Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

AGENTIC SIMHEURISTIC: INTEGRATING GENERATIVE AI AND SIMHEURISTIC FOR A TEAM ORIENTEERING PROBLEM

Mohammad Peyman[1,2], and Yusef Ahsini[1,3]

[1]Bechained Artificial Intelligence Technologies, Barcelona, Spain
[2]Research Center on Production Mgmt. & Eng., Universitat Politècnica de València, Alcoy, Spain
[3]Pure & Applied Math. University Research Institute, Universitat Politècnica de València, Valencia, Spain

ABSTRACT

Addressing complex stochastic optimization problems often requires hybridization of search and evaluation methods. Simheuristics combine metaheuristics and simulation but typically rely on static control logic. Meanwhile, large language models (LLMs) offer advanced reasoning but lack robust mechanisms for constrained optimization. We propose the agentic simheuristic framework, a novel architecture that leverages an LLM as a high-level coordinator for simheuristic components. Applied to the team orienteering problem (TOP) under uncertainty, the framework employs an LLM to manage an exploratory agent for broad solution search and an exploitative agent for intensive refinement. Both agents integrate Monte Carlo simulation to evaluate solutions under uncertainty. The LLM guides the process by selecting promising and diverse exploratory solutions to seed refinement, enabling intelligent coordination within simheuristics. We present the framework architecture and provide initial empirical results on TOP benchmark instances, illustrating operational feasibility as a proof of concept and highlighting potential for explainable, AI-driven optimization.

1 INTRODUCTION

Modern decision-making scenarios, particularly in operations, logistics, and resource management, are increasingly characterized by large-scale, complex constraints, and widespread uncertainty (Juan et al. 2015). While traditional optimization techniques are effective under deterministic conditions, the presence of stochastic factors such as variable travel times, fluctuating demand, or unpredictable resource availability, necessitates methods that explicitly account for variability and risk. Simulation-based optimization approaches, particularly simheuristics, have emerged as a powerful paradigm for tackling such challenges (Juan et al. 2023; Abdullahi et al. 2025). By integrating metaheuristic search algorithms with simulation techniques, depending on the specific characteristics of the system under investigation, simheuristics may employ either MCS techniques (Gonzalez-Neira et al. 2017) or discrete-event simulation methods (Rabe et al. 2020) for the exploration of vast solution spaces while evaluating candidate solutions under realistic, uncertain conditions, leading to solutions that are not only high-performing on average but also robust against potential disruptions. Despite their success, conventional simheuristics often rely on pre-defined algorithmic structures and parameter settings. The strategic decisions within the search such as balancing exploration and exploitation, selecting promising regions to investigate further, or adapting heuristic parameters based on search progress, typically depend on static rules or require extensive, instance-specific tuning by domain experts. This limits their adaptability and potential autonomy in complex, dynamic environments.

Additionally, the emergence of powerful generative AI, particularly large language models (LLMs) like Gemini, GPT-4, and Llama 2, has opened new frontiers in automated reasoning, planning, and complex instruction following (Touvron et al. 2023; Gemini Team et al. 2023). These generative AI models possess remarkable capabilities for processing diverse information streams, understanding nuanced context, generating explanatory rationales, and making high-level strategic judgments when appropriately prompted and guided (Wei et al. 2022). However, attempts to apply LLMs directly to fine-grained combinatorial optimization problems often face difficulties in properly satisfying hard constraints, navigating complex search spaces effectively, and achieving near-optimal solution quality (Huang et al. 2025; Surendar 2025).

This paper bridges the gap between the robust evaluation capabilities of simheuristics and the strategic reasoning potential of Generative AI. We introduce the agentic simheuristic framework, a novel methodology where an LLM acts as an intelligent orchestrator, coordinating specialized simheuristic "agents" rather than performing the low-level optimization search itself. The LLM receives structured feedback summarizing the performance and characteristics of solutions generated by different agents, analyzes this information, and makes high-level strategic decisions to guide the overall search process. This includes dynamically allocating computational budget, and selecting promising candidate solutions generated by an exploration-focused agent to seed an exploitation-focused agent. An embedded simulation component remains critical, providing the stochastic evaluation required by the agents and guiding the LLM's assessment of solution robustness. We demonstrate the effectiveness of this framework on the team orienteering problem (TOP) under uncertainty, a prototypical NP-hard routing challenge applicable in domains such as disaster relief, tourism planning, and autonomous mission scheduling (Chao et al. 1996; Panadero et al. 2020). Formally, the TOP involves designing routes for a limited fleet of vehicles that begin and end their journeys at predefined depots. Each potential location offers a reward when visited, and traveling between locations incurs a cost, typically measured by time or distance. The objective is to maximize the total collected reward while ensuring that the travel cost of each vehicle's route remains within a predetermined maximum limit. The main contributions of this work are:

- The conceptualization and implementation of the agentic simheuristic framework, employing an LLM to orchestrate distinct exploration-focused and exploitation-focused simheuristic agents for solving stochastic optimization problems like the TOP under uncertainty.
- A novel mechanism for LLM-driven strategic control in simheuristics, demonstrating how generative AI can analyze structured feedback (performance metrics, solution diversity) to make informed decisions (e.g., seed selection) that guide the search trajectory.
- An empirical evaluation of the proposed framework on benchmark TOP instances with stochastic travel times.
- A design of the framework with a view toward improved explainability in future extensions.

The remainder of this paper is organized as follows: Section 2 provides background information and reviews relevant literature on simheuristics, and the emerging use of LLMs in optimization contexts. Section 3 presents the definition of the stochastic TOP. Section 4 details the proposed agentic simheuristic framework, outlining its architecture, the roles of the LLM orchestrator and (meta)heuristic agents, and the operational workflow through its distinct phases. Section 5 describes the experimental setup, including the benchmark instances used, parameter settings, and performance metrics, followed by a presentation and analysis of the computational results. Finally, Section 6 concludes the paper, summarizing the key findings, discussing the implications and limitations of the agentic approach, and suggesting directions for future research.

2 BACKGROUND AND LITERATURE REVIEW

2.1 SIMHEURISTICS

Variables in practical scenarios often exhibit uncertainty—e.g., travel delays or fluctuations in demand. Travel times between nodes in urban networks, typically based on speed and distance, are influenced by factors such as traffic and weather, resulting in temporal variability. This can be modeled as a stochastic variable following distributions like lognormal or Weibull, which effectively represent skewed data. In this context, travel times comprise ideal components plus stochastic delays. Other problem uncertainties can be modeled using fuzzy variables (Lootsma 2013), requiring solution methods that incorporate both probabilistic constraints and stochastic/fuzzy modeling. Simheuristics address this by integrating simulation with optimization. The approach begins by replacing uncertain variables with expected values to solve a deterministic version, then evaluates candidate solutions through Monte Carlo or discrete-event simulation, iteratively refining an elite solution set until convergence (Rabe et al. 2020; Juan et al. 2023). Final elite solutions undergo extensive simulation for robust analysis. Numerous studies attest to simheuristics' power under uncertainty: iterated local search for preventive maintenance (Gruler et al. 2021), fuzzy-stochastic extensions (Juan et al. 2023), and discrete-event models for warehouse layout (Leon et al. 2023). In the stochastic TOP, adaptive MDP heuristics (Tricoire et al. 2010), BR-simheuristics for hospital logistics during COVID-19 (Rabe et al. 2021), chance-constrained programming (Herrera et al. 2022), and urban drone routing (Peyman et al. 2025) demonstrate its practical impact.

2.2 GENERATIVE AI AND LLMs

The rapid advancement of large language models (LLMs), a key branch of generative AI, has transformed artificial intelligence. Built on the transformer architecture (Vaswani et al. 2017), models such as OpenAI's GPT series (Brown et al. 2020), Meta's LLaMA family (Touvron et al. 2023), and Google's PaLM and Gemini (Gemini Team et al. 2023) demonstrate strong capabilities in natural language tasks. Trained on vast corpora, LLMs exhibit emergent abilities such as few-shot and zero-shot learning (Brown et al. 2020), as well as improved reasoning and planning via techniques like Chain-of-Thought prompting (Kojima et al. 2022). These capabilities position LLMs for roles beyond text generation, including decision-support and complex task execution. Recent "agentic" LLM-based systems autonomously decompose tasks, utilize external tools, and reflect on outcomes (Xi et al. 2025; Park et al. 2023), often via a reasoning–action–observation loop as in ReAct (Yao et al. 2023) or Reflexion (Shinn et al. 2023). This capability underpins our LLM orchestrator. Simheuristics—hybrids of stochastic simulation and metaheuristics—have proven effective in uncertain domains (Chica et al. 2020), yet seldom incorporate AI. Notable exceptions include deep RL for orienteering problems (Gama and Fernandes 2021), neural algorithmic reasoning frameworks (Wu et al. 2024), and a recent LLM-assisted simheuristic for project portfolio selection (Saiz and Calvet 2024).

However, direct integration involving LLM-driven generative methods with simheuristics remains significantly underexplored. Our proposed agentic simheuristic framework represents a novel synthesis within this landscape. Unlike direct optimization attempts, it leverages established, powerful simheuristic agents for the core search and evaluation, ensuring solution feasibility and robust performance assessment via MCS. Unlike simple parameter tuning or code generation, the LLM acts as an active, online orchestrator during the search process. Its key role is strategic coordination between distinct, complementary heuristic agents. By processing structured feedback summarizing agent performance and solution characteristics (quality, diversity), the LLM makes informed, high-level decisions—specifically, selecting diverse, high-potential seeds from the explorer to guide the exploiter. This contrasts with single-agent LLM frameworks or approaches where the LLM selects a single heuristic upfront. Furthermore, the tight integration with simheuristics allows the LLM's strategic decisions to be informed by stochastic performance estimates, and its generative nature offers inherent potential for explainability. This work, therefore, explores the unique potential of using LLMs specifically for the high-level strategic coordination between multiple specialized optimization-simulation components within a dynamic search process.

3 PROBLEM DEFINITION

As introduced earlier, we demonstrate and evaluate our agentic simheuristic framework using the TOP under uncertainty. The TOP, a well-known NP-hard routing challenge (Chao et al. 1996), involves designing routes for a limited fleet of vehicles (or operational teams) to maximize collected rewards from visited locations, subject to resource constraints. While the conceptual understanding of the TOP was outlined in the introduction, this section provides its formal mathematical definition, specifically tailored to incorporate stochastic transition costs.

The problem is defined on a directed graph $G = (V,A)$. The set of nodes $V = \{0,1,2,\ldots,n+1\}$ includes a designated initial depot (node 0), a final depot (node $n+1$), and a set $N = \{1,\ldots,n\}$ of intermediate operational points where rewards can be collected. Arcs $(i, j) \in A$ (where $i, j \in V, i \neq j$) represent feasible transitions. We consider a set K of M available vehicles/teams. Each vehicle $k \in K$ initiates its route from node 0, visits a subset of nodes in N, and makes an eventual return to node $n+1$. Each visited node $j \in N$ yields a deterministic reward $u_j \geq 0$; rewards for the depots are $u_0 = u_{n+1} = 0$. The primary objective is to maximize the total reward collected by all vehicles/teams, as shown in Equation (1). For each arc $(i, j) \in A$ and each vehicle $k \in K$, a binary decision variable x^k_{ij} is 1 if vehicle k traverses arc (i, j), and 0 otherwise.

This objective is maximized subject to several constraints. Firstly, each intermediate node $j \in N$ can be visited at most once across all vehicles/teams (2). Each vehicle $k \in K$ must start its route at node 0 (3) and end at node $n+1$ (4). Route continuity for each vehicle k is ensured by flow conservation constraints at each intermediate node $j \in N$ (5). To prevent sub-tours, auxiliary variables $y^k_j \geq 0$ are introduced to track the position of node j in the route of vehicle k (6 and 7). Constraint (8) caps the total travel time of every vehicle $k \in K$ at its endurance limit $C_{\max}$. Each edge $(i, j) \in A$ has an associated (stochastic) travel time S_{ij}, whose distribution is estimated from data (e.g., Log-Normal). This formulation enforces the time budget while explicitly accounting for uncertainty in travel durations. The evaluation of this chance constraint necessitates simulation. Finally, the decision variables x^k_{ij} are binary (9).

$$\max \sum_{k\in K} \sum_{(i,j)\in A} u_j x^k_{ij} \tag{1}$$

$$\sum_{k\in K} \sum_{i\in V} x^k_{ij} \leq 1 \quad \forall j \in N \tag{2}$$

$$\sum_{j\in N} x^k_{0j} = 1 \quad \forall k \in K \tag{3}$$

$$\sum_{i\in N} x^k_{i,n+1} = 1 \quad \forall k \in K \tag{4}$$

$$\sum_{i\in V} x^k_{ij} = \sum_{l\in V} x^k_{jl} \quad \forall j \in N, \forall k \in K \tag{5}$$

$$y^k_i - y^k_j + 1 \leq (1 - x^k_{ij}) \cdot |N| \quad \forall i, j \in N, i \neq j, (i, j) \in A, \forall k \in K \tag{6}$$

$$y^k_j \geq 0 \quad \forall j \in N, \forall k \in K \tag{7}$$

$$\sum_{(i,j)\in A} S_{ij}\, x^k_{ij} \;\leq\; C_{\max}, \qquad \forall\, k \in K, \tag{8}$$

$$x^k_{ij} \in \{0,1\} \quad \forall (i, j) \in A, \forall k \in K \tag{9}$$

Our proposed agentic simheuristic approach, detailed in the following section, is designed to address such complexities by leveraging an LLM for high-level strategic coordination of specialized simheuristic agents. While we focus on the TOP with its vehicle routing interpretation for this proof-of-concept, the underlying principles of LLM-coordinated exploration and exploitation agents within a simheuristic loop are envisioned to be generalizable to a broader class of complex stochastic optimization problems.

4 AGENTIC SIMHEURISTIC APPROACH

This section details the proposed agentic simheuristic framework, a novel methodology leveraging the strategic reasoning capabilities of LLMs to orchestrate simheuristic optimization processes. The core idea is to use an LLM not for direct optimization but as a high-level controller coordinating specialized heuristic agents, whose solutions are evaluated under uncertainty using MCS. Specifically, in this work, the framework coordinates a biased-randomized multi-start (BR-MS) agent focused on exploration and a biased-randomized iterated local search (BR-ILS) agent focused on exploitation.

Figure 1 schematically presents our three-component architecture. At its core, an LLM orchestrator (e.g., GPT-4, Gemini, Llama 2) governs the workflow, interprets agent feedback, and makes strategic calls. It communicates with two specialized (meta)heuristics: a BR-MS exploration agent (Martí et al. 2013) (solid arrows) that quickly generates diverse candidate solutions, and a BR-ILS exploitation agent (Lourenço et al. 2010) (dashed arrows) that applies perturbation-and-local-search cycles to intensify high-quality solutions. Both agents feed candidate solutions into an embedded MCS for risk-aware evaluation. This combination leverages BR-MS's speed and BR-ILS's refinement under LLM guidance to efficiently navigate stochastic search spaces. The framework executes through a sequence of phases orchestrated by the LLM. While the architecture permits more dynamic control flows, our current implementation adopts a structured multi-phase strategy to systematically balance exploration and exploitation for the stochastic TOP. Initially, the LLM orchestrator launches the exploration phase. It decides on an appropriate initial time allocation for the BR-MS agent, considering the total time budget, the need to reserve time for final evaluation, and potentially the complexity of the instance. It then tasks the agent with exploring the solution space within this budget, providing the instance details and simulation parameters. The BR-MS agent proceeds iteratively, using biased randomization and short MCS evaluations to generate and assess a diverse pool of candidate solutions. Upon completion, it reports its findings (best deterministic solution, elite simulated solutions) back to the LLM orchestrator.

Subsequently, the framework enters the analysis and seed selection phase, where the LLM orchestrator actively makes strategic decisions. Processing the feedback, the LLM analyzes the characteristics of the unique candidate solutions provided by the explorer. Based on its configured strategy (e.g., prioritizing high simulated reward, potentially considering solution robustness or structural features), it ranks the candidates. Before the LLM is queried, the Python controller performs a deterministic *diversity pre-filtering* step. For every candidate produced by the BR-MS explorer we extract the set of visited customer nodes and compute its Jaccard similarity with the node sets of the survivors selected so far. A candidate is retained only if the maximum similarity does not exceed a defined threshold (0.85$ in our experiments). After this screening, the LLM ranks the remaining candidates and selects up to *k* of them as seeds for the subsequent exploitation phase. The LLM finalizes this phase by outputting the validated list of seed-solutions. Its implementation is conducted using the following prompt:

```
Final Seed Selection from Pre-Filtered Candidates.
Total solver time budget: {max_time:.2f}s. Remaining solver time:
{remaining_time:.2f}s.
Goal: Select the best seeds for focused Biased-Randomized ILS runs from
the provided pre-filtered list.
Available PRE-FILTERED candidates (index: summary):
{candidate_summaries}
Diversity Threshold (already applied in pre-filtering): {diversity_threshold:.2f}.
Select up to {max_seeds} indices from the list above. Prioritize candidates
with the best reward potential (consider both deterministic and simulated
rewards).
Respond with:
1. A comma-separated list of integer indices (e.g. 0,2,4) on the first
line.
```

```
2. The separator '{separator}' on the next line.
3. A brief explanation for your final selection choices on the following
lines.
```

Because the list that the LLM receives has already been diversity-screened, the model is relieved from computing pairwise similarity itself. The prompt therefore simply reminds it that the "`Diversity Threshold`" has been applied; its task is to pick the *k* highest-reward seeds subject to that cardinality limit. Along with this diversity information, the LLM also receives the total solver time budget, remaining solver time, and maximum number of seeds to select (typically 3 in our instances). For each solution in the list, the LLM receives a concise structured summary containing: (1) a deterministic reward value measuring performance under expected conditions, (2) a simulation reward value capturing performance under stochastic evaluation, and (3) structural information about the number of routes. These three metrics enable the LLM to make informed decisions prioritizing solutions with the best overall performance potential while considering time constraints. In other words, the diversity constraint is enforced algorithmically by the controller, whereas the LLM performs strategic selection from this pre-filtered, diverse candidate pool with awareness of both solution quality and computational resource constraints.

In other words, the diversity constraint is enforced algorithmically by the controller, whereas the LLM merely operates on the pre-filtered, hence already diverse, set of candidates. The exploitation Phase is then initiated based on the LLM's decisions. The LLM orchestrator determines how to allocate the remaining computational budget among the selected seed-solutions. This could be a simple equal division or potentially a more nuanced allocation based on the perceived potential of each seed. The prompt used is:

```
Focused Exploitation Time Allocation (Biased-Randomized ILS Loops).
Total solver time budget: {max_time:.2f}s. Remaining solver time:
{remaining_time:.2f}s.
Number of final selected seeds: {num_seeds}.
Final selected seed summaries:
{seed_summaries}
Goal: Allocate remaining solver time among {num_seeds} final seeds for
Standard ILS.
Task: First, explain your reasoning for choosing a time allocation
strategy (consider seed quality/diversity, remaining time). Then,
state the chosen strategy ('EQUAL' or 'WEIGHTED').
Respond with:
1. Your reasoning on the first lines.
2. The separator '{separator}' on the next line.
3. The strategy name ('EQUAL' or 'WEIGHTED') on the final line.
```

It then launches sequential instances of the BR-ILS agent, providing each with its assigned seed and corresponding time budget. Each BR-ILS agent instance executes its iterative search, alternating between perturbation (configured by the LLM – e.g., biased-randomized) and Local Search, using short MCS evaluations to guide its internal acceptance criterion and track promising solutions. Upon completion, each agent reports its best findings back to the LLM orchestrator. Finally, the final evaluation and selection phase concludes the process. The LLM orchestrator aggregates all results. It identifies the best overall deterministic solution and the top unique elite candidates based on short simulations from all exploitation runs. This final pool undergoes high-fidelity evaluation via long MCS runs, performed within the reserved time budget. Based on these reliable results, the LLM makes the final determination of the final best result in deterministic and stochastic situations. Throughout the process, the LLM can be prompted to provide explanations justifying its key strategic decisions regarding seed selection and potentially time allocation.

The orchestrator queries the LLM through a stateless HTTPS API, pinning requests to a deployment region close to the simulation server to curb network latency; it employs the Gemini 2.0 Flash model, whose fast inference facilitates real-time optimisation. Each call is issued with a low temperature to limit stochasticity, and the prompt is strongly typed, specifying exact syntax for floats, comma-separated indices and a sentinel separator so that the parser can deterministically extract actionable data. The returned string is immediately vetted by range checks, regular-expression matching and hash-based diversity guards, and any deviation triggers an automatic fallback to tuned heuristic defaults. This blend of low-latency deployment, fast-response model, conservative temperature and strict prompt-and-verify protocol keeps LLM guidance both rapid and reliable, while leaving the solver's time budget intact.

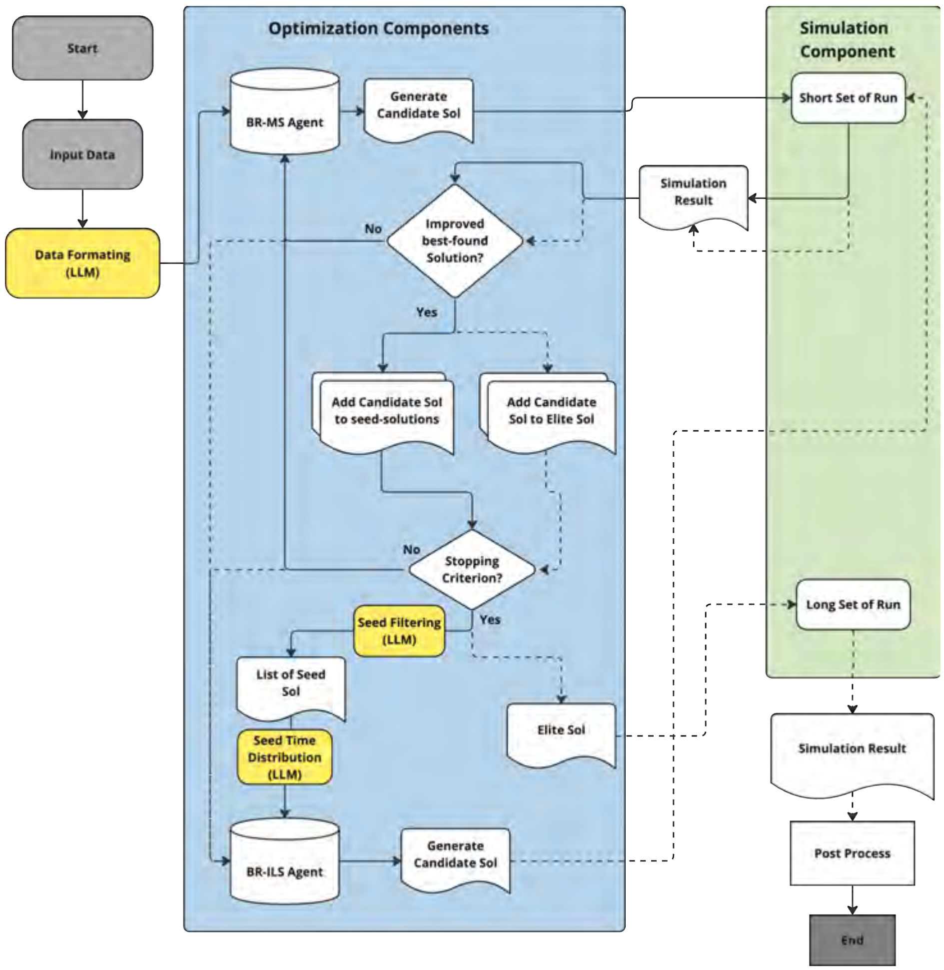

Figure 1: Schema of the agentic simheuristic framework.

.

5 COMPUTATIONAL RESULTS

The computational experiments were conducted on Apple Silicon M2 CPU with 16 GB RAM. The agentic simheuristic framework were implemented using Python 3.12. The LLM selected in our implementation is Gemini 2.0 Flash. The framework's computational time was set to a time limit of 60 seconds for all instances. The β parameter for the geometric distribution was randomly assigned from the interval $(0.1, 0.3)$ after a quick tuning process over a random sample of instances which established a good performance. For the exploratory and refinement stages of the simulation phase, we conducted 100 and 1000 simulation runs, respectively. To validate the proposed methodology, we randomly selected 10 instances from a well-known benchmark proposed by (Chao et al. 1996) and conducted 10 separate executions for each instance, utilizing varying initial seeds for the algorithm. Among the results from these runs, we reported the best solutions based on the highest collected reward. This benchmark set is composed of a total of 320 instances, divided into seven main groups. Each instance is labeled in the pattern px.y.z, where x is the group number, y is the number of vehicles, which varies from 2 to 4 depending on the instance, and z indicates the instance number. These instances have been widely used in previous research to assess the efficiency of algorithms in handling the deterministic TOP. In the stochastic scenario, we introduce uncertainty into the travel times between nodes. Each edge $(i,j) \in A$ in the directed graph $G = (V,A)$ is now defined by a travel time, $T_{ij} = T_{ji} > 0$, which is not deterministic but follows a best-fit probability distribution function with a mean value $E[t_{ij}] > 0$. In our computational experiments, we used the Log-Normal probability distribution to model the random travel times. The Log-Normal distribution is preferred over the Normal distribution when modeling non-negative random variables. It has two parameters, namely the location parameter μ and the scale parameter σ. These parameters can be determined based on the properties of the Log-Normal distribution considering the stochastic travel times between nodes i and j are assumed to be as follows: $E[T_{ij}] = t_{ij}$ (i.e., the travel costs of the deterministic instances), and $\mathrm{Var}[T_{ij}] = c \cdot t_{ij}$ for all $i, j \in \{0, 1, 2, \ldots, n+1\}$. The parameter c serves as a design parameter enabling us to control the level of uncertainty. As c approaches zero, the results of the stochastic scenario are expected to converge with those obtained in the deterministic scenario. For our analysis, we have employed the value $c = 1$, which introduces a significant level of uncertainty as it uses the deterministic travel time as variance. A more comprehensive discussion on the implications of varying the parameter c can be found in (Panadero et al. 2023). In our analysis, the LLM selection policy in the Seed Filtering is strikingly uniform, when only one candidate remains (32% of cases) it is chosen; when two remain (18%), both are chosen; when three remain (41%), all three are chosen; and when more than three pass the filter (9%), the top three seeds by combined deterministic and simulated reward are selected. For the subsequent time allocation, the model applies an equal-distribution strategy in 94% of runs, resorting to a weighted allocation in the remaining 6%. In the roughly 6% of cases where a weighted allocation is used, the decision always reflects a clear performance gap among the seeds, one seed's simulated or deterministic reward substantially exceeds its peers, so the agent concentrates a larger share of ILS iterations on that top performer while still reserving a smaller portion of time for the other candidates to maintain diversity.

Tables 1 presents the computational results obtained by applying the proposed agentic simheuristic framework respectively. The first column identifies the problem instances, while the subsequent columns present the results under both deterministic and stochastic evaluation scenarios. The *BKS* column lists the best-known solutions (BKS) for the deterministic variant of the problem, obtained from the literature. The *OBD-R* column reports the total collected reward for the best-found solutions in the deterministic scenario. The *OBD-S-R* column indicates the collected reward when the deterministic solutions are evaluated in a stochastic environment, reflecting their performance under uncertainty. The *OBS-R* column presents the collected reward for the best-found solutions in the stochastic scenario.

Firstly, over the full set of instances (average BKS = 778.25), the agentic simheuristic's deterministic outputs (average OBD-R = 756.10) consistently match or exceed the standalone BR-MS baseline (average OBD-R = 685.50) and often tie the BKS. This deterministic advantage is driven by the insertion of a BR-ILS exploitation phase immediately after the broad BR-MS exploration, enabling more intensive refinement of the most promising candidate solutions. Secondly, when those deterministic solutions are re-evaluated stochastically, the agentic framework again outperforms BR-MS—achieving an average OBD-S-R of 705.46 versus 643.68 for BR-MS—and its directly optimized stochastic solutions (OBS-R = 711.46 on average) markedly exceed the BR-MS average of 647.38. These results demonstrate that LLM-guided coordination and focused exploitation not only yield high-quality deterministic solutions but also deliver substantially greater expected rewards under uncertainty, underscoring the robustness of the agentic simheuristic in stochastic settings.

Table 1: Comparison of BKS, agentic simheuristics, and BR-MS results.

		Agentic Simheuristics			**BR-MS Simheuristic**		
Instance	**BKS [1]**	**OBD-R [2]**	**OBD-S-R**	**OBS-R [3]**	**OBD-R [4]**	**OBD-S-R**	**OBS-R [5]**
p2.3.h	165	165	153.50	153.75	165	153.45	153.66
p2.4.f	105	105	95.46	99.53	105	95.18	98.23
p3.2.k	550	550	520.59	526.15	500	465.19	473.68
p3.3.t	760	760	691.08	694.12	710	670.48	671.68
p3.4.t	670	670	637.76	640.21	670	636.40	639.40
p3.4.r	600	600	563.26	569.70	520	493.48	493.89
p4.2.q	1268	1246	1181.21	1182.98	865	846.53	846.53
p4.3.m	1063	929	872.14	872.76	756	721.16	723.46
p4.4.p	1124	1094	992.47	1008.46	817	765.22	781.92
p5.2.z	1680	1660	1573.10	1573.59	1660	1564.33	1550.98
p5.2.u	1460	1425	1337.69	1349.06	1425	1335.36	1346.81
p5.3.h	260	250	233.35	234.15	240	226.11	228.68
p5.3.k	495	470	445.97	455.93	435	398.46	402.39
p5.4.g	140	140	130.89	131.23	140	130.44	130.77
p5.4.x	1450	1450	1348.65	1367.85	1415	1335.13	1342.57
p6.3.j	828	828	754.30	756.64	816	732.54	746.89
p6.3.n	1170	1164	1097.03	1100.07	1002	932.07	932.07
p6.4.k	528	516	468.15	470.41	486	433.80	439.09
p7.2.p	1002	875	798.72	828.68	772	739.22	745.68
p7.3.f	247	225	214.03	214.06	211	199.13	199.34
Average	778.25	756.10	705.46	711.46	685.50	643.68	647.38

The performance gap relative to the BKS is shown in Figure 2. The boxplots display the distribution of percentage gaps of OBD, OBD-S, and OBS solutions for both the standalone BR-MS baseline (red boxes) and the agentic simheuristic (blue boxes). For the OBD Gap, the agentic approach exhibits a mean gap of 2.69% compared to 10.20% for BR-MS. Likewise, for the OBD-S Gap the agentic mean is 9.32% versus 16.08% for BR-MS, and for the OBS Gap the agentic mean is 8.46% versus 15.44% for BR-MS. We applied a two-tailed Wilcoxon signed-rank test to the paired OBS rewards ($d_i = \text{OBS\_Agentic}_i - \text{OBS\_BR-MS}_i$) over 20 instances. All $d_i > 0$, yielding $W = 0$, $p < 0.0001$. Hence, we reject H_0 and conclude that the agentic simheuristic delivers significantly higher stochastic rewards than BR-MS.

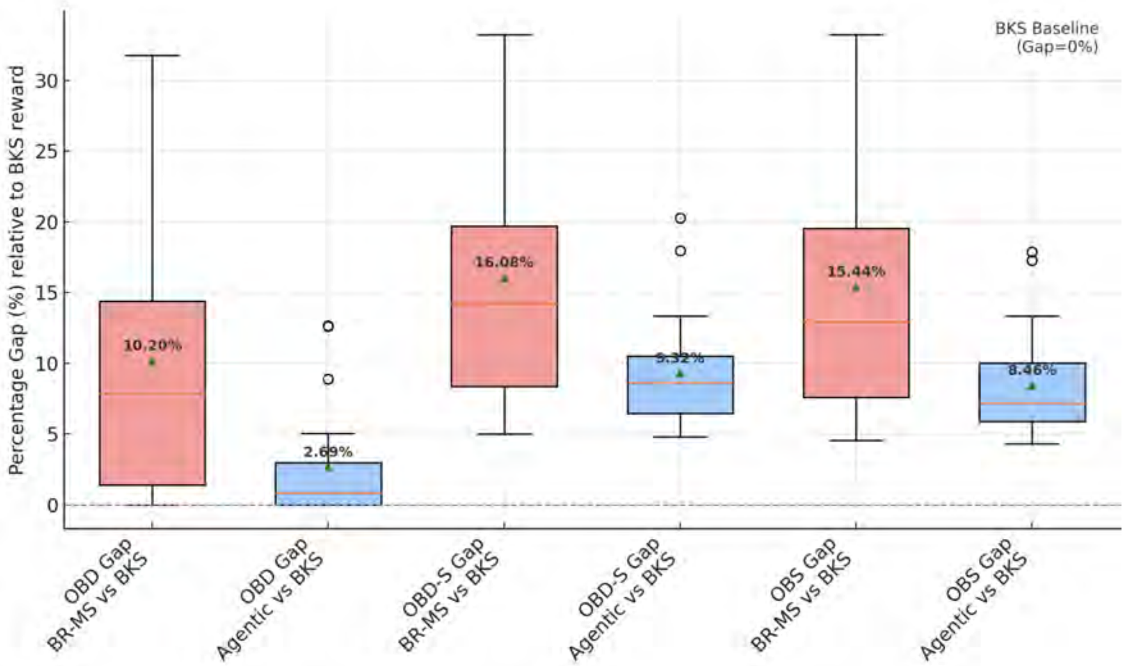

Figure 2: Gaps with respect to BKS for the different approaches.

6 CONCLUSION AND DISCUSSION

We have presented an agentic simheuristic framework in which an LLM orchestrator guides BR-MS exploration and BR-ILS exploitation with embedded MCS for risk-aware evaluation. Applied to the stochastic TOP, our proof-of-concept implementation reduced the average deterministic gap from 10.20 % to 2.69 % and the average stochastic gap from 16.08 % to 9.32 %, while producing significantly higher expected rewards. These results confirm that LLM-guided coordination can both improve solution quality and reduce variability compared to a standalone BR-MS baseline. The LLM orchestrator managed the interplay between a BR-MS agent for exploration and a BR-ILS agent for exploitation, with both agents utilizing embedded MCS for risk-aware solution evaluation. By processing diverse information streams—ranging from quantitative performance metrics and solution characteristics to, in future iterations, qualitative problem descriptions—the LLM applies flexible, explainable reasoning. This separation of concerns offers a promising alternative to the static control logic and manual parameter tuning typical of traditional simheuristics, fostering greater robustness across varied instances and enabling dynamic adaptation of search strategies based on real-time feedback.

Nonetheless, this initial study remains a proof-of-concept with certain limitations. The current implementation uses a structured yet simple summary-based protocol for seed selection and inter-phase time allocation, confining strategic intervention to discrete decision points rather than continuous control over parameters or operator choice. Moreover, overall performance still depends on the efficacy of the underlying BR-MS and BR-ILS agents and the fidelity of the embedded MCS. Future work will extend this framework by enabling richer LLM–agent dialogues for dynamic parameter tuning and operator selection, integrating continuous monitoring of search state, and validating the approach on other stochastic optimization domains.

REFERENCES

Abdullahi, H., L. Reyes-Rubiano, D. Ouelhadj, J. Faulin, and A. A. Juan. 2025. “A Reliability-Extended Simheuristic for the Sustainable Vehicle Routing Problem with Stochastic Travel Times and Demands”. *Journal of Heuristics* 31(2):1–39.

Brown, T. B., B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, *et al*. 2020. “Language models are few-shot learners”. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS ’20. Red Hook, NY, USA.

Chao, I.-M., B. L. Golden, and E. A. Wasil. 1996. “The team orienteering problem”. *European Journal of Operational Research* 88(3):464–474 https://doi.org/https://doi.org/10.1016/0377-2217(94)00289-4.

Chica, M., A. A. Juan, C. Bayliss, O. Cordón, and W. D. Kelton. 2020. “Why Simheuristics?: Benefits, Limitations, and Best Practices When Combining Metaheuristics with Simulation”. *SORT: Statistics and Operations Research Transactions* 44(2):311–334.

Gama, R., and H. L. Fernandes. 2021. “A Reinforcement Learning Approach to the Orienteering Problem with Time Windows”. *Computers & Operations Research* 133:105357.

Gemini Team *et al*. 2023. “Gemini: A Family of Highly Capable Multimodal Models”. *arXiv preprint arXiv: 2312.11805*.

Gonzalez-Neira, E. M., D. Ferone, S. Hatami, and A. A. Juan. 2017. “A Biased-Randomized Simheuristic for the Distributed Assembly Permutation Flowshop Problem with Stochastic Processing Times”. *Simulation Modelling Practice and Theory* 79:23–36.

Gruler, A., C. L. Quintero-Araujo, L. Calvet, and A. A. Juan. 2021. “Waste Collection Under Uncertainty: A Simheuristic Based on Variable Neighbourhood Search”. *European Journal of Industrial Engineering* 15(1):1–27.

Herrera, E. M., J. Panadero, P. Carracedo, A. A. Juan, and E. Perez-Bernabeu. 2022. “Determining Reliable Solutions for the Team Orienteering Problem with Probabilistic Delays”. *Mathematics* 10(20):3788.

Huang, L., W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, *et al*. 2025. “A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions”. *ACM Transactions on Information Systems* 43(2):1–55.

Juan, A. A., J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira. 2015. “A Review of Simheuristics: Extending Metaheuristics to Deal With Stochastic Combinatorial Optimization Problems”. *Operations Research Perspectives* 2:62–72.

Juan, A. A., P. Keenan, R. Martí, S. McGarraghy, J. Panadero, P. Carroll *et al*. 2023. “A Review of the Role of Heuristics in Stochastic Optimisation: From Metaheuristics to Learnheuristics”. *Annals of Operations Research* 320(2):831–861.

Juan, A. A., M. Rabe, M. Ammouriova, J. Panadero, D. Peidro, and D. Riera. 2023. “Solving NP-Hard Challenges in Logistics and Transportation under General Uncertainty Scenarios Using Fuzzy Simheuristics”. *Algorithms* 16(12):570.

Kojima, T., S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa. 2022. “Large language models are zero-shot reasoners”. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS ’22. Red Hook, NY, USA.

Leon, J. F., Y. Li, M. Peyman, L. Calvet, and A. A. Juan. 2023. “A Discrete-Event Simheuristic for Solving a Realistic Storage Location Assignment Problem”. *Mathematics* 11(7):1577.

Lootsma, F. A. 2013. *Fuzzy Logic for Planning and Decision Making*, Volume 8 of *Applied Optimization*. Dordrecht, The Netherlands: Springer Science & Business Media.

Lourenço, H. R., O. C. Martin, and T. Stützle. 2010. *Iterated Local Search: Framework and Applications*, 363–397. Boston, MA: Springer US https://doi.org/10.1007/978-1-4419-1665-5_12.

Martí, R., M. G. Resende, and C. C. Ribeiro. 2013. “Multi-Start Methods for Combinatorial Optimization”. *European Journal of Operational Research* 226(1):1–8.

Panadero, J., A. A. Juan, C. Bayliss, and C. Currie. 2020. “Maximising Reward From A Team of Surveillance Drones: A Simheuristic Approach to the Stochastic Team Orienteering Problem”. *European Journal of Industrial Engineering* 14(4):485–516.

Panadero, J., A. A. Juan, E. Ghorbani, J. Faulin, and A. Pagès-Bernaus. 2023. “Solving the Stochastic Team Orienteering Problem: Comparing Simheuristics with the Sample Average Approximation Method”. *International Transactions in Operational Research* 31(6):3036–3060.

Park, J. S., J. O’Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. 2023. “Generative Agents: Interactive Simulacra of Human Behavior”. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 1–22. San Francisco, USA.

Peyman, M., X. Martin, J. Panadero, and A. A. Juan. 2025. “A Discrete-Event Simheuristic for Enhancing Urban Mobility”. *Simulation Modelling Practice and Theory* 140:103084.

Rabe, M., J. L. Chicaiza-Vaca, R. D. Tordecilla, L. d. C. Martins, and A. A. Juan. 2021, December. “Supporting Hospital Logistics During the First Months of the COVID-19 Crisis: A Simheuristic for the Stochastic Team Orienteering Problem”. In *Proceedings of the 2021 Winter Simulation Conference (WSC)*, 1–10 https://doi.org/10.1109/WSC52266.2021.9715337.

Rabe, M., M. Deininger, and A. A. Juan. 2020. “Speeding Up Computational Times in Simheuristics Combining Genetic Algorithms with Discrete-Event Simulation”. *Simulation Modelling Practice and Theory* 103:102089 https://doi.org/doi.org/10.1016/j.simpat.2020.102089.

Saiz, M., and L. Calvet. 2024. “Combining LLMs and Simheuristics: An Application to the Project Portfolio Selection Problem”. In *Proceedings of the Decision Science Alliance International Summer Conference (DSAISC)*, 3–13. Barcelona, Spain: Springer, Cham https://doi.org/10.1007/978-3-031-78238-1_1.

Shinn, N., F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao. 2023. "Reflexion: Language Agents With Verbal Reinforcement Learning". *Advances in Neural Information Processing Systems* 36:8634–8652.

Surendar, A. 2025. "Hybrid LLM-Algorithm Optimization: Iterated Fine-Tuning for Combinatorial Problems". *Frontiers in Mathematical and Computational Research* 1(1):32–39.

Touvron, H., L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, *et al*. 2023. "Llama 2: Open Foundation and Fine-Tuned Chat Models". *arXiv preprint arXiv:2307.09288*.

Tricoire, F., M. Romauch, K. F. Doerner, and R. F. Hartl. 2010. "Heuristics for the Multi-Period Orienteering Problem with Multiple Time Windows". *Computers & Operations Research* 37(2):351–367.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, *et al*. 2017. "Attention Is All You Need". In *Advances in Neural Information Processing Systems*, Volume 30. Neural Information Processing Systems Foundation https://doi.org/10.5555/3295222.3295349.

Wei, J., X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. H. Chi, *et al*. 2022. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models". *Advances in Neural Information Processing Systems* 35:24824–24837 https://doi.org/10.5555/3600270.3602070.

Wu, Z., L. Qiu, A. Ross, E. Akyürek, B. Chen, B. Wang, *et al*. 2024. "Reasoning or Reciting? Exploring the Capabilities and Limitations of Language Models Through Counterfactual Tasks". In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 1819–1862. Association for Computational Linguistics https://doi.org/10.18653/v1/2024.naacl-long.102.

Xi, Z., W. Chen, X. Guo, W. He, Y. Ding, B. Hong, *et al*. 2025. "The Rise and Potential of Large Language Model Based Agents: A Survey". *Science China Information Sciences* 68(2):121101 https://doi.org/doi.org/10.1007/s11432-024-4222-0.

Yao, S., J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan *et al*. 2023. "React: Synergizing Reasoning and Acting in Language Models". In *International Conference on Learning Representations (ICLR)*.

AUTHOR BIOGRAPHIES

MOHAMMAD PEYMAN is a Data Scientist and Machine Learning Engineer at Bechained. He is concurrently a PhD candidate and researcher at the Universitat Politècnica de València, specializing in Statistics and Optimization. His primary research interests lie at the intersection of simulation-optimization, artificial intelligence, data science, and machine learning algorithms, with a focus on developing intelligent systems for complex decision-making under uncertainty. His email address is mpeyman@upv.es.

YUSEF AHSINI is a Machine Learning Engineer at Bechained. He is concurrently an AI Researcher at the Universitat Politècnica de València, specializing in machine learning and data analytics . His primary research interests lie at the intersection of machine learning, data analytics, deep learning, and artificial intelligence, with a focus on developing optimization algorithms for complex routing and logistics problems under uncertainty. His e-mail address is yahsoua@upv.edu.es.