Island-DR-ALNS: A Diversity-Driven Multi-Agent Framework for Deep Reinforcement Learning-based Combinatorial Optimization

Adam Żychowski[1][0000−0003−0026−5183], Xin Yao[2][0000−0001−8837−4442], and Jacek Mańdziuk[1,3][0000−0003−0947−028X]

[1] Faculty of Mathematics and Information Science, Warsaw University of Technology, Poland
[2] School of Data Science, Lingnan University, Hong Kong SAR
[3] Faculty of Computer Science, AGH University of Krakow, Krakow, Poland
{adam.zychowski,jacek.mandziuk}@pw.edu.pl, xinyao@ln.edu.hk

Abstract. Reinforcement Learning-based Combinatorial Optimization (RLCO) has emerged as a powerful alternative to handcrafted heuristics in CO. Most existing RLCO approaches rely on a monolithic learning agent that is prone to premature convergence and lack of behavioral diversity. To address this, we propose Island-DR-ALNS, a novel framework that integrates Deep Reinforcement Learning with a decentralized island model architecture. Instead of a single agent, the proposed method deploys a population of heterogeneous agents that cooperatively optimize the parameters of the Adaptive Large Neighborhood Search (ALNS). To increase agents diversity, Negative Correlated Search is incorporated as a regularization mechanism, which explicitly rewards agents for developing distinct search policies. Furthermore, we introduce a socially-aware state representation that enables agents to contextualize their performance relative to others. Extensive experiments on the Capacitated Vehicle Routing Problem demonstrate that Island-DR-ALNS outperforms state-of-the-art baselines. The framework also exhibits strong generalization capabilities, enabling effective zero-shot transfer to distinct problems such as TSP and mTSP without retraining.

1 Introduction

In recent years the approach to combinatorial optimization has evolved from manually engineered heuristics toward Reinforcement Learning-based Combinatorial Optimization (RLCO) [6,8,22]. This evolution is driven by the need for adaptive, data-driven solvers capable of navigating complex problem spaces, such as those in Vehicle Routing Problems (VRP) or Traveling Salesman Problems (TSP). RLCO methods replace handcrafted decision rules with learned search policies that can automatically extract structural features from problem instances [13]. This shift not only reduces the need for expert knowledge and manual parameter tuning but also enables the discovery of sophisticated search strategies that traditional heuristics often fail to capture.

However, a significant challenge remains at the heart of the RLCO paradigm: most current approaches rely on a monolithic learning agent. While these neural models excel at optimizing a specific search trajectory, they are prone to converging toward deterministic, dominant policies. This often leads to a lack of behavioral diversity, with the agent

effectively "over-specializing" in a particular region of the strategy space, leaving other potentially superior regions unexplored.

Evolutionary Computation (EC) has addressed the challenge of effective exploration through the use of island-based (multi-deme) models [36,2]. These frameworks distribute the search across multiple independent populations (islands) to maintain global diversity and prevent the search from being trapped in local optima. While island models provide a robust structural mechanism for diversification, they traditionally rely on static, non-adaptive rules to govern the search within each island.

In this work, we bridge these two domains. The intuition behind this integration is rooted in a natural synergy: RLCO provides the local information and adaptability mechanism, while island models provide the global *structure* for diversification. Combining them allows us to move from a single, dominant search policy to a heterogeneous population of learners [19]. This framework allows agents to specialize in distinct regions of the search space or different phases of the optimization process (e.g., some focusing on aggressive exploration and others on fine-grained exploitation) without converging into a single, mediocre strategy.

Specifically, using Adaptive Large Neighborhood Search (ALNS) [27] as an underlying search mechanism, we introduce Island Deep Reinforcement Learning optimization algorithm (Island-DR-ALNS)–a novel framework that fills the above gap. Unlike the standard single-agent approach (e.g. [26] or [12]), our method employs a population of heterogeneous agents, each controlling its own local search process. To prevent these agents from converging to similar behaviors, we incorporate Negative Correlated Search (NCS) [32] as a regularizer, explicitly rewarding the learning of distinct search policies. By distributing a computational budget across cooperating islands and facilitating the exchange of solutions via a migration mechanism, we demonstrate that learned search policies lead to more robust, high-quality solutions than a single, monolithic controller.

The main contributions of this paper are as follows:

- **Multi-Agent Island Architecture for Hyperheuristics.** We propose replacing the monolithic DRL agent with an ensemble of independent learners. We show that a distributed island model significantly outperforms the single-agent baseline [26] and other SOTA methods (e.g., DRLH [12]) within the same computational budget.
- **Behavioral Diversity via Negative Correlated Search.** We integrate an NCS-based loss function that penalizes policy similarity between agents. We empirically demonstrate that this forces the emergence of diverse search strategies without manual engineering, leading to higher robustness and solution quality.
- **Socially-Aware State Representation.** We extend the problem-agnostic state space of DR-ALNS with *social* features (e.g., island rank, distance to neighbors), allowing the agents to contextualize their performance relative to the population and adjust their strategies dynamically.
- **Zero-Shot Cross-Problem Transferability.** We demonstrate that the learned policy ensemble generalizes beyond the problem it was trained on. An Island-DR-ALNS model trained on Capacitated VRP (CVRP) instances can be successfully applied to TSP or mTSP problems without retraining, achieving competitive results. It confirms that agents learn *generalizable search mechanisms* rather than problem-specific rules.

2 Related Work

Learning to Search in ALNS. Adaptive Large Neighborhood Search (ALNS), introduced by [27], remains a dominant metaheuristic for routing and scheduling problems. Its performance relies heavily on the adaptive layer that selects destroy and repair operators. While traditional implementations use handcrafted scoring mechanisms [21], recent trends have shifted towards replacing these heuristics with data-driven policies, a paradigm known as Learning to Search (L2S) [33].

Early works, such as [11], utilized Deep Neural Networks to learn repair operators directly. However, learning low-level actions can be computationally expensive and problem-specific. A more generalizable approach involves learning high-level decisions (operator selection). The direct predecessor of our work, DR-ALNS [26], employs a DRL agent to dynamically select operators and adjust parameters (e.g., Simulated Annealing temperature) based on a problem-agnostic state representation. Similarly, approaches like DRLH [12] use DRL as a hyper-heuristic controller. These methods outperform classical ALNS, however they rely on a single agent that tends to converge to a deterministic policy, potentially getting trapped in local optima regardless of their search strategy.

Island Models and Cooperative Search. To overcome the limitations of single-population search, island models partition the population into sub-populations that evolve independently and interact via migration [36,14]. This setup promotes maintaining diverse genetic material and enables parallel exploration of different regions of the search space. It was shown that the way migration is handled (specifically how often and how much data is migrated) highly affects the results [30,29]. Similarly, [19] highlighted the trade-offs between topology, migration policies, and the homogeneity of the islands, proving the efficiency of heterogeneous setups.

The island model was initially used in Genetic Algorithms, and subsequently in other metaheuristics like Differential Evolution [3] or Particle Swarm Optimization [1]. Modern developments have introduced dynamic setups. Some models use clustering to better match the problem's structure [23], while other automatically adjust island connections and migration rules based on how the population is performing [9,4]. A recent work [40] proposed a Diversity-driven Cooperating Portfolio of Metaheuristics (DdCPM), demonstrating that heterogeneous islands exchanging solutions based on diversity metrics visibly outperform monolithic baselines. This direction was further extended by the adaptive metaheuristic island assignement [39] and adaptive fitness-and-diversity based migrations, outperforming traditional fixed-interval migration strategies [41]. However, existing island models rely on predefined metaheuristic portfolios, rather than employing learning-based agents that adapt to the local search dynamics.

Diversity and Negative Correlated Search. A critical challenge in both EC and RL is preventing premature convergence. In the context of ensemble learning and population-based RL, simply running multiple agents in parallel often leads to learning identical or very similar policies [24]. To address this, explicit diversity preservation mechanisms are required [7]. Negative Correlated Search (NCS) [32,15], serves as a powerful regularization technique. NCS explicitly penalizes correlations between the behaviors of in-

dividual learners, facilitating coordinated parallel exploration [37] and encouraging the population to adapt to different modes of the solution space. This concept has been successfully adapted in DRL–in game playing or continuous control–to train diverse populations of agents that cooperatively exhibit robust behavior [17]. Furthermore, competitive co-evolution strategies mitigating the few-shots learning challenge [31] and surrogate-assisted ensembles of solvers [38] have been recently shown to be highly effective in generating diverse and generalizable parallel algorithm portfolios for combinatorial optimization problems.

Novelty of the Proposed Approach. Island-DR-ALNS extends DR-ALNS in the following 3 aspects: (1) It presents a multi-agent approach (compared to single-agent DR-ALNS); (2) It extends the online learning capabilities of DRL by adding island-based diversity schemes; (3) While multi-population architectures and NCS are individually well established, our novelty lies in combining them to mitigate policy homogenization in RLCO. By integrating NCS into a multi-agent DR-ALNS framework, we propose the hyper-heuristic that not only learns to configure ALNS online but also enforces a heterogeneous population of search strategies, combining the adaptability of RL with the exploration power of diversity-driven island models.

3 Proposed Island-DR-ALNS

Our method (see Figure 1) extends a single-agent approach by introducing a population of N heterogeneous agents, denoted as $\mathcal{A} = \{A_1, ..., A_N\}$. Each agent controls an independent ALNS process within a separate population (island). Agents are trained cooperatively to maximize individual performance while simultaneously maximizing the behavioral diversity of the entire population through NCS. Unlike traditional evolutionary island models that rely on crossover and mutation, the search within each island is driven exclusively by the local DR-ALNS agent iteratively refining its solution.

3.1 Island Maintenance

The agent maintains the island population through a Markov Decision Process. Each agent's interaction within its specific search environment is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P})$, where $\mathcal{S}$ denotes the **state space**, defined through *Island Search Dynamics* and *Social Context* (see below); $\mathcal{A}$ represents the **action space**, comprising the discrete choices for heuristic selection and parameter configuration; $\mathcal{R}$ is the **reward function**, which provides a scalar feedback signal to guide the agent towards optimal search behaviors; $\mathcal{P}$ signifies the **state transition probability function**, describing the likelihood of transitioning to a subsequent state s_{t+1} given a state s_t and action a_t.

State Space. The standard DR-ALNS state space focuses solely on the local search dynamics, described in the upper part of Table 1. To enable agents to reason about their relative performance vs. the global population state, we extend the state definition with *social features* (lower part of Table 1), which are broadcasted asynchronously to strictly preserve the non-blocking, parallel execution of the framework. The extended state definition s_t consists of these two groups of features.

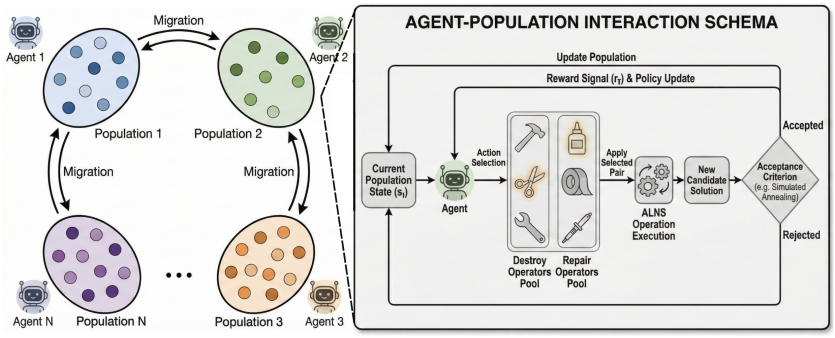

Fig. 1. Island-DR-ALNS framework. The left panel depicts a decentralized population structure organized in a ring topology, with islands cooperating through a migration mechanism. The right panel presents the interaction schema between an agent and the respective population, detailing the local loop where the agent selects an action based on the current search state to iteratively refine the solution.

Action Space. The action space $\mathcal{A}$ follows the configuration of the ALNS heuristics. At each step t, the agent selects a composite action a_t consisting of four discrete components. To maintain consistency for a fair comparison with the previous single-agent approach, our framework utilizes the action space defined in DR-ALNS [26]. Further implementation details can be found in the original work, including: **Destroy Operator ($d \in \Omega^-$) -** selects a heuristic to remove parts of the solution (e.g., Random Removal, Worst Removal, Related Removal); **Repair Operator ($r \in \Omega^+$) -** selects a heuristic to reconstruct the solution (e.g., Greedy Insertion, Regret Insertion); **Destroy Severity (δ) -** a discrete value (1-10) determining the percentage of the solution to be destroyed (from 10% to 100%); **Acceptance Parameter (T) -** the temperature for the Simulated Annealing (SA) criterion, selected from a discrete set of 50 values.

Reward Function. The reward function $\mathcal{R}$ is tailored to the decentralized island architecture. We implement a hierarchical base reward structure: a reward of $+1$ is assigned for finding a solution that improves upon the current individual candidate, $+5$ for discovering a new best solution within the local island population, and $+10$ for identifying a new global best solution across the entire framework. This multi-level feedback ensures that agents are motivated for both steady local refinement and significant global breakthroughs. During training, this base reward is augmented by a diversity term derived from NCS, which explicitly encourages agents to explore distinct regions of the strategy space and prevents policy homogenization.

3.2 Training via Negative Correlated Search

To ensure behavioral diversity among agents and prevent the population from converging into a single, homogeneous policy, we utilize the NCS mechanism [32], which explicitly penalizes similarity between the agent policies.

Table 1. State space features. The top section lists local (within-island) search dynamics features. The bottom section introduces social (inter-island) features.

Feature	Description
	Island Search Dynamics
Best Improved	Has the best global solution improved?
Current Accepted	Has the last candidate solution been accepted?
Current Improved	Is the current solution better than the previous one?
Is Current Best	Is the current solution the best found so far?
Cost Difference	Relative gap between current and best solution objective.
Stagnation Count	Number of iterations without improvement.
	Social Context (New)
Island Rank (Best)	Rank of this island's best solution within the population.
Island Rank (Avg)	Rank of this island's average population fitness.
Island Distance	Normalized distance to the closest neighboring island.
Migration Time	Iterations since the last migration event.

The training objective is to minimize a total loss function $L(\theta_i)$ for each agent i parameterized by θ_i, which is a weighted combination of the Proximal Policy Optimization (PPO) loss [28] and the correlation penalty:

$$L(\theta_i) = L_{PPO}(\theta_i) - \lambda \mathbb{E}_{s\sim\rho}[C_i(s)] \tag{1}$$

where λ is a regularization coefficient that controls the strength of the diversity pressure. L_{PPO} refers to the standard Proximal Policy Optimization objective [28], which ensures stable and efficient policy updates by clipping large policy changes to prevent performance collapse.

The correlation penalty C_i for agent i is defined as the average distance between its policy and the policies of all other agents $j \neq i$:

$$C_i = \frac{1}{N-1} \sum_{j\neq i} D(\pi_{\theta_i}, \pi_{\theta_j}) \tag{2}$$

D denotes a distance metric used to quantify the behavioral difference between policies. This distance is calculated over trajectories consisting of m sequential decision steps. In each step, an agent observes a state s_t, executes an action a_t, and transitions to a new state s_{t+1}. D measures the distance between the distributions of actions that given agents' policies would perform across all individuals (in all islands) when executing these sequences. It fosters the emergence of specialized strategies such as aggressive exploration or fine-grained exploitation.

3.3 Migration Mechanism

Migration is a fundamental component of island-based architectures, acting as a structural mechanism for global cooperation by facilitating the exchange of high-quality solutions between isolated sub-populations. While NCS ensures diversity at the level of

search behavior, migration operates at the level of *search results*. This dual approach prevents the population from being trapped in local optima: heterogeneous policies explore the search space via distinct strategies, while migration ensures that breakthroughs made by one agent can be exploited by others to re-ignite exploration in stagnant islands. Following the findings regarding the effective migration strategies from [40], we implemented the following diversity-driven migration strategy (see [40] for details).

- **Topology (*Where*).** We employ a *Ring* topology, where migration occurs exclusively between immediate neighbors. This restricted connectivity enables maintaining global genetic diversity and preventing the premature convergence, often occurring in fully connected models.
- **Timing (*When*).** Migration is triggered asynchronously based on the internal state of each island. Specifically, an island initiates a request when it detects a local stagnation, defined as k consecutive iterations without improvement of its (local) best solution. While NCS prevents policy homogenization globally, individual agents may still converge to isolated local optima within their specific search trajectories, necessitating migration.
- **Content (*What*).** To maintain the structural diversity of the solution pools, we avoid the greedy selection of the best-fitted individual. Instead, we select a solution that maximizes a weighted sum of the fitness and contribution to the target island's population diversity.
  Specifically, the solution x^* from the source island $\mathcal{P}_{src}$ is selected for migration to the target island $\mathcal{P}_{tgt}$ as:

$$x^* = \arg\max_{x \in \mathcal{P}_{src}} \left[(1-\alpha) \cdot \text{Fit}(x) + \alpha \cdot \text{Div}(x, \mathcal{P}_{tgt})\right]$$

  where $\text{Fit}(x)$ represents the normalized fitness of solution x, $\text{Div}(x, \mathcal{P}_{tgt})$ denotes its contribution to the diversity of the target island (calculated as the average distance to all individuals in $\mathcal{P}_{tgt}$), and $\alpha \in [0, 1]$ is a weighting coefficient.
  This ensures that migrated solutions provide novel information rather than merely cloning identical structures across the entire framework, which inherently dictates the diversity and selection pressure of the parallel populations.

3.4 Agent Architecture

Each agent is modeled as a Multi-Layer Perceptron (MLP) with two hidden layers of 64 units and *tanh* activation functions. The output layer consists of 4 heads (with *softmax* function) corresponding to the four discrete action components described in Section 3.1.

4 Experimental Setup

To evaluate the efficacy of the proposed Island-DR-ALNS framework, we conducted extensive experiments on the CVRP [25,20] - a standard benchmark for RLCO. The source code is available on the website [42].

Training Set. The models were trained on procedurally generated CVRP instances with the number of customers in a range of $[20, 150]$. Coordinates were sampled uniformly from the unit square $[0, 1]^2$, and demands were sampled uniformly from $\{1, \ldots, 9\}$. This generation process ensures that the agent learns a general search policy rather than overfitting to specific graph structures.

Test Set. For evaluation, we utilized 15 established benchmark instances from the CVRPLIB repository [35,34]. To analyze performance across different complexity levels, we categorized these instances into three groups of 5 based on the number of customers (c_n): *Small* ($c_n \in [20, 50]$), *Medium* ($c_n \in [51, 100]$), and *Large* ($c_n \in [101, 150]$).

Baselines and Comparison. Our comparison focuses on data-driven and adaptive control of ALNS to ensure a fair evaluation of the learning component. Consequently, we excluded general population-based metaheuristics and highly specialized CVRP solvers that rely on fundamentally different search mechanisms, as introducing entirely different underlying operators would make it difficult to separate the impact of handcrafted local actions from the efficacy of the high-level learning policy. Hence, we compared our method against the following 4 baselines:

1. **ALNS-Vanilla.** The standard ALNS implementation (without training) with fixed parameters based on [27].
2. **ALNS-BO.** ALNS tuned via Bayesian Optimization using the SMAC3 library [16], representing a strong, offline-tuned heuristic.
3. **DRLH.** A Deep Reinforcement Learning Hyper-heuristic framework [12].
4. **DR-ALNS.** The single-agent state-of-the-art method [26], which serves as a direct baseline for our multi-agent extension.

Training and Hyperparameters. The agents were trained using the PPO algorithm [28] augmented by the NCS-based diversity objective. The "Social Context" features (see Table 1) were updated in real-time, enabling agents to dynamically perceive and react to the search progress of other agents. Overall, the training process spanned 300,000 steps globally (divided equally among the N agents to match the single-agent baseline budget) - 30 search runs $\times$ 10^4 evaluations, where each evaluation triggered a reward calculation and policy update. The regularization coefficient for NCS was set to $\lambda = 0.2$ based on preliminary experiments. For the island model, we utilized $N = 6$ islands connected in a Ring topology. To quantify the behavioral difference between policies we used Bhattacharyya distance as a distance metric (D in Eq. 2) with behavior trajectories computed over $m = 3$ sequential decision steps. While the multi-agent framework introduces a few additional parameters, it is important to note that they do not require per-instance tuning. As detailed in the Supplementary Material [42], the method is robust to these choices and the default configuration generalizes effectively across different problem sizes and routing variants.

To ensure a fair comparison, all methods were restricted to the same total computational budget of 10^6 objective function evaluations. For the island model, the budget was distributed equally among the N agents. Detailed discussion of hyperparameter configurations and their influence on results is reported in the Supplementary Material [42].

Evaluation Metrics and Statistical Analysis. During inference, the learned policies are fixed, and the agents cooperated solely through migration to collectively solve test instances. We report the performance using the *Optimality Gap* (%), defined as:

$$Gap_{alg} = \frac{Cost_{alg} - Cost_{opt}}{Cost_{opt}} \times 100\% \tag{3}$$

where $Cost_{alg}$ is the best solution found by the algorithm and $Cost_{opt}$ is the known optimal (or best known) solution. Optimality Gap was preferred over raw objective values, as it provides a scale-independent measure of quality, allowing aggregation across instances of varying sizes.

To verify the statistical significance of results, we performed the Wilcoxon signed-rank test with a significance level of $p < 0.05$. All results reported in the following section are averages over 20 independent runs with different random seeds. All experiments were conducted on an Intel Xeon Silver 4116 @2.10GHz.

5 Results and Analysis

In this section, we present the performance results of the proposed Island-DR-ALNS framework. We analyze the solution quality across different instance sizes, examine the convergence behavior, and validate the contribution of individual components through an ablation study. Finally, we investigate the generalization capabilities of our agents across different problem sizes and combinatorial problems (Zero-Shot Transfer).

Main Performance Comparison. Table 2 compares the Island-DR-ALNS against the baselines on the CVRP test set. To provide a granular analysis, we report the optimality gap (%) aggregated into three categories: *Small*, *Medium*, and *Large*. Detailed results for each benchmark instance are presented in the Supplementary Material [42].

Table 2. Comparison of optimality gaps (Eq. 3) on 15 CVRP test instances averaged over 20 runs per instance and aggregated by category (5 instances per size).. "Computation time" indicates the average wall-clock time over all instances for the inference (search) process. Best results are **bolded**. Detailed per instance results are presented in the Supplementary Material [42]. The statistical analysis (Wilcoxon signed-rank test with $p < 0.05$ evaluated per instance across runs

) shows that Island-DR-ALNS yields significantly superior solutions in 11 out of the 15 benchmark instances (2 Small, 4 Medium, and 5 Large) compared to the strongest baseline (DR-ALNS).

Method	**Optimality Gap (%)**			**Computation**	**Training**
	Small	**Medium**	**Large**	**time [s]**	**time [min]**
ALNS-Vanilla	4.15 ± 0.48	5.12 ± 0.62	6.42 ± 0.85	$\mathbf{31.2 \pm 1.5}$	$\mathbf{0.0 \pm 0.0}$
ALNS-BO	3.55 ± 0.42	4.20 ± 0.51	5.21 ± 0.68	34.5 ± 2.1	265.5 ± 12.4
DRLH	2.10 ± 0.31	3.15 ± 0.45	4.50 ± 0.58	73.3 ± 4.8	1472.2 ± 45.6
DR-ALNS	1.45 ± 0.24	2.05 ± 0.31	2.71 ± 0.44	63.4 ± 3.5	147.6 ± 8.2
Island-DR-ALNS	$\mathbf{0.85 \pm 0.15}$	$\mathbf{1.18 \pm 0.21}$	$\mathbf{1.63 \pm 0.36}$	78.4 ± 5.2	180.3 ± 10.5

The results demonstrate that Island-DR-ALNS achieves the lowest optimality gap across all categories, with an average gap of **1.22%**, significantly outperforming DR-ALNS (2.07%). The single-agent DR-ALNS performs relatively well on Small instances, but its performance degrades on larger graphs (the gap increases to 2.71%).

Despite the natural performance decline associated with the problem complexity, Island-DR-ALNS proves to be a more robust solver, consistently achieving significantly lower optimality gaps regardless of the instance size. Although the proposed method introduces a moderate computational overhead in sequential execution, this is a favorable trade-off for the substantial improvements in solution quality. Furthermore, the decentralized island architecture natively supports parallel execution and in practical deployments running the agents concurrently across multiple threads would effectively eliminate this wall-clock time difference.

Convergence and Stability To analyze the search dynamics, Figure 2 presents the training and inference (search) convergence.

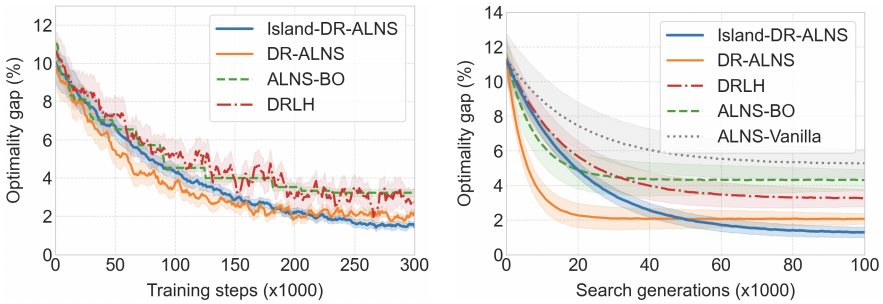

Fig. 2. Left: Training convergence of the compared methods. The optimality gap is reported as the average across all test instances, evaluated periodically throughout the training process. Right: Search convergence of compared methods (inference phase, after training).

The convergence plots confirm that Island-DR-ALNS reaches lower optimality gaps in both training and testing performance. Single-agent baselines (DR-ALNS, DRLH) exhibit faster initial convergence but plateau earlier at sub-optimal levels. This behavior is not surprising, as distributing the fixed evaluation budget across N islands slows down the initial per agent experience accumulation, though this diversity ultimately enables the ensemble to escape local optima and reach a superior final solution.

Beyond convergence speed, we analyzed the stability of the proposed framework, showing that Island-DR-ALNS exhibits lower optimality gap variance compared to the single-agent baseline. Specifically, regarding *training consistency*, the standard deviation of the final gap over 20 independent trainings (across all training instances) dropped from 0.101 for DR-ALNS to 0.076 for our method. This trend is even more visible in *inference repeatability*, where the standard deviation of 20 search processes (averaged over all test instances) decreased from 0.332 to 0.242. The results confirm that the coop-

erative population acts as a stabilizing mechanism, mitigating the inherent stochasticity of DRL and ensuring more predictable performance.

Diversity Analysis. To verify that NCS induces meaningful behavioral diversity rather than random noise, we analyzed the agents' policy distributions collected during the inference phase on the CVRP test instances. Figure 3 visualizes the aggregated action probabilities (top) and a t-SNE projection [18] of the agents' action vectors (bottom).

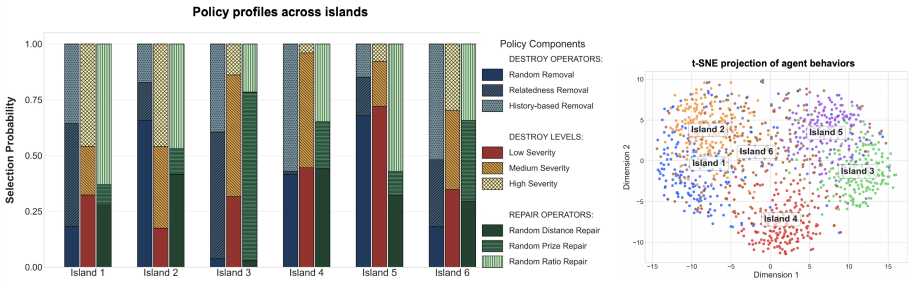

Fig. 3. Visualization of the emergent behavioral diversity across the Island-DR-ALNS population. The left figure illustrates the selection probabilities for actions, highlighting how different islands specialize in distinct search tactics (e.g., Island 1 focuses on high-severity exploration while Island 3 prioritizes low-severity refinement).The right figure shows a t-SNE projection of the action distributions, confirming that agents occupy non-overlapping behavioral regions due to the NCS diversity pressure.

The analysis reveals clear emergent specialization. The histograms show that the agents spontaneously adopt distinct roles, e.g., Island 1 prioritizes *High Severity* destroy operators (aggressive exploration), whereas Island 5 favors *Low Severity* moves (fine-grained exploitation). This division emerges in the training process and is not hard-coded. The t-SNE projection confirms this by displaying distinct clusters. The lack of significant overlap between the clusters indicates that the NCS mechanism successfully forced the agents to occupy different regions of the strategy space. Island-DR-ALNS relies on a robust portfolio of heuristics, capable of handling various problem structures which is a key advantage over the monolithic, single-policy approach.

Ablation Study. To delve deeper into the source of the performance gains, we performed a step-by-step ablation study (see Table 3). We started with the single-agent baseline and gradually added our contributions: (1) The Island architecture (without NCS and Social features), (2) The Diversity objective (NCS), and (3) The Socially-Aware State.

It comes from Table 3 that simply parallelizing the agents into islands (with migrations) but agents training is independent) improves the optimality gap to 1.74% compared to initial 2.07% for the DR-ANLS setup. A similar gain (a gap decrease to 1.37%) comes from introducing NCS. This confirms that explicit diversity pressure is one of the key performance drivers. Finally, adding the Social State features (informing the agents

Table 3. Ablation study. Each row represents a cumulative extension of the previous configuration with the specified component, culminating in the full Island-DR-ALNS framework. Shaded cells indicate results that are statistically significantly better relative to the configuration without the extension (the row above).

Configuration	Optimality Gap (%)	Time [s]
Baseline (DR-ALNS)	2.07	**63.4**
+ Islands	1.74	73.3
+ Diversity (NCS)	1.37	75.2
+ Social State (Full Method)	**1.22**	78.4

about the population performance) further reduces the gap to 1.22%. Note that the '+ Islands' configuration effectively represents independent DR-ALNS runs, confirming the performance gain is driven by our explicit diversity mechanisms rather than a multi-start ensemble effect.

Analysis of Social State Features. To further check the impact of the proposed Socially-Aware State representation, we conducted a granular ablation study focusing on the four novel features introduced in Table 1: Island Rank based on best solution (IB), Island Rank based on population average (IA), Distance to closest island (D), and Time since last migration (M). We evaluated the contribution of these features using two complementary strategies: *Additive Analysis* - starting from the standard problem-agnostic state (denoted as O), we gradually added each social feature one by one to assess its marginal gain; *Subtractive Analysis:* - starting from the full state representation (ALL), we gradually removed features one by one to measure the performance drop (feature importance). The *Baseline* (B) baseline corresponds to the Island model with NCS but without social context (1.37% gap), while ALL represents the complete version of the proposed Island-DR-ALNS method (1.22% gap).

Table 4. Impact of individual Social State features on the Optimality Gap (denoted as "Gap"). The "Baseline" (B) configuration refers to the state definition from standard DR-ALNS, while IB, IA, D, and M denote the added social features (cf. Table 1).

Additive Strategy		Subtractive Strategy	
Config	**Gap (%)**	**Config**	**Gap (%)**
B	1.37	ALL	**1.22**
$B + IB$	1.34	$ALL - IB$	1.24
$B + IA$	1.31	$ALL - IA$	1.27
$B + D$	1.33	$ALL - D$	1.26
$B + M$	1.31	$ALL - M$	1.27

The analysis reveals that while all social features contribute to the performance improvement, they are not equally significant. In the additive test, including the *Rank*

(Avg) (IA) or *Migration Time* (M) yields the largest initial reduction in the optimality gap (from 1.37% to 1.31%). This observation is confirmed by the subtractive analysis: removing either IA or M from the Island-DR-ALNS formulation results in the most severe performance degradation (gap increases to 1.27%). The results suggest that an agent's awareness of its population average quality (relative to others) and the "freshness" of its genetic pool (time since migration) are the most critical factors.

Generalization and Transferability. A key promise of Learning to Search is the ability to generalize beyond the training distribution. We evaluate this aspect in two scenarios: *Within-Problem Generalization* and *Zero-Shot Cross-Problem Transfer*. The details regarding the experimental setups, including the list of problem instances, and the complete results are provided in the Supplementary Material [42].

Within-Problem Generalization. To assess the robustness of the learned policies, we evaluated their generalization performance on two problem scales *Small* (20-50 nodes) and *Large* (101-150 nodes) in two directions: *Upward* (training on Small, testing on Large) and *Downward* (training on Large, testing on Small). Table 5 summarizes the comparative results of Island-DR-ALNS vs. DR-ALNS. In the Upward scenario, while both methods naturally degrade on unseen larger graphs, Island-DR-ALNS exhibits better performance (+0.52 vs. +1.14 decay). In the Downward scenario, the single-agent baseline struggles to adapt its strategies to simpler problems (a gap increases by 0.40), whereas our ensemble demonstrates only a slight increase of 0.06. The results indicate that a single agent overfits to the strategies suitable for given graphs sizes, whereas a diverse ensemble maintains a broader set of heuristics, some of which can potentially generalize better to different-scale problems.

Table 5. Cross-Size Generalization Results. Left: Upward (Train Small → Test Large). Right: Downward (Train Large → Test Small). Decay quantifies performance degradation relative to training and testing on the same instance size.

Upward: Trained on Small			
Method	**Test: Small**	**Test: Large**	**Decay**
DR-ALNS	1.45	3.85	+1.14
Island-DR-ALNS	**0.88**	**2.15**	**+0.52**

Downward: Trained on Large			
Method	**Test: Large**	**Test: Small**	**Decay**
DR-ALNS	2.71	1.85	+0.40
Island-DR-ALNS	**1.63**	**0.94**	**+0.06**

Cross-Problem Transfer. Next, we evaluated the zero-shot transferability by applying the model trained on CVRP directly to solving TSP [10] and mTSP (Multiple TSP) [5] instances, without any fine-tuning. For mTSP, we addressed the Min-Max Single-Depot variant, which focuses on minimizing the length of the longest route among all salesmen. While TSP is structurally a subset of CVRP (effectively a single-vehicle problem with infinite capacity), the Min-Max mTSP objective represents a fundamental shift from total cost minimization. As detailed in Supplementary Material [42], successfully adapting to this rigorous test effectively validates the framework's broader applicability to distinct routing variants. We compared a model trained directly on the target problem

(TSP or mTSP, resp.) using the Island-DR-ALNS algorithm, and two models transferred from CVRP: a single-agent DR-ALNS one and an Island-DR-ALNS one.

Table 6. Zero-Shot Transfer. Gap (%) on TSP and mTSP using a model trained on CVRP.

Configuration	**TSP gap**	**mTSP gap**
Island-DR-ALNS (Trained on the target problem)	*0.05*	*0.42*
DR-ALNS (Transferred from CVRP)	1.12	2.35
Island-DR-ALNS (Transferred from CVRP)	**0.35**	**1.15**

The results presented in Table 6 show that the Island-DR-ALNS model trained on CVRP achieves a gap of 0.35% on TSP and 1.15% on mTSP, which is competitive with specialized heuristics and significantly better than the single-agent transfer (1.12% and 2.35%, resp.). This suggests that the island population learns universal "meta-strategies" of local search that are useful across different routing problems.

All improvements of Island-DR-ALNS over DR-ALNS reported in Tables 5 and 6 are statistically significant according to the Wilcoxon signed-rank test ($p < 0.05$).

6 Conclusions

This work introduces Island-DR-ALNS, a novel multi-agent Deep Reinforcement Learning framework for the online control of Adaptive Large Neighborhood Search. By replacing the traditional monolithic agent with a heterogeneous population of cooperating learners, the proposed method addresses the critical issue of behavioral diversity loss often observed in single-agent approaches. The integration of Negative Correlated Search as an intrinsic diversity objective, coupled with a socially-aware state representation, enables individual agents to autonomously differentiate their search strategies, leading to a robust optimization approach.

The extensive evaluation of Island-DR-ALNS on CVRP benchmarks demonstrates that the method clearly outperforms state-of-the-art baselines, reducing the average optimality gap from 2.07% (single-agent DR-ALNS) to 1.22% while maintaining the same total computational budget. Specifically, as confirmed by the ablation analysis, the explicit diversity pressure from NCS drives the parallel exploration of disjoint heuristic spaces, while the socially-aware state and migrations act as complementary exploitation mechanisms, allowing the ensemble to rapidly capitalize on breakthrough solutions without suffering from premature convergence. Furthermore, the learned policies not only scale effectively to problem instances of different sizes but also exhibit zero-shot transferability to distinct problems such as TSP and mTSP. This zero-shot transferability indicates that the island population learns universal, high-level search heuristics rather than problem-specific strategies.

Limitations A certain limitation of the proposed framework is the increased configuration complexity due to additional hyperparameters.

Acknowledgments Xin Yao's work was partially supported by an internal grant from Lingnan University. Jacek Mańdziuk was partially supported by the National Science Centre grant number 2023/49/B/ST6/01404.

References

1. Abadlia, H., Smairi, N., Ghedira, K.: Particle swarm optimization based on dynamic island model. In: 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI). pp. 709–716. IEEE (2017)
2. Akay, R., Basturk, A., Kalinli, A., Yao, X.: Parallel population-based algorithm portfolios: An empirical study. Neurocomputing **247**, 115–125 (2017)
3. Apolloni, J., Leguizamón, G., García-Nieto, J., Alba, E.: Island based distributed differential evolution: an experimental study on hybrid testbeds. In: 8th International Conference on Hybrid Intelligent Systems. pp. 696–701. IEEE (2008)
4. Araujo, J.N., Batista, L.S.: A diversity-driven migration strategy for distributed evolutionary algorithms. Swarm and Evolutionary Computation **82**, 101361 (2023)
5. Bektas, T.: The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega **34**(3), 209–219 (2006)
6. Bello, I., Pham, H., Le, Q.V., Norouzi, M., Bengio, S.: Neural combinatorial optimization with reinforcement learning. In: International Conference on Learning Representations (2017)
7. Eysenbach, B., Gupta, A., Ibarz, J., Levine, S.: Diversity is all you need: Learning skills without a reward function. In: International Conference on Learning Representations (2019)
8. Garmendia, A.I., Ceberio, J., Mendiburu, A.: Neural combinatorial optimization: a new player in the field. arXiv preprint arXiv:2205.01356 (2022)
9. Gozali, A.A., Fujimura, S.: DM-LIMGA: dual migration localized island model genetic algorithm—a better diversity preserver island model. Evolutionary Intelligence **12**(4), 527–539 (2019)
10. Gutin, G., Punnen, A.P.: The traveling salesman problem and its variations, vol. 12. Springer Science & Business Media (2006)
11. Hottung, A., Tierney, K.: Neural large neighborhood search for the capacitated vehicle routing problem. In: European Conference on Artificial Intelligence, pp. 443–450. IOS Press (2020)
12. Kallestad, J., Hasibi, R., Hemmati, A., Sörensen, K.: A general deep reinforcement learning hyperheuristic framework for solving combinatorial optimization problems. European Journal of Operational Research **309**(1), 446–468 (2023)
13. Kool, W., van Hoof, H., Welling, M.: Attention, learn to solve routing problems! In: International Conference on Learning Representations (2019)
14. Leitão, A., Pereira, F.B., Machado, P.: Island models for cluster geometry optimization: how design options impact effectiveness and diversity. Journal of Global Optimization **63**, 677–707 (2015)
15. Li, Y., Lu, X., Yao, X.: Negatively correlated search for constrained optimization. In: 2023 IEEE Congress on Evolutionary Computation (CEC). pp. 1–10. IEEE (2023)
16. Lindauer, M., Eggensperger, K., Feurer, M., Biedenkapp, A., Deng, D., Benjamins, C., Ruhkopf, T., Sass, R., Hutter, F.: SMAC3: A versatile bayesian optimization package for hyperparameter optimization. Journal of Machine Learning Research **23**(54), 1–9 (2022)
17. Liu, S., Tang, K., Yao, X.: Generative adversarial construction of parallel portfolios. IEEE Transactions on Cybernetics **52**(2), 784–795 (2020)

18. Maaten, L.v.d., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research **9**(Nov), 2579–2605 (2008)
19. Mambrini, A., Sudholt, D., Yao, X.: Homogeneous and heterogeneous island models for the set cover problem. In: International Conference on Parallel Problem Solving from Nature. pp. 11–20. Springer (2012)
20. Mańdziuk, J.: New shades of the vehicle routing problem: Emerging problem formulations and computational intelligence solution methods. IEEE Transactions on Emerging Topics in Computational Intelligence **3**(3), 230–244 (2018)
21. Mara, S.T.W., Norcahyo, R., Jodiawan, P., Lusiantoro, L., Rifai, A.P.: A survey of adaptive large neighborhood search algorithms and applications. Computers & Operations Research **146**, 105903 (2022)
22. Martins, M.S., Sousa, J.M., Vieira, S.: A systematic review on reinforcement learning for industrial combinatorial optimization problems. Applied Sciences **15**(3), 1211 (2025)
23. Meng, Q., Wu, J., Ellis, J., Kennedy, P.J.: Dynamic island model based on spectral clustering in genetic algorithm. In: International Joint Conference on Neural Networks (IJCNN). pp. 1724–1731. IEEE (2017)
24. Oroojlooy, A., Hajinezhad, D.: A review of cooperative multi-agent deep reinforcement learning. Applied Intelligence **53**(11), 13677–13722 (2023)
25. Ralphs, T.K., Kopman, L., Pulleyblank, W.R., Trotter, L.E.: On the capacitated vehicle routing problem. Mathematical Programming **94**(2), 343–359 (2003)
26. Reijnen, R., Zhang, Y., Lau, H.C., Bukhsh, Z.: Online control of adaptive large neighborhood search using deep reinforcement learning. In: Proceedings of the International Conference on Automated Planning and Scheduling. vol. 34, pp. 475–483 (2024)
27. Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science **40**(4), 455–472 (2006)
28. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
29. Skolicki, Z.: An analysis of island models in evolutionary computation. In: 7th Annual Workshop on Genetic and Evolutionary Computation. pp. 386–389 (2005)
30. Skolicki, Z., De Jong, K.: The influence of migration sizes and intervals on island models. In: 7th Annual Conference on Genetic and Evolutionary Computation. pp. 1295–1302 (2005)
31. Tang, K., Liu, S., Yang, P., Yao, X.: Few-shots parallel algorithm portfolio construction via co-evolution. IEEE Transactions on Evolutionary Computation **25**(3), 595–607 (2021)
32. Tang, K., Yang, P., Yao, X.: Negatively correlated search. IEEE Journal on Selected Areas in Communications **34**(3), 542–550 (2016)
33. Tang, K., Yao, X.: Learn to optimize—a brief overview. National Science Review **11**(8), nwae132 (2024)
34. Uchoa, E., Fukasawa, R., Poggi, M., et al.: CVRPLIB: A digital library of CVRP instances. `http://vrp.atd-lab.inf.puc-rio.br/` (2025), accessed: 2025-12-28
35. Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., Subramanian, A.: New benchmark instances for the capacitated vehicle routing problem. European Journal of Operational Research **257**(3), 845–858 (2017)
36. Whitley, D., Rana, S., Heckendorn, R.B.: The island model genetic algorithm: On separability, population size and convergence. Journal of Computing and Information Technology **7**(1), 33–47 (1999)
37. Yang, P., Yang, Q., Tang, K., Yao, X.: Parallel exploration via negatively correlated search. Frontiers of Computer Science **15**(5), 155333 (2021)
38. Zakrzewski, G., Yao, X., Mańdziuk, J.: Accelerating parallel algorithm portfolio construction. Procedia Computer Science **270**, 1159–1168 (2025)
39. Żychowski, A., Yao, X., Mańdziuk, J.: Adaptive metaheuristic selection in island-based optimization. Procedia Computer Science **270**, 1119–1128 (2025)

40. Żychowski, A., Yao, X., Mańdziuk, J.: Diversity-driven cooperating portfolio of metaheuristic algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 863–871 (2025)
41. Żychowski, A., Yao, X., Mańdziuk, J.: Migration timing in hybrid island-based metaheuristic algorithms. In: International Conference on Artificial Intelligence and Soft Computing. pp. 292–302 (2025)
42. Żychowski, A., Yao, X., Mańdziuk, J.: Island-DR-ALNS: A Diversity-Driven Multi-Agent Framework for Deep Reinforcement Learning-based Combinatorial Optimization - Supplementary Material and Source Code. https://github.com/zychowskia/Island-DR-ALNS/ (2026), accessed: 2026-03-15