

LLM-Guided Genetic Search for the Home Healthcare Routing and Scheduling Problem

Roberta Angioni
Department of Mathematics and
Computer Science
University of Cagliari
Cagliari, Italy
roberta.angioni@unica.it

Sergio Consoli
European Commission
Ispra, Italy
sergio.consoli@ec.europa.eu

Silvia Maria Massa
Department of Mathematics and
Computer Science
University of Cagliari
Cagliari, Italy
Silviam.massa@unica.it

Diego Reforgiato Recupero
Department of Mathematics and
Computer Science
University of Cagliari
Cagliari, Italy
diego.reforgiato@unica.it

Daniele Riboni
Department of Mathematics and
Computer Science
University of Cagliari
Cagliari, Italy
riboni@unica.it

Abstract

This paper investigates whether Large Language Models (LLMs) can effectively support combinatorial optimization in home healthcare scheduling. As a case study, we consider the Home Health Care Routing and Scheduling Problem (HHCRSP), a challenging healthcare logistics problem with strict temporal and skill constraints. Our experiments show that a standalone LLM is unreliable on this problem, but becomes effective when integrated as a perturbation component within a Biased Random-Key Genetic Algorithm (BRKGA). In the proposed hybrid approach, called LLM-BRKGA, the LLM provides high-level guidance by identifying spatial outliers and hotspots for targeted search perturbations. Results on benchmark instances show that LLM-BRKGA improves solution quality over the BRKGA baseline, especially on medium-to-large and large instances, suggesting that LLMs can effectively complement traditional optimization methods by injecting structured, problem-aware perturbations beyond purely stochastic operators.

CCS Concepts

• **Theory of computation** → **Discrete optimization**; • **Computing methodologies** → **Genetic algorithms**; **Artificial intelligence**; *Natural language processing*.

Keywords

Combinatorial optimization, metaheuristics, Home Healthcare Routing and Scheduling Problem, Generative AI, Genetic algorithms, Large Language Models

ACM Reference Format:

Roberta Angioni, Sergio Consoli, Silvia Maria Massa, Diego Reforgiato Recupero, and Daniele Riboni. 2026. LLM-Guided Genetic Search for the

Home Healthcare Routing and Scheduling Problem. In *International Conference on Information Technology for Social Good (GoodIT '26)*, September 02–04, 2026, Pisa, Italy. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3794786.3830755>

1 Introduction

This paper explores how Large Language Models (LLMs) can support combinatorial optimization in healthcare by complementing traditional objective-driven search with high-level reasoning on structural patterns.

We study the application of LLMs to this problem from two perspectives: first, as standalone solvers that generate schedules directly; second, as components integrated into a classical metaheuristic. Traditional algorithms such as Genetic Algorithms (GAs) [1] are effective optimization tools, but they do not exploit semantic information about the problem structure. By integrating LLMs into the search process, we aim to complement stochastic exploration with high-level guidance based on spatial patterns.

A particularly relevant application for this approach is the Home Health Care Routing and Scheduling Problem (HHCRSP) [7, 16]; indeed, improving the efficiency and fairness of home-care planning can directly support timely care delivery for fragile patients and more sustainable use of healthcare resources. As the world's population ages, Home Health Care (HHC) has become an essential alternative to hospital care, improving resource utilization while allowing patients to receive treatment at home. Efficient routing is therefore crucial to provide timely and equitable care [8].

However, managing the HHCRSP is a major logistical challenge: it is classified as NP-hard [16] and, unlike simple delivery routes, it requires coordinating staff with different medical skills to visit patients at specific times. The problem is also made harder by human factors, such as the need for two caregivers to arrive simultaneously for certain tasks, or the requirement that specific treatments follow a predefined order, as highlighted in the formulation of Mankowska et al. [16].

GAs are widely used to solve NP-hard problems and are among the most established population-based metaheuristic optimization methods. These methods are inspired by the Darwinian theory



This work is licensed under a Creative Commons Attribution 4.0 International License. *GoodIT '26, Pisa, Italy*

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2483-1/26/09

<https://doi.org/10.1145/3794786.3830755>

of natural selection and the biological process of “survival of the fittest”, where candidate solutions evolve from one generation to the next. Prior reviews show that GAs are versatile and effective in large-scale scheduling and other combinatorial settings [1, 10].

However, by relying on stochastic processes, this kind of algorithm can miss obvious improvements, such as simple spatial clusters, that a human planner would recognize instantly, or lead to premature convergence where the algorithm gets stuck in local optima. To bridge this gap, the reasoning capabilities of LLMs can be adopted in the approach we propose here, which aims to inspect instance summaries and route configurations and identify semantic outliers and “hotspots”, allowing domain-aware guidance to steer the combinatorial search performed by a Biased Random-Key Genetic Algorithm (BRKGA) with Multi-Parent (MP) and Implicit Path-Relinking (IPR), proposed by Kummer et al. [11]. For instance, the LLM may identify a patient geographically isolated within a route, suggesting its reassignment to a closer cluster. This recommendation is then translated into a targeted removal and biased reinsertion of the corresponding gene.

The main contributions of this study are as follows:

- *Hybrid optimization framework*: we propose a hybrid LLM-BRKGA method for the HHCRSP, where the LLM provides strategic guidance without replacing the feasibility-preserving search mechanism.
- *LLM-guided perturbation*: we design a mechanism in which the LLM analyzes route summaries and patient proximity information to identify promising substructures for targeted ruin-and-recreate perturbations.
- *Evaluation on standard benchmarks*: we evaluate the proposed method on the benchmark instances introduced by Mankowska et al. in [16].
- *Empirical findings*: we show that the proposed method is competitive with BRKGA on small instances and improves average solution quality on several larger subsets, while a standalone LLM solver remains unreliable.

To the best of our knowledge, this is the first work where: (i) an LLM is used as a semantic analyzer of routing structures, (ii) its output is systematically translated into structured ruin-and-recreate operators within a BRKGA framework, and (iii) this integration is evaluated on the HHCRSP benchmark.

2 Related Work

The core of modern optimization research is increasingly centered on the use of LLMs as autonomous agents for algorithm design, as autonomous solvers for complex problems, or as active heuristics inside optimization routines. This transition is part of a broader breakthrough in LLM releases: as noted by Cascella et al. [5], LLMs have expanded their role to sophisticated diagnostic and clinical decision support systems. According to the structured review by Ghanbarzadeh and Mirjalili [9], LLMs have evolved into active components for problem formulation, parameter tuning, and the innovation of entirely new search strategies. This shift is remarked in the LLM-EA paradigm, where Yu and Liu [22] describe the synergy between LLMs as dynamic strategy generators and Evolutionary Algorithms (EAs) for efficient exploration.

Several frameworks have recently explored LLM-assisted optimization, including LLaMEA [18], ReEvo [20], FunSearch [17], LLM-LNS [21], and LLMO [24], demonstrating the potential of LLMs for heuristic design, neighborhood generation, and optimization.

In healthcare, LLMs have been applied to resource allocation [19] and patient care [4], although their integration into optimization pipelines remains limited.

Another key development is TextGrad [23], which allows LLMs to self-refine via textual gradients, potentially providing them with an integrated optimization pipeline.

While these experiments are promising, GAs remain a reliable, mathematically validated approach for a search process that ensures both efficiency and diversity, as discussed in the theoretical reviews by Katoch et al. [10] and Alhijawi and Awajan [1].

Regarding the HHCRSP, it can be defined as a complex generalization of a vehicle routing problem. Early foundational work by Cheng and Rich [6] provided a Mixed-Integer Linear Programming (MILP) model that accounts for staff skills and patient compatibility, using simple heuristics for improvement. Mankowska et al. [16] provided a seminal definition of the problem by introducing synchronization and precedence constraints, proposing an Adaptive Variable Neighborhood Search (AVNS) to generate solutions for realistic instances. To further increase the realism of the model, Liu et al. [13] expanded the problem to include practical constraints such as synchronized visits and mandatory lunch breaks for caregivers, utilizing a hybrid metaheuristic approach. Additionally, uncertainty in travel and care times was addressed by Bazirha et al. [3], who proposed a stochastic programming model with recourse to maintain schedule adherence despite stochastic delays.

To address the computational intractability of the various constraints of the problem, Kummer Neto et al. [12] proposed a fix-and-optimize metaheuristic that iteratively optimizes small subsets of routes. Finally, Kummer et al. [11] developed the BRKGA-MP-IPR approach (from now on, referred to as BRKGA for short) that separates the solution structure from the search space using chromosomes and a decoder that builds solutions from them. Our work builds directly upon these structures, taking this last formulation cited as a starting point to explore how LLMs can further enhance the optimization processes in the case of HHCRSP.

3 Background

To establish the necessary background for this study, we first overview the problem formulation: the HHCRSP can be viewed as an extension of the classic Vehicle Routing Problem with Time Windows (VRPTW), and we adopt the specific problem formulation and benchmark dataset by Mankowska et al. [16], comprising 70 instances grouped into seven subsets (Table 1).

Approximately 70% of patients require a single service, while the remaining 30% require double services involving either synchronization or precedence constraints. These double services introduce significant coordination challenges through two types of temporal constraints: in each instance, 15% of the patients requires simultaneous visits, where two caregivers must be present at the same time, and another 15% of the visits are linked by precedence, where one service must be completed before the next can begin. All the coordinates of the problem, so those of the central depot and the

Table 1: Characteristics of the benchmarking dataset [16]

Subset	Instances	Patients per instance ($ C $)	Caregivers per instance ($ \mathcal{V} $)
A (from inst. 10_1 to 10_10)	10	10	3
B (from inst. 25_1 to 25_10)	10	25	5
C (from inst. 50_1 to 50_10)	10	50	10
D (from inst. 75_1 to 75_10)	10	75	15
E (from inst. 100_1 to 100_10)	10	100	20
F (from inst. 200_1 to 200_10)	10	200	30
G (from inst. 300_1 to 300_10)	10	300	40
Total	70	-	-

patients' locations, are generated within a fixed 100×100 grid, regardless of the number of patients, with travel times assumed to be symmetric. Each patient is assigned a soft time window of two hours within a total daily planning horizon of 10 hours, and caregivers are divided into two equal groups, with each caregiver qualified to perform a specific set of services to ensure that every task can be done by at least one qualified caregiver. Finally, the quality of a schedule is evaluated with a multi-objective cost function that minimizes a weighted sum of the total travel distance, the total accumulated tardiness (defined as the delay when a visit begins later than the patient's preferred time slot) and the maximum observed tardiness, ensuring that the solution balances efficiency with fair service delivery for the patients. The remainder of this section provides the necessary theoretical foundations: Section 3.1 formally details the HHCRSP, including the objective function and the specific constraints. Then, Section 3.2 describes in detail the BRKGA that serves as the structural baseline for our hybrid framework.

3.1 Home Healthcare Routing and Scheduling Problem

The HHCRSP is modeled as a minimization problem where the goal is to organize home visits for a diverse team of caregivers, ensuring they reach patients spread across the area within their scheduled appointment times. Formally, it can be defined on a graph where the node set $C^0 = C \cup \{0\}$ includes a set of patients C and a central office (depot) denoted by 0, where the task is coordinating the fleet of caregivers $\mathcal{V}$, where individual qualifications are defined by a

binary parameter a_{vs} , which equals 1 if caregiver $v \in \mathcal{V}$ is qualified to perform service type $s \in \mathcal{S}$.

The core complexity of the formulation considered lies in the division of patients based on service requirements: the set of patients C can be partitioned into those requiring a *single service* (C^s) and those requiring a *double service* (C^d) and, in addition, double services enforce temporal dependencies, where some double services need to be performed simultaneously and others are linked by precedence, according to the following categorization.

- **Simultaneous Services (C^{sim})**: two caregivers must start their respective service operations at the exact same time. Mathematically, the time distance between the start of the two services is constrained by $\delta^{min} = \delta^{max} = 0$.
- **Services with Precedence (C^{prec})**: these services require a temporal separation. The second operation must start within a specific time lag $[\delta^{min}, \delta^{max}]$ after the start of the first operation, where $\delta^{min} > 0$.

Time slots are modeled via time windows $[e_i, l_i]$ for each patient i . In this specific formulation, the earliest start time e_i acts as a hard constraint; if a caregiver arrives before e_i , they must wait until the window opens. Conversely, the latest start time l_i is treated as a soft constraint, so services starting after l_i are feasible but incur a tardiness penalty z_{is} . In addition, caregivers must return to the depot after finishing their visits, with each patient node being visited only once per route, and a caregiver cannot perform two consecutive services for the same patient. The objective function is a weighted sum designed to optimize three performance measures simultaneously. As mentioned earlier, the optimization goal is to minimize Z :

$$\min Z = \lambda_1 D + \lambda_2 T + \lambda_3 T^{max} \quad (1)$$

where:

- D is the total distance traveled by the fleet.
- T is the total tardiness, representing the aggregate lateness of all services relative to the patients' preferred time windows.
- T^{max} is the maximum tardiness, representing the single highest delay observed in the schedule among all patients.

The inclusion of the T^{max} term is crucial to ensure fairness, preventing scenarios where a minority of patients suffer extreme delays. The weights $\lambda_1, \lambda_2, \lambda_3$ allow the planner to flexibly prioritize between minimizing distance traveled and service quality.

3.2 Biased Random-Key Genetic Algorithm

To design an effective solution approach for HHCRSP, it is essential to first give a clear understanding of the metaheuristic framework used as a baseline. At its core, a GA [1] is a population-based search method for optimization where a population, defined as a set of candidate solutions to the problem, is initialized and evolved within the algorithm to find the best solutions. Each of these candidates, known as a chromosome or individual, consists of a set of genes that represent specific decision variables. Usually, the GA fills the initial population with random solutions and then develops the optimal one from one generation to the next through the use of the following genetic operators [1]:

- **Selection:** this process identifies the "fittest" individuals from the current population to act as parents for the next generation.
- **Crossover:** this operator generates offspring by inheriting each gene from the selected parents according to a specific probability. It aims to improve individual quality by combining the best traits of previous candidates.
- **Mutation:** this operator introduces random variations into individual's genes. Its primary role is to maintain population diversity and prevent the algorithm from becoming trapped in a local optimum.

These operators work together to refine the population, with crossover and mutation typically performed according to fixed probabilities, until a termination condition is met, and the best solution among all is considered the final solution of the modeled problem. The BRKGA brings various enhancements to the classical GA approach starting from the selection process: the population is partitioned into an elite set and a non-elite set, where the elite set is copied directly into the next generation. The remaining chromosomes of the next generation are created by mating individuals selected from both sets and, in our case, a loginverse bias function is used to weight the influence of the fittest parents during gene inheritance. All these strategies keep both diversity and high quality genetic material. In detail, BRKGA explores the problem's solution space indirectly through an auxiliary search space of random keys, represented as vectors of real values in the interval $[0, 1)$. So, each individual in the population is represented by a chromosome of length $|C| + 2$, where the first $|C|$ keys determine the priority or sequence in which patients are inserted into the schedule, thus, each of the first $|C|$ keys corresponds to a patient, and the associated key values determine the insertion priority. The core of the algorithm is a heuristic-powered decoder that transforms these random keys into a feasible HHC schedule using a greedy cheapest-insertion strategy. For each task in the sequence, the decoder evaluates all possible pairs of qualified caregivers and assigns the patient to the combination that minimizes the weighted multi-objective function defined in Eq. 1). In the implementation of this model, the multiplicative factors for the objective function are set as $\lambda_1 = \lambda_2 = \lambda_3 = \frac{1}{3}$ to provide equal weight to each criterion.

The final two chromosome keys activate the convex-hull and workload-balancing heuristics introduced by Kummer et al. [11]. The mathematical model behind this approach ensures that each vehicle v departs from and returns to the depot, and that all other constraints are respected. An important technical component of the BRKGA is the specialized logic used to handle route interdependency constraints. When evaluating an insertion for simultaneous services, the algorithm calculates the start time as the maximum arrival time of both caregivers to ensure both are present before processing begins:

$$t_{iv_1s_1} = t_{iv_2s_2} = \max(\text{arrival}_{v_1}, \text{arrival}_{v_2}) \quad (2)$$

For precedence constraints, the decoder enforces minimum ($\delta_i^{\min}$) and maximum ($\delta_i^{\max}$) separation times, mathematically defined by the following inequalities:

$$t_{iv_2s_2} - t_{iv_1s_1} \geq \delta_i^{\min} \text{ and } t_{iv_2s_2} - t_{iv_1s_1} \leq \delta_i^{\max} \quad (3)$$

If the maximum separation time is violated because the second caregiver arrives too late, the algorithm introduces a delay to the first caregiver's start time to maintain feasibility, calculated as:

$$t_{iv_1s_1} = \max(\text{arrival}_{v_1}, \text{arrival}_{v_2} - \delta_i^{\max}) \quad (4)$$

Furthermore, IPR mechanism periodically refines elite solutions across independent populations (islands) by exploring the path between a base and a guide chromosome: by swapping keys to align their visit orders, it generates intermediate offspring that, if any of them outperforms the base chromosome, it replaces the base solution in the population. Also, an immigration mechanism is used, where a fixed number of chromosomes are periodically transferred from one island to another. The search process stops based on a dynamic stopping criterion proportional to the instance size. Specifically, the genetic algorithm terminates the search after completing $2 \times |C|$ generations without improvement in the best fitness value found.

3.3 TextGrad

TextGrad [23] is an LLM-based algorithm for optimization that we used to solve the HHCRSP problem, in such a way to build a pure LLM-only solver. Specifically, TextGrad is an approach that introduces a paradigm shift in how compound LLM systems are optimized: while traditional deep learning relies on numerical gradients and backpropagation to update model weights, TextGrad implements Automatic Differentiation via Text, treating the entire system as a computation graph where variables can be text snippets (solutions, prompts, or code) and "gradients" are structured feedback provided by LLMs. In our solver, this mechanism is integrated into the iterative loop to refine the LLM-generated solutions to the HHCRSP, that follows these steps:

- **Forward pass:** the LLM receives a prompt containing a JSON version of the instance and directives that explain the task, so it generates a candidate schedule. This solution is then evaluated by a validator module, which identifies constraint violations and calculates the objective cost.
- **Backward Pass:** the validator's feedback and the current solution are passed to TextGrad through a LLM call, that performs a "backward pass" to generate backward text. This text consists of concrete operational hints, such as reassigning patients, adjusting visit orders, or modifying caregiver assignments.
- **Iterative Refinement:** these textual gradients (that is, backward texts) are appended to the validator's feedback provided to the LLM in the next iteration. This provides the model with a clear direction for improvement.

After a fixed number of optimization steps, the best feasible solution found is retained. In our case, the number of steps is set to 10, and the LLM used is Qwen 3 30B [14] served locally by LM Studio¹. These settings provide a reasonable trade-off between reasoning capability and computational cost. However, the limitations met by this standalone approach during the experimental phase led to a shift in research toward a hybrid model that combines the reasoning potential of the LLM with the reliability of the BRKGA.

¹LM Studio is a desktop application for running local LLMs, available at <https://lmstudio.ai/>

Both the details regarding the hybrid approach and the limitations of the standalone solver are discussed in dedicated sections.

4 The Hybrid Approach (LLM-BRKGa)

In this section we describe the proposed LLM-BRKGa hybrid framework, which integrates instance-aware semantic guidance from an LLM with the robust exploration capabilities of the BRKGa. This approach leverages the ability of LLMs to capture coarse-grained spatial patterns that are not explicitly modeled in the objective function.

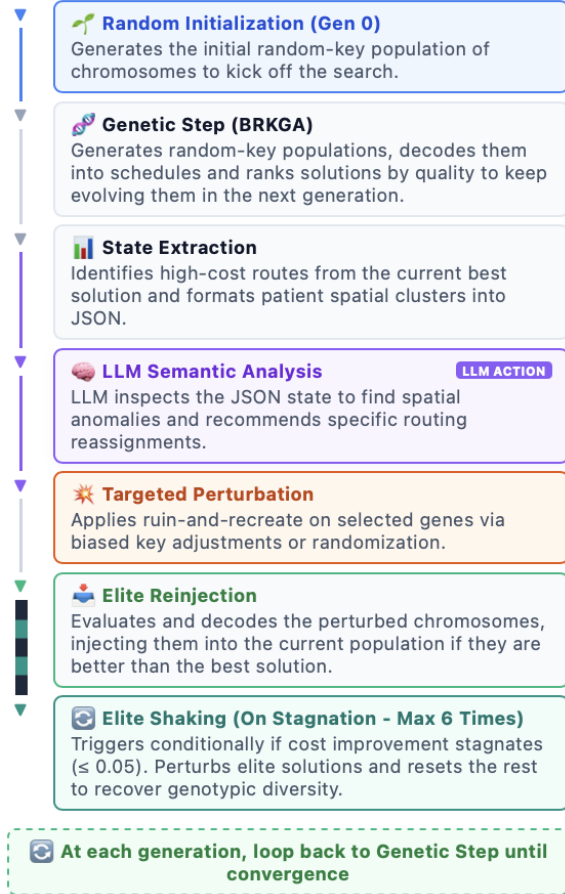


Figure 1: S

chematic workflow of the LLM-BRKGa hybrid optimization loop.

As shown in Figure 1, this architecture uses BRKGa as its structural foundation, evolving a population of chromosomes initialized with random keys to solve the HHCRSP. At each generation, the LLM acts as a high-level semantic director. It receives and inspects a small, instance-specific summary of the current best solution, which includes:

- A subset of caregiver routes (ordered visit sequences) that contribute the most to the solution cost. This requires computing the contribution of each route to the overall solution cost before calling the LLM.

- Some proximity groups of patients, to give the model an idea of how the patients are placed in the area.

By analyzing this data, the model identifies spatial outliers, namely small but impactful substructures where the current scheduling may be sub-optimal. Specifically, the LLM we use for this task is again the Qwen 3 30B model² that runs locally through LM Studio, which again allows optimizing resources: Qwen 3 provides sufficient reasoning capacity to capture complex scheduling patterns but also efficient enough to be called many times without making the total execution time prohibitive, and to encourage diverse yet meaningful guidance, we use the following settings:

- *Temperature*: we choose to use a moderate temperature value of 0.4 to provide a balance between exploration and consistency, which helps the mechanism avoid getting stuck in local optima while tending to suggest variable reinsertion ideas, to a certain extent.
- *Top-P Sampling*: a value of 0.95 for this setting, where P refers to the cumulative probability of candidate tokens, ensures the model stays focused and keeps its suggestions clear and useful.

The rationale is that long travel distances often arise from suboptimal spatial assignments that conventional search operators may fail to recognize. As a result, they often fail to identify spatial clusters of patients that could be treated consecutively to reduce the distance traveled by caregivers when moving from one patient to another, constraints permitting. During the *LLM Semantic Analysis* in Figure 1, the LLM is asked to generate a structured recommendation that specifies a limited subset of patients whose assignment needs to be changed, defining a space for targeted optimization. In the following step, an optimizer module along with the original BRKGa’s decoder translates the LLM’s recommendation into a targeted *ruin-and-recreate* operation. This process generates a family of perturbed candidate chromosomes that originate from the chromosome of the current best solution, by systematically manipulating the genes associated with the removed patients. These manipulations include:

- *Allele Biasing*: adjusting keys to favor reinsertion toward the start or the end of a route, once the solution is decoded, starting from the chromosome. A key value close to 1 biases insertion toward earlier positions in the route, while a value close to 0 promotes insertion toward the end.
- *Randomization*: assigning randomized values to explore entirely new positions in the chromosome.
- *Combinatorial Variants*: forming global variants that test different combinations of these reinsertion patterns.

These perturbed chromosomes are then injected into the next genetic population if the solution decoded from them is better than the current best solution (according to the exact same multi-objective function used in BRKGa); specifically, they are injected in place of the worst chromosomes of the new generation (that is, those leading to the most costly solutions). The control keys governing the convex-hull and workload-balancing heuristics remain unchanged. This ensures the system can preserve its original functioning, except for the injection, at each generation, of a family of perturbed

²Version: qwen3-30b-a3b-2507.

chromosomes. The LLM-BRKGa approach also implements an additional elite-group shaking operator that is invoked when the search approaches a stagnation threshold. Rather than producing local offspring as it happens with LLM-driven perturbations during each generation, this operator perturbs the elite set directly: a controlled subset of elite chromosomes is modified according to a configurable intensity and move type (e.g., allele inversion, random reassignment, or adjacent swaps, which is the move type adopted in our HHCRSP experiments), while the remainder of the population outside the elite group is reinitialized with fresh random keys to restore genotypic diversity. After perturbation the entire population is re-evaluated, and the procedure may also optionally reinject a historically best chromosome to preserve high-quality genetic material. When the algorithm observes sustained stagnation and the maximum permitted number of shakes has been already performed, then the stopping criterion is triggered and the execution of the algorithm stops after $2 \times |C|$ generations without improvement. However, it is important to note that, in both the shaking phase and the stopping criterion, any cost improvement smaller than or equal to 0.05 is not taken into account. This prevents the algorithm from running additional generations when they are not truly necessary to obtain a better solution. Consequently, shaking or stopping criterion are triggered only after a certain number of generations without a significant improvement, rather than after every strictly positive but negligible cost decrease.

5 Experimental Evaluation

In this section, we present our computational experiments to compare the different methods explained before: the native LLM-based optimization approach (i.e. *TextGrad*), the original *BRKGa* by Kummer et al. [11], and our proposed hybrid approach (*LLM-BRKGa*) that combines LLMs with GAs. The parameter configuration used for both *BRKGa* and *LLM-BRKGa* was obtained using *irace* [15] on subset G, following the original *BRKGa* implementation, with the goal of identifying a robust configuration for large-scale instances. The shared *BRKGa* configuration is summarized below:

- *Population & generations*: 1,200 individuals, with 20.64% preserved as elite and 15% random mutants injected per generation. The maximum number of generations is 3000.
- *Mating*: new offspring generated from 3 parents (2 elite, 1 non-elite) using a loginverse bias function.
- *Islands & migration*: 3 independent populations with 10 immigrants exchanged every 125 generations.
- *IPR*: every 49 generations, 95 of the best pairs are mixed to find better paths, but only if they are different enough to avoid redundancy.

This configuration is also adopted in the proposed hybrid LLM-BRKGa approach for all parameters shared with *BRKGa*, ensuring a fair comparison. The following parameters are unique to the LLM-BRKGa and were determined empirically for each subset (Table 2).

- *Maximum Shakes* (–maxshakes): this is the total number of times the algorithm is allowed to perform a shake during a single run. This value is set to 6 for all subsets.

- *Shake Trigger* (–shakettrigger): this defines the threshold of generations without any significant improvement that triggers a shake.
- *Shake Intensity* (–shakeint): this represents the strength of the perturbation, or how many changes are applied to each of the elite chromosomes during a shake, where change is a random swap between two alleles of a chromosome.

Table 2: Parameter configuration for the shaking operator in LLM-BRKGa.

Subset	–shakettrigger	–shakeint
A	3	5
B	12	20
C	10	45
D	29	27
E	45	27
F	45	40
G	45	40

The primary metric for solution quality is the average cost over 10 runs (referred to as *cost*), which represents the mean value of the multi-objective function $f(x) = \lambda_1 D + \lambda_2 T + \lambda_3 T^{max}$. This metric is complemented by the standard deviation (*SD*)[2], which measures the stability of the algorithm, where a lower value indicates that the approach consistently converges to high-quality regions of the solution space; and by the average generation over 10 runs in which the best solution was first identified (referred to as *best g*), revealing the efficiency of the approach in helping the algorithm to converge. We evaluate the ability to converge by tracking the best generations instead of the actual computation time, since the integration of LLM API calls leads to longer computation times. Although the integration of LLM calls substantially increases the overall execution time with respect to the original *BRKGa*, the objective of this work is to assess the effectiveness of LLM-guided search in improving optimization quality rather than computational efficiency. As partially mentioned before, all the computational experiments, whose results are reported in Table 3 and Table 4, were conducted using the Qwen 3 30B model as the LLM engine within a machine equipped with an Apple M1 Max processor³. Comparing *BRKGa* and *LLM-BRKGa* reveals the efficacy of integrating semantic guidance into genetic search. While both perform similarly on small-scale scenarios (Subsets A and B, Table 3), the hybrid approach’s benefits grow with instance size. On the largest instances (Subset G, Table 4), it achieved a significant reduction in average multi-objective cost, from 2397.87 to 2065.01, alongside a stabilization of the search process, with standard deviation dropping from 502.22 to 76.51. This indicates that LLM-driven directives help reach better, more consistent solutions, likely because larger instances contain more spatial structure for the LLM to exploit.

³Mac Studio equipped with an Apple M1 Max chip (10-core CPU, up to 3.2 GHz; 24-core GPU), 32 GB of unified memory, running macOS Tahoe (version 26.3)

Table 3: Complete comparison between TextGrad, BRKGA, and LLM-BRKGA (Part I: Subsets A-D)

Inst.	TextGrad	BRKGA		LLM-BRKGA	
	cost $\pm$ SD	cost $\pm$ SD	best g $\pm$ SD	cost $\pm$ SD	best g $\pm$ SD
10_1	1,010.67 $\pm$ 722.34	218.20 $\pm$ 0.00	10.00 $\pm$ 1.00	218.20 $\pm$ 0.00	7.00 $\pm$ 3.00
10_2	1,209.00 $\pm$ 505.86	246.60 $\pm$ 0.00	6.00 $\pm$ 3.00	246.60 $\pm$ 0.00	10.00 $\pm$ 4.00
10_3	703.00 $\pm$ 98.50	305.90 $\pm$ 0.00	7.00 $\pm$ 1.00	309.90 $\pm$ 0.00	5.00 $\pm$ 2.00
10_4	916.00 $\pm$ 100.56	186.90 $\pm$ 0.00	6.00 $\pm$ 2.00	186.90 $\pm$ 0.00	5.00 $\pm$ 1.00
10_5	569.00 $\pm$ 111.08	189.50 $\pm$ 0.00	5.00 $\pm$ 1.00	189.50 $\pm$ 0.00	8.00 $\pm$ 2.00
10_6	763.00 $\pm$ 97.86	200.20 $\pm$ 0.04	6.00 $\pm$ 1.00	200.10 $\pm$ 0.00	8.00 $\pm$ 3.00
10_7	1,306.00 $\pm$ 102.75	225.40 $\pm$ 0.00	5.00 $\pm$ 0.00	225.40 $\pm$ 0.00	3.00 $\pm$ 2.00
10_8	744.67 $\pm$ 56.04	232.00 $\pm$ 0.00	7.00 $\pm$ 0.00	232.00 $\pm$ 0.00	6.00 $\pm$ 1.00
10_9	1,104.00 $\pm$ 376.88	222.30 $\pm$ 0.00	8.00 $\pm$ 0.00	222.30 $\pm$ 0.00	7.00 $\pm$ 1.00
10_10	915.00 $\pm$ 124.52	225.00 $\pm$ 0.00	2.00 $\pm$ 0.00	225.00 $\pm$ 0.00	4.00 $\pm$ 2.00
AVG A	924.03 $\pm$ 229.64	225.20 $\pm$ 0.01	6.20 $\pm$ 0.90	225.59 $\pm$ 0.00	6.30 $\pm$ 2.10
25_1	592.00 $\pm$ 125.08	430.28 $\pm$ 2.96	46.10 $\pm$ 9.60	431.49 $\pm$ 3.09	61.00 $\pm$ 17.32
25_2	582.00 $\pm$ 97.43	483.69 $\pm$ 2.71	42.00 $\pm$ 8.40	483.44 $\pm$ 3.28	48.70 $\pm$ 15.34
25_3	6,470.00 $\pm$ 3,299.07	405.73 $\pm$ 5.30	48.00 $\pm$ 5.59	405.77 $\pm$ 5.85	55.60 $\pm$ 18.06
25_4	8,060.00 $\pm$ 4,588.00	431.82 $\pm$ 10.35	62.22 $\pm$ 18.48	428.86 $\pm$ 11.55	73.40 $\pm$ 17.18
25_5	3,690.00 $\pm$ 536.90	373.35 $\pm$ 3.25	49.10 $\pm$ 11.09	375.43 $\pm$ 2.44	66.11 $\pm$ 12.06
25_6	4,784.00 $\pm$ 2,354.71	471.46 $\pm$ 2.02	64.20 $\pm$ 13.44	471.77 $\pm$ 5.59	65.90 $\pm$ 19.47
25_7	684.00 $\pm$ 56.02	330.92 $\pm$ 4.78	54.00 $\pm$ 9.48	328.96 $\pm$ 0.88	63.77 $\pm$ 13.37
25_8	9,115.00 $\pm$ 3,106.97	358.28 $\pm$ 0.97	54.00 $\pm$ 13.27	357.78 $\pm$ 0.31	64.00 $\pm$ 16.22
25_9	10,284.00 $\pm$ 6,682.45	409.60 $\pm$ 5.56	56.00 $\pm$ 16.35	408.52 $\pm$ 3.98	72.60 $\pm$ 23.30
25_10	7,123.00 $\pm$ 5,478.03	472.28 $\pm$ 2.41	43.50 $\pm$ 17.13	470.70 $\pm$ 3.13	59.00 $\pm$ 14.93
AVG B	5,139.40 $\pm$ 2,332.76	416.74 $\pm$ 4.03	51.91 $\pm$ 12.28	416.27 $\pm$ 4.01	63.01 $\pm$ 16.73
50_1	14,845.33 $\pm$ 2,767.01	992.57 $\pm$ 4.40	122.28 $\pm$ 9.30	982.10 $\pm$ 13.50	105.00 $\pm$ 8.74
50_2	8,771.33 $\pm$ 2,408.08	627.75 $\pm$ 13.31	128.50 $\pm$ 22.60	620.63 $\pm$ 24.60	116.30 $\pm$ 26.65
50_3	11,548.00 $\pm$ 7,195.93	567.86 $\pm$ 6.61	144.55 $\pm$ 27.21	576.36 $\pm$ 21.76	124.60 $\pm$ 30.45
50_4	15,769.67 $\pm$ 3,896.12	557.74 $\pm$ 52.86	155.77 $\pm$ 40.23	607.92 $\pm$ 77.19	125.00 $\pm$ 21.21
50_5	9,388.67 $\pm$ 1,457.90	749.79 $\pm$ 16.75	146.10 $\pm$ 27.18	748.84 $\pm$ 26.15	123.00 $\pm$ 20.40
50_6	18,424.00 $\pm$ 4,991.06	875.69 $\pm$ 25.16	185.50 $\pm$ 54.49	881.32 $\pm$ 19.00	134.00 $\pm$ 30.00
50_7	25,927.33 $\pm$ 8,757.78	554.57 $\pm$ 5.44	119.55 $\pm$ 22.36	550.00 $\pm$ 10.00	119.50 $\pm$ 16.39
50_8	10,784.33 $\pm$ 3,247.40	494.38 $\pm$ 10.43	141.60 $\pm$ 43.66	493.35 $\pm$ 6.49	125.40 $\pm$ 20.20
50_9	10,888.00 $\pm$ 2,217.75	562.05 $\pm$ 10.80	122.00 $\pm$ 17.30	573.40 $\pm$ 15.20	108.70 $\pm$ 22.26
50_10	10,885.67 $\pm$ 1,567.86	690.36 $\pm$ 14.68	122.50 $\pm$ 21.30	688.40 $\pm$ 22.62	127.00 $\pm$ 23.60
AVG C	13,723.23 $\pm$ 3,753.69	667.28 $\pm$ 16.04	138.84 $\pm$ 30.56	672.23 $\pm$ 24.15	120.85 $\pm$ 22.04
75_1	–	1,222.40 $\pm$ 30.50	234.50 $\pm$ 58.20	1,215.30 $\pm$ 5.82	170.40 $\pm$ 35.20
75_2	–	762.95 $\pm$ 20.66	190.20 $\pm$ 48.49	768.58 $\pm$ 23.31	253.85 $\pm$ 43.10
75_3	–	684.92 $\pm$ 12.53	198.30 $\pm$ 58.77	686.14 $\pm$ 18.26	250.00 $\pm$ 42.11
75_4	–	865.80 $\pm$ 10.78	189.50 $\pm$ 64.34	868.82 $\pm$ 9.77	204.40 $\pm$ 52.64
75_5	–	768.61 $\pm$ 24.68	240.00 $\pm$ 66.44	767.70 $\pm$ 22.68	251.60 $\pm$ 39.64
75_6	–	792.78 $\pm$ 12.10	209.90 $\pm$ 43.69	788.73 $\pm$ 14.44	244.50 $\pm$ 51.20
75_7	–	616.40 $\pm$ 15.31	228.40 $\pm$ 42.50	621.25 $\pm$ 15.60	242.80 $\pm$ 56.70
75_8	–	711.58 $\pm$ 8.46	205.80 $\pm$ 50.85	707.93 $\pm$ 11.01	219.70 $\pm$ 29.60
75_9	–	788.52 $\pm$ 45.37	154.80 $\pm$ 99.97	757.57 $\pm$ 17.57	253.60 $\pm$ 46.15
75_10	–	1,930.40 $\pm$ 345.85	251.50 $\pm$ 87.95	1,842.13 $\pm$ 271.29	262.80 $\pm$ 48.49
AVG D	–	914.44 $\pm$ 52.62	210.29 $\pm$ 62.12	902.42 $\pm$ 40.97	235.37 $\pm$ 44.48

Table 4: Complete comparison between TextGrad, BRKGA, and LLM-BRKGA (Part II: Subsets E-G)

Inst.	TextGrad	BRKGA		LLM-BRKGA	
	cost $\pm$ SD	cost $\pm$ SD	best g $\pm$ SD	cost $\pm$ SD	best g $\pm$ SD
100_1	–	1,457.91 $\pm$ 72.14	388.88 $\pm$ 109.30	1,457.04 $\pm$ 62.79	489.40 $\pm$ 67.55
100_2	–	945.02 $\pm$ 24.52	354.70 $\pm$ 68.17	949.19 $\pm$ 32.85	455.00 $\pm$ 45.08
100_3	–	869.58 $\pm$ 19.90	298.70 $\pm$ 68.45	869.37 $\pm$ 15.30	387.00 $\pm$ 88.08
100_4	–	778.32 $\pm$ 13.19	314.60 $\pm$ 80.94	792.97 $\pm$ 33.61	409.80 $\pm$ 68.99
100_5	–	932.87 $\pm$ 15.90	297.22 $\pm$ 77.17	921.99 $\pm$ 21.73	397.50 $\pm$ 54.00
100_6	–	885.73 $\pm$ 17.59	319.00 $\pm$ 73.86	875.07 $\pm$ 29.39	446.80 $\pm$ 68.59
100_7	–	762.79 $\pm$ 14.88	279.55 $\pm$ 55.58	758.59 $\pm$ 8.97	370.40 $\pm$ 75.12
100_8	–	794.59 $\pm$ 10.13	345.60 $\pm$ 99.32	799.48 $\pm$ 11.75	388.70 $\pm$ 78.27
100_9	–	1,045.14 $\pm$ 37.02	318.60 $\pm$ 78.86	1,034.78 $\pm$ 37.36	414.10 $\pm$ 112.51
100_10	–	903.32 $\pm$ 20.65	318.33 $\pm$ 72.14	909.56 $\pm$ 19.43	430.40 $\pm$ 132.86
AVG E	–	937.53 $\pm$ 24.59	323.52 $\pm$ 78.38	936.80 $\pm$ 27.32	418.91 $\pm$ 79.11
200_1	–	1,748.23 $\pm$ 34.45	690.20 $\pm$ 221.70	1,705.35 $\pm$ 69.84	725.00 $\pm$ 125.42
200_2	–	1,626.25 $\pm$ 63.83	758.30 $\pm$ 170.67	1,600.37 $\pm$ 37.97	732.10 $\pm$ 116.00
200_3	–	1,469.75 $\pm$ 58.49	624.80 $\pm$ 117.07	1,462.90 $\pm$ 86.82	627.40 $\pm$ 123.00
200_4	–	1,331.41 $\pm$ 43.03	787.80 $\pm$ 240.02	1,397.61 $\pm$ 69.63	623.22 $\pm$ 155.13
200_5	–	1,605.48 $\pm$ 55.83	736.70 $\pm$ 197.34	1,622.85 $\pm$ 48.40	690.70 $\pm$ 203.30
200_6	–	1,630.10 $\pm$ 38.04	676.60 $\pm$ 141.07	1,580.27 $\pm$ 52.48	617.50 $\pm$ 122.90
200_7	–	1,312.69 $\pm$ 39.11	702.40 $\pm$ 136.33	1,301.82 $\pm$ 22.77	681.50 $\pm$ 60.37
200_8	–	1,400.66 $\pm$ 70.00	706.42 $\pm$ 108.90	1,422.50 $\pm$ 59.64	634.70 $\pm$ 88.83
200_9	–	1,593.32 $\pm$ 32.75	694.40 $\pm$ 92.44	1,561.70 $\pm$ 36.87	766.42 $\pm$ 165.01
200_10	–	1,832.34 $\pm$ 25.77	810.89 $\pm$ 33.62	1,796.20 $\pm$ 30.77	765.89 $\pm$ 88.34
AVG F	–	1,555.02 $\pm$ 46.13	718.85 $\pm$ 145.92	1,545.16 $\pm$ 51.52	686.44 $\pm$ 124.83
300_1	–	2,125.23 $\pm$ 75.24	1,030.00 $\pm$ 216.00	2,110.54 $\pm$ 53.60	1,024.27 $\pm$ 142.32
300_2	–	2,190.17 $\pm$ 80.53	1,102.50 $\pm$ 374.88	2,143.51 $\pm$ 110.64	1,061.40 $\pm$ 246.84
300_3	–	1,731.38 $\pm$ 22.40	1,037.10 $\pm$ 122.00	1,846.18 $\pm$ 70.37	926.70 $\pm$ 149.03
300_4	–	1,844.51 $\pm$ 33.03	943.20 $\pm$ 190.23	1,780.86 $\pm$ 79.53	1,106.50 $\pm$ 307.84
300_5	–	2,028.52 $\pm$ 26.85	1,114.50 $\pm$ 224.40	2,102.03 $\pm$ 109.39	1,027.50 $\pm$ 244.48
300_6	–	2,058.04 $\pm$ 65.67	1,041.00 $\pm$ 367.85	2,114.61 $\pm$ 48.33	985.20 $\pm$ 180.56
300_7	–	1,961.24 $\pm$ 92.76	1,035.30 $\pm$ 404.43	1,952.12 $\pm$ 49.33	915.30 $\pm$ 149.53
300_8	–	1,822.47 $\pm$ 47.04	976.90 $\pm$ 160.35	1,801.41 $\pm$ 56.58	906.70 $\pm$ 189.74
300_9	–	5,905.85 $\pm$ 4,503.20	902.40 $\pm$ 354.10	2,527.38 $\pm$ 131.64	1,082.55 $\pm$ 324.58
300_10	–	2,311.32 $\pm$ 75.51	986.60 $\pm$ 260.26	2,271.45 $\pm$ 55.65	954.80 $\pm$ 230.90
AVG G	–	2,397.87 $\pm$ 502.22	1,016.95 $\pm$ 267.45	2,065.01 $\pm$ 76.51	999.09 $\pm$ 216.58

Furthermore, on the largest subsets the hybrid model also shows competitive or improved convergence behavior in terms of the generation at which the best solution is first identified. In Subset G, the optimal solution was identified, on average, at generation 999.09, slightly earlier than the 1016.95 generations required by the baseline; in addition, a similar improvement was observed for Subset F. This suggests that the LLM-based guidance is particularly beneficial on larger instances, and this aligns with the expected behavior of LLMs in capturing high-level structure: within the limits of their token processing capacity, these models should be able to provide a more significant contribution as the complexity and scale of the problem increase. Regarding the results obtained by TextGrad, they show that the approach struggles significantly to comprehend and process the fully combinatorial nature of the problem, so it produces results that are far worse than those obtained in the other experimental settings. Moreover, it was possible to conduct tests only up to Subset C, as larger instances could not be solved due to exceeding the model’s token limits or requiring

computational times that are not viable (that is, on the order of two hours per generation, meaning two hours for each optimization step with Textgrad, which amounts to approximately 20 hours per independent run, given 10 optimization steps). It should also be emphasized that the standalone LLM does not consistently yield valid solutions. Therefore, the TextGrad averages reported in the tables are computed only over the subset of runs that produced valid solutions, and should therefore not be directly compared with the BRKGA-based methods without considering this limitation. As a consequence, the average values reported for each instance are not computed consistently over 10 independent runs, but only over those runs, among the 10 attempted, that produced a valid solution. In most cases, the number of independent runs that return a valid solution ranges between 5 and 8. These structural limitations and the significant lack of reliability in standalone reasoning further validate the necessity of a hybrid approach, where the LLM’s high-level semantic capabilities are coupled with the BRKGA’s robust mechanism to ensure both feasibility and high quality performance.

Although a full ablation study, including isolating the effects of the LLM-guided perturbations and the elite-shaking operator, is left for future work, the consistent improvements observed on larger instances, where random perturbations alone are less effective, suggest that semantic guidance plays a key role.

6 Conclusion

In this work, we investigated the application of LLMs to solve the HHCRSP, a complex combinatorial optimization problem in health-care. We compared the performance of three different strategies: the BRKGA proposed in the literature, a native LLM-based optimization solution based on TextGrad, and our proposed hybrid approach (LLM-BRKGA) that uses the LLM as a high-level director for local search. Our findings demonstrate that while the LLM alone (TextGrad) is unable to solve the problem effectively due to the complexity of the problem and further limitations in scalability, the LLM-BRKGA shows meaningful improvements over the BRKGA baseline on several medium-to-large and large instances. Overall, the results indicate that LLM-guided perturbations can improve solution quality on medium-to-large instances, suggesting that LLMs are promising components of hybrid optimization frameworks. The findings obtained with LLM-BRKGA suggest that delegating feasibility management and evolutionary search to a GA, while using an LLM for strategic guidance, represents a promising design pattern for other complex combinatorial optimization problems. Future work will focus on validating the proposed framework in real-world home healthcare settings, where operational constraints such as data quality, system integration, and organizational requirements may introduce additional challenges beyond those captured by benchmark instances. We also plan to investigate the trade-off between optimization quality and the computational cost of LLM-guided search, including its implications for the sustainability of large-scale deployments. Future work will also investigate the robustness of the proposed framework across different LLMs, assessing how the choice of the underlying model affects optimization performance and solution quality.

References

- [1] Bushra Alhijawi and Arafat Awajan. 2024. Genetic algorithms: theory, genetic operators, solutions, and applications. *Evolutionary Intelligence* 17, 3 (Jun 2024), 1245–1256. doi:10.1007/s12065-023-00822-6
- [2] Richard S. Barr, Bruce L. Golden, James P. Kelly, Mauricio G. C. Resende, and William R. Stewart. 1995. Designing and Reporting on Computational Experiments with Heuristic Methods. *Journal of Heuristics* 1, 1 (1995), 9–32. doi:10.1007/BF02430363
- [3] Mohammed Bazirha, Abdeslam Kadrani, and Rachid Benmansour. 2023. Stochastic home health care routing and scheduling problem with multiple synchronized services. *Annals of Operations Research* 320, 2 (Jan 2023), 573–601. doi:10.1007/s10479-021-04222-w
- [4] Felix Busch, Lena Hoffmann, Christopher Rueger, Elon H. C. van Dijk, Rawen Kader, Esteban Ortiz-Prado, Marcus R. Makowski, Luca Saba, Martin Hadamitzky, Jakob Nikolas Kather, Daniel Truhn, Renato Cuocolo, Lisa C. Adams, and Ken K. Bressem. 2025. Current applications and challenges in large language models for patient care: a systematic review. *Communications Medicine* 5, 1 (Jan 2025), 26. doi:10.1038/s43856-024-00717-2
- [5] Marco Cascella, Federico Semeraro, Jonathan Montomoli, Valentina Bellini, Ornella Piazza, and Elena Bignami. 2024. The Breakthrough of Large Language Models Release for Medical Applications: 1-Year Timeline and Perspectives. *Journal of Medical Systems* 48, 1 (Feb 2024), 22. doi:10.1007/s10916-024-02045-3
- [6] Eve Cheng and Jennifer L. Rich. 1998. *A home health care routing and scheduling problem*. Technical Report TR98-04. Rice University, USA.
- [7] Mohamed Cissé, Semih Yalçındağ, Yannick Kergosien, Evren Şahin, Christophe Lenté, and Andrea Matta. 2017. OR problems related to Home Health Care: A review of relevant routing and scheduling problems. *Operations Research for Health Care* 13-14 (2017), 1–22. doi:10.1016/j.orhc.2017.06.001
- [8] Patrik Eveborn, Mikael Rönnqvist, Helga Einarssdóttir, Mats Eklund, Karin Lidén, and Marie Almroth. 2009. Operations research improves quality and efficiency in home care. *Interfaces* 39, 1 (2009), 18–34.
- [9] Reza Ghanbarzadeh and Seyedali Mirjalili. 2025. A structured review of large language models in metaheuristic optimisation. *Decision Analytics Journal* 15 (2025), 100587. doi:10.1016/j.dajour.2025.100587
- [10] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. 2021. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications* 80, 5 (Feb 2021), 8091–8126. doi:10.1007/s11042-020-10139-6
- [11] Alberto F. Kummer, Olinto C.B. de Araújo, Luciana S. Buriol, and Mauricio G.C. Resende. 2024. A biased random-key genetic algorithm for the home health care problem. *International Transactions in Operational Research* 31, 3 (2024), 1859–1889. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/itor.13221 doi:10.1111/itor.13221
- [12] Alberto Kummer Neto, Luciana Buriol, and Olinto Araújo. 2019. A Matheuristic Algorithm Applied to the Home Health Care Problem. In *Proceedings of the LI Simpósio Brasileiro de Pesquisa Operacional (SBPO)*. Sobrappo, Limeira, SP, 1–11. doi:10.59254/sbpo-2019-106759
- [13] Wenheng Liu, Mahjoub Dridi, H. Fei, and Amir Hajjam. 2021. Hybrid Metaheuristics for Solving a Home Health Care Routing and Scheduling Problem with Time Windows, Synchronized Visits and Lunch Breaks. *Expert Systems with Applications* 183 (06 2021), 115307. doi:10.1016/j.eswa.2021.115307
- [14] lmstudio community. 2025. Qwen3-30B-A3B-Instruct-2507-MLX-4bit. https://huggingface.co/lmstudio-community/Qwen3-30B-A3B-Instruct-2507-MLX-4bit.
- [15] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Thomas Stützle, and Mauro Birattari. 2011. The irace Package: Iterated Racing for Automatic Algorithm Configuration. *Operations Research Perspectives* 3 (01 2011). doi:10.1016/j.orp.2016.09.002
- [16] Dorota Slawa Mankowska, Frank Meisel, and Christian Bierwirth. 2024. The home health care routing and scheduling problem with interdependent services. *Health Care Management Science* 17, 1 (Mar 2024), 15–30. doi:10.1007/s10729-013-9243-1
- [17] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. 2023. Mathematical discoveries from program search with large language models. *Nature* 625, 7995 (2023), 468–475. doi:10.1038/s41586-023-06924-6
- [18] Niki van Stein, Haoran Yin, Anna V. Kononova, Thomas Bäck, and Gabriela Ochoa. 2025. Behaviour Space Analysis of LLM-driven Meta-heuristic Discovery. arXiv:2507.03605 [cs.NE] https://arxiv.org/abs/2507.03605
- [19] Fang Wan, Kezhi Wang, Tao Wang, Hu Qin, Julien Fondrevelle, and Antoine Duclos. 2025. Enhancing healthcare resource allocation through large language models. *Swarm and Evolutionary Computation* 94 (2025), 101859. doi:10.1016/j.swevo.2025.101859
- [20] Haoran Ye, Jiarui Wang, Zhiguang Cao, Federico Berto, Chuanbo Hua, Haeyeon Kim, Jinkyoo Park, and Guojie Song. 2024. ReEvo: Large Language Models as Hyper-Heuristics with Reflective Evolution. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 43571–43608. doi:10.52202/079017-1381
- [21] Huigen Ye, Hua Xu, An Yan, and Yaoyang Cheng. 2024. Large Language Model-driven Large Neighborhood Search for Large-Scale MILP Problems. *arXiv preprint arXiv:2404.03693* (2024).
- [22] He Yu and Jing Liu. 2024. Deep Insights into Automated Optimization with Large Language Models and Evolutionary Algorithms. *arXiv preprint arXiv:2410.20848* (2024).
- [23] Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. 2024. TextGrad: Automatic "Differentiation" via Text. arXiv:2406.07496 [cs.CL] https://arxiv.org/abs/2406.07496
- [24] Jincheng Zhang. 2025. From Language to Optimization: Leveraging LLM Principles to Design a New Generation of Metaheuristics. *Preprint submitted to ResearchGate* (2025). doi:10.13140/RG.2.2.33127.56487