EMILI-VRP

Pablo Contreras Estrada Thomas Stützle
Leslie Pérez Cáceres

1 EMILI-VRP framework

EMILI [5] is a general framework for the automatic design of metaheuristics, a context-free grammar describes a large family of possible methods, and each valid derivation of this grammar corresponds to a complete algorithm with a well-defined structure and parameters [2].

In the CVRP setting, EMILI is instantiated as *EMILI-VRP*. Its design space includes:

- single-solution, trajectory-based schemes (e.g., VNS/VND, RVNS, LKH [1],ILS,AILS [4], TS-like schemes, skewed VNS variants),
- and population-based hybrid genetic algorithms, including a family based on the open-source Hybrid Genetic Search for the CVRP (HGS-CVRP) [6].

At the C++ level, all algorithms produced by EMILI-VRP:

- operate on the same CVRP solution representation (sets of routes),
- use the same cost and feasibility evaluation routines,
- and rely on the same library of neighborhoods, construction heuristics, and speed-up mechanisms.

The grammar exposes high-level building blocks such as construction heuristics, local search structures, perturbation mechanisms, acceptance criteria, population management rules, and termination conditions. EMILI translates each configuration into a concrete algorithm, and evaluates it on CVRP instances. The exploration of this design space is handled by the automatic configurator *irace* [3], which iteratively samples, tests, and refines configurations on a training set of instances.

1.1 CVRP model and solution representation

We address the classical symmetric Capacitated Vehicle Routing Problem (CVRP). Let $V = \{0, 1, \dots, n\}$ be the set of nodes, where node 0 is the depot and nodes $1, \dots, n$ are customers. Each customer i has a non-negative demand d_i, and a fleet of identical vehicles with capacity Q is available. For each pair of nodes

(i, j), a non-negative travel cost c_{ij} is given, typically derived from Euclidean or road distances.

A solution is a set of routes, each route being an ordered sequence of customers starting and ending at the depot, such that:

- every customer appears in exactly one route,
- the sum of demands in each route does not exceed Q,
- and the total travel cost over all routes is minimized.

Internally, EMILI-VRP stores, for each solution:

- the sequence of customers in each route,
- the load and cost of each route,
- auxiliary data structures for fast evaluation, such as nearest-neighbor lists, don't-look bits, and incremental delta-cost information.

1.2 Algorithmic components in EMILI-VRP

1.2.1 Construction heuristics

Each algorithm derived from the EMILI-VRP grammar starts from one or more initial solutions generated by a construction component. The framework includes several CVRP constructors:

- **Random construction**: customers are randomly permuted and greedily inserted into the current last route, opening a new route when the capacity constraint would be violated.
- **Nearest-neighbor construction**: routes are grown by repeatedly inserting the closest feasible customer to the current end of the route, until no more customers fit and a new route is started.
- **Insertion heuristics**: at each step, the algorithm considers a set of candidate customer–position pairs and selects the insertion with minimum marginal cost, optionally combining cost and distance-to-depot or demand terms.

The grammar encodes which constructor is used, with which parameters (e.g., savings formula weights, candidate list sizes), and whether multiple initial solutions are generated. This choice is handled automatically by the configuration process; the framework itself does not hard-code any particular constructor as the best one.

1.2.2 Neighborhoods and local search

The local search component is described in the grammar as a sequence of neighborhoods together with a search strategy. EMILI-VRP provides a broad catalogue of neighborhoods:

Inter-route moves

- *Shift(1,0)* and *Shift(2,0)*
- *Swap(1,1)*, *Swap(2,1)* and *Swap(2,2)*
- *CROSS* and Cross-Exchange variants
- *Swap**

Intra-route moves

- single and double reinsertion within a route,
- pairwise customer exchanges within a route,
- 2-opt and 3-opt segment reversals.

Removal–insertion moves (RI)

- a destructive phase, in which one or several customers are removed according to a rule (random removal, worst-cost removal, route-based removal),
- a repair phase, in which the removed customers are reinserted using best insertion, regret-based insertion or randomized variants.

1.2.3 Perturbation and diversification

To escape local minima, algorithms produced by EMILI-VRP include perturbation steps, which are also grammar-controlled:

- **Random shaking**: a small number of random moves, usually using stronger neighborhoods (e.g., CROSS, 3-opt, or large RI moves).
- **Adaptive intensity**: the number and magnitude of perturbation moves can increase after many non-improving iterations.

1.2.4 Speed-up mechanisms

To keep runtimes under control, especially on large-scale CVRP instances, EMILI-VRP integrates standard speed-ups:

- **Don't-look bits**: customers that have failed to yield improvements for some iterations are temporarily skipped during neighborhood exploration.
- **Nearest-neighbor lists**: candidate moves are restricted to a limited set of closest customers, reducing the effective neighborhood size.

Thresholds and list lengths are tuned by the configurator, allowing EMILI to adapt the aggressiveness of pruning to the problem size.

1.2.5 Search schemes encoded in the grammar

At the highest level, the grammar controls the overall search scheme. EMILI-VRP can instantiate:

- **Trajectory-based metaheuristics**, where a single current solution is iteratively improved by alternating local search, perturbation and acceptance steps. By combining the components described above in different ways, EMILI can express classical VNS, skewed VNS, ILS and TS-like patterns, all operating on the same CVRP neighborhoods and cost structures.

- **Population-based hybrid genetic algorithms**, implemented by wrapping a hybrid genetic search of the HGS-CVRP family as an additional algorithm template in the design space. The underlying engine follows the open-source implementation of HGS-CVRP [6].

References

[1] Keld Helsgaun. An effective implementation of the Lin–Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000.

[2] Carlos López-Ibáñez and Thomas Stützle. Automatic design of metaheuristic algorithms. In Michel Gendreau and Jean-Yves Potvin, editors, *Handbook of Metaheuristics*, pages 273–328. Springer, 3 edition, 2019.

[3] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari, and Thomas Stützle. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58, 2016.

[4] Vinícius R Máximo, Jean-François Cordeau, and Mariá CV Nascimento. Ails-ii: An adaptive iterated local search heuristic for the large-scale capacitated vehicle routing problem. *INFORMS Journal on Computing*, 36(4):974–986, 2024.

[5] Federico Pagnozzi and Thomas Stützle. Automatic design of hybrid stochastic local search algorithms for permutation flowshop problems. *European journal of operational research*, 276(2):409–421, 2019.

[6] Thibaut Vidal. Hybrid genetic search for the cvrp: Open-source implementation and new benchmark results. *Computers & Operations Research*, 2022. Preprint and code available online.