0.10)**
hfe-6	3.94(0.00)(=)(-)(-)(-)(-)(-)	5.82(10.15)(=)(-)(-)(-)(-)	3.92(0.04)(-)(-)(-)(-)	1.48(0.37)(+)(+)(-)	2.44(0.19)(+)(-)	3.88(0.03)(-)	**1.21(0.09)**
hfe-7	3.77(0.00)(+)(=)(-)(-)(=)(-)	3.90(0.51)(-)(-)(-)(=)(-)	3.78(0.03)(-)(-)(=)(-)	1.67(0.98)(+)(+)(-)	2.24(0.10)(+)(-)	3.78(0.04)(-)	**1.24(0.11)**
hfe-8	4.05(0.00)(+)(+)(-)(-)(+)(-)	4.11(0.04)(=)(-)(-)(=)(-)	4.13(0.05)(-)(-)(=)(-)	1.47(0.21)(+)(+)(-)	2.45(0.12)(+)(-)	4.14(0.05)(-)	**1.19(0.09)**
hfe-9	2.85(0.00)(=)(=)(-)(-)(=)(-)	29.04(80.37)(=)(-)(-)(=)(-)	2.85(0.04)(-)(-)(=)(-)	1.03(0.05)(+)(+)(-)	1.43(0.07)(+)(-)	2.85(0.03)(-)	**0.84(0.05)**
hfe-10	4.10(0.00)(=)(=)(-)(-)(=)(-)	4.14(0.03)(=)(-)(-)(=)(-)	4.11(0.04)(-)(-)(=)(-)	1.48(0.08)(+)(+)(-)	2.51(0.18)(+)(-)	4.12(0.04)(-)	**1.27(0.10)**
Average	3.80	6.97	3.82	1.46	2.26	3.82	**1.17**
Average rank	5.07(=)(=)(-)(-)(=)(-)	5.68(=)(-)(-)(=)(-)	5.63(-)(-)(=)(-)	2.04(+)(+)(-)	2.97(+)(-)	5.56(-)	**1.04**

Table 6: Comparing the `hfe-1` test performances of the individually learned rules and the individual rules of the best, median (16th), and worst GP-GA rules run alongside the other manually designed rules.

DX	Best GP-DX	Mean GP-DX	Best GP-DA	16th GP-DA	Worst GP-DA
D1	3.89	3.98	3.97	3.97	3.97
D2	3.89	3.98	3.94	3.97	3.96
D3	1.32	1.39	1.48	1.42	1.50
D4	2.23	2.38	2.83	3.11	3.33
D5	3.90	3.98	4.00	3.88	3.91

uses `NWA` 32.92% more than GP-DA (**D3**). Intuitively, if we do not send a transport ambulance to another waiting for backup, we may indefinitely lose a member of the fleet. Increased use of `NWA` aligns with countering this potential error. Figure 5 shows a high-performing GP-D3 rule, simplified to the case of only considering advanced-skill ambulances. Here, if the skilled ambulance has transport capacity, the rule returns a negative value if `NWUT` is positive, thus will choose to attend a waiting task (i.e. `D4`). Else, if the ambulance does not have transport capacity, the output is either one (if `NWA` = 0) or equivalent to a transport ambulance (if `NWA` > 0). This behaviour aligns with practice: use transport ambulances as much as possible, and reserve the more mobile, non-transport agents in cases of relaxed pressure on the fleet. GP-D4 uses `AC` 44.56% more than GP-DA (**D4**). Given we learn GP-D4 alongside the manual rule r^m_{g3} (always dispatch), and the frequent use of `RTA`, we deduce that GP-D4 adequately identifies the emergencies that require backup,

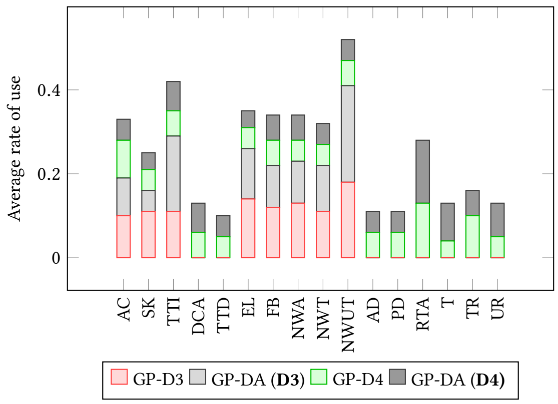

Figure 4: The rates at which the learned rules of GP-D3, GP-DA (D3), GP-D4, and GP-DA (D4) use each terminal on average on the EGL-G and HFE dataset.

aligning with the point above. GP-D4 uses T 56.02% less than GP-DA (**D4**). Given the generic nature of T and the specialisation of GP-D4, we hypothesise that T is not valuable to EMD. GP-D4 uses TR 42.73% more than GP-DA (**D4**). TR offers an *approximation* as to how much longer an ambulance will take before completing the demand of an emergency (given historical information). As such, we expect GP-D4 rules to avoid sending ambulances to emergencies that have little time remaining. GP-D4 uses UR 36.67% less than GP-DA (**D4**), a *highly* unexpected observation. From rule analysis, we make two observations: rules that use UR frequently, as expected (i.e. prioritising urgent emergencies), and rules that ignore UR and focus on features that maintain fleet availability (i.e. providing backup). Consider Figure 6, showing a high-performing GP-D4 rule of the latter type, learned on the hfe-10 instance. If RTA = 0, the rule prioritises emergencies requiring minimal additional resources. If RTA = 1, the rule prioritises emergencies requiring maximal additional resources. Further, the rule prioritises emergencies with nearby ambulances, maximising the likelihood of receiving backup, scaled by the current pressure on the fleet. Finally, the rule prefers emergencies with a larger *expected* time remaining, maximising dispatch value. In general, while this rule does not utilise the UR terminal, it does balance the remaining terminals in a manner pursuant to our expectations of fleet management. Further, we observe that the **D3** and **D4** rules use more primitives per tree, on average. However, GP-DA can learn more succinct trees across all decisions, while still achieving better performance than all GP-DX algorithms.

4.2.3 Summary. Through a comprehensive analysis, we confirm that GPHH is capable of simultaneously learning effective decision rules for EMD. On instances that incur few urgent emergencies (the EGL dataset), deciding whether or not to immediately attend an emergency (**D1**) or send an idle ambulance to a waiting emergency (**D3**) are the least valuable decisions and identifying the most suitable emergency for an idle ambulance is the most valuable (**D4**). For instances that incur many urgent emergencies (the EGL-G and HFE datasets), while **D4** remains pertinent, **D3** is the most valuable. Finally, we show that we can learn all rules simultaneously and obtain both more succinct and performant heuristics.

Nonetheless, our implementation leaves room for improvement. We believe GP-D1, GP-D2, and GP-D5 perform poorly due to an

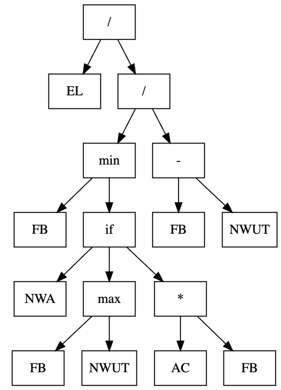

Figure 5: A learned GP-D3 rule, simplified for advanced skill ambulances, learned on the hfe-10 instance.

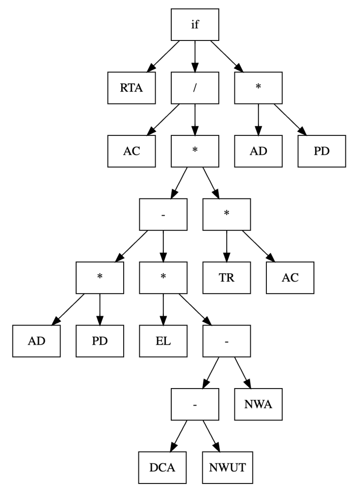

Figure 6: A learned GP-D4 rule learned on the hfe-10 instance.

insufficient representation. Future research will improve the DMP and identify additional terminal features to ensure these decisions contribute to improving the system performance of GP-DA.

5 CONCLUSIONS

This paper aims to determine whether GPHH could successfully learn dispatch rules for EMD to outperform those that approximate human dispatchers. We have proposed a modularised DMP framework that facilitated thorough analysis which showed GPHH is an algorithm capable of learning such rules. We make three specific conclusions from this work: a) two idle ambulance decision points are of high value to EMD (**D3** and **D4**), b) our current representation likely disadvantages the new task decision points (**D1** and **D5**), and c) learning all decisions simultaneously can overcome the flaws of the manually designed rule set, performing better via succinct rules. Our future work will simplify the decision set, expand the problem representation, and consider paramedic workload restrictions [11].

ACKNOWLEDGMENTS

To Wellington Free Ambulance, for their continued practical advice on this project, and service to our community.

REFERENCES

[1] Ramon Alanis, Armann Ingolfsson, and Bora Kolfal. 2013. A Markov chain model for an EMS system with repositioning. *Production and operations management* 22, 1 (2013), 216–231.

[2] Tobias Andersson and Peter Värbrand. 2007. Decision support tools for ambulance dispatch and relocation. *Operational Research for Emergency Planning in Healthcare* 1 (2007), 36–51.

[3] Jacques Benders. 1962. Partitioning Procedures for Solving Mixed-Variable Programming Problems. *Numerische Matkematic* 4 (1962), 238–252.

[4] Patrizia Beraldi and Maria Elena Bruni. 2009. A probabilistic model applied to emergency service vehicle location. *European Journal of Operational Research* 196, 1 (2009), 323–331.

[5] José Brandão and Richard Eglese. 2008. A deterministic tabu search algorithm for the capacitated arc routing problem. *Computers & Operations Research* 35, 4 (2008), 1112–1126.

[6] Edmund K Burke, Mathew R Hyde, Graham Kendall, Gabriela Ochoa, Ender Ozcan, and John R Woodward. 2009. Exploring hyper-heuristic methodologies with genetic programming. In *Computational intelligence*. Springer, 177–201.

[7] Xinan Chen, Ruibin Bai, Rong Qu, and Haibo Dong. 2022. Cooperative Double-Layer Genetic Programming Hyper-Heuristic for Online Container Terminal Truck Dispatching. *IEEE Transactions on Evolutionary Computation* (2022).

[8] Richard Church and Charles ReVelle. 1974. The maximal covering location problem. In *Papers of the Regional Science Association*, Vol. 32. Springer, 101–118.

[9] Federal Communications Commission et al. 2020. Wireless E911 location accuracy requirements. *Federal Register* 07-114 (2020).

[10] Tan N Doan, Brendan V Schultz, Stephen Rashford, and Emma Bosley. 2020. Surviving out-of-hospital cardiac arrest: the important role of bystander interventions. *Australasian Emergency Care* 23, 1 (2020), 47–54.

[11] Shakiba Enayati, Osman Y. Özaltın, Maria E. Mayorga, and Cem Saydam. 2018. Ambulance redeployment and dispatching under uncertainty with personnel workload limitations. *IISE Transactions* 50, 9 (2018), 777–788.

[12] Huali Fan, Hegen Xiong, and Mark Goh. 2021. Genetic programming-based hyper-heuristic approach for solving dynamic job shop scheduling problem with extended technical precedence constraints. *Computers & Operations Research* 134 (2021), 105401.

[13] Michel Gendreau, Ola Jabali, and Walter Rei. 2016. 50th anniversary invited article–future research directions in stochastic vehicle routing. *Transportation Science* 50, 4 (2016), 1163–1173.

[14] Michel Gendreau, Gilbert Laporte, and Frédéric Semet. 1997. Solving an ambulance location model by tabu search. *Location science* 5, 2 (1997), 75–88.

[15] Vincent Guigues, Anton Kleywegt, and Victor Hugo Nascimento. 2022. Operation of an ambulance fleet under uncertainty. *arXiv preprint arXiv:2203.16371* (2022).

[16] Johan Holmén, Johan Herlitz, Sven-Erik Ricksten, Anneli Strömsöe, Eva Hagberg, Christer Axelsson, and Araz Rawshani. 2020. Shortening ambulance response time increases survival in out-of-hospital cardiac arrest. *Journal of the American Heart Association* 9, 21 (2020), e017048.

[17] Cheng Hua and Tauhid Zaman. 2020. Optimal dispatch in emergency service system via reinforcement learning. In *INFORMS International Conference on Service Science*. Springer, 75–87.

[18] Zhixing Huang, Fangfang Zhang, Yi Mei, and Mengjie Zhang. 2022. An Investigation of Multitask Linear Genetic Programming for Dynamic Job Shop Scheduling. In the *European Conference on Genetic Programming*. Springer, 162–178.

[19] Caroline J Jagtenberg, Sandjai Bhulai, and Robert D van der Mei. 2017. Dynamic ambulance dispatching: is the closest-idle policy always optimal? *Health Care Management Science* 20, 4 (2017), 517–531.

[20] Håkon Leknes, Eirik Skorge Aartun, Henrik Andersson, Marielle Christiansen, and Tobias Andersson Granberg. 2017. Strategic ambulance location for heterogeneous regions. *European Journal of Operational Research* 260, 1 (2017), 122–133.

[21] Leon Y O Li. 1992. *Vehicle Routeing for Winter Gritting*. Ph. D. Dissertation. Lancaster University.

[22] Leon Y O Li and Richard W Eglese. 1996. An interactive algorithm for vehicle routeing for winter—gritting. *Journal of the Operational Research Society* 47, 2 (1996), 217–228.

[23] Kunpeng Liu, Xiaolin Li, Cliff C Zou, Haibo Huang, and Yanjie Fu. 2020. Ambulance dispatch via deep reinforcement learning. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*. 123–126.

[24] Yuxin Liu, Yi Mei, Mengjie Zhang, and Zili Zhang. 2020. A predictive-reactive approach with genetic programming and cooperative coevolution for the uncertain capacitated arc routing problem. *Evolutionary computation* 28, 2 (2020), 289–316.

[25] Jordan MacLachlan, Yi Mei, Juergen Branke, and Mengjie Zhang. 2020. Genetic programming hyper-heuristics with vehicle collaboration for uncertain capacitated arc routing problems. *Evolutionary computation* 28, 4 (2020), 563–593.

[26] Jordan MacLachlan, Yi Mei, Fangfang Zhang, and Mengjie Zhang. 2022. Genetic Programming for Vehicle Subset Selection in Ambulance Dispatching. In *2022 IEEE Congress on Evolutionary Computation*. IEEE, 1–8.

[27] Jordan MacLachlan, Yi Mei, Fangfang Zhang, Mengjie Zhang, and Jessica Signal. 2023. A novel set of Emergency Medical Dispatch datasets. Retrieved April 11, 2023 from https://tinyurl.com/emd23

[28] Matthew S Maxwell, Shane G Henderson, and Huseyin Topaloglu. 2009. Ambulance redeployment: An approximate dynamic programming approach. In the *Winter Simulation Conference*. IEEE, 1850–1860.

[29] Matthew S Maxwell, Mateo Restrepo, Shane G Henderson, and Huseyin Topaloglu. 2010. Approximate dynamic programming for ambulance redeployment. *INFORMS Journal on Computing* 22, 2 (2010), 266–281.

[30] Richard McCormack and Graham Coates. 2015. A simulation model to enable the optimization of ambulance fleet allocation and base station location for increased patient survival. *European Journal of Operational Research* 247, 1 (2015), 294–309.

[31] Laura A McLay and Maria E Mayorga. 2013. A model for optimally dispatching ambulances to emergency calls with classification errors in patient priorities. *IIE Transactions* 45, 1 (2013), 1–24.

[32] Yi Mei and Mengjie Zhang. 2018. Genetic programming hyper-heuristic for multi-vehicle uncertain capacitated arc routing problem. In the *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 141–142.

[33] Mateo Restrepo. 2008. *Computational Methods for Static Allocation and Real-Time Redeployment of Ambulances*. Ph. D. Dissertation. Cornell University.

[34] Andrew Runka. 2011. Genetic programming for the RoboCup Rescue simulation system. (2011).

[35] Christian Schaack and Richard C Larson. 1989. An N server cutoff priority queue where arriving customers request a random number of servers. *Management Science* 35, 5 (1989), 614–634.

[36] Verena Schmid and Karl F Doerner. 2010. Ambulance location and relocation problems with time-dependent travel times. *European Journal of Operational Research* 207, 3 (2010), 1293–1303.

[37] Lihua Sun, L Jeff Hong, and Zhaolin Hu. 2014. Balancing exploitation and exploration in discrete optimization via simulation through a Gaussian process-based search. *Operations Research* 62, 6 (2014), 1416–1438.

[38] C Swoveland, D Uyeno, I Vertinsky, and R Vickson. 1973. Ambulance location: a probabilistic enumeration approach. *Management Science* 20, 4-part-ii (1973), 686–698.

[39] Ke Tang, Juan Wang, Xiaodong Li, and Xin Yao. 2016. A scalable approach to capacitated arc routing problems based on hierarchical decomposition. *IEEE Transactions on Cybernetics* 47, 11 (August 2016), 3928–3940.

[40] Joseph Tassone and Salimur Choudhury. 2020. A comprehensive survey on the ambulance routing and location problems. *arXiv preprint arXiv:2001.05288* (2020).

[41] IDS Taylor and James GC Templeton. 1980. Waiting time in a multi-server cutoff-priority queue, and its application to an urban ambulance service. *Operations Research* 28, 5 (1980), 1168–1188.

[42] Constantine Toregas, Ralph Swain, Charles ReVelle, and Lawrence Bergman. 1971. The location of emergency service facilities. *Operations research* 19, 6 (1971), 1363–1373.

[43] Tonguç Ünlüyurt and Yasir Tunçer. 2016. Estimating the performance of emergency medical service location models via discrete event simulation. *Computers & Industrial Engineering* 102 (2016), 467–475.

[44] Juan Wang, Ke Tang, Jose A Lozano, and Xin Yao. 2016. Estimation of the distribution algorithm with a stochastic local search for uncertain capacitated arc routing problems. *IEEE Transactions on Evolutionary Computation* 20, 1 (2016), 96–109.

[45] Shaolin Wang, Yi Mei, Mengjie Zhang, and Xin Yao. 2021. Genetic programming with niching for uncertain capacitated arc routing problem. *IEEE Transactions on Evolutionary Computation* (2021).

[46] Wei Yang, Qiang Su, Samuel H Huang, Qiugen Wang, Yanhong Zhu, and Ming Zhou. 2019. Simulation modeling and optimization for ambulance allocation considering spatiotemporal stochastic demand. *Journal of Management Science and Engineering* 4, 4 (2019), 252–265.

[47] Wei Yang, Qiang Su, Ming Zhou, and Xinghong Qin. 2020. Ambulance allocation considering the spatial randomness of demand. *Computers & Industrial Engineering* 139 (2020), 106202.

[48] Soovin Yoon and Laura A Albert. 2018. An expected coverage model with a cutoff priority queue. *Health Care Management Science* 21, 4 (2018), 517–533.

[49] Soovin Yoon and Laura A Albert. 2020. A dynamic ambulance routing model with multiple response. *Transportation Research Part E: Logistics and Transportation Review* 133 (January 2020), 101807.

[50] Soovin Yoon and Laura A Albert. 2021. Dynamic dispatch policies for emergency response with multiple types of vehicles. *Transportation Research Part E: Logistics and Transportation Review* 152 (2021), 102405.

[51] Soovin Yoon, Laura A Albert, and Veronica M White. 2021. A stochastic programming approach for locating and dispatching two types of ambulances. *Transportation Science* 55, 2 (2021), 275–296.

[52] Fangfang Zhang, Yi Mei, Su Nguyen, Mengjie Zhang, and Kay Chen Tan. 2021. Surrogate-assisted evolutionary multitask genetic programming for dynamic flexible job shop scheduling. *IEEE Transactions on Evolutionary Computation* 25, 4 (2021), 651–665.