Learning to Evolve: Diffusion-Based Evolutionary Algorithm for Maximum Independent Sets

JOAN SALVÀ SOLER (TU WIEN)

This work introduces the Diffusion-based Evolutionary Algorithm (DEA), a novel framework that integrates denoising diffusion models with evolutionary algorithms (EAs) for combinatorial optimization. Focusing on the Maximum Independent Set (MIS) problem, DEA utilizes diffusion-based initialization and a learned recombination operator to enhance exploration. Experiments on Erdős-Rényi graphs show that DEA outperforms Difusco in solution quality, a state-of-the-art diffusion solver, and surpasses Gurobi on larger instances under the same time constraints. These results highlight the synergy between diffusion models and EAs, demonstrating a promising direction for hybrid machine learning and metaheuristic optimization.

Supervision: Günther Raidl (TU Wien). **Implementation**: We open-source the code in https://github.com/jsalvasoler/difusco_dea.

1 INTRODUCTION

Combinatorial Optimization (CO) problems involve finding the best solution from a finite but exponentially large set of possibilities. Many of these problems are NP-hard, making exact solutions computationally infeasible for large instances. Traditional approaches include exact methods, heuristics, and metaheuristics (e.g., Evolutionary Algorithms (EA)), each with their own strengths and limitations.

Recently, ML approaches have been applied to CO problems, with denoising diffusion models emerging as particularly promising. Difusco (5) represents a significant advancement, using a GNN-based diffusion model to generate high-quality solutions. Difusco offers two key advantages: (1) its multimodal property allows sampling diverse high-quality solutions, and (2) it provides solutions in a single denoising pass, making it computationally efficient (vs. e.g., autoregressive models).

However, diffusion-based approaches like Difusco face important limitations: (1) they often rely on problem-specific heuristics for constraint satisfaction and solution refinement, and (2) they lack the robust exploration capabilities of traditional metaheuristics. These limitations motivate our work, which aims to combine the generative power of diffusion models with the exploratory strengths of EAs.

2 DIFFUSION-BASED EVOLUTIONARY ALGORITHM

We propose a Diffusion-based Evolutionary Algorithm (DEA) framework that integrates pre-trained diffusion models into the core operators of an EA. Figure 1 provides an overview of the DEA architecture, and we now describe the two key components.

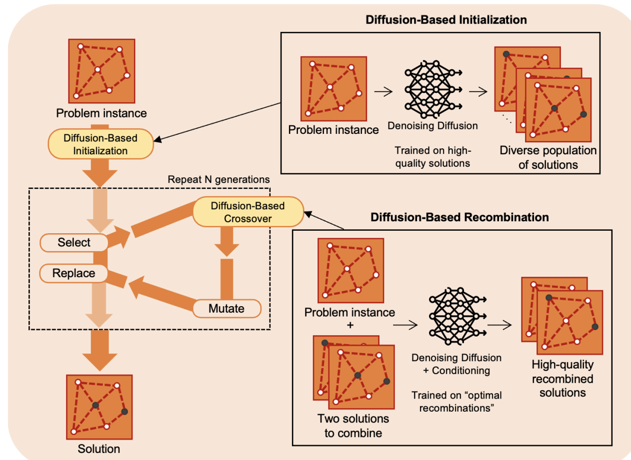

Fig. 1. Overview of the DEA framework.

Diffusion-based initialization. We leverage the multimodal property of a pre-trained Difusco model to parallelly sample a diverse initial population of high-quality solutions. The architecture details are as follows: we use an anisotropic GNN backbone, with an embedding dimension of 256 and 12 layers, for a total of 5.1M parameters. We use categorical diffusion instead of Gaussian diffusion, as in (5). The inference schedule is a cosine schedule with $T_{train} = 1000$ and $T_{inf} = 50$ steps for training and inference, respectively.

Diffusion-based recombination. We develop a novel recombination operator using a pre-trained diffusion model via imitation learning. The model learns from an expert demonstrator, which is formulated as an Integer Linear Program that finds the optimal recombination solving the MIS, with an additional constraint (3) that centers the search space around the parents:

$$\max \quad \sum_{i \in V} z_i \tag{1}$$

$$\text{s.t.} \quad z_u + z_v \leq 1 \qquad \forall (u, v) \in E \tag{2}$$

$$h(z, x) + h(z, y) \leq \lambda \cdot h(x, y) \tag{3}$$

$$z_i \in \{0, 1\} \qquad \forall i \in V \tag{4}$$

Author's address: Joan Salvà Soler (TU Wien), jsalvasoler@hotmail.com.

Where z is the child solution, x and y are the parent solutions, h is the Hamming distance, and λ is a hyperparameter controlling the search space size.

We propose a more general embedding of the input features of the Difusco GNN in order to condition on extra node features (the parent solutions), which allows the model to effectively leverage information from both parents to generate higher-quality offspring. Other architecture details are kept the same as in the initialization model. The resulting model has 5.3M parameters.

3 EXPERIMENTS

We evaluate DEA on the MIS using Erdős-Rényi random graphs of different sizes: 50-100 nodes, 300-400 nodes, and 700-800 nodes. Our experiments focus on: (1) evaluating the diffusion recombination operator, (2) an ablation study on the diffusion components, and (3) a final comparison of DEA against other solvers.

3.1 Evaluation of the Diffusion-Based Recombination

We train the diffusion recombination operator with 160k examples coming from 20k unique graph instances on each dataset. We train with a cosine-decayed learning rate scheduler for 50 epochs.

In Table 1, we evaluate the performance of the diffusion recombination considering different conditioning strategies: EA parents (real high-quality from observed evolutionary trajectories), heuristic parents (medium-quality), and random (infeasible and low-quality). The diffusion recombination shows the best gaps, and performance degrades with the worsening of the conditioning solutions, providing evidence of cost monotonicity.

Table 1. Average gap to the optimal recombination label of different recombination approaches and conditioning strategies.

Method	Parent Type	Gap (%)		
		ER-50-100	ER-300-400	ER-700-800
Diffusion Recomb.	EA Parents	**1.28**	**6.10**	**5.18**
	Heuristic	2.86	12.18	16.81
	Random	3.10	26.28	26.60
Difusco	–	6.36	8.51	5.44
Classic	EA Parents	6.38	20.52	24.13

3.2 Ablation Study

We compare the performance of DEA with and without the diffusion components. When both are removed, we get a naive EA baseline inspired by standard MIS evolutionary solvers (1, 4). Table 2 compares the full DEA gaps to the optimal labels with those from the ablations. Population size P and number of generations G are selected in a way that all resulting algorithms have equivalent runtime to DEA with $P = 16$ and $G = 20$. Results show that both diffusion components are key for DEA's performance, and the diffusion recombination operator is more critical.

Table 2. Ablation study: Impact of diffusion components on DEA.

Method	Gap (%)		
	ER-50-100	ER-300-400	ER-700-800
Full DEA	**0.07**	**0.864**	**2.81**
w/o Diff. Recomb.	0.55	3.57	5.75
w/o Diff. Init.	0.11	1.39	4.52
Naive EA (w/o Both)	0.25	8.58	13.36

3.3 Comparing DEA with Other Solvers

We compare the solution quality and runtime of DEA with other solvers: (a) Difusco with different sampling strategies, (b) T2T (3), (c) Gurobi, and (d) KaMIS (2). Table 3 shows that DEA marginally outperforms Difusco and T2T in solution quality, at the expense of a higher runtime. When given the same runtime budget, DEA achieves better solution quality than Gurobi. Finally, DEA is close to the quality of KaMIS, but with much longer runtimes.

Table 3. Average cost and runtime for different MIS solvers across datasets.

Method	ER-50-100		ER-300-400		ER-700-800	
	Cost	Time (s)	Cost	Time (s)	Cost	Time (s)
Difusco x4	19.86	0.6	34.28	2.5	40.74	11.1
Difusco x16	20.32	0.7	35.53	8.7	42.12	43.6
Difusco x32	20.52	0.86	35.89	17.2	42.51	85.3
DEA (ours)	20.73	12.7	36.70	109.1	43.43	475.1
T2T x4	–	–	–	–	41.37	13.1
KaMIS	20.75	60.0	37.05	60.0	44.86	60.0
Gurobi	20.75	0.1	35.57	109.2	41.36	475.5

4 CONCLUSION

Our work demonstrates the effectiveness of integrating diffusion models with EAs for CO. The DEA framework significantly outperforms standalone Difusco and Gurobi and poses a promising approach to enhance the robustness of generative models in CO tasks.

Limitations include limited scalability due to the high runtime of the diffusion recombination operator. The resource-intensive data generation pipeline for the diffusion recombination could be improved by leveraging a GPU-free synthetic approach. Future work could explore the application of DEA to different problem domains and hybridize DEA with other metaheuristics.

REFERENCES

[1] Thomas Bäck and Sami Khuri. 1994. An evolutionary heuristic for the maximum independent set problem. *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence* (1994), 531–535 vol.2. https://api.semanticscholar.org/CorpusID:16300272

[2] Sebastian Lamm, Peter Sanders, Christian Schulz, Darren Strash, and Renato F. Werneck. 2017. Finding near-optimal independent sets at scale. *J. Heuristics* 23, 4 (2017), 207–229. https://doi.org/10.1007/s10732-017-9337-x

[3] Yang Li, Jinpei Guo, Runzhong Wang, and Junchi Yan. 2023. From Distribution Learning in Training to Gradient Search in Testing for Combinatorial Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[4] Xingzhao Liu, Akio Sakamoto, and T. Shimamoto. 1996. A genetic algorithm for maximum independent set problems. *1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No.96CH35929)* 3 (1996), 1916–1921 vol.3. https://api.semanticscholar.org/CorpusID:117744111

[5] Zhiqing Sun and Yiming Yang. 2023. DIFUSCO: Graph-based Diffusion Solvers for Combinatorial Optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 36. 3706–3731.