Evaluating Assisted Reinforcement Learning Evolutionary Algorithms for Constraint Driven Floor-Plan Generation

LaTeX template adapted from:
European Conference on Artificial Intelligence

Mikolaj Mikuliszyn[1]

Abstract. Translating natural language descriptions into coherent floor-plan layouts requires handling spatial ambiguity, geometric constraints, and incomplete specifications. This paper evaluates a hybrid optimization approach that combines an Evolutionary Algorithm (EA) with reinforcement learning guided seeding, implemented through a Multi Armed Bandit model. The EA refines candidate layouts using constraint driven mutation, crossover, and repair, while the bandit selects seeding strategies that consistently yield stronger initial populations.

A large scale study of 500 paired runs compares the EA only baseline against the EA+RL hybrid under identical conditions. The RL assisted variant achieves lower final fitness in 83.8% of runs, reduces variance across seeds, and improves median performance by 6.96%. Convergence benefits are modest, indicating that RL primarily enhances the starting population rather than altering long term evolutionary behavior. These findings demonstrate the value of learning based initialization for constraint aware layout generation.

1 Introduction

Modern applications increasingly require systems that transform high level natural language instructions into structured spatial layouts. Floor-plan generation exemplifies this challenge: textual descriptions are often ambiguous or incomplete, yet the resulting designs must satisfy geometric, spatial, and functional constraints. Conventional supervised approaches rely on paired text layout datasets and struggle when instructions omit detail or require complex spatial reasoning.

This project investigates an alternative optimization based pipeline for generating constraint aware floor plans from natural language descriptions. The system combines an Evolutionary Algorithm (EA) with reinforcement learning guided seeding, where a Multi Armed Bandit selects among several initialization strategies. The EA then refines candidate layouts through mutation, crossover, and repair steps that enforce constraints such as adjacency, directional placement, compactness, and mask conformity.

The aim is to assess whether RL guided seeding improves the EA's ability to satisfy high level layout constraints and produces more coherent plans than an EA only baseline. This work contributes: (1) a hybrid EA/RL approach for text to layout generation, (2) custom mutation and repair operators tailored to spatial coherence, and (3) a large scale empirical evaluation across diverse layout configurations.

[1] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: mm6659o@gre.ac.uk

The remainder of this report presents the theoretical background (Section 2), describes the implementation and experimental setup (Section 3), analyses the results (Section 4), and concludes with limitations and future directions (Section 5).

2 Background

Evolutionary Algorithms (EAs) are population based optimization methods that iteratively refine a set of candidate solutions through variation and selection. Unlike gradient based approaches, EAs do not require differentiability or smoothness, making them well suited for combinatorial problems defined by interacting geometric and logical constraints. In this project, floor-plan genomes are represented as discrete sets of occupied grid cells for each room, allowing mutation operators to reposition or reshape rooms while respecting masks, directions, and connectivity requirements.

Formally, the EA minimizes a composite fitness function:

$$F(\mathbf{x}) = \sum_{i=1}^{K} w_i C_i(\mathbf{x}) \tag{1}$$

Where each C_i corresponds to a constraint component (e.g., overlap, adjacency, mask conformity, compactness). For example, mask violation is computed as:

$$C_{\text{mask}}(\mathbf{x}) = \sum_{(r,c)\in\mathbf{x}} \mathbb{1}[M_{r,c} = 0] \tag{2}$$

Mutation acts as a stochastic transformation M on candidate layouts:

$$\mathbf{x}' = \mathcal{M}(\mathbf{x}, \theta, \epsilon) \tag{3}$$

where θ denotes configuration parameters (light/heavy mode, target sizes) and $\epsilon \sim \mathcal{U}(0,1)$ controls activation of individual mutation steps. Many of the operators used in this work, such as centroid pull shifts and connectivity repair, can be expressed as bounded grid translations:

$$(r, c) \mapsto (r + \Delta r,\ c + \Delta c), \quad \Delta r, \Delta c \in \{-1, 0, 1\}. \tag{4}$$

Reinforcement Learning provides a complementary mechanism for guiding search. Instead of a full RL formulation, this project uses a Multi Armed Bandit (MAB) model to select which seeding strategy to use at the start of each evolutionary run.

For each seed strategy a, the bandit maintains an empirical reward estimate and selects strategies using an ε-greedy rule.

$$\hat{\mu}_a = \frac{1}{n_a} \sum_{i=1}^{n_a} R_i, \quad R_i = -F(\mathbf{x}_i) \tag{5}$$

Hybrid approaches have been categorized into EA assisted RL, RL assisted EA, and fully synergistic models [2]. The present work follows the RL assisted EA pattern: the EA conducts optimization, while the MAB adaptively selects promising initial populations, improving convergence speed and reducing stagnation across episodes.

3 Experiments and Results

A total of 500 paired experiments were conducted to evaluate the effect of reinforcement learning guided seeding. Each pair contained an EA-only baseline and an EA+RL variant, where a Multi-Armed Bandit (MAB) selected among five seeding strategies. All runs used identical evolutionary parameters (population size 52, mutation rate 0.3, crossover rate 0.8, tournament size 3, elitism 0.05, stagnation threshold 20) to ensure a fair comparison. Experiments were executed on (Intel i9-9900K, 32 GB RAM) without GPU acceleration. The dataset used is the FloorPlans970Dataset [1].

Fitness combined geometric, topological, and realism constraints, with lower scores indicating better solutions. The RL reward was the negative final fitness, enabling the bandit to favor seeding strategies that consistently generated stronger initial populations.

To compare optimization quality, Figure 1 plots EA only best fitness against EA+RL best fitness for all 500 runs. The diagonal marks equal performance.

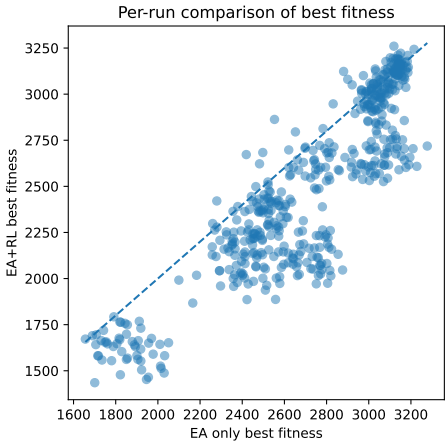

Figure 1. Per run comparison of best fitness for the EA only (x-axis) and EA+RL (y-axis) variants. Points below the diagonal indicate runs where RL guided seeding achieves a lower (better) fitness. EA+RL outperforms EA in 419 out of 500 runs (83.8%).

Distributional differences are shown in Figure 2.

The EA+RL distribution shifts downward, shows tighter inter quartile bounds, and contains fewer high fitness outliers, suggesting both stronger and more reliable optimization.

Figure 3 shows the distribution of per run fitness differences (EA best - RL best).

The improvement distribution is positively skewed. Many runs show moderate gains, while a smaller subset shows very large improvements, typically for challenging floor-plans where a good ini-

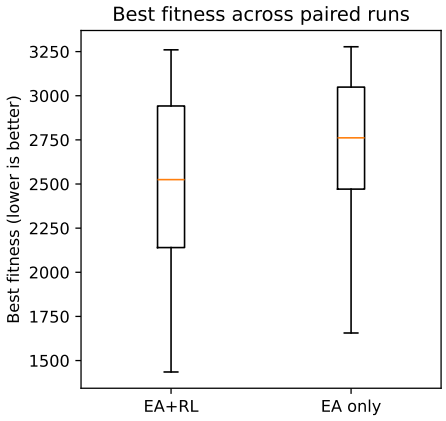

Figure 2. Distribution of best fitness across 500 paired runs. The EA+RL variant achieves a substantially lower median fitness (2524.86 vs. 2761.57) and reduced variance, indicating improved quality and greater optimization stability.

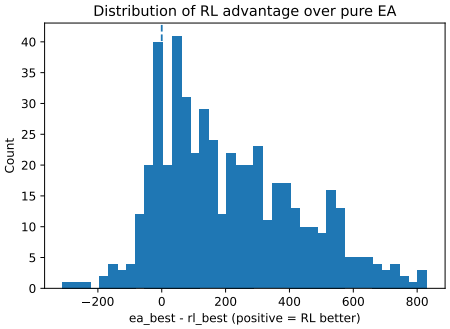

Figure 3. Histogram of fitness differences (EA best - RL best). Positive values indicate an RL advantage. Most improvements fall between 0 and 400, with several outliers exceeding 800. EA only wins are rare and typically small.

tialization substantially accelerates progress. Table 1 summaries the main statistics.

Table 1. Summary of performance across 500 paired runs.

Metric	EA+RL	EA
Runs better (count)	419	81
Runs better (%)	83.8	16.2
Mean best fitness	2489.06	2704.86
Median best fitness	2524.86	2761.57
Mean relative improvement (%)	8.27	-
Median relative improvement (%)	6.96	-
Mean generation at best	67.33	77.08
Mean duration (s)	458.50	309.49

Overall, the results show that RL guided seeding substantially improves optimization quality across a diverse set of floor-plans. Improvements are consistent, statistically large, and supported by both distributional and point wise comparisons.

In terms of convergence behavior, the EA+RL variant tends to reach its best solution slightly earlier on average than the EA only baseline (67.33 vs. 77.08 generations). This difference is modest in magnitude but consistent across trials. Examination of run histories suggests that RL seeding primarily provides a better initial position in the search space rather than fundamentally accelerating later evolutionary dynamics. Both variants display overlapping convergence rates and occasional slow converging outliers, particularly on layouts with challenging mask or compactness constraints.

Taken together, the evidence indicates that RL guided seeding offers a mild but reliable head start in convergence, with its primary benefit being improved final optimization quality rather than major changes to long term evolutionary behavior.

4 Discussion

The results demonstrate that RL guided seeding provides a clear and consistent advantage over the EA only baseline, with improvements observed across most floors and under identical evolutionary settings. The primary benefit arises from the quality of the initial population: bandit selected seeders more frequently generate layouts with coherent spatial structure, reducing the search effort required for the EA to reach competitive solutions. This explains both the strong improvement in final best fitness and the reduced variability observed in EA+RL outcomes.

However, the convergence analysis suggests that these benefits do not translate into a substantial acceleration of the overall evolutionary process. The mean generation at best differs by only around ten generations, and both methods display overlapping convergence rates with similar outlier behavior. Thus, RL guided seeding primarily enhances solution quality rather than fundamentally altering optimization dynamics.

Several limitations influence these results. The fitness function is highly composite, meaning improvements in one constraint may mask degradations in others. Some floor-plans also exhibit extreme mask or compactness penalties, which can dominate the search landscape and produce atypical runs. Finally, the bandit model considers only final fitness as reward; incorporating intermediate feedback or context aware selection could potentially yield further gains.

5 Conclusion and Future Work

This work evaluated a hybrid optimization pipeline that combines an Evolutionary Algorithm with reinforcement learning guided seeding for natural language driven floor-plan generation. Across 500 paired runs, the RL assisted variant consistently achieved lower final fitness and reduced outcome variability, demonstrating that informed initialization can meaningfully improve solution quality even under identical evolutionary settings. Convergence benefits were modest, indicating that RL primarily enhances the starting population rather than altering long term optimization behavior.

Several directions offer potential for improvement. A richer reward signal incorporating intermediate progress rather than only final fitness may enable more adaptive bandit behavior. Context aware seeding strategies conditioned on floor plan properties could further reduce variance on difficult layouts. Finally, exploring multi objective formulations or integrating learned repair operators may

strengthen the system's ability to navigate highly constrained geometric spaces.

REFERENCES

[1] HamzaWajid1. Floorplans970dataset. `https://huggingface.co/datasets/HamzaWajid1/FloorPlans970Dataset`, 2025. Accessed: 24-10-2025.

[2] Pengyi Li, Jianye Hao, Hongyao Tang, Xian Fu, Yan Zheng, and Ke Tang, 'Bridging evolutionary algorithms and reinforcement learning: A comprehensive survey on hybrid algorithms', *IEEE Transactions on Evolutionary Computation*, **29**(5), 1707–1728, (2025).

A Additional Figures

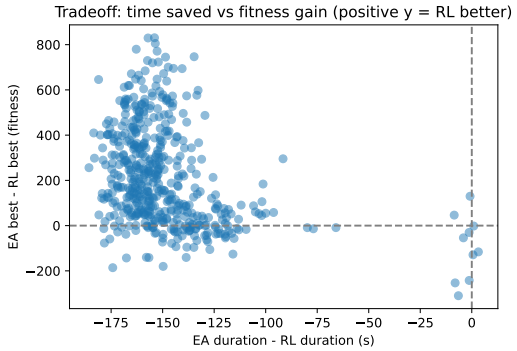

Figure 4. Trade off between runtime difference and fitness improvement. Points above the horizontal line indicate runs where RL achieves lower (better) fitness.

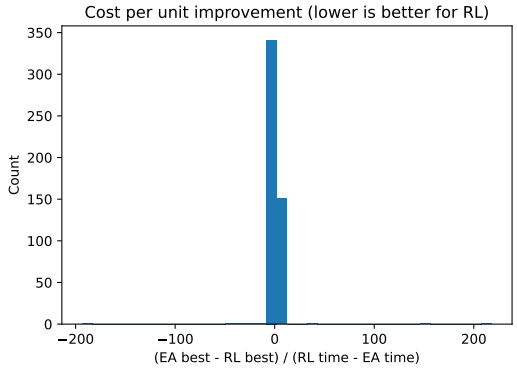

Figure 5. Distribution of cost per unit improvement $(EA_{\text{best}} - RL_{\text{best}})/(RL_{\text{time}} - EA_{\text{time}})$. Lower values indicate more efficient RL gains.