# A Budget-Aware Hybrid ACO–SA Approach for Constructive Triangle Reassembly under Geometric and Combinatorial Constraints

Damir Hasanspahić[1], Adis Alihodžić[2], and Damir Hasić[3]

[1] University of Sarajevo, Sarajevo, Bosnia and Herzegovina
damir.hasanspahic@gf.unsa.ba
[2] University of Sarajevo, Sarajevo, Bosnia and Herzegovina
adis.alihodzic@pmf.unsa.ba
[3] University of Sarajevo, Sarajevo, Bosnia and Herzegovina
d.hasic@pmf.unsa.ba

**Abstract.** We model triangle reassembly as a geometric–combinatorial optimization problem and evaluate several canonical constructive metaheuristics together with a lightweight budget-aware hybrid under strict geometric feasibility constraints and fixed wall-clock budgets. Following the formulation introduced in [1], we start from triangles generated by a Delaunay triangulation of the container, independently rotate and translate them, discard all adjacency information, and then seek to reassemble a non-overlapping subset inside the original container so as to maximize covered area. The search space combines hard geometric feasibility with rapidly growing combinatorial ambiguity and is highly sensitive to early constructive decisions. We evaluate canonical baseline based on Ant Colony Optimization (ACS, MMAS) [9,22,11], Simulated Annealing [15,16], and tabular Q-learning [26,23]. We also introduce a lightweight budget-aware hybrid ACO–SA method in which one SA-based constructive ant, operating under a fixed sub-budget, is embedded into an ACO colony cycle and contributes directly to pheromone updates. All methods use the same geometric primitive—the length-matching attachment rule (Strategy S4 in [1])—and are compared under identical geometric operators, feasibility checks, and matched wall-clock budgets. Experiments on unit-square instances ranging from 94 to 1994 triangles show that, across 30 independent runs, the hybrid achieves the highest coverage on 4 of 6 instance sizes and remains close to the best baseline on the other two. On the largest instance, it improves coverage by 6.09 percentage points over MMAS under the same budget.

**Keywords:** Geometric reassembly · Computational geometry · Triangle packing · Combinatorial optimization · Hybrid metaheuristics · Ant colony optimization · Simulated annealing

## 1 Introduction

Packing and reassembly problems in computational geometry study how geometric objects can be placed inside bounded domains under strict feasibility

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_17

constraints while optimizing a global objective. When the objects are irregular and their mutual adjacency information is missing—as in the reassembly of a scattered triangulation—the search space combines continuous geometric constraints with a combinatorial explosion of feasible attachment structures. More broadly, irregular packing and nesting problems arise in cutting and manufacturing, logistics, robotics, and computer graphics, and are often addressed through metaheuristic search strategies or geometry-based optimization models [17,13].

Our focus is the reconstruction formulation introduced in [1]. We start from a Delaunay triangulation [8] of the container, whose triangles tile the domain exactly. After independent random rigid motions are applied and all adjacency information is discarded, reassembly becomes a nontrivial search problem. Our earlier study proposed four lightweight geometry-only strategies and found that the length-matching attachment rule was the most effective, since it attaches triangles only along equal or nearly equal edge lengths [1]. Building on that formulation, the present work implements and evaluates canonical metaheuristics, introduces a lightweight budget-aware hybrid, and extends the experimental study from the earlier limit of 114 triangles to unit-square instances with up to 1994 triangles, while constraining all methods to the same geometric primitive.

Scaling amplifies two coupled difficulties: edge-length matching becomes increasingly ambiguous in large Delaunay-derived sets, where many edge lengths are near-duplicates, while feasibility checking becomes increasingly costly as the partial packing grows [4,2]. Under fixed budgets, the algorithm must discover long and mutually consistent match chains while avoiding expensive dead ends. This may induce stagnation in purely constructive procedures.

To isolate the impact of search control, we evaluate canonical metaheuristics for constructive combinatorial optimization—Ant Colony Optimization (ACO), Simulated Annealing (SA), and a minimal reinforcement-learning baseline based on Q-learning—while restricting every method to the same length-matching attachment rule inherited from Strategy S4 [1]. Our main algorithmic proposal is a lightweight budget-aware hybrid, ACO–SA: within the ACO colony cycle, we replace one standard ant with one SA-based constructive ant operating under a fixed sub-budget and allow its solution to participate directly in pheromone learning. The design follows a standard intensification–diversification rationale while remaining compact enough to preserve the shared wall-clock budget.

*Contributions.*

1. **A rescaled evaluation setting with matched budgets:** We extend the experimental range of scattered-triangulation reassembly from the earlier limit of 114 triangles to unit-square instances with up to 1994 Delaunay-derived triangles, while keeping a shared constructive geometry pipeline and matched wall-clock budgets across canonical ACO, SA, and Q-learning baselines.
2. **A lightweight budget-aware hybrid:** We replace one standard ant with a SA-based constructive ant operating under a fixed sub-budget. Its solution participates directly in pheromone updating rather than serving only as a post-construction improvement step.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_17

## 2 Related Work

Irregular packing and nesting are well-studied problem classes with a substantial body of models and heuristic approaches, motivated by applications in manufacturing and cutting, logistics, robotics, and computer graphics [17,13]. From a computational complexity standpoint, many geometric packing variants are intractable in general. For example, packing identical simple polygons is NP-hard [3]. Triangle packing problems exhibit related computational hardness phenomena [7,12], and specialized triangle packing has been studied in discrete optimization and approximation settings [27,14,18,6,25].

Our setting differs from most standard packing formulations. The input triangles are not arbitrary parts but originate from a triangulation that exactly tiles the container, and the task is to reconstruct compatible adjacency relations after scattering. To our knowledge, this "reassembly of scattered triangulations" perspective has not been studied systematically beyond our initial formulation and geometry-only strategies in [1].

The metaheuristic families used here are standard. ACO was introduced as Ant System [10] and refined into ACS and MMAS [9,22]; a comprehensive reference is [11]. SA is based on Metropolis acceptance and remains a canonical metaheuristic for stochastic search [19,15,16]. Q-learning is a classical model-free reinforcement-learning algorithm [26,23]. In this paper we keep these methods canonical and restrict them to the same length-matching attachment rule, so differences primarily reflect search control rather than different placement primitives.

Hybrid metaheuristics combining ACO with local search or annealing-based components have been explored in several optimization settings. In the MMAS line, local search is applied after ants construct candidate tours [21], whereas in dynamic-TSP variants pheromone information is transferred across successive problem iterations [20].

Within our reconstruction setting, we have not found an equivalent budget-matched design in which a single solution produced by an SA-based constructive ant under one constructive geometry rule is injected directly into pheromone updating rather than used only as an external improvement phase.

## 3 Problem Formulation

Let the container be the unit square $R = [0,1] \times [0,1]$ with area $Area(R) = 1$. For a given number of points $n$, define a point set $X = \{x_1, \ldots, x_n\} \subset R$ such that four points are the corners of the square $R$, and the remaining $n-4$ points lie strictly in the interior (no three are collinear) [1]. The Delaunay triangulation of the set $X$ partitions $R$ into non-overlapping triangles $T = \{T_1, \ldots, T_m\}$ whose union is exactly $R$ [8,4]. A planar triangulation for a square whose hull has size $h = 4$ generates $m = 2n - 6$ triangles [4]. Each triangle $T_i$ is then moved arbitrarily in the plane by an arbitrary translation and an arbitrary rotation. This produces an unordered set of triangles $T'$ in which all positional and adjacency information is discarded [1].

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_17

A solution is a subset $S \subseteq T'$ together with one rigid placement for each triangle in $S$. Let $\widetilde{T}_i$ denote triangle $T_i$ mapped by a rigid motion into the container $R$. The placed triangles must satisfy the constraints

$$\widetilde{T}_i \cap \widetilde{T}_j = \emptyset \quad \forall T_i \neq T_j \in S, \tag{1}$$

$$\widetilde{T}_i \subseteq R \quad \forall T_i \in S. \tag{2}$$

The objective is to maximize the covered area, expressed through the coverage measure

$$P(S) = 100 \cdot \frac{\sum_{T_i \in S} Area(T_i)}{Area(R)} \in [0, 100]. \tag{3}$$

Since $T$ tiles $R$, the theoretical optimum is 100%.

## 4 Methods

### 4.1 Shared Constructive Procedure

All methods share the same state representation (a feasible partial packing in $\mathbb{R}^2$), the same candidate generation based on edge lengths (4) inherited from Strategy S4 in [1], the same feasibility checks (1)–(2), and the same initialization policy, in which construction begins by selecting a triangle that has one of the longest edges in the input set and aligning that longest edge with a boundary edge of the unit square. An anchor is an exposed edge of an already placed triangle, and candidate placements are generated only by matching edges of unused triangles to such anchors.

Let $\ell(e)$ denote the length of edge $e$, and let $\varepsilon > 0$ be a fixed numerical tolerance. A triangle $U$ may be attached to an already placed triangle $V$ only if there exist edges $e_U \subset U$ and $e_V \subset V$ whose lengths are compatible according to

$$\left|\ell(e_U) - \ell(e_V)\right| \leq \varepsilon. \tag{4}$$

For such a compatible edge pair, the candidate placement is defined by the unique rigid motion (with possible swapping of the edge endpoints) that aligns $e_U$ with $e_V$, subject to (1)–(2).

This shared procedure ensures a fair and controlled comparison, because the only differences arise from search strategies rather than from geometric operator or initialization effects.

### 4.2 Canonical Baseline Methods

**Ant Colony Optimization (ACO): ACS and MMAS** Ant Colony Optimization (ACO) is a canonical family of constructive metaheuristics [10,11]. In our setting, an ant constructs a packing by repeatedly (i) selecting an anchor

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_17

on the current boundary, defined as an exposed edge of an already placed triangle, and (ii) selecting a compatible unused triangle–edge pair from the length-matching neighborhood induced by (4). Candidate selection follows the standard pheromone–heuristic rule

$$p(i \to j) = \frac{\tau_{ij}^{\alpha}\, \eta_j^{\beta}}{\sum_{k \in \mathcal{N}(i)} \tau_{ik}^{\alpha}\, \eta_k^{\beta}}, \tag{5}$$

where $\tau_{ij}$ denotes the pheromone value associated with the attempted anchor–candidate relation $(i \to j)$, $\eta_j$ is a simple desirability score (we use triangle area as a size-aware heuristic component, consistent with common geometry-based heuristics in irregular packing [17,13]), and $\mathcal{N}(i)$ is the length-compatible neighborhood defined by (4); full geometric feasibility is checked separately through (1)–(2). Pheromone evaporation is performed once per iteration, and pheromone reinforcement is proportional to solution quality, following canonical ACO schemes [11]. The two baselines differ only in their standard, literature-defined update mechanisms. ACS employs a pseudo-random proportional rule with local pheromone updates during construction in addition to global updates [9,11], whereas MMAS stabilizes learning by bounding pheromone values and reinforcing only the best solution(s), typically iteration-best or global-best [22,11]. In our implementation, both variants are kept faithful to their canonical definitions and differ only through the shared feasibility checks and the shared length-matching attachment rule.

**Simulated Annealing (SA)** Simulated Annealing (SA) is implemented in its canonical Metropolis form with a geometric cooling schedule [19,15,16]. A state corresponds to a feasible partial packing $S$. The neighborhood is defined primarily through ADD moves, which attempt to extend $S$ by attaching a single triangle via the length-matching attachment rule and are accepted only if feasibility is preserved, and occasional REMOVE moves, which delete a previously placed triangle to restore flexibility when the construction becomes locked. Given a proposal $S \to S'$ with objective difference $\Delta = P(S') - P(S)$, the move is accepted if $\Delta \geq 0$ and otherwise with probability $\exp(\Delta/T)$ at temperature $T$ [15,16].

**Tabular Q-learning (QRL)** To provide a minimal reinforcement-learning baseline, we model constructive reassembly as a Markov decision process (MDP). The state is represented by the current anchor context on the boundary of the partial packing, an action selects a candidate attachment from the length-matching pool, and the reward is the incremental coverage gain together with a small penalty for failed attachment attempts. Learning follows canonical one-step tabular Q-learning with $\varepsilon$-greedy exploration [26,23]. To keep the baseline computationally tractable, the policy is evaluated on uniformly sampled subsets of eligible anchors and feasible actions, and episode length is capped. This baseline is simple: geometric feasibility is handled through the environment dynamics rather than being encoded explicitly in the policy representation.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_17

### 4.3 Budget-Aware Hybrid ACO–SA

Hybrid metaheuristics typically combine population-level guidance with local intensification [24,5]. In our setting, the two components play complementary roles: ACO provides reusable colony memory across ants, whereas SA can occasionally accept short-term deteriorations and escape early locking within a single constructive trajectory. Under the length-matching attachment rule (4), large instances contain many plausible near-matches, so early stochastic choices may reinforce relations that are frequent yet geometrically unproductive. Purely constructive ACO can then spend an increasing fraction of its budget on rejected feasibility checks, whereas pure SA may discover useful relations without propagating them as reusable population-level memory [11,19,16]. The proposed ACO–SA hybrid addresses this gap with minimal intervention: one SA-based constructive trajectory is embedded inside the colony and its output participates immediately in pheromone learning. The distinctive element is therefore not generic ACO–SA hybridization, but a budget-matched constructive integration in which a single solution produced by the SA-based constructive ant enters the same pheromone update under the same geometric rule and within the same wall-clock budget.

**Adapted and Reused Components** The hybrid retains a lightweight ACO construction loop and replaces exactly one standard constructive ant with one SA-based constructive ant within the same colony cycle:

- *Relative to ACS*, the hybrid shares the general pheromone-guided construction principle (5) but does *not* introduce ACS-specific step-wise local updates or the $q_0$ pseudo-random proportional rule; ACS is evaluated separately as a baseline [9,11].
- *In the spirit of MMAS*, we retain only a minimal stabilization safeguard: after evaporation, the global pheromone scale is bounded below by a small $\tau_{\min}$ to prevent numerical collapse. We omit MMAS-specific $\tau_{\max}$ clipping and elitist best-only reinforcement so that the hybrid remains lightweight within a common ACO framework [22,11].
- *At the global-update level*, we use a simple lightweight colony update: after evaporation, every solution constructed in the current colony cycle reinforces the relation multipliers associated with the attachments it actually used, in proportion to solution quality. The hybrid-specific addition is only that the solution produced by the SA-based constructive ant enters this same update, so relations discovered by SA become immediately reusable colony memory. This keeps the ACO–SA–0SA ablation directly comparable to a lightweight non-hybrid colony variant [10,11].

Thus, the key adaptation beyond canonical ACO is *where intensification enters*: one SA-based constructive trajectory, guided by the current pheromone model and operating under a fixed sub-budget, feeds directly into the same pheromone update and reusable colony memory. Budget-awareness is achieved by replacement rather than addition: the SA trajectory takes the place of one standard ant inside the common wall-clock budget.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_17

**Algorithm 1** ACO–SA (Budget-Aware Hybrid with an SA-Based Constructive Ant)

**Require:** Time budget $B$; total colony size $N_{\text{col}}$; ACO parameters $(\alpha, \beta, \rho, Q, \tau_{\min})$; SA budget $b_{\text{SA}}$

Initialize pheromones $\tau \leftarrow \tau_0$ and best-so-far solution $S^\star \leftarrow \emptyset$
**while** time $< B$ **do** ▷ one ACO iteration
  **for** $k = 1$ to $N_{\text{col}}$ **do** ▷ construct $N_{\text{col}}$ solutions under the shared operator (4)
    **if** $k = 1$ **then**
      $S_k \leftarrow \textsc{FastSAConstruct}(\tau, b_{\text{SA}})$
    **else**
      $S_k \leftarrow \textsc{ConstructAnt}(\tau)$
    **end if**
    **if** $P(S_k) > P(S^\star)$ **then**
      $S^\star \leftarrow S_k$
    **end if**
  **end for**
  Evaporation: $\tau \leftarrow (1-\rho)\tau$; lower bound: $\tau \leftarrow \max(\tau, \tau_{\min})$
  Reinforcement (iteration-level colony update): for each solution $S_h$ constructed in the current iteration and each relation $(i \to j)$ used in $S_h$, update $\tau_{ij} \leftarrow \tau_{ij} + Q \cdot P(S_h)$
**end while**
**return** $S^\star$

**Fast SA-Based Constructive Ant** The SA agent constructs a solution under the same strict feasibility procedure as all other methods and is explicitly tuned to operate on the time scale of a single ant. It uses a short cooling schedule, proposes mainly ADD moves, and allows occasional REMOVE moves to recover flexibility when the construction becomes locked. Within an ADD move, candidate ordering is biased by the same pheromone–heuristic preference $\tau^\alpha \eta^\beta$ used in ACO (5), so SA trajectories remain aligned with colony memory while still exploring through Metropolis acceptance [19,15,16]. The SA component is not a post-processor: its constructed solution participates in pheromone learning in the same way as the solutions produced by the other ants. Numerical settings are summarized in Section 5.

## 5 Experimental Setup

*Instance families and budget schedule.* The instances are generated as described in Section 3. The study uses six point-set sizes: $n = 50$, 100, 300, 500, 700, and 1000. These correspond to $m = 94$, 194, 594, 994, 1394, and 1994 triangles, with matched wall-clock budgets of 15, 30, 90, 150, 210, and 300 seconds. Each algorithm is executed 30 times with different random seeds.

*Shared execution protocol.* All compared algorithms use identical geometric precomputations, candidate generation, feasibility checks, and deterministic initialization. The wall-clock timer starts immediately before the first constructive

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_17

decision, and each run stops as soon as the common budget $B$ is exhausted; no method receives additional post-processing time beyond this limit. In the hybrid, one standard ant is replaced by one SA-based constructive ant inside the same colony cycle, so the SA effort is paid from the common wall-clock budget rather than added on top of it.

*Metrics.* For all main methods, we report best-of-30 coverage together with the run-level mean and standard deviation; in the tables, the latter is written compactly as mean(std). The primary objective is coverage (3). To expose where computation is spent, we record the number of *attachment attempts* $A$ (candidate placements tested) and the number of *overlap tests* $O$ (invocations of the non-overlap check). We further report the normalized ratios $A/P$ and $O/P$. Since $P \in [0, 100]$ varies within a relatively narrow band in practice while $A$ and $O$ increase sharply with instance size, these ratios should be interpreted as per-unit-coverage operational cost; the scaling signal is driven mainly by the numerators rather than by variation in achieved coverage. For the largest instance we provide a mean Pareto view (coverage versus overlap tests). For the ablation study we report both best-of-30 and mean(std) coverage, and we use mean $O/P$ to summarize quality–cost behavior. We additionally include a small budget-sensitivity check at $m = 194$ under a doubled wall-clock budget. Because no normality assumption is imposed on the run-level outcomes, inferential comparison is based on two-sided Mann–Whitney $U$ tests on the 30 coverage values per method, with Holm-adjusted $p$-values for the size-wise comparisons summarized in Table 3.

*Parameter settings.* Method-specific parameters were fixed in short pilot experiments conducted before the reported 30-run evaluation and then kept unchanged across all instance sizes; no per-instance retuning was performed. The tolerance in the shared length-matching attachment rule (4) was set to $\varepsilon = 10^{-10}$. Hybrid ACO–SA used a total colony size of 10 solutions per iteration, with 9 standard ants and 1 SA-based constructive ant; its remaining parameters were $\alpha = 1.0$, $\beta = 2.0$, $\rho = 0.10$, $Q = 1.0$, $\tau_0 = 1.0$, and $\tau_{\min} = 10^{-4}$. Its embedded fast SA used $T_0 = 0.5$, cooling factor 0.85, $k = 10$ temperature levels, and remove probability $p_{\text{rem}} = 0.06$. MMAS used $n_{\text{ants}} = 10$, $\alpha = 1.0$, $\beta = 2.5$, $\rho = 0.20$, and $Q = 1.0$. ACS used $n_{\text{ants}} = 10$, $\alpha = 1.0$, $\beta = 2.5$, $\rho = 0.12$, $\xi = 0.10$, $q_0 = 0.80$, and $\tau_0 = 1.0$. QRL used learning rate 0.10, discount factor 0.97, exploration rate 0.12, fail penalty 0.05, and a capped episode length of $5m$ for $m \leq 300$ and $\min(3.2m, 3200)$ otherwise. The standalone SA baseline used $T_0 = 1.0$, cooling factor 0.995, $k = 1000$ temperature levels, $L = 5m$, and $p_{\text{rem}} = 0.06$.

*Implementation environment.* All experiments were executed on a single machine to ensure consistent wall-clock measurements across all compared methods. The implementation was written in Python 3.11 and executed in single-threaded mode on a standard x86–64 laptop equipped with an Intel® Core™ i5–3360M CPU running at 2.80 GHz and 8 GB of RAM. Since all algorithms were evaluated under identical time budgets on the same hardware and software environment,

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_17

**Table 1.** Coverage $P$ (%) over 30 runs by instance size and wall-clock budget; rows report best-of-30 and mean(std)

| | | Instance size ($m = 2n - 6$) / Budget $B$ (s) | | | | | |
|---|---|---|---|---|---|---|---|
| Method | Statistic | $m = 94$ (15s) | $m = 194$ (30s) | $m = 594$ (90s) | $m = 994$ (150s) | $m = 1394$ (210s) | $m = 1994$ (300s) |
| ACO–SA | best | 90.27 | 88.02 | 89.61 | 93.36 | 90.17 | 92.09 |
| | mean(std) | 85.75(2.44) | 84.38(1.63) | 85.16(1.83) | 87.31(2.83) | 84.99(2.12) | 86.83(2.34) |
| ACS | best | 87.71 | 88.56 | 83.51 | 91.92 | 85.47 | 85.20 |
| | mean(std) | 85.43(1.49) | 83.54(1.95) | 82.56(0.52) | 86.90(2.23) | 82.46(1.32) | 81.82(1.43) |
| MMAS | best | 86.82 | 83.64 | 85.79 | 90.21 | 90.24 | 86.00 |
| | mean(std) | 84.46(1.96) | 82.71(0.59) | 82.93(1.10) | 85.60(1.90) | 85.74(1.84) | 84.19(0.96) |
| QRL | best | 82.80 | 83.64 | 81.52 | 83.42 | 80.37 | 82.03 |
| | mean(std) | 80.85(1.84) | 79.00(2.74) | 78.30(2.27) | 80.28(2.18) | 76.85(1.77) | 79.87(1.43) |
| SA | best | 73.15 | 79.35 | 78.47 | 79.62 | 75.88 | 80.22 |
| | mean(std) | 68.31(2.62) | 71.15(4.74) | 69.82(3.85) | 73.48(3.01) | 68.47(3.51) | 70.82(4.51) |

the reported results reflect differences in search behavior and learning dynamics rather than absolute computational throughput.

## 6 Results and Discussion

### 6.1 Coverage

Table 1 reports both best-of-30 and mean(std) coverage across all instance sizes, while Fig. 1 visualizes the same comparison with the best-run view on the left and the mean-over-30 view on the right. Descriptive summaries are complemented by the corrected run-level tests reported in Table 3. Under the best-run view, the proposed ACO–SA hybrid attains the highest coverage on four instance sizes ($m \in \{94, 594, 994, 1994\}$). On the remaining two sizes, it stays very close to the strongest baseline: at $m = 194$ ACS is ahead by 0.54 percentage points (88.56% vs. 88.02%), and at $m = 1394$ MMAS is ahead by 0.07 points (90.24% vs. 90.17%). Averaged over all six sizes, ACO–SA achieves the highest best-run coverage profile (90.59%). The mean view comparison is consistent and slightly stronger in favor of the hybrid: ACO–SA has the highest run-level mean on five of the six sizes, trailing only MMAS at $m = 1394$. On the largest instance, ACO–SA improves best-of coverage by 6.09 points over MMAS (92.09% vs. 86.00%) and mean coverage by 2.65 points (86.83% vs. 84.19%) under the same budget. Together, the two views show that the hybrid improves both peak and typical behavior at scale without changing the shared geometric primitive.

### 6.2 Operational Cost, Normalized Efficiency, and Scaling

Figure 2 reports mean operational counters over 30 runs: attachment attempts $A$ and overlap tests $O$. Both counters rise sharply with instance size because the shared constructive pipeline must resolve more ambiguous length matches while checking feasibility against a progressively larger partial packing; in the present regime, the observed growth appears steeper than linear. A simple visual ratio

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_17

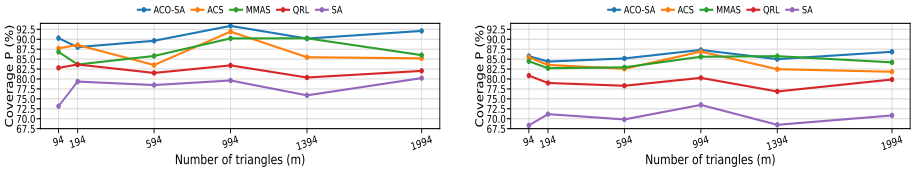


**Fig. 1.** Coverage $P$ across sizes: best-of-30 (left) and mean over 30 runs (right)

check from $m = 994$ to $m = 1994$ is already indicative: the instance size grows by a factor of 2.01, while the cost curves in Fig. 2 rise by clearly more than twofold for the ACO-based methods. We nevertheless retain the same linear schedule for all methods, as it keeps the study computationally manageable while preserving a fair comparison under identical hardware, operators, feasibility checks, and stopping rules. The mean-cost curves show that ACO–SA remains substantially cheaper than ACS and MMAS in attachment effort at large sizes. At $m = 1994$, ACO–SA uses $7.99 \times 10^5$ attachment attempts on average, compared with $1.15 \times 10^6$ for MMAS and $1.56 \times 10^6$ for ACS. Mean overlap counts at the same size are $6.63 \times 10^5$ for ACO–SA, $6.46 \times 10^5$ for MMAS, and $7.90 \times 10^5$ for ACS. Thus, the hybrid reaches the highest coverage with overlap effort that remains close to MMAS and clearly below ACS.

Figure 3 normalizes these counters by achieved coverage. Because $P$ stays within a relatively narrow band whereas $A$ and $O$ grow by orders of magnitude with $m$, the ratios $A/P$ and $O/P$ are driven mainly by the growth of the counters themselves. They therefore indicate how quickly attachment effort and feasibility checking expand per unit of realized coverage as the instances become larger. QRL remains the most economical method by both ratios, but this economy comes with a clear coverage deficit across all sizes. Among the high-coverage methods, ACO–SA offers the strongest $A/P$ profile from $m \geq 594$ onward and, on the largest instance, also the best $O/P$ among ACO–SA, ACS, and MMAS (7645.6 vs. 7677.0 vs. 9657.5). The main exception is $m = 1394$, where MMAS is slightly stronger than the hybrid in both mean coverage and $O/P$; this is the same size at which MMAS is also the only non-hybrid method to lead the mean-coverage panel in Fig. 1. Overall, the mean-cost view supports the same conclusion as the coverage plots: at large scale the hybrid spends its budget more selectively than the canonical ACO baselines.

A small supplementary check at fixed size $m = 194$ with a doubled budget (30 s $\rightarrow$ 60 s), using 10 runs per method, shows that only MMAS slightly exceeds its previous best-of value (83.64 $\rightarrow$ 83.69), while ACS and ACO–SA remain essentially unchanged. This suggests that additional time can help in some cases, but moderate budget increases do not reliably improve realized best-of outcomes at every size, so the main study retains the shared linear schedule.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_17

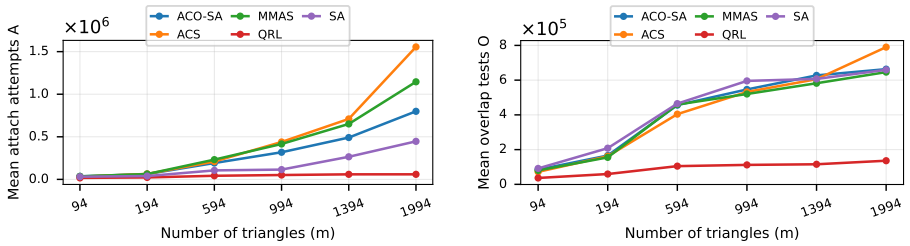


**Fig. 2.** Mean operational counters across sizes: $A$ (left) and $O$ (right)

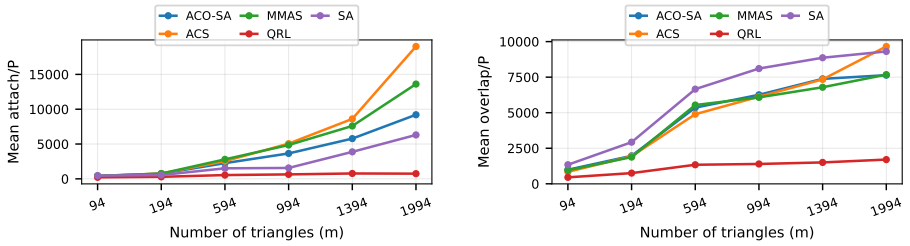


**Fig. 3.** Mean normalized counters: $A/P$ (left) and $O/P$ (right)

### 6.3 Pareto View for the Largest Instance

Figure 4 gives the mean coverage–overlap Pareto view at $m = 1994$. The hybrid occupies the most favorable high-coverage region: it attains the highest mean coverage, keeps overlap effort almost identical to MMAS, and remains clearly below ACS in overlap cost. This is consistent with the interpretation that one SA-based constructive trajectory provides useful guidance to the colony, helping it spend more of the budget on productive match chains rather than on repeatedly exploring ambiguous dead ends.

### 6.4 Ablation Study

The ablation study isolates three effects: (i) removing SA entirely (ACO–SA–0SA), (ii) retaining SA while excluding SA-generated solutions from pheromone updates (ACO–SA–no-update), and (iii) increasing the fraction of SA-based construction to 50% (ACO–SA–50%SA and ACO–SA–50%SA–no-update). Table 2 reports both best-of and mean(std) coverage, while Fig. 5 focuses on the mean view through coverage and $O/P$.

Across both views, the full hybrid remains the strongest overall variant. The mean rows are especially informative. At $m = 194$, for example, ACO–SA–no-update attains the best single run (88.66%) but its mean coverage (82.35%) is clearly below the full hybrid (84.38%), showing that removing pheromone learning from SA-generated solutions weakens typical behavior even when an

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_17

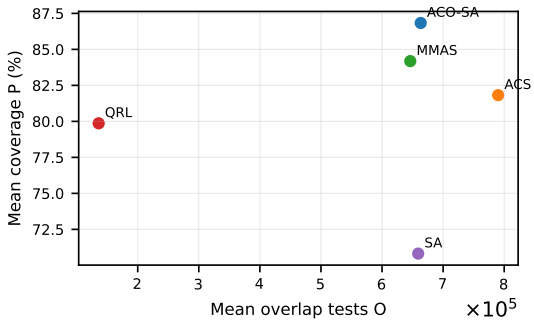


**Fig. 4.** Mean coverage–overlap Pareto view at $m = 1994$

**Table 2.** Ablation of SA usage and pheromone participation; rows report best-of-30 and mean(std) coverage

| | | Instance size ($m = 2n - 6$) | | | | | |
|---|---|---|---|---|---|---|---|
| Variant | Statistic | $m = 94$ | $m = 194$ | $m = 594$ | $m = 994$ | $m = 1394$ | $m = 1994$ |
| ACO–SA-full | best | 90.27 | 88.02 | 89.61 | 93.36 | 90.17 | 92.09 |
| | mean(std) | 85.75(2.44) | 84.38(1.63) | 85.16(1.83) | 87.31(2.83) | 84.99(2.12) | 86.83(2.34) |
| ACO–SA–0SA | best | 87.40 | 88.28 | 85.21 | 86.33 | 88.00 | 90.84 |
| | mean(std) | 82.19(2.47) | 83.61(2.09) | 81.83(1.59) | 81.78(2.15) | 84.00(1.80) | 85.99(2.52) |
| ACO–SA–no-update | best | 86.39 | 88.66 | 83.96 | 89.44 | 89.68 | 90.45 |
| | mean(std) | 82.09(2.46) | 82.35(2.62) | 81.30(1.38) | 83.91(2.23) | 86.06(2.33) | 84.93(2.57) |
| ACO–SA–50%SA | best | 85.27 | 85.88 | 84.22 | 87.69 | 87.98 | 88.43 |
| | mean(std) | 79.61(3.11) | 80.62(3.88) | 81.07(1.40) | 84.26(1.56) | 84.24(2.22) | 83.51(2.18) |
| ACO–SA–50%SA–no-update | best | 84.02 | 83.73 | 83.92 | 86.71 | 85.84 | 86.78 |
| | mean(std) | 79.35(2.24) | 79.37(1.98) | 81.84(1.39) | 82.16(1.94) | 82.42(1.46) | 82.93(2.05) |

occasional strong run still occurs. The $O/P$ panel leads to the same reading: the heavier 50%SA variants do not yield a systematically cleaner quality–cost profile at larger sizes, but mostly spend more operational effort per unit of achieved coverage without proportional gains. These results support the interpretation of the hybrid: a small, budget-matched SA component is most useful when it acts as an information source within pheromone learning rather than as a heavier parallel search component.

### 6.5 Statistical Tests

Table 3 reports size-wise significance tests comparing ACO–SA with the strongest non-hybrid baseline at each size, using the 30 run-level coverage values. After Holm correction, only the comparisons at $m = 594$ and $m = 1994$ remain statistically significant. This is consistent with the magnitude of the observed gaps: the larger-scale advantages of ACO–SA persist under run-to-run variability, whereas the small run-level gaps at $m = 94$, 194, 994, and 1394 do not remain significant after correction. The case $m = 194$ remains instructive: ACS attains the slightly

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_17

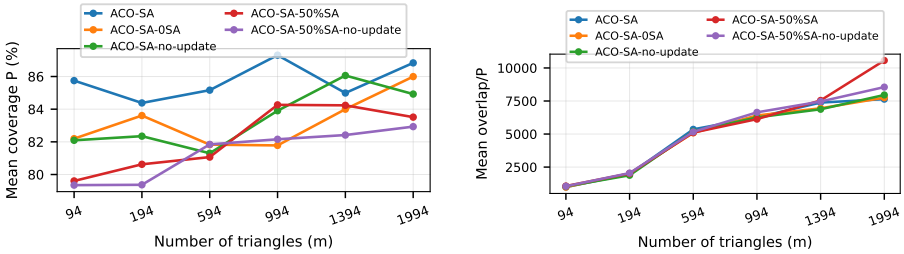


**Fig. 5.** Ablation means across sizes: coverage (left) and $O/P$ (right)

**Table 3.** Size-wise two-sided Mann–Whitney $U$ tests comparing ACO–SA with the strongest non-hybrid baseline at each size; $p$-values are Holm-corrected across the six tests

| $m$ | Baseline | $U$ | $p$ | $p_{\text{Holm}}$ |
|---|---|---|---|---|
| 94 | ACS | 460.5 | 0.882 | 1.000 |
| 194 | ACS | 586.0 | 0.045 | 0.181 |
| 594 | MMAS | 785.5 | $7.31 \times 10^{-7}$ | $4.38 \times 10^{-6}$ |
| 994 | ACS | 469.0 | 0.784 | 1.000 |
| 1394 | MMAS | 363.0 | 0.201 | 0.603 |
| 1994 | MMAS | 765.0 | $3.32 \times 10^{-6}$ | $1.66 \times 10^{-5}$ |

better best-of value, but ACO–SA has the slightly higher mean, so the corrected test does not support a stable difference in typical performance.

## 7 Conclusion

We studied large-scale reassembly of Delaunay-derived triangles in the unit square, extending the reconstruction formulation of [1] from the earlier 114-triangle limit to instances with up to 1994 triangles. The problem was treated as a geometric–combinatorial optimization task shaped by strict feasibility constraints, combinatorial growth of construction choices, and limited computational budgets.

Within this setting, we showed that minimal and carefully integrated hybridization can be effective. All methods were constrained to the same length-matching attachment rule derived from Strategy S4, ensuring that performance differences arise purely from search control rather than from different geometric operators. Under fixed wall-clock budgets, the proposed lightweight budget-aware hybrid ACO–SA—which replaces one standard ant with one SA-based constructive ant operating under a fixed sub-budget and allows its solution to participate in pheromone learning—achieves the strongest overall coverage profile and a favorable quality–cost trade-off at scale. Run-level means, standard deviations, and corrected nonparametric tests confirm that the larger gains at

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_17

$m = 594$ and $m = 1994$ are statistically significant, whereas the sub-1% gaps remain indistinguishable under 30-run variability. The ablation study indicates that the dominant gain stems from allowing SA to contribute information within pheromone learning: removing SA entirely or excluding SA-generated solutions from pheromone updates eliminates a substantial portion of the improvement.

Future work may explore adaptive and multi-level hybrid variants, alternative hybridization strategies, and broader classes of stochastic and deterministic constructive optimization approaches within geometric reassembly and related packing problems.

## References

1. Alihodžić, A., Hasanspahić, D., Tuba, E., Hasić, D.: Triangle packing strategies within a rectangle. In: Proc. IEEE International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA). Boracay, Philippines (Feb 2026), to appear
2. Alihodžić, A.: Exploring Computational Geometry: Theory and Python Implementations. Springer Nature, Cham, Switzerland (2026)
3. Allen, S.R., Iacono, J.: Packing identical simple polygons is NP-hard. CoRR **abs/1209.5307** (2012), https://arxiv.org/abs/1209.5307
4. de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: Computational Geometry: Algorithms and Applications. Springer, Berlin, Heidelberg, 3 edn. (2008). https://doi.org/10.1007/978-3-540-77974-2
5. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys **35**(3), 268–308 (2003). https://doi.org/10.1145/937503.937505
6. Chen, C., He, D.: A heuristic method for solving triangle packing problem. Journal of Zhejiang University-SCIENCE A **6**(6), 565–570 (2005). https://doi.org/10.1631/jzus.2005.A0565
7. Chou, A.: NP-hard triangle packing problems. Tech. rep., MIT Research Science Institute (Jan 2016), https://math.mit.edu/research/highschool/rsi/documents/2015Chou.pdf
8. Delaunay, B.: Sur la sphère vide. A la m'emoire de Georges Voronoï. Bulletin de l'Acad'emie des Sciences de l'URSS. Classe des sciences math'ematiques et naturelles (6), 793–800 (1934)
9. Dorigo, M., Gambardella, L.M.: Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation **1**(1), 53–66 (1997). https://doi.org/10.1109/4235.585892
10. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **26**(1), 29–41 (1996). https://doi.org/10.1109/3477.484436
11. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press, Cambridge, MA, USA (2004)
12. Goldreich, O.: P, NP, and NP-Completeness: The Basics of Computational Complexity. Cambridge University Press, Cambridge, UK (2012)
13. Guo, B., Zhang, Y., Hu, J., Li, J., Wu, F., Peng, Q., Zhang, Q.: Two-dimensional irregular packing problems: A review. Frontiers in Mechanical Engineering **8** (2022). https://doi.org/10.3389/fmech.2022.966691

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_17

14. Hassin, R., Rubinstein, S.: An approximation algorithm for maximum triangle packing. In: Algorithms – ESA 2004. Lecture Notes in Computer Science, vol. 3221, pp. 403–413. Springer (2004). https://doi.org/10.1007/978-3-540-30140-0_37
15. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983). https://doi.org/10.1126/science.220.4598.671
16. van Laarhoven, P.J.M., Aarts, E.H.L.: Simulated Annealing: Theory and Applications. Springer Dordrecht (1987). https://doi.org/10.1007/978-94-015-7744-1
17. Leao, A.A.S., Toledo, F.M.B., Oliveira, J.F., Carravilla, M.A., Alvarez-Valdés, R.: Irregular packing problems: A review of mathematical models. European Journal of Operational Research **282**(3), 803–822 (2020). https://doi.org/10.1016/j.ejor.2019.04.045
18. Mathieson, L., Prieto, E., Shaw, P.: Packing edge disjoint triangles: A parameterized view. In: Parameterized and Exact Computation. IWPEC 2004. Lecture Notes in Computer Science, vol. 3162, pp. 127–137. Springer (2004). https://doi.org/10.1007/978-3-540-28639-4_12
19. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. The Journal of Chemical Physics **21**(6), 1087–1092 (1953). https://doi.org/10.1063/1.1699114
20. Stodola, P., Michenka, K., Nohel, J., Rybanský, M.: Hybrid algorithm based on ant colony optimization and simulated annealing applied to the dynamic traveling salesman problem. Entropy **22**(8), 884 (2020). https://doi.org/10.3390/e22080884
21. Stützle, T., Hoos, H.H.: Max–min ant system and local search for the traveling salesman problem. In: Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC 1997). pp. 309–314. IEEE (1997). https://doi.org/10.1109/ICEC.1997.592327
22. Stützle, T., Hoos, H.: MAX–MIN ant system. Future Generation Computer Systems **16**(8), 889–914 (2000). https://doi.org/10.1016/S0167-739X(00)00043-1
23. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, USA, 2 edn. (2018)
24. Talbi, E.G.: A taxonomy of hybrid metaheuristics. Journal of Heuristics **8**(5), 541–564 (2002). https://doi.org/10.1023/A:1016540724870
25. Wang, R., Luo, Y., Dong, J., Liu, S., Qi, X.: A heuristic algorithm for solving triangle packing problem. Discrete Dynamics in Nature and Society (2013). https://doi.org/10.1155/2013/686845
26. Watkins, C., Dayan, P.: Q-learning. Machine Learning **8**, 279–292 (1992). https://doi.org/10.1007/BF00992698
27. Zhang, Y., Fan, Y.: Packing and covering a unit equilateral triangle with equilateral triangles. Electronic Journal of Combinatorics **12**(1), R55 (2005). https://doi.org/10.37236/1952

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29909-3_17