

CIPHEUR: LLM-Guided Structural Search Control for Conflict-Graph Optimization

Yuqing Cheng*
National University of Defense
Technology
Changsha, China
chengyuqing@nudt.edu.cn

Yan Jia*
National University of Defense
Technology
Changsha, China
jia.yan20@nudt.edu.cn

Yanjie Song
Dalian Maritime University
Dalian, China
songyj_2017@163.com

Xiaolu Liu
National University of Defense
Technology
Changsha, China
lxl_sunny@nudt.edu.cn

Lining Xing
Jiangsu University of Technology
Changzhou, China
xln@jsut.edu.cn

Feng Yao
National University of Defense
Technology
Changsha, China
yaofeng@nudt.edu.cn

Abstract

Conflict-graph search can stagnate despite effective operators when interventions fail to address the conflicts blocking progress. A disruptive intervention may also require further search before its value emerges. We propose CIPHEUR, a large language model (LLM)-guided framework that connects conflict evidence to coordinated intervention and recovery through Structured Intervention Units (SIUs). The shared control interface supports offline synthesis of executable policies and online contextual decisions. Asynchronous online control sustains native search during inference, while all policies preserve the original objective and feasibility constraints. Across eight satellite-ground scheduling conflict views with five runs per view and a 900-second evaluation cap, CIPHEUR-online achieves the highest mean objective among 11 configurations and increases mean scheduled contact duration by 14.84% over the greedy baseline GWMIN. It exceeds the strongest external per-view mean in seven of eight views and outperforms CHILS in all 40 paired runs. Within the same runs, its normalized objective improvement from 360 to 900 seconds is 1.50–1.86 times that of the two programmatic controllers. A separately frozen, one-shot LLM-generated controller also improves the mean objective over CHILS without runtime model calls. Together, these results demonstrate how structural search control translates LLM design knowledge and contextual reasoning into improved solution quality and sustained search progress.

Keywords

Large language models, Search control, Combinatorial optimization, Hyper-heuristics, Maximum-weight independent set

ACM Reference Format:

Yuqing Cheng, Yan Jia, Yanjie Song, Xiaolu Liu, Lining Xing, and Feng Yao. 2027. CIPHEUR: LLM-Guided Structural Search Control for Conflict-Graph

*Yuqing Cheng and Yan Jia contributed equally to this work.



This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 26th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2027), M. Baldoni, F. Fang, W. Yeoh, N. Yorke-Smith (eds.), 3–7 May 2027, Hanoi, Vietnam. © 2027 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

Optimization. In *Proc. of the 26th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2027)*, Hanoi, Vietnam, 3–7 May 2027, IFAAMAS, 10 pages.

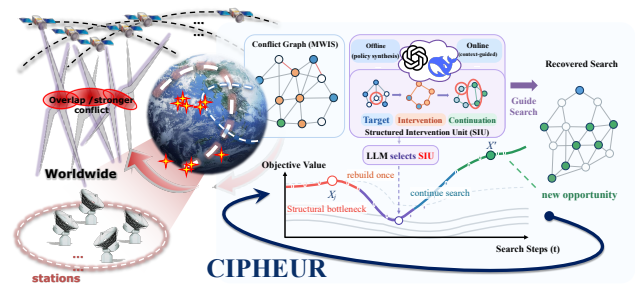


Figure 1: CIPHEUR motivation. Conflict evidence guides structural intervention and recovery in satellite-ground scheduling.

1 Introduction

Effective combinatorial optimization depends on matching search actions to the opportunities available in the current state. Learning-assisted optimization and selection hyper-heuristics have made such decisions an explicit target of algorithm design [3, 10, 12]. Learning to schedule existing primal heuristics, for example, improves the allocation of search effort within a solver [8]. In conflict-graph optimization, this allocation also has a structural dimension: the value of an operation depends on the activities it displaces and the combinations it makes accessible. Exploiting this relationship calls for control decisions that connect observed conflicts to the subsequent course of search.

We study maximum-weight independent set (MWIS) optimization on resource-labelled conflict graphs. Weighted vertices represent candidate activities, and edges encode pairwise incompatibility. In satellite-ground scheduling, resource and temporal labels describe the conflicts among communication opportunities. Two search trajectories can attain similar objective values while excluding different combinations of activities, making the choice of

intervention target consequential. A reconstruction can also reduce a trajectory’s immediate value while creating opportunities for later improvement or exchange. These dependencies motivate the joint choice of *how to intervene*, *where to act*, and *how search should continue*. The resulting control problem links structural targeting with the delayed effects of intervention (Figure 1).

Large language models (LLMs) offer complementary capabilities for addressing this problem. They can construct executable heuristics from problem descriptions and evaluative feedback [22, 26, 38], and make state-dependent heuristic choices during optimization [35, 40]. Bringing these capabilities to conflict-graph search requires an actionable connection between contextual judgment and structural change. A useful decision must identify its target, organize the intervention and its continuation, and remain meaningful as the solver progresses. This motivates a common control abstraction through which both generated policies and runtime reasoning can direct a continuing search.

We propose **CIPHEUR**, an LLM-guided structural search-control framework for conflict-graph optimization. The framework translates conflict evidence and search experience into coordinated decisions about how to redirect an ongoing search. To express these decisions, we introduce the **Structured Intervention Unit (SIU)**, which jointly specifies a structural target, an intervention, and its continuation. Candidate SIUs are grounded in the conflicts affecting current search trajectories. The solution-quality changes observed across executed intervention–recovery periods inform subsequent choices, closing the loop between structural evidence and search behavior.

Two working modes realize control over this common SIU space. In the *offline mode*, an LLM translates problem semantics and the control specification into customized executable policies. These policies select SIUs from evolving observations and may update their internal estimates from accumulated execution feedback. In the *online mode*, an LLM makes the same type of SIU choice directly from current structural evidence and search history. Offline synthesis provides reusable control logic, while online reasoning retains contextual judgment during a solve. Both modes act through a shared execution interface that preserves the original objective and feasibility constraints; asynchronous online decisions allow native search to progress throughout model inference.

The contributions are threefold. *First*, we propose CIPHEUR, an LLM-guided structural search-control framework for conflict-graph optimization that turns conflict evidence and search experience into adaptive control of ongoing search. *Second*, we design SIUs that couple structural targets with intervention and recovery, using feedback from executed intervention–recovery periods to guide subsequent choices. *Third*, we develop two policy realizations—offline controller synthesis and online contextual decision-making—over a shared executable interface, with asynchronous online execution sustaining native search and both modes preserving the original objective and feasibility constraints.

Experiments on eight constraint views of one satellite–ground scheduling source, with five runs per view and a 900-second evaluation cap, establish the competitiveness of CIPHEUR across 11 configurations. CIPHEUR-online increases mean scheduled contact duration by 14.84% over the greedy baseline GWMIN. It also outperforms CHILS in all 40 paired runs. A separately frozen, one-shot

LLM-generated controller improves the mean objective over CHILS without runtime model calls, while matched mechanism studies and cross-source evaluations further characterize the framework’s effectiveness.

2 Related Work

2.1 Independent-Set Search and Structural Guidance

MWIS heuristics exploit solution structure to combine intensification with exploration. METAMIS uses efficient local moves and path relinking within a GRASP framework [9], while CHILS combines iterated local search with cooperation based on differences among solutions [13]. Learning-based graph optimization [5, 19] offers another form of structural guidance: Li et al. [20] use graph-convolutional predictions to guide tree search for independent sets. These approaches share the goal of using graph and solution information to expose promising combinations [2].

CIPHEUR studies this relationship at the level of search interventions. Resource-specific blocking evidence connects candidate activities to the trajectories that exclude them, providing a basis for choosing where to redirect search. A coordinated continuation then explores the opportunities created by the intervention. The emphasis is on converting structural evidence into decisions about the course of a population search, complementing work on local moves, solution exchange, and vertex-level guidance.

2.2 Adaptive and Temporally Extended Search Control

Selection hyper-heuristics adapt the use of low-level heuristics to search state and history [10]; dynamic algorithm configuration learns policies for changing algorithm settings during execution [1]. Learned heuristic scheduling similarly allocates effort among existing solver components [8]. Adaptive large neighborhood search coordinates competing removal and insertion heuristics through performance feedback [27], and learned neighborhood selection chooses variables for solver-based reoptimization [15, 34, 39]. Together, these studies establish policy adaptation, effort allocation, and intervention targeting as complementary dimensions of search control [32, 37].

Temporal abstraction provides a further dimension by treating an extended course of action as a decision unit, as formalized by the options framework [30]. SIUs combine structural targeting with an explicit intervention–recovery sequence. A conflict anchor guides reconstruction priorities across the candidate activities, and the accompanying continuation specifies how the altered trajectory contributes to further search. Feedback summarizes quality changes over the executed sequence. This coupling makes the structural scope and observed effects of an intervention jointly available to the controller.

2.3 LLM-Based Algorithm Design and Search Decisions

LLM-based algorithm design couples program generation with empirical evaluation [6, 31]. FunSearch evolves executable programs using systematic scoring [26]; EoH jointly evolves heuristic ideas

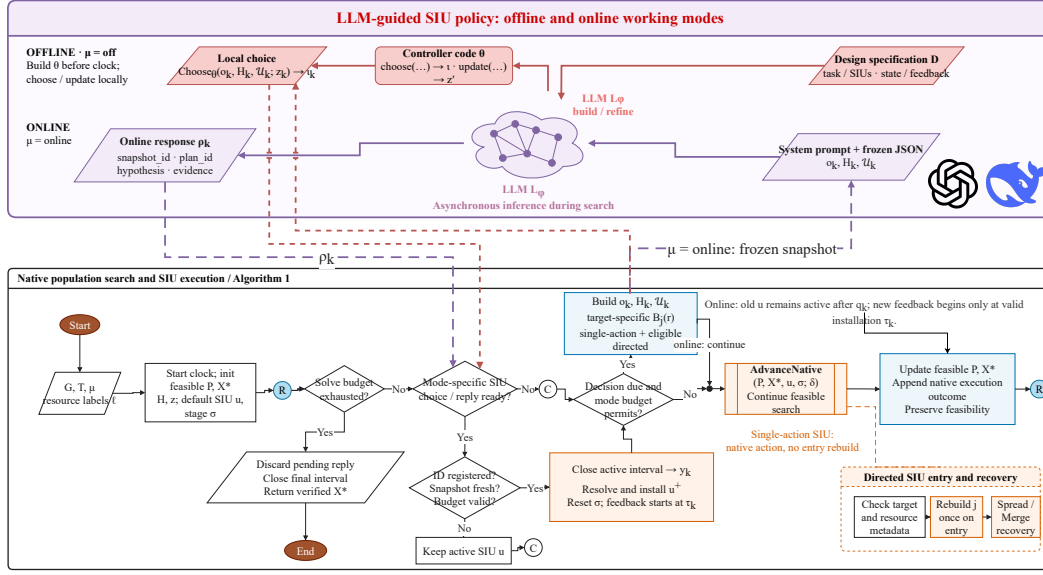


Figure 2: CIPHEUR overview. Both LLM modes select from one SIU registry; native search continues during online inference.

and code [22, 36]; and ReEvo uses reflective feedback to guide this search [24, 33, 38]. MCTS-AHD broadens exploration of heuristic programs through tree search [41], while EoH-S evolves complementary sets of heuristics [21]. These methods retain LLM design knowledge in reusable heuristic programs and portfolios [14, 18, 25].

A related line places model decisions within optimization [7, 16, 17]. LAOS combines search-state information, operator credit, and replayed experience for adaptive operator selection [40]. LLMOA constructs sequences of low-level operators for continuous optimization [42]. HeurAgenix combines heuristic evolution with model-guided filtering and Monte Carlo continuation estimates for selection, and also trains a lightweight selector [35].

CIPHEUR makes the joint choice of target, intervention, and recovery the common task for generated policies and online LLMs. The design focus is a control policy over structurally grounded interventions: directed SIUs specify which trajectory to redirect, which conflict relationships to emphasize, and how search should proceed afterward. Offline synthesis turns this decision task into a reusable controller, while online inference resolves it from the current search context. Both modes therefore operate over the same structural and temporal decision space.

Feedback relates policy decisions to changes in solution quality across their intervention–recovery periods. In online mode, asynchronous inference lets the optimizer continue searching while the model chooses its next intervention. The resulting design principle is to make structurally targeted search behavior the adaptable object, while retaining objective evaluation and feasibility enforcement in the native solver.

3 CIPHEUR

CIPHEUR uses LLM capabilities to construct search controllers offline or choose interventions online. Both modes operate through

a common loop: structural observation, SIU selection, native execution, and feedback (Figure 2). The work mode is fixed before a solve.

3.1 Problem and Control State

Let $G = (V, E, w, \ell)$ be a resource-labelled conflict graph with positive weights w and metadata ℓ . Its feasible family and objective are

$$\begin{aligned} \mathcal{I}(G) &= \{X \subseteq V : \{u, v\} \not\subseteq X \ \forall \{u, v\} \in E\}, \\ F(X) &= \sum_{v \in X} w_v. \end{aligned} \quad (1)$$

Each scheduling vertex is a complete contact $v = (s_v, a_v, b_v, e_v)$ with integer duration $w_v = e_v - b_v$. For $u \prec v$ in start-time order, let $d_{uv} = b_v - e_u$. Contacts conflict when $(a_u = a_v \wedge d_{uv} < \delta_{au}^A)$ or $(s_u = s_v \wedge d_{uv} < \delta_{su}^S)$; equality with a threshold is feasible. Resource/time labels form ℓ , and G remains fixed throughout a solve.

The native state contains $m = 4$ persistent trajectories $\mathcal{P}_t = (X_{t,j})_{j=0}^{m-1}$ and a historical incumbent X_t^* . The initial feasible set is replicated with distinct random states; each trajectory retains its solution and internal search state. We use CHILS local-search kernels [13] within the CIPHEUR backend and report $F(X_T^*)$ at budget T .

At decision opportunity k , the common control task is

$$u_k = \pi(o_k, \mathcal{H}_k, \mathcal{U}_k; z_k), \quad u_k \in \mathcal{U}_k, \quad (2)$$

where o_k is the current observation, $\mathcal{H}_k$ contains recent execution records, $\mathcal{U}_k$ is the SIU registry, and z_k is optional policy memory. Observation time q_k and installation time τ_k are distinct: online inference can overlap continued execution of the preceding SIU.

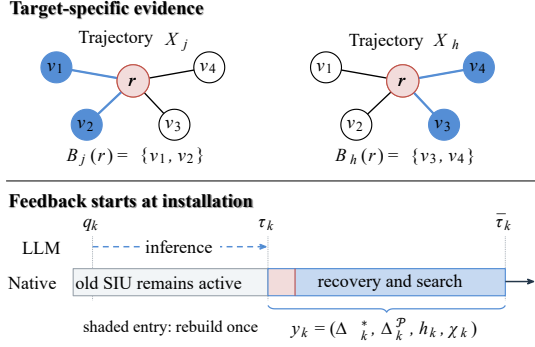


Figure 3: SIU grounding. (a) One anchor has different blockers across trajectories. (b) Online inference overlaps the old SIU; new feedback starts at installation au_k .

3.2 From Conflict Evidence to SIUs

Target-specific evidence. For anchor r and trajectory j , selected blockers and their weight are

$$B_j(r) = N_G(r) \cap X_{q_k, j}, \quad W_j(r) = \sum_{v \in B_j(r)} w_v. \quad (3)$$

The observer records their counts, weights and resource relationships, together with the target’s value gap, overlap with the population best, and whether r is already selected. Figure 3(a) illustrates why evidence is recomputed for each target.

The observer considers the $L = 128$ largest-weight vertices, retains those excluded and blocked by the population best, and ranks them by $w_r / \max\{1, W_{j_{\text{best}}}(r)\}$. The first $R = 6$ define the anchor set $\mathcal{R}_k$. Counts and weights use complete blocker sets; displayed blocker identities are capped at 16 and shared-blocker measures use these samples. Donor cross-conflicts sample up to 64 donor-exclusive vertices. Pairwise overlap is Jaccard similarity; population diversity is $|\bigcup_j X_j \setminus \bigcap_j X_j| / |\bigcup_j X_j|$. Progress, stagnation and resource use complete o_k . The anchor ratio ranks observations; executed outcomes evaluate plans.

Joint intervention representation. An SIU has the form

$$u = (i, \omega, \mathcal{S}), \quad \omega = (j, r, c), \quad \mathcal{S} = ((a_1, d_1), \dots, (a_{n_u}, d_{n_u})), \quad (4)$$

where i is its identifier, ω binds the trajectory and anchor to resource-priority template c , and $\mathcal{S}$ specifies actions with nominal dwell times. Directed SIUs reconstruct j once on entry, then continue with SPREAD or MERGE: the former distributes search across trajectories, while the latter emphasizes exchange with a structurally different donor. Single-action SIUs have $\omega = \emptyset$.

Eligible targets $\mathcal{J}_k$ exclude slot zero and the population best. Given templates $\mathcal{C}$ and continuations $\mathcal{A}_{\text{rec}}$, we construct

$$\mathcal{U}_k = \mathcal{U}_{\text{single}} \cup \{u(j, r, c, a) : j \in \mathcal{J}_k, r \in \mathcal{R}_k, c \in \mathcal{C}, a \in \mathcal{A}_{\text{rec}}\}_{\text{eligible}}. \quad (5)$$

Eligibility requires valid targets, target-specific evidence and the resource identities required by c . Descriptors are frozen with their observation for both policy modes. The nine single-action alternatives are exploit, spread, merge, relay, antenna, satellite,

kick, reseed, and the native adaptive fallback. They cover intensification, diversification, exchange, resource-guided moves, entry perturbation and ordinary reseeding.

Reconstruction and continuation. Let $I_A(v, r)$ and $I_S(v, r)$ indicate shared resources and $t(v, r) = |b_v - b_r|$. For antenna, satellite and mixed templates, respectively, set

$$\begin{aligned} (M_A, g_A) &= (1 + 2I_A, \delta_{a_v}^A), \quad (M_S, g_S) = (1 + 2I_S, \delta_{s_v}^S), \\ (M_M, g_M) &= (1 + I_A + I_S, \max\{\delta_{a_v}^A, \delta_{s_v}^S\}), \\ p_c(v; r) &= \text{sdiv}(w_v M_c, \text{sdiv}(t(v, r), w_v + g_c)). \end{aligned} \quad (6)$$

Here $\text{sdiv}(a, b) = \text{trunc}_0[a / (1 + |b|)]$; arithmetic saturates at magnitude $2^{61} - 1$. All vertices are sorted by p_c , with seeded tie-breaking, then inserted greedily when compatible. Only a completed feasible construction replaces the target; interruption preserves the old set and ordering random state. Original weights determine F .

Directed SIUs use 8 s for entry/search and 16 s for recovery. The one entry reconstruction receives $\min(0.25, \delta/2, \text{remaining})$ seconds and at most 10^7 work units within a native quantum δ . Remaining time continues population search. The final action persists until replacement or termination. During donor exchange, independent candidates A compete with $B = N(A) \cap X_j$ in a weighted bipartite subproblem, preserving $X_j \setminus B$. A completed flow solves this fixed exchange; bounded interruption retains feasibility. Accepted exchanges have nonnegative original-objective gain.

3.3 Feedback over Executed Interventions

An interval starts at installation τ_k and closes at the next local decision checkpoint, accepted online replacement, fallback, or termination $\bar{\tau}_k$. With $Q_t = \sum_j F(X_{t, j})$, its outcome is

$$\begin{aligned} y_k &= (\Delta_k^*, \Delta_k^P, h_k, \chi_k), \quad h_k = \bar{\tau}_k - \tau_k, \\ \Delta_k^* &= F(X_{\bar{\tau}_k}^*) - F(X_{\tau_k}^*), \quad \Delta_k^P = Q_{\bar{\tau}_k} - Q_{\tau_k}. \end{aligned} \quad (7)$$

where χ_k records stages reached and execution status. Incumbent gain measures objective progress; population change exposes disruption and recovery. The record retains the initiating observation, while its gain and duration start at actual installation (Figure 3(b)).

Recent records and native-progress summaries inform subsequent decisions. Ongoing intervals carry an explicit unfinished status. Offline numerical controllers transform incumbent gain into a log-scaled improvement rate; online LLMs receive the underlying gains, durations and context. These are alternative uses of the same executed outcomes, with final evaluation governed by Eq. (1).

3.4 Two LLM Policy Realizations

Offline: LLM construction of executable control. Before deployment, a design LLM $\mathcal{L}_\psi$ receives specification $\mathcal{D}$: the MWIS objective, observation semantics, SIU interface and feedback contract. Construction produces program θ , with optional runtime memory z_k :

$$\begin{aligned} \theta &\leftarrow \text{CONSTRUCTWITHLLM}(\mathcal{L}_\psi, \mathcal{D}), \\ u_k &= \text{CHOOSE}_\theta(o_k, \mathcal{H}_k, \mathcal{U}_k; z_k), \\ z_{k+1} &= \text{UPDATE}_\theta(z_k, o_k, u_k, y_k). \end{aligned} \quad (8)$$

Construction precedes the timed solve; choice and optional updates execute locally (Figure 4(a)), with memory reset between runs. The

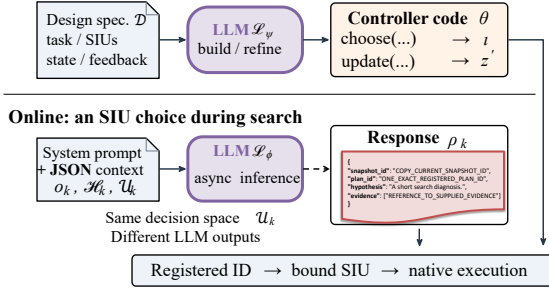
Offline: LLM-built controller code

Figure 4: Offline controller construction and online SIU selection. Both modes share the native executor.

recorded one-shot realization is off-GEN. Archived off-EXP (KNN) and off-VAL (LinUCB) implement the same interface, separately from that generation record.

For these numerical realizations, let f_k^{obs} be the initiating incumbent and $x = \phi(o_k, u) \in \mathbb{R}^{59}$. Features comprise 10 global statistics, 15 action/template indicators, 10 target quantities and 24 interactions; identifiers are excluded. Feedback is

$$r_k = \log \left(1 + \frac{10^6 \Delta_k^*}{\max(1, f_k^{\text{obs}}) \max(0.01, h_k)} \right). \quad (9)$$

EXP keeps the latest 160 samples, uses five nearest neighbors with $a_i = (0.2 + \|x - x_i\|_2)^{-1}$, and scores $\sum_i a_i r_i / \sum_i a_i + 0.25 / \sqrt{1 + \sum_i a_i}$. VAL starts from $A = I, b = 0$, updates $A \leftarrow A + xx^\top$, $b \leftarrow b + r_k x$, and scores $b^\top A^{-1} x + 0.25 \sqrt{x^\top A^{-1} x}$. Both start with SPREAD and retain its preference $0.1/(1+n)$ after n updates. Nonfinite features map to bounded neutral values.

Online: LLM reasoning over current SIUs. A fixed instruction s_{sys} and the current control context form the prompt; the model returns a data-only choice:

$$\begin{aligned} p_k &= \text{PROMPT}(s_{\text{sys}}; o_k, \mathcal{H}_k, \mathcal{U}_k), \\ \rho_k &\leftarrow \mathcal{L}_\phi(p_k), \quad u_k = \text{RESOLVE}(\rho_k, \mathcal{U}_k). \end{aligned} \quad (10)$$

The instruction asks the LLM to compare continued search with structural alternatives using progress, diversity, blocking evidence, recovery outcomes and remaining time. The JSON packet includes bounded summaries and previous choices; repeated SIU fields use shared tables. Figure 4(b) shows the response contract: the snapshot and plan IDs determine the executable SIU, while the diagnosis and evidence references are recorded as context. RESOLVE returns the registered descriptor for a correctly bound choice and $\perp$ otherwise. The full graph and selected sets remain native. Each fresh two-message request includes at most 12 action episodes, eight completed SIUs and eight previous choices, plus progress windows of 10, 40 and 120 s. JSON factorization retains every registered choice. Explanatory fields are recorded as context; the selected registry identity controls execution.

Algorithm 1 CIPHEUR: LLM-guided SIU control

Require: $G, T, \mu \in \{\text{off}, \text{online}\}$; quantum ε
Specification $\mathcal{D}$; instruction s_{sys} ; LLMs $\mathcal{L}_\psi, \mathcal{L}_\phi$

Ensure: Best verified independent set X^*

```

1: if  $\mu = \text{off}$  then
2:    $\theta \leftarrow \text{CONSTRUCTWITHLLM}(\mathcal{L}_\psi, \mathcal{D})$ 
3: end if
4: Start clock; initialize feasible  $\mathcal{P}, X^*, \mathcal{H}, z$ 
5: Install default SIU  $u$ ; reset stage  $\sigma$ 
6: while elapsed time  $< T$  do
7:   if  $\mu = \text{online}$  and LLM reply  $\rho_k$  is ready then
8:      $u^+ \leftarrow \text{RESOLVE}(\rho_k, \mathcal{U}_k)$ 
9:     if  $u^+ \neq \perp$  and freshness/budget checks pass then
10:      Close and record interval
11:      Install  $u^+$ ; reset  $\sigma$  to entry stage
12:    end if
13:    Release the LLM inference lane
14:  end if
15:  if a decision is due and mode-specific budgets allow then
16:    if  $\mu = \text{off}$  then
17:      Close interval; update  $z$ 
18:      Build  $o_k, \mathcal{H}_k, \mathcal{U}_k$ 
19:       $u_k \leftarrow \text{CHOOSE}_\theta(o_k, \mathcal{H}_k, \mathcal{U}_k; z)$ 
20:      Validate/install  $u_k$ ; reset  $\sigma$ ; advance schedule
21:    else if the LLM inference lane is idle then
22:      Build and freeze  $o_k, \mathcal{H}_k, \mathcal{U}_k$ 
23:       $p_k \leftarrow \text{PROMPT}(s_{\text{sys}}; o_k, \mathcal{H}_k, \mathcal{U}_k)$ 
24:      Launch  $\rho_k \leftarrow \mathcal{L}_\phi(p_k)$  asynchronously
25:      Retain  $\mathcal{U}_k$ ; advance request schedule
26:    end if
27:  end if
28:  if the solve budget is exhausted then
29:    break
30:  end if
31:  Advance due stages in  $\sigma$ ; retain the final action
32:   $\delta \leftarrow \min\{\varepsilon, \text{stage remainder}, \text{solve remainder}\}$ 
33:   $\text{ADVANCENATIVE}(\mathcal{P}, X^*, u, \sigma; \delta)$ 
34:  Append the native execution outcome
35: end while
36: Discard pending replies; close the final interval
37: return  $X^*$  after original-graph and objective checks

```

3.5 Asynchronous Execution and Complete Procedure

Algorithm 1 realizes Eq. (2) through the LLM-built program or online model; predeployment construction is outside the timed solve. Its local branch closes the preceding interval before updating and selecting; its online branch retains a frozen registry while native search continues. Returned choices undergo snapshot/descriptor checks, a 60 s freshness limit and remaining-budget checks; request timeout is 45 s. An idle inference lane and over 47 s remaining are required to submit. The main thread alone installs plans and owns native state.

The LLM steps are explicit: line 2 produces controller code; lines 23–24 form and submit an online decision; line 8 maps the reply to a frozen SIU. Native advancement at line 33 proceeds while inference is pending. Before a targeted entry, the executor rechecks slot protection and metadata. Rejection advances to the registered continuation, or installs the default plan when none remains. It retains the requested target’s identity throughout. Interval closure uses Eq. (7); policy updates skip the initial fallback interval and occur only when feedback is available.

Feasibility and cost. Starting from feasible states, native moves preserve independence and reconstruction commits only completed compatible sets. The incumbent accepts only a better feasible set.

Induction over completed operations therefore preserves population feasibility and nondecreasing incumbent value, while allowing individual trajectories to lose value during diversification. With p eligible targets, R anchors, C templates, A continuations and S single-action plans, registry size is at most $S + pRCA$ (117 in this realization). Full reconstruction costs $O(|V| \log |V| + |E|)$ before work/time limits. Observation and local execution count toward the solve; online inference overlaps native search. Checks between bounded operations yield a soft wall-clock budget. All decisive observation, priority, feedback and execution choices are specified above; full helper code accompanies the submission.

4 Experiments

We examine terminal quality, the development of online gains, and the contributions of structural control. The historical comparison, fresh matched screen, completed control extensions, and new-source tests remain separate evaluation frames.

4.1 Instances, Comparators and Protocol

CP-SCALE-AU-L002 supplies eight views of the same contacts and weights. Homogeneous views H_a use antenna gaps of 340, 680, 1200 and 1800 s; mixed views $M_{a/b}$ use western/eastern gaps (340, 1200), (680, 1200), (1200, 340) and (1200, 680) s. Each view uses seeds 67, 71, 73, 79 and 83 under a 900 s outer wall cap. The registered CIPHEUR/CHILS campaign matches four trajectories and one native thread per solve. Online requests begin at 20 s, recur at 40 s opportunities and have a cap of 21; the archived model identifier is gpt-6-1una. Model inference uses additional compute. Observation and local selection consume solve time; inference overlaps native search.

Table 1 retains all 11 completed configurations. External methods include CHILS-p4-custom [13], Feasibility Jump [23], FastWVC [4], GRASP [11], StableSolver LS and LNS [29], and GWMIN [28]. Four CIPHEUR realizations complete the comparison. LS and LNS ignore seed arguments and therefore use five fresh process repeats. LS has a 9 GiB limit, requests 850 s native time and records 856.8–874.4 s wall time; LNS requests 890 s and records 891.1–892.5 s. GWMIN stops before the cap. All accepted results passed recorded graph-feasibility and integer-objective audits. External-table comparisons are descriptive; pairwise wins below use matched frames.

EXP and VAL are the archived numerical controls defined in Section 3.4; *off* denotes zero runtime model calls. GEN follows a separate frozen generation protocol. The released EXP/VAL features clip elapsed and remaining times after division by 360 s; that convention is retained in this 900 s comparison. Means give equal weight to views after averaging their runs. For terminal comparison, $E_v = \max_{b \in \mathcal{B}} \bar{F}_{b,v}$ is the best mean among all seven external rows. Table 1 shows $100(\bar{F}_{m,v}/E_v - 1)$, absolute overall means and mean within-view ranks.

4.2 Final Quality and Late-Search Improvement

Competitiveness across constraint views. CIPHEUR-online attains the highest mean, 1,138,760.369 contact-seconds, and best average rank, 1.75. Online, EXP and VAL each exceed the strongest external mean in seven views. CHILS defines that reference in six views, LS in H_{1200} and Feasibility Jump in H_{1800} ; online trails the latter

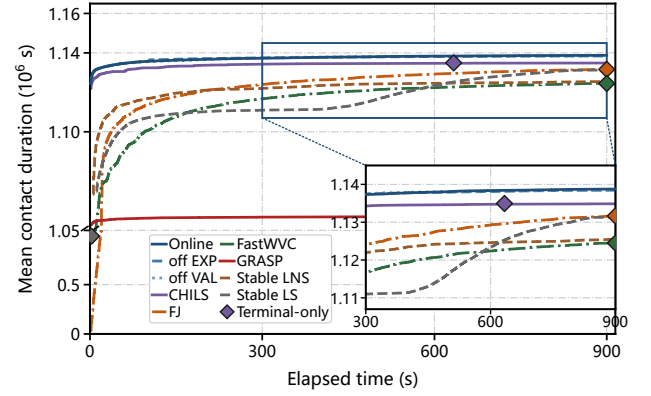


Figure 5: L002 progress (40 runs). The main axis uses 20:1 compression below 1.05×10^6 contact-s; inset: 300–900 s. CHILS, FJ and FastWVC are separate reruns.

by 0.148%. Against CHILS alone, online gains 3,817.583 contact-seconds (0.336%) and wins all 40 graph-seed pairs. Its additional mean gains over EXP and VAL are 308.457 and 342.692 seconds, with 24/40 wins against each: a smaller, variable improvement within the common framework.

Figure 5 shows the recorded method trajectories and a late-search enlargement. The horizontal coordinate follows each method’s recorded solver or outer clock. Filled diamonds identify frozen terminal-only means; their positions mark the recorded end or budget cap, not first attainment.

Same-run improvement and attainment. For trajectories, R_v is the best audited individual terminal value among the original online/EXP/VAL/CHILS runs in view v . Define the empirical-reference shortfall

$$g_{m,v,s}(t) = 100 \frac{R_v - F_{m,v,s}(t)}{R_v}, \quad \bar{g}_m(t) = \frac{1}{40} \sum_{v,s} g_{m,v,s}(t). \quad (11)$$

R_v is a feasible reference, distinct from E_v and from a certified optimum. Figure 7 uses saved epoch-end incumbents from the same 900 s runs, displaying all 12 released checkpoints. CHILS supplies terminal values only.

At 360 s, online has larger shortfall than EXP and VAL; by 900 s this ordering reverses (Table 2). Its $\bar{g}(360) - \bar{g}(900)$ improvement is **1.50 times EXP’s and 1.86 times VAL’s**, a ratio of quality gains rather than runtime speedups. Online attains the strictest positive target, 0.05%, in 10/40 runs versus 6/40 for each comparator. The complete nine-threshold grid also preserves the exception at 0.4% (online 38, EXP 39) and equal attainment at 0.6% and 1%. The recorded time-normalized 20–900 s shortfall areas (AOC) are 0.24065%, 0.24177% and 0.25226%: online’s terminal advantage coexists with near-equal whole-run area against EXP.

Table 1: L002 terminal quality (five runs/view). View entries are gains over E_v (%); E_v and Mean use 10^3 contact-s. Rank averages within-view ranks; bold marks bests. [†] GEN was separately frozen; LS/LNS use process repeats.

Method E_v (10^3 s)	Mean [†] —	H_{340} 1624.295	H_{680} 1246.823	H_{1200} 884.035	H_{1800} 684.860	$M_{340/1200}$ 1259.316	$M_{680/1200}$ 1064.712	$M_{1200/340}$ 1261.095	$M_{1200/680}$ 1062.766	Rank _↓ —
<i>External baselines</i>										
CHILS	1,134.943	0.00	0.00	-0.63	-0.41	0.00	0.00	0.00	0.00	5.38
Feasibility Jump [23]	1,131.650	-0.06	-0.25	-1.40	0.00	-0.31	-0.59	-0.27	-0.43	6.12
StableSolver LS [29]	1,131.306	-0.80	-0.62	0.00	-0.61	-0.28	-0.26	-0.38	-0.14	6.38
StableSolver LNS [29]	1,125.482	-1.08	-1.05	-0.81	-0.63	-0.85	-0.98	-1.01	-0.75	8.38
FastWVC [4]	1,124.488	-1.41	-1.59	-0.20	-0.02	-1.14	-0.94	-1.19	-0.75	7.38
GRASP [11]	1,057.130	-6.21	-8.68	-4.77	-10.95	-6.00	-7.19	-6.21	-7.00	10.00
GWMIN [28]	991.619	-12.08	-14.54	-8.46	-28.54	-10.24	-11.62	-10.74	-11.20	11.00
<i>CIPHEUR family</i>										
CIPHEUR-off-EXP	1,138.452	+0.13	+0.23	+0.24	-0.19	+0.25	+0.36	+0.13	+0.49	2.50
CIPHEUR-off-VAL	1,138.418	+0.13	+0.23	+0.09	-0.18	+0.27	+0.38	+0.20	+0.47	2.50
CIPHEUR-off-GEN [†]	1,136.347	+0.03	+0.14	-0.21	-0.31	+0.05	+0.11	-0.04	+0.31	4.62
CIPHEUR-online	1,138.760	+0.17	+0.24	+0.22	-0.15	+0.32	+0.36	+0.21	+0.48	1.75

Table 2: Late progress, attainment, and matched control effects. Control Δ is online minus control.

Late shortfall/AOC (%) and improvement (pp)									
Mode	$\bar{g}_{360} \downarrow$			$\bar{g}_{900} \downarrow$		$\Delta_{360:900} \uparrow$		AOC $\downarrow$	
Online	0.23746			0.12698		0.11048		0.24065	
Off-EXP	0.22471			0.15113		0.07359		0.24177	
Off-VAL	0.21849			0.15914		0.05936		0.25226	
Terminal attainment (counts out of 40)									
ϵ (%)	0	0.05	0.10	0.15	0.20	0.30	0.40	0.60	1.00
Online	3	10	17	27	33	37	38	40	40
Off-EXP	3	6	12	23	31	36	39	40	40
Off-VAL	2	6	16	22	31	35	36	40	40
Matched control split (contact-s; wins/6)									
Homogeneous									
Control	Mean Δ			Wins		Mixed gaps			
						Mean Δ	Wins		
SIU-Off	+3,416.7			6/6		+5,065.7			6/6
Fixed-Spread	+576.9			3/6		+2,210.4			6/6
Random-SIU	-338.3			1/6		+1,415.1			5/6
No-Struct	-797.0			2/6		+1,060.3			6/6
Single-Op	-836.1			2/6		+1,023.4			6/6

4.3 Control Contributions and Observed Recovery

The fresh mechanism screen matches graph, seed and host partition on H_{340} , H_{1800} , $M_{340/1200}$ and $M_{1200/340}$, with three seeds each. SIU-Off retains the initial diversity-adaptive native plan throughout; Fixed-Spread keeps decision checkpoints but always selects SPREAD. Random-SIU samples legal units, Single-Op restricts the registry to single actions, and No-Struct masks resource witnesses and plan context while preserving legal IDs and bindings. Thus No-Struct retains cues in action identifiers. All use 900 s and the same objective checks.

Figure 6 retains every paired gain and loss. Full online control gains **4,241.212 contact-seconds over SIU-Off** (12/12 wins) and

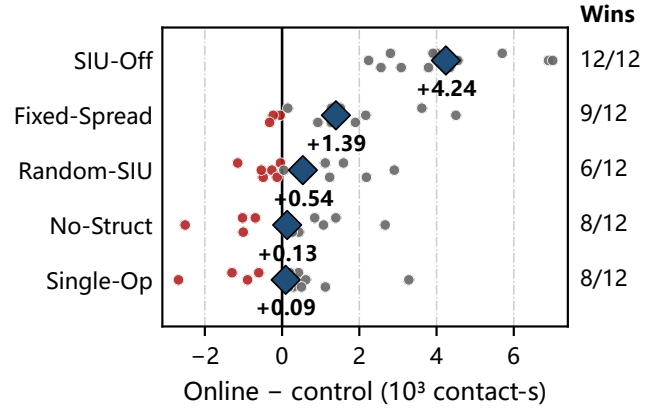


Figure 6: Matched mechanism screen. Each dot is an online-minus-control difference (12 pairs per control); diamonds mark means and the right column lists positive pairs.

1,393.654 over Fixed-Spread (9/12). Fixed-Spread changes both adaptivity and the action mix, so the result supports the complete control configuration. Finer differences are +538.377 for Random-SIU, +131.678 for No-Struct and +93.655 for Single-Op, with 6/12, 8/12 and 8/12 wins. These smaller, seed-dependent contrasts provide limited component-specific evidence. Separately completed 40-position controls average 1,134,403.037 (SIU-Off) and 1,137,567.142 (Fixed-Spread) seconds. Their matched online extension was incomplete; historical online runs are not substituted as partners.

The bottom panel of Table 2 localizes the fine-control differences. On mixed-gap views, full control exceeds No-Struct by 1,060.3 seconds and Single-Op by 1,023.4 seconds, with 6/6 positive pairs for each. Both comparisons reverse in the homogeneous-group mean (2/6 positive). Structure evidence and coordinated recovery therefore have their clearest observed benefit in the mixed-constraint

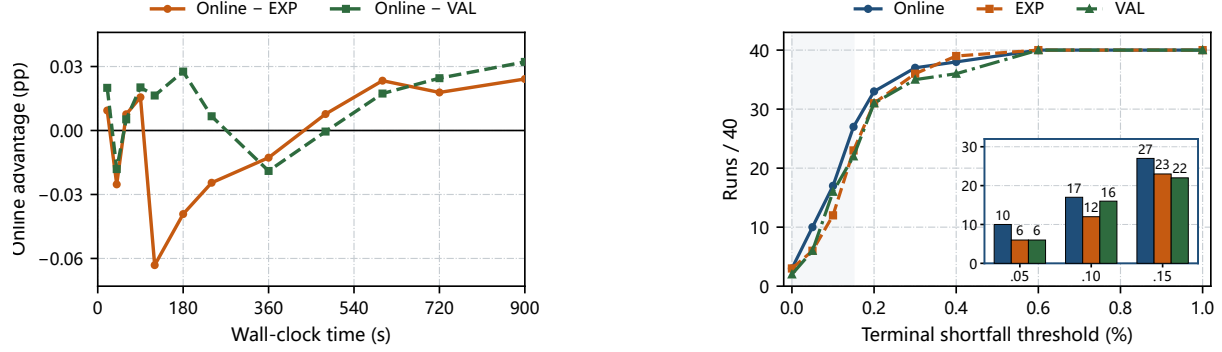


Figure 7: Search progress and attainment. (a) Mean online-minus-programmatic quality at 12 saved checkpoints. (b) Counts across nine terminal thresholds (40 runs); inset: three strictest positive thresholds.

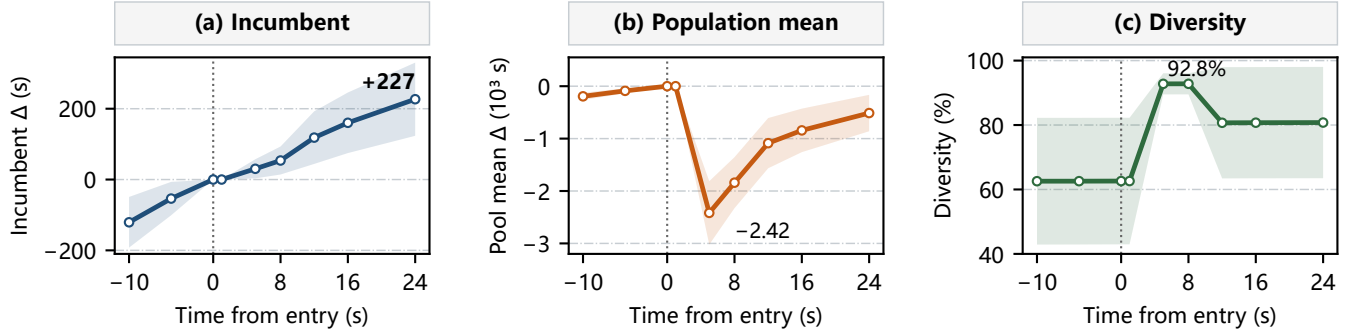


Figure 8: Observed recovery after 401 directed reseeds. Curves average 40 run means; bands show descriptive ± 1 sample SD. Panel (b) is the population mean.

subset; the two-view groups support a descriptive explanation, not a general interaction claim.

A measured intervention–recovery pattern. Of 406 recorded directed intervals, 401 install a native reseed. Figure 8 averages events within each run and then weights the 40 runs equally. At +5 s, mean population value is 2,417.631 seconds below entry and diversity rises from 62.6% to 92.8%. At +24 s, the deficit is 512.513 seconds: **78.8% has recovered**, while the historical incumbent improves by 226.823 seconds. All 401 events remain installed through +24 s; 400 remain at +30 s. The population curve describes the pool mean, not only the targeted slot. These observational changes explain why immediate trajectory loss can coexist with later progress; they do not isolate the causal effect of reseeding alone.

Native work continues during inference. Among 840 online calls, 838 install a valid plan. Request-to-receipt latency has median 11.06 s and 90th percentile 13.87 s. Pending intervals total 9,845.261 s and overlap 9,837.360 s of native work: **99.92% overlap**, or 245.9 s per solve, with 2,720,049 native cycles. HTTP-only overlap is 99.99%. These intervals lie inside the solve budget; they establish sustained execution during model latency, not an additive time saving or measured synchronous-control speedup.

4.4 Offline Synthesis and Cross-Source Use

The prospective GEN test issued one recorded gpt-6-luna request at low reasoning effort for a deterministic `select_plan(packet, plans)` function. Only the supplied state and legal descriptors were accessible. The first candidate was frozen without objective-guided revision. All 84 saved interface packets yielded legal IDs, deterministic repeats and unchanged inputs. These packets came from four L002 views at seed 83, so those four graph–seed positions overlap the subsequent 40-position evaluation; validation tested the interface rather than tuning objective performance. GEN completes all 40 runs with zero deployment model calls and a mean of 1,136,347.357 seconds: **+1,404.572 over CHILS (0.124%)**. Its lower mean than EXP/VAL shows that a usable one-shot policy need not dominate established control realizations.

L003 adds AU and AP regional contact sets, each under homogeneous and mixed constraints with three seeds.

The authors attest to excluding these sources from prompt design, tuning and controller selection. Frozen EXP/VAL programs use fresh per-solve memory. Table 3 brings per-view means into the main evaluation; the sample SDs remain in the supplement. EXP and VAL improve the overall CHILS mean by 0.197% and 0.182%, lead all four view means, and win 11/12 and 12/12 pairs. Only the three

Table 3: L003 cross-source means (10^3 contact-s; three runs per view). Gains and matched wins are relative to CHILS.

View	CHILS	Off-EXP	Off-VAL
AU-H	1,623.290	1,627.711	1,627.468
AU-M	1,259.639	1,264.050	1,263.209
AP-H	1,690.812	1,692.549	1,692.806
AP-M	1,283.948	1,284.905	1,284.875
Mean↑	1,464.422	1,467.304	1,467.090
Gain (%)	—	+0.197	+0.182
Wins	—	11/12	12/12

completed arms are included; online and GEN cross-source runs remain incomplete.

Scope. The evidence covers a few contact sources and a 12-position fine-control screen. Time-feature scaling remains a configuration constraint, L003 non-use is author-attested, and synchronous-control/online cross-source tests are unfinished. Missing token receipts prevent complete inference-cost accounting.

5 Conclusion

CIPHEUR treats conflict-graph search control as a joint decision over structural targets, interventions and their continuation. SIUs give generated programs and online LLMs a common executable task, while native search preserves the original optimization semantics. The completed experiments show broad terminal competitiveness, stronger late-search progress for the online configuration, and a reconstruction–recovery pattern that motivates interval-level feedback. A frozen one-shot controller further demonstrates deployment without runtime model calls. Together, these findings identify structural search control as a practical way to use LLM design knowledge and contextual reasoning in combinatorial optimization.

References

- [1] Steven Adriaensen, André Biedenkapp, Gresa Shala, Noor Awad, Theresa Eimer, Marius Lindauer, and Frank Hutter. 2022. Automated Dynamic Algorithm Configuration. *Journal of Artificial Intelligence Research* 75 (2022), 1633–1699. doi:10.1613/jair.1.13922
- [2] Ismail Alkhouri, Cedric Le Denmat, Yingjie Li, Cunxi Yu, Jia Liu, Rongrong Wang, and Alvaro Velasquez. 2025. Differentiable Quadratic Optimization For the Maximum Independent Set Problem. In *Proceedings of the 42nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 267)*. PMLR, 1102–1127. <https://proceedings.mlr.press/v267/alkhouri25a.html>
- [3] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. 2021. Machine learning for combinatorial optimization: A methodological tour d’horizon. *European Journal of Operational Research* 290, 2 (2021), 405–421. doi:10.1016/j.ejor.2020.07.063
- [4] Shaowei Cai, Yuanjie Li, Wenying Hou, and Haoran Wang. 2019. Towards faster local search for minimum weight vertex cover on massive graphs. *Information Sciences* 471 (2019), 64–79. doi:10.1016/j.ins.2018.08.052
- [5] Semih Cantürk, Thomas Sabourin, Frederik Wenkel, Michael Perlmutter, and Guy Wolf. 2026. Can Computational Reducibility Lead to Transferable Models for Graph Combinatorial Optimization?. In *Proceedings of the 43rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 306)*. PMLR, 11063–11082. <https://proceedings.mlr.press/v306/canturk26a.html>
- [6] Bin Chen, Shouliang Zhu, Beidan Liu, Yong Zhao, Tianle Pu, Huichun Li, and Zhengqiu Zhu. 2026. A₂DEPT: Large Language Model–Driven Automated Algorithm Design via Evolutionary Program Trees. In *Proceedings of the 43rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 306)*. PMLR, 18654–18693. <https://proceedings.mlr.press/v306/chen26hb.html>
- [7] Shengkai Chen, Zhiguang Cao, Jianan Zhou, Yaoxin Wu, Senthilnath Jayavelu, Zhuoyi Lin, Xiaoli Li, and Shili Xiang. 2026. DRAGON: LLM-Driven Decomposition and Reconstruction Agents for Large-Scale Combinatorial Optimization. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2809–2817. doi:10.65109/SBAY6258
- [8] Antonia Chmiela, Elias B. Khalil, Ambros Gleixner, Andrea Lodi, and Sebastian Pokutta. 2021. Learning to Schedule Heuristics in Branch and Bound. In *Advances in Neural Information Processing Systems*, Vol. 34. 24235–24246. <https://proceedings.neurips.cc/paper/2021/hash/cb7c403aa312160380010ee3dd4bfc53-Abstract.html>
- [9] Yuanyuan Dong, Andrew V. Goldberg, Alexander Noe, Nikos Parotsidis, Mauricio G. C. Resende, and Quico Spaen. 2022. A Local Search Algorithm for Large Maximum Weight Independent Set Problems. In *30th Annual European Symposium on Algorithms (ESA 2022) (Leibniz International Proceedings in Informatics, Vol. 244)*. 45:1–45:16. doi:10.4230/LIPIcs.ESA.2022.45
- [10] John H. Drake, Ahmed Kheiri, Ender Özcan, and Edmund K. Burke. 2020. Recent advances in selection hyper-heuristics. *European Journal of Operational Research* 285, 2 (2020), 405–428. doi:10.1016/j.ejor.2019.07.073
- [11] Thomas A. Feo and Mauricio G. C. Resende. 1995. Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization* 6, 2 (1995), 109–133. doi:10.1007/BF01096763
- [12] Chengrui Gao, Haopu Shang, Ke Xue, and Chao Qian. 2025. Neural Solver Selection for Combinatorial Optimization. In *Proceedings of the 42nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 267)*. PMLR, 18528–18549. <https://proceedings.mlr.press/v267/gao25l.html>
- [13] Ernestine Großmann, Kenneth Langedal, and Christian Schulz. 2025. Concurrent Iterated Local Search for the Maximum Weight Independent Set Problem. In *23rd International Symposium on Experimental Algorithms (SEA 2025) (Leibniz International Proceedings in Informatics, Vol. 338)*. 22:1–22:18. doi:10.4230/LIPIcs.SEA.2025.22
- [14] Oguzhan Gungordu, Siheng Xiong, and Faramarz Fekri. 2026. PathWise: Planning through World Model for Automated Heuristic Design via Self-Evolving LLMs. In *Proceedings of the 43rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 306)*. PMLR, 37779–37816. <https://proceedings.mlr.press/v306/gungordu26a.html>
- [15] Achref Jaziri, Thibaut Cuvelier, and Bruno Adrien De Backer. 2026. Generative Large Neighborhood Search: Scalable Set Cover Optimization via Discrete Diffusion. In *Proceedings of the 43rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 306)*. PMLR, 51046–51082. <https://proceedings.mlr.press/v306/jaziri26a.html>
- [16] Xuemei Jiang, Jiting Li, Rammohan Mallipeddi, Guangyin Jin, Jian Wu, Lining Xing, and Yanjie Song. 2026. Integrating Adaptive Divide-and-Conquer and Large Language Model for Scheduling Large-Scale Tasks in Electromagnetic Satellite Systems. *Expert Systems with Applications* 318 (2026), 131978. doi:10.1016/j.eswa.2026.131978
- [17] Xia Jiang, Yaoxin Wu, Minshuo Li, Zhiguang Cao, and Yingqian Zhang. 2025. Large Language Models as End-to-end Combinatorial Optimization Solvers. In *Advances in Neural Information Processing Systems*, Vol. 38. doi:10.52202/085713-5495
- [18] Nguyen Viet Tuan Kiet, Tung Dao, Cong Dao Tran, and Huynh Thi Thanh Binh. 2026. MOTIF: Multi-strategy Optimization via Turn-based Interactive Framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 37000–37008. doi:10.1609/aaai.v40i43.41028
- [19] Jintao Li, Yong-Yi Wang, Zheng-An Wang, and Heng Fan. 2026. LoRe: Adaptive Interaction-Evaluation Routing with Per-step Interaction Budgets for Iterative Graph Solvers. In *Proceedings of the 43rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 306)*. PMLR, 70969–70982. <https://proceedings.mlr.press/v306/li26gz.html>
- [20] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. 2018. Combinatorial Optimization with Graph Convolutional Networks and Guided Tree Search. In *Advances in Neural Information Processing Systems*, Vol. 31. <https://papers.nips.cc/paper/2018/hash/8d3bba7425e7c98c50f52ca1b52d3735-Abstract.html>
- [21] Fei Liu, Yilu Liu, Qingfu Zhang, Xialiang Tong, and Mingxuan Yuan. 2026. EoH-S: Evolution of Heuristic Set Using LLMs for Automated Heuristic Design. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 37090–37098. doi:10.1609/aaai.v40i43.41038
- [22] Fei Liu, Xialiang Tong, Mingxuan Yuan, Xi Lin, Fu Luo, Zhenkun Wang, Zhichao Lu, and Qingfu Zhang. 2024. Evolution of Heuristics: Towards Efficient Automatic Algorithm Design Using Large Language Model. In *Proceedings of the 41st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 235)*. 32201–32223. <https://proceedings.mlr.press/v235/liu24bs.html>
- [23] Bjornar Luteberget and Giorgio Sartor. 2023. Feasibility Jump: an LP-free Lagrangian MIP heuristic. *Mathematical Programming Computation* 15, 2 (2023), 365–388. doi:10.1007/s12532-023-00234-8
- [24] Manuj Malik, Jianan Zhou, Shashank Reddy Chirra, and Zhiguang Cao. 2026. PyVRP+: LLM-Driven Metacognitive Heuristic Evolution for Hybrid Genetic Search in Vehicle Routing Problems. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2800–2808. doi:10.65109/ISFV2587
- [25] Junhao Qiu, Xin Chen, Liang Ge, Liyong Lin, Zhichao Lu, and Qingfu Zhang. 2026. Evolving Interdependent Operators with Large Language Models for

- Multi-Objective Combinatorial Optimization. In *Proceedings of the 43rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 306)*. PMLR, 101861–101898. <https://proceedings.mlr.press/v306/qui26a.html>
- [26] Bernardino Romera-Paredes, Mohammadamin Barekatin, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. 2024. Mathematical discoveries from program search with large language models. *Nature* 625 (2024), 468–475. doi:10.1038/s41586-023-06924-6
- [27] Stefan Ropke and David Pisinger. 2006. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science* 40, 4 (2006), 455–472. doi:10.1287/trsc.1050.0135
- [28] Shuichi Sakai, Mitsunori Togasaki, and Koichi Yamazaki. 2003. A note on greedy algorithms for the maximum weighted independent set problem. *Discrete Applied Mathematics* 126, 2–3 (2003), 313–322. doi:10.1016/S0166-218X(02)00205-6
- [29] StableSolver Contributors. 2026. StableSolver: Algorithms for the Maximum-Weight Independent Set Problem. <https://github.com/fontanf/stablesolver>. Open-source software repository. Accessed: 2026-10-01.
- [30] Richard S. Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112, 1–2 (1999), 181–211. doi:10.1016/S0004-3702(99)00052-1
- [31] Niki van Stein and Thomas Bäck. 2025. LLaMEA: A Large Language Model Evolutionary Algorithm for Automatically Generating Metaheuristics. *IEEE Transactions on Evolutionary Computation* 29, 2 (2025), 331–345. doi:10.1109/TEVC.2024.3497793
- [32] Federico Julian Camerota Verdù, Lorenzo Castelli, and Luca Bortolussi. 2025. Scaling Combinatorial Optimization Neural Improvement Heuristics with On-line Search and Adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 27135–27143. doi:10.1609/aaai.v39i25.34921
- [33] Yang Wu, Junran Pan, Yifan Zhang, Ning Xu, Fanshuo Zeng, and Jian Cheng. 2026. RefineEvo: Planning-Guided Heuristic Evolution with Bidirectional Experience. In *Proceedings of the 43rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 306)*. PMLR, 136582–136624. <https://proceedings.mlr.press/v306/wu26ap.html>
- [34] Yaixin Wu, Wen Song, Zhiguang Cao, and Jie Zhang. 2021. Learning Large Neighborhood Search Policy for Integer Programming. In *Advances in Neural Information Processing Systems*, Vol. 34. 30075–30087. <https://proceedings.neurips.cc/paper/2021/hash/fc9e62695def29ccdb9eb3fed5b4c8c8-Abstract.html>
- [35] Xianliang Yang, Ling Zhang, Haolong Qian, Lei Song, and Jiang Bian. 2025. HeurAgenix: Leveraging LLMs for Solving Complex Combinatorial Optimization Challenges. arXiv preprint, version 2. arXiv:2506.15196 [cs.AI] <https://arxiv.org/abs/2506.15196>
- [36] Shunyu Yao, Fei Liu, Xi Lin, Zhichao Lu, Zhenkun Wang, and Qingfu Zhang. 2025. Multi-Objective Evolution of Heuristic Using Large Language Model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 27144–27152. doi:10.1609/aaai.v39i25.34922
- [37] Furong Ye, Chuan Luo, and Shaowei Cai. 2025. Better Understandings and Configurations in MaxSAT Stochastic Local Search Solvers via Anytime Performance Analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 27153–27160. doi:10.1609/aaai.v39i25.34923
- [38] Haoran Ye, Jiarui Wang, Zhiguang Cao, Federico Berto, Chuanbo Hua, Haeyeon Kim, Jinkyoo Park, and Guojie Song. 2024. ReEvo: Large Language Models as Hyper-Heuristics with Reflective Evolution. In *Advances in Neural Information Processing Systems*, Vol. 37. 43571–43608. https://proceedings.neurips.cc/paper_files/paper/2024/hash/4ced59d480e07d290b6f29fc8798f195-Abstract-Conference.html
- [39] Huigen Ye, Hua Xu, An Yan, and Yaoyang Cheng. 2025. Large Language Model-driven Large Neighborhood Search for Large-Scale MILP Problems. In *Proceedings of the 42nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 267)*. PMLR, 72131–72180. <https://proceedings.mlr.press/v267/ye25j.html>
- [40] Yisong Zhang and Guoxing Yi. 2025. LAOS: Large Language Model-Driven Adaptive Operator Selection for Evolutionary Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 517–526. doi:10.1145/3712256.3726450
- [41] Zhi Zheng, Zhuoliang Xie, Zhenkun Wang, and Bryan Hooi. 2025. Monte Carlo Tree Search for Comprehensive Exploration in LLM-Based Automatic Heuristic Design. In *Proceedings of the 42nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 267)*. 78338–78373. <https://proceedings.mlr.press/v267/zheng25o.html>
- [42] Rui Zhong, Abdelazim G. Hussien, Jun Yu, and Masaharu Munetomo. 2025. LLMOA: A novel large language model assisted hyper-heuristic optimization algorithm. *Advanced Engineering Informatics* 64 (2025), 103042. doi:10.1016/j.aei.2024.103042