
ON SEARCHING FOR MINIMAL INTEGER REPRESENTATION OF UNDIRECTED GRAPHS

A PREPRINT

 **Victor Parque***

Department of Modern Mechanical Engineering
Waseda University
3-4-1 Okubo, Shinjuku, Tokyo, Japan 169-8555
parque@aoni.waseda.jp

 **Tomoyuki Miyashita**

Department of Modern Mechanical Engineering
Waseda University
3-4-1 Okubo, Shinjuku, Tokyo, Japan 169-8555

December 15, 2023

Abstract

Minimal and efficient graph representations are key to store, communicate, and sample the search space of graphs and networks while meeting user-defined criteria. In this paper, we investigate the feasibility of gradient-free optimization heuristics based on Differential Evolution to search for minimal integer representations of undirected graphs. The class of Differential Evolution algorithms are population-based gradient-free optimization heuristics having found a relevant attention in the nonconvex and nonlinear optimization communities. Our computational experiments using eight classes of Differential Evolution schemes and graph instances with varying degrees of sparsity have shown the merit of attaining minimal numbers for graph encoding/representation rendered by exploration-oriented strategies within few function evaluations. Our results have the potential to elucidate new number-based encoding and sample-based algorithms for graph representation, network design and optimization.

Keywords undirected graphs · graph representation · enumerative representation · differential evolution · optimization.

1 Introduction

Graphs allow to compute with dependencies in several fields. Devising succinct graph representations is essential to realize efficient mechanisms for storing, transmission and sampling graphical structures ubiquitously. Although matrices and lists are useful data structures allowing the simple and straightforward representation of graphs, they are unable to meet the information-theoretic tight bounds. In a seminal paper, Turan outlined the concept of succinctness when dealing with graph representations, and proposed the succinct representations of both labeled and unlabeled planar graphs using size up to $12n$ bits (n is the number of nodes in the graph) [Turan(1984)]. And Farzan and Munro proposed the representation of arbitrary graphs with the size being a multiplicative factor $1 + \epsilon$ above the information-theoretic minimum, for small positive ϵ [Farzan and Munro(2008)]. Finding efficient representation of arbitrary graphs implies finding regularities in graphs that can be polynomially encoded/decoded. As such, the community has used regular features in the graph topology such as planarity [Turan(1984), Jacobson(1989), Barbay et al.(2012)Barbay, Castelli Aleardi, He, and Munro, Aleardi et al.(2006)Aleardi, Devillers, and Schaeffer, He et al.(1999)He, Kao, and Lu], triangularity [Barbay et al.(2012)Barbay, Castelli Aleardi, He, and Munro, Aleardi et al.(2006)Aleardi, Devillers, and Schaeffer, He et al.(2000)He, Kao, and Lu], separability [He et al.(2000)He, Kao, and Lu, Belloch and Farzan(2010), Blandford et al.(2003)Blandford, Belloch, and Kash] and symmetry [Katebi et al.(2012)Katebi, Sakallah, and Markov]. When using regularity in graph representation, it is often useful find label encodings

*Corresponding author.

that allow tests for node adjacency [Erdős and Evans(1989), Gallian(2018)]. Generally speaking, when regularity in the graph topology is known, the search space of tailored graph representations is reduced; thus it becomes feasible to investigate tight bounds on the representation mechanism.

The community has also explored the feasibility of compressing real-world networks [Claude and Navarro(2010), Boldi et al.(2011)Boldi, Rosa, Santini, and Vigna, Buehrer and Chellapilla(2008), Brisaboa et al.(2014)Brisaboa, Ladra, and Navarro], in which compression efficiency is often reported in bits per edge. It is also possible to realize the number-based representation of graphs by using combinatorial ideas [Kreher and Stinson(1999), Parque et al.(2014)Parque, Kobayashi, and Higashi, Parque and Miyashita(2017a), Parque and Miyashita(2018a)], in which a graph coding scheme computes an integer number representing the graph, and a decoding scheme generates the graph topology from the integer number. Representing graphs with numbers is advantageous when a-priori knowledge of graph regularity is unknown. Also, the number representing the graph meets the information-theoretic tight bound over all possible graph representations in the same class. On the other hand, when the problem is to find an optimal graph topology, the number based representation enables the one-dimensional (integer) search space and the implicit parallelization of search.

One of the key questions in number-based graph representation is whether exploring the landscape of enumerative encodings would render representations with fewer digits, and thus smallest possible integers. Although approaches for succinct graph encodings have relied on a-priori knowledge of graph invariants such as planarity, separability, symmetry, and sparsity, the study of representing arbitrary graphs with the smallest possible integers has received little attention in the literature. In this paper, we study the feasibility of using gradient-free population-based optimization heuristics inspired by Differential Evolution for smallest/minimal integer representation of graphs, and present an analysis of the convergence performance. In particular, our contributions are as follows: (1) We formulate the problem of searching for minimal integer representation of graphs, portray examples of the search space, and evaluate the performance of eight gradient-free optimization algorithms with varying forms of exploration-exploitation based on Differential Evolution schemes. (2) Our computational experiments using graph instances with varying degrees of sparsity has shown the merit of exploration strategies to attain better convergence with few function evaluations.

The rest of this paper is organized as follows. Section 2 provides the preliminaries on enumerative and minimal graph representation, section 3 describes the computational experiments, and section 4 concludes our paper.

2 Graphs with Minimal Integers

2.1 Enumerative Representation of Graphs

Let $\mathcal{G}_{n,m}$ be the class of undirected graphs with n nodes and m edges. When graphs in the class $\mathcal{G}_{n,m}$ are represented by integer numbers g , the following condition holds: $g \in [0, |\mathcal{G}_{n,m}| - 1]$, in which,

$$|\mathcal{G}_{n,m}| = \binom{n}{m}. \quad (1)$$

The above implies that graphs in the class $\mathcal{G}_{n,m}$ are combinatorial structures that involves the combinations of $\binom{n}{m}$ edges taken m at the time. Let the graph instance $G = (V, E) \in \mathcal{G}_{n,m}$, with $n = |V|$ and $m = |E|$; a numerical representation (or numerical encoding) of G is the tuple $(g, CODE, DECODE)$ as shown by Fig. 1:

$$CODE : G \rightarrow g, \quad (2)$$

$$DECODE(CODE(g)) \rightarrow G. \quad (3)$$

For a defined labeling of nodes with the tuple $(1, 2, 3, \dots, n)$, it is possible to define the *CODE* function as the following algebraic relation [Kreher and Stinson(1999), Parque et al.(2014)Parque, Kobayashi, and Higashi]:

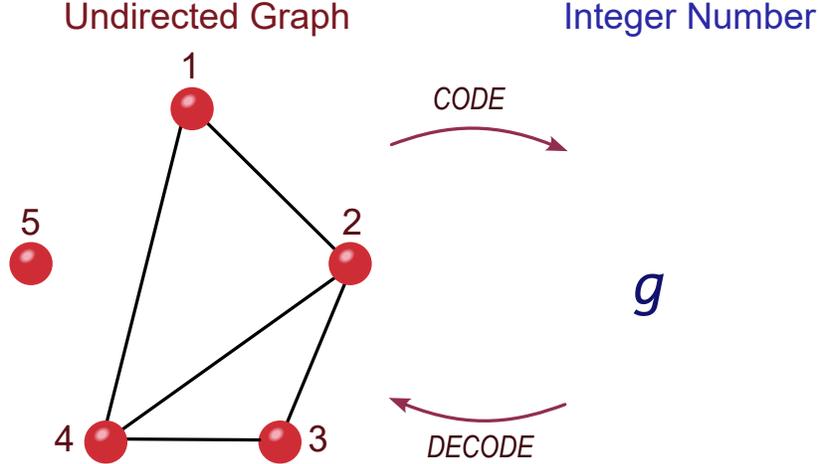


Figure 1: Basic concept of representing graphs as numbers. Given an graph, the *CODE* function generates the number g representing the graph and, viceversa, the *DECODE* function renders the graph from the integer number g .

$$g(x) = \sum_{c=1}^m (-1)^{m-c} \left[\binom{i_c}{c} - 1 \right], \quad (4)$$

where i_c is the numerical label (integer number) of the c -th edge $(u, v) \in E$, computed as follows [Parque et al.(2014)Parque, Kobayashi, and Higashi]:

$$i_c = \binom{u-1}{2} + v, \text{ for } u > v, \quad (5)$$

where $u \in [1, n]$ and $v \in [1, n]$ are labels of nodes from the set V .

The reader may note that the number representation of a graph is contingent upon the values of n and m , and the above-mentioned definitions consider a defined labeling order of the set of nodes V . It is possible to define the *DECODE* function with a decoding algorithm based on the revolving door order [Kreher and Stinson(1999),Parque et al.(2014)Parque, Kobayashi, and Higashi].

2.2 Minimal Integer Representation

Graphs can be represented with minimal integers by solving the following:

$$\begin{aligned} & \underset{x}{\text{Minimize}} && L(x) \\ & \text{subject to} && x \in \mathbf{F}(n) \end{aligned} \quad (6)$$

where the objective function is

$$L(x) = \log(g(x)), \quad (7)$$

where x denotes the representation of the ordering of labels in the graph, and $\mathbf{F}(n)$ is the search space related to the factorial numbering system of n . The problem described above is NP-hard, and without further knowledge of the graph topology or regularities in the structure, computing the gradient of L is unfeasible; thus, the use of a gradient-free and stochastic optimization heuristic is a desirable choice to sample the search space of $F(x)$. The logarithmic function in Eq. (7) facilitates computing the number of digits in number-based representations of graphs, thus it provides an intuitive idea on the space needed to encode undirected graph. Furthermore, the use of the factorial (or *factoradic*) space $\mathbf{F}(n)$ in Eq. (6) is due to the permutation of node labels in terms of $n!$.

To exemplify the landscape of the above-mentioned objective function, we consider the following examples of graphs:

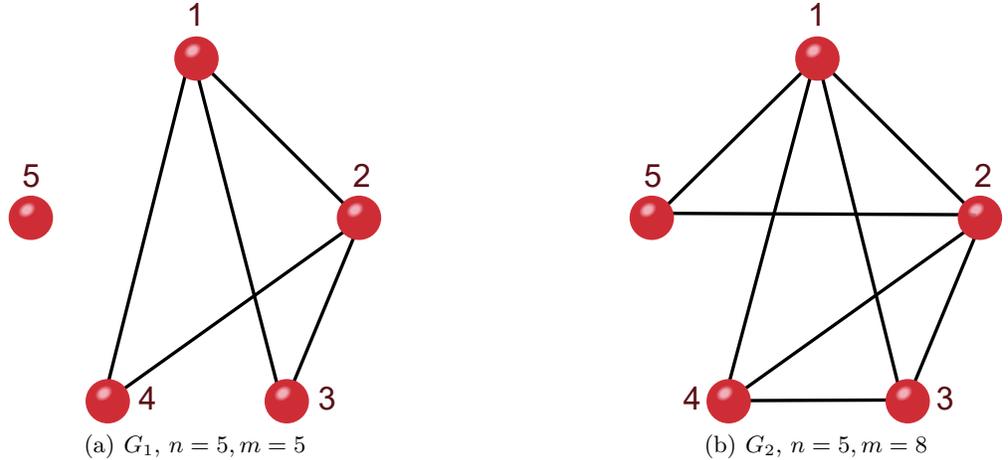


Figure 2: Example of undirected graph topologies.

- G_1 : an undirected graph with $n = 5$ nodes and $m = 5$ edges, and
- G_2 : an undirected graph with $n = 5$ nodes and $m = 8$ edges.

Fig. 2 shows the topologies of graphs G_1 and G_2 . For both graphs, $n = 5$, yet the reader may note that a node with label 5 in G_1 is not connected to any other node. Then, one can compute the objective function L over the factorial space $\mathbf{F}(n)$, for each permutation of the node labels. For $n = 5$, there exists $5! = 120$ possible orders of the labels, thus 120 possible values of the metric L .

In order to show the landscape of the objective function, Fig. 3 shows the rendering of the metric L for both graphs G_1 and G_2 when evaluating all possible 120 permutations of labels. Here, the x-axis shows the factorial number system corresponding to the ordering of the label (for $n = 5$, the x-axis shows numbers in the range 1-120), and the y-axis shows the metric L . By observing Fig. 3, one can note that the landscape is nonsmooth, and it is possible to find regions with smallest values of L . Such regions (labeled with green color) are of interest to find node and edge labels that render smallest/minimal integer representation. Examples of such node labels are provided in Fig. 3-(a) and Fig. 3-(b), for each graph G_1 and G_2 , respectively. Sampling through the factorial number system allows to explore possible configurations of graphs whose representation may lead to the use of fewer number of digits.

In order to show the overall landscape for distinct and arbitrary number of edge configurations, Fig. 4 shows the landscapes of the metric L for arbitrary graphs when considering varying number of edges. Here, we show the the case of undirected graphs with $n = 5$, $m \in [2, 8]$, and the case of undirected graphs with $n = 6$ and number of edges $m \in [2, 14]$. Compared to Fig. 3, the landscape of in Fig. 4 shows the larger number of instances in which it is possible to attain smallest integer representation. This is due to the graph configurations arising from larger number of edges and nodes. Also, sparse undirected graphs with fewer edges compared to number of nodes will allow the larger number of combinatorial permutations with smallest integer representation. Furthermore, although the minimization of the objective function in Eq. (7) implies searching for smallest numbers, it is possible to explore other objective functions to search for graphs with tailored properties (e.g., by using factorization and the fundamental theorem of arithmetic).

3 Computational Experiments

To evaluate the performance of the proposed approach, we conducted computational experiments using diverse graph topologies, configuration of nodes and edges, with varying degrees of sparsity. Fig. 5 shows the set of graph instances. For ease of reference, plots correspond to arbitrary ordering/labeling of nodes (in the range of 10 to 20), considering the modeling of labeled trees as well. The key motivation for using the settings in Fig. 5 is due to our interest in finding compact representations for multiagent communication networks, where the number of agents is expected to be distributed in indoor environments (e.g., robots, drones, sensors distributed in the map). As such, exploring the compact representation of graphs is desirable to communicate and share the topology of the graph among members of a robot swarm. The reader may also

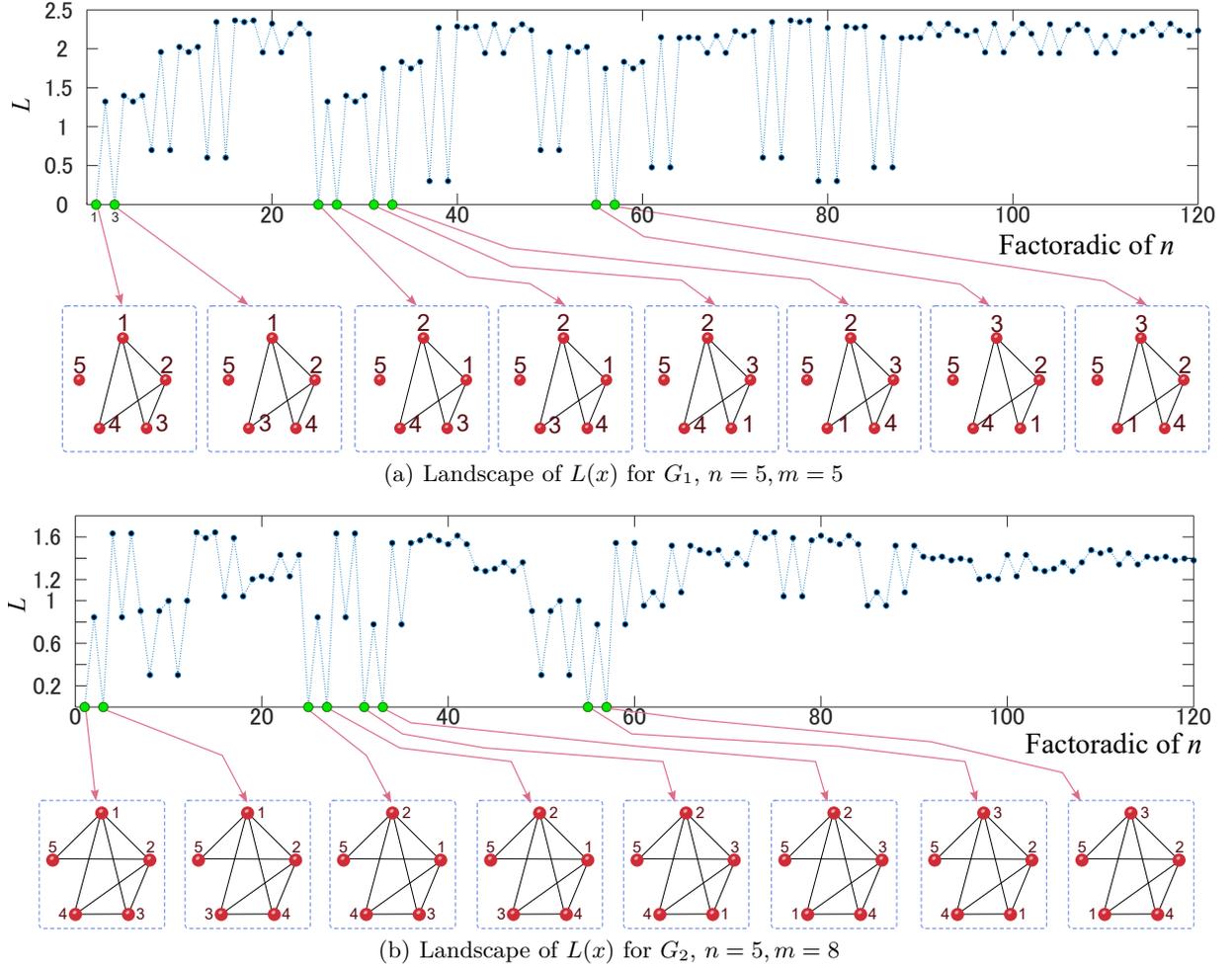


Figure 3: Example of the landscapes of the fitness functions for representing of graphs by smallest numbers. Green marks imply the lowest smallest integers.

note that some graphs in Fig. 5 are not connected to the entire network. This observation (and situation) is especially relevant when agents (nodes) are independent of the group (swarm); thus, communication and information gathering is not essential at all times.

To evaluate the feasibility in attaining quick convergence towards graph representations with smaller integers, we use a representative class of optimization algorithms derived from Differential Evolution, which is a population-based gradient-free optimization algorithm extending the difference of vectors:

1. DESPS: Differential Evolution with Successful Parent Selection [Guo et al.(2015)Guo, Yang, Hsu, and Tsai]
2. OBDE: Differential Evolution with Opposition-based Learning [Rahnamayan et al.(2008)Rahnamayan, Tizhoosh, and Salama]
3. JADE: Adaptive Differential Evolution External Archive [Zhang and Sanderson(2009)]
4. DCMAEA: Differential Covariance Matrix Adaptation [Ghosh et al.(2012)Ghosh, Das, Roy, Minhazul Islam, and Suganthan]
5. DERAND: DE/rand/1/bin Strategy [Storn and Price(1997)],
6. DEBEST: DE/best/1/bin Strategy [Storn and Price(1997)],
7. RBDE: Rank-based Differential Evolution [Sutton et al.(2007)Sutton, Lunacek, and Whitley],
8. DESIM: Differential Evolution with Similarity Based Mutation [Segredo et al.(2018)Segredo, Lalla-Ruiz, and Hart].

The key motivation of using the above set is due to our interest in evaluating distinct forms of selection pressure, parameter adaptation, exploration and exploitation mechanisms during search. As for dimensionality of

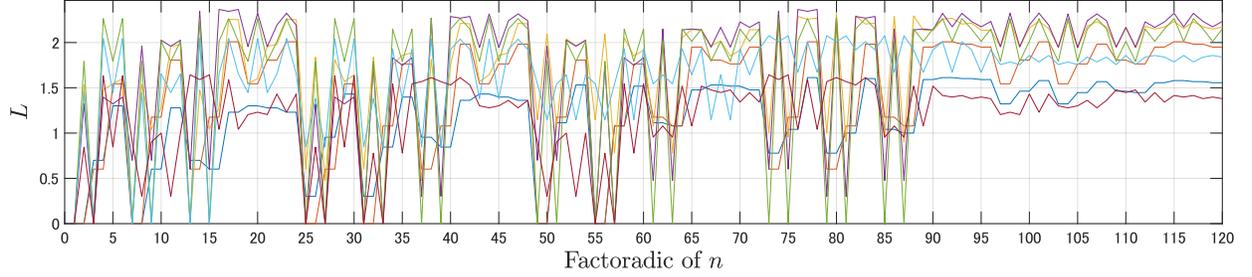
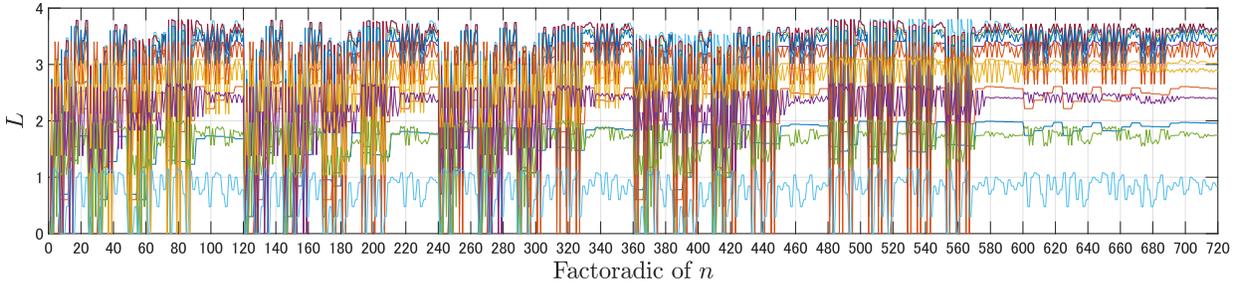
(a) Landscape of $L(x)$ for arbitrary graphs with $n = 5$, $m = 2, \dots, 8$ (b) Landscape of $L(x)$ for arbitrary graphs with $n = 6$, $m = 2, \dots, 14$

Figure 4: Examples of landscapes of the function L for arbitrary graphs. (a) The case of $n = 5, m \in [2, 8]$. (b) The case of $n = 6$ and $m \in [2, 14]$.

the problem, since $x \in \mathbf{F}(n)$, the dimensionality is related to n due to node labelings in the factorial search space. For simplicity and without loss of generality, we used the factoradic representation to encode solutions in the factorial search space. As for algorithm parameters, we used the probability of crossover $CR = 0.5$, scaling factor $F = 0.7$, population sizes $POP = 10$, the bias term in RBDE $\rho = 3$, and the termination criterion is $\Omega = 1000$ function evaluations. In DESIM, the similarity is based on the Euclidean distance, the learning parameters F and CR are adapted using JADE [Zhang and Sanderson(2009)], and initialization uses Opposition Based Learning [Rahnamayan et al.(2008)Rahnamayan, Tizhoosh, and Salama].

Also, due to the stochastic nature of the above mentioned metaheuristics, 10 independent runs were evaluated for each algorithm and each configuration. Other parameters followed the suggested values of the above-mentioned references. The fine-tuning of optimization parameters is out of the scope of this paper. The key motivations for using the above parameters are as follows: Crossover probability with $CR = 0.5$ implements the equal importance and consideration to historical search directions up to the current number of iterations t . Small population size $NP = 10$ and number of evaluations up to 1000 allow evaluating the (frontier) performance of the gradient-free algorithms under tight computational budgets.

Computational experiments were conducted in Matlab 2018b. Since computing the number g that represents a graph involves the summation of binomial coefficients, we rather use the logarithm of g to efficiently compute the reduction of large binomial coefficients. In this paper, the computation of logarithmic function of the number g is realized through the parallel reductions in a GPU (NVIDIA GeForce GTX TITAN X). The reason of using a GPU is due to the efficiency in the reduction scheme; however, since we are constrained by the GPU hardware, we can only explore the possibility of computing the logarithms within the bound of `realmax` in Matlab (1.8×10^{308}). As such, exploring the compact number-based representations for large graphs is out of the scope of this paper (due to hardware limitations) and is left for future work in our agenda (potentially through a parallel distributed computing environment).

3.1 Results and Discussion

Fig. 6 shows the mean convergence performance through distinct graph instances and independent runs. Also, the x-axis of each plot in Fig. 6 shows the number of function evaluations, whereas the y-axis shows the value of the objective function L . By observing Fig. 6, we note the following facts: (1) convergence and stagnation in flat regions often occur after 200 function evaluations with graphs with ten nodes and 20 edges, whereas convergence occurs after 400 functions in other graph instances; (2) the minimal objective values corresponding to each graph instance differ from one to another. It is possible to observe the pattern that

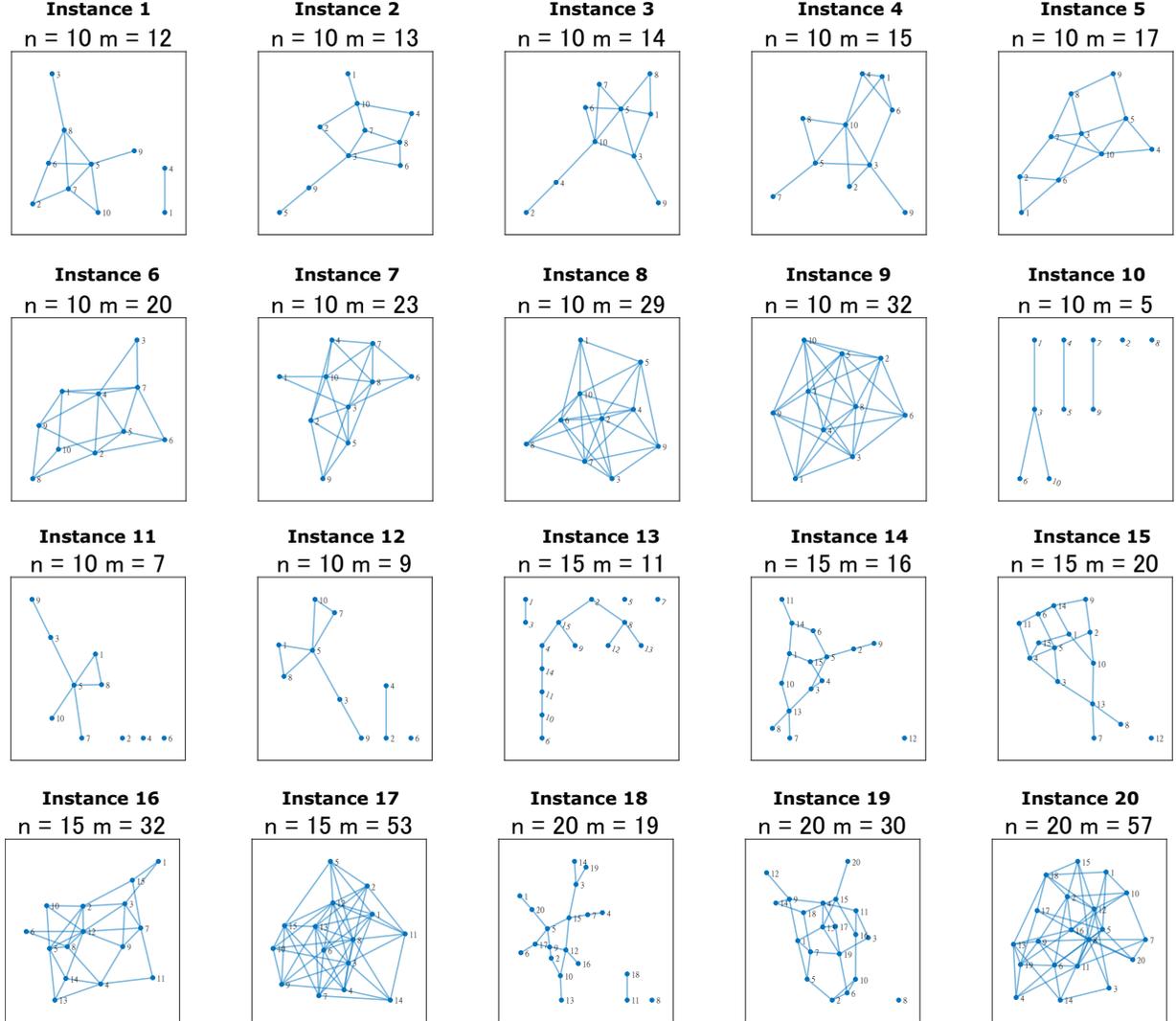


Figure 5: Graph instances used in this study.

a large (small) number of edges implies higher (lower) objective values. This is due to the fact that graphs encode dependencies; as such, more information will be necessary to encode graphs with a larger number of edges, and the integer number representing the graph will be larger due to Eq. 4 depending on m on the summation component.

Among all studied graph instances, the objective value of graph instance 10-11 is lower compared to the objective values of other graph instances. Yet, for all cases, it is possible to attain lower objective values for all cases. Overall cases, the algorithms relying on exploration, such as DERAND, and the algorithm relying on heavy exploration during the early stages of the search, such as DESIM show better convergence performance. This observation is due to the noisy landscape of the problem, such as those depicted in Fig. 3 and Fig. 4. Despite the large factorial search space, the convergence figures show that it is possible to find potential areas of low objective values using few function evaluations.

The observations above show it is possible to find convergence regions quickly and that exploration strategies are relevant for better performance. To show the comparison of the performance of the algorithms across graph instances, we performed statistical test at 5% significance level (Wilcoxon). Here, our goal is to compute instances in which an algorithm is significantly better(+), equal(=), or worse(-) compared to another algorithm. As such, Fig. 7 shows the overall performance comparison among algorithms based on the statistical significance. Here, the bars in the plot show the number of times an algorithm is significantly better (outperformance), equal (equal performance), or worse (underperformance) compared to other algorithms across all graph instances overall independent runs. For clarity of exposition, the counts (numbers depicted

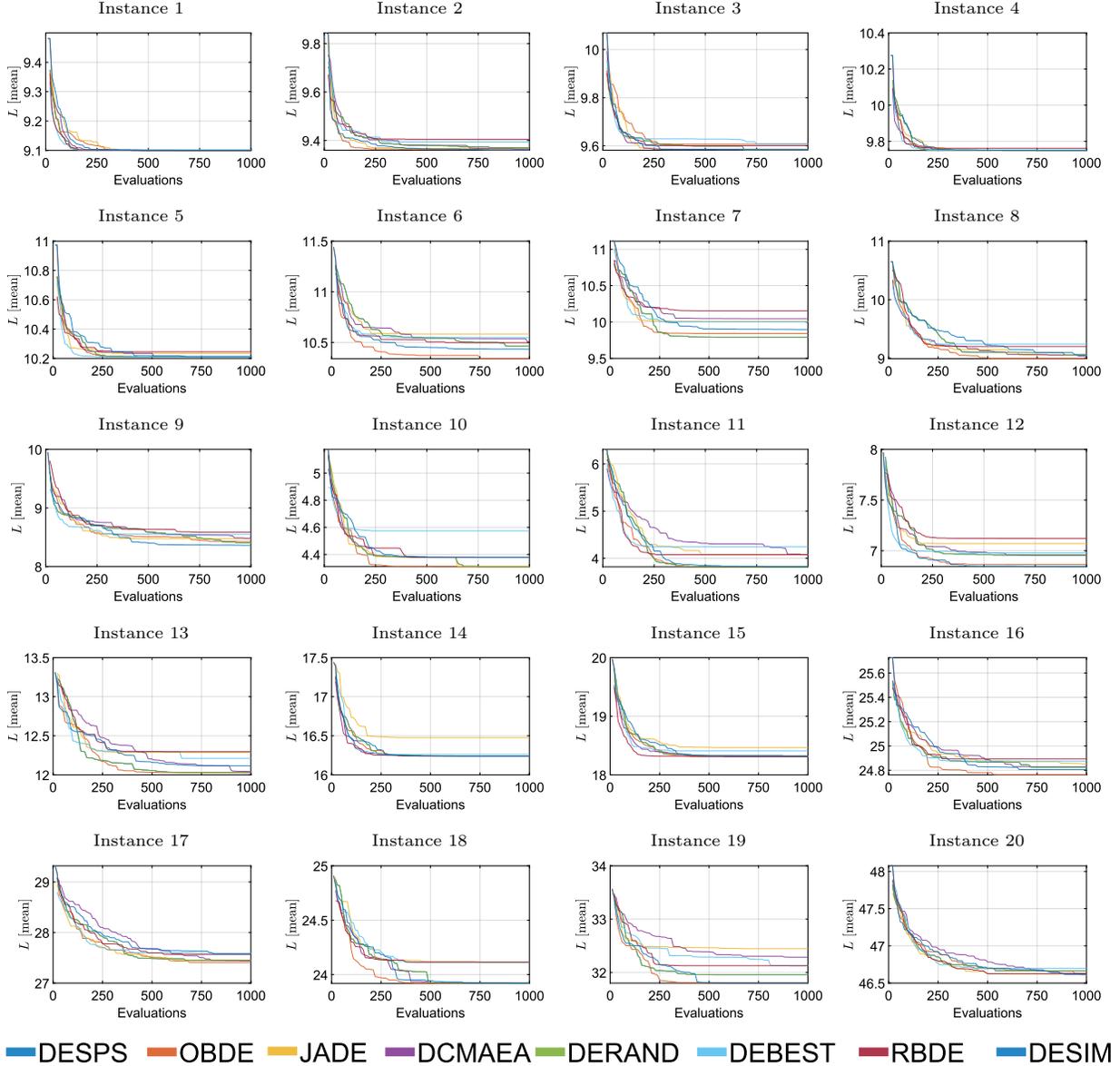


Figure 6: Mean convergence of the objective function L across independent runs.

over each bar) are computed by accumulating the number of events in which $[+]$, $[=]$, and $[-]$ occur from the Wilcoxon test. By observing Fig. 7, we note that algorithms with an explicit exploration strategy outperform other algorithms. For instance, DESPS, DERAND, OBDE and DESIM outperform other algorithms in 12-14 problem instances. Also, algorithms with exploitation strategies such as DEBEST and RBDE perform equally in most problem scenarios, yet underperform in 13-26 instances.

The above results show that using gradient-free heuristics can enable the search for compact integer representations of graphs using few evaluations, which may find use in modeling and sampling graphs and networks with utmost efficiency. The future work will aim at studying the tailored optimization algorithms extending adaptation and explorative strategies, and the modular schemes for large graphs [Parque and Miyashita(2018b)]. It also remains to be answered whether different labeling strategies than the revolving door [Parque et al.(2014)Parque, Kobayashi, and Higashi] are able to achieve smaller values of the objective function (smaller numbers representing graphs). Furthermore, it remains to be answered how to achieve minimal number representation for weighted graphs, directed graphs [Parque and Miyashita(2017a)], loopy graphs [Parque and Miyashita(2017b)], and other relevant graph architectures. On the application side, we aim at using the insights from this study to extrapolate towards large-scale minimal networks in the

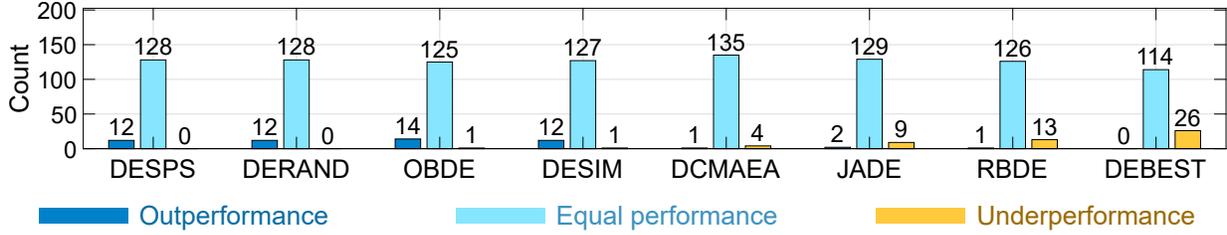


Figure 7: Number of cases of outperformance, equal performance and underperformance derived from pairwise Wilcoxon tests at 5% significance level.

plane [Parque(2021),Parque and Miyashita(2018c)], and the further works in network design and optimization.

4 Conclusion

We have studied the feasibility of gradient-free population-based optimization heuristics based on Differential Evolution variants for minimal integer representations of graphs. Our computational experiments using graph instances with varying degrees of sparsity and eight optimization algorithms with distinct forms of adaptation and exploration-exploitation have shown the merit of exploration-oriented strategies to attain better convergence with fewer function evaluations. Investigating the role of adaptation, exploration and modular strategies in large graphs, and evaluating other forms of graph encoding strategies is left for future work. Our results have the potential to elucidate new number-based approaches for graph representation, network design and optimization.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 20K11998.

References

- [Turán(1984)] György Turán. On the succinct representation of graphs. *Discrete Applied Mathematics*, 8(3):289–294, 1984.
- [Farzan and Munro(2008)] Arash Farzan and J Ian Munro. Succinct representations of arbitrary graphs. In *Esa*, volume 5193, pages 393–404. Springer, 2008.
- [Jacobson(1989)] Guy Jacobson. Space-efficient static trees and graphs. In *30th annual symposium on foundations of computer science*, pages 549–554. IEEE Computer Society, 1989.
- [Barbay et al.(2012)Barbay, Castelli Aleardi, He, and Munro] Jérémy Barbay, Luca Castelli Aleardi, Meng He, and J Ian Munro. Succinct representation of labeled graphs. *Algorithmica*, 62:224–257, 2012.
- [Aleardi et al.(2006)Aleardi, Devillers, and Schaeffer] Luca Castelli Aleardi, Olivier Devillers, and Gilles Schaeffer. Optimal succinct representations of planar maps. In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 309–318, 2006.
- [He et al.(1999)He, Kao, and Lu] Xin He, Ming-Yang Kao, and Hsueh-I Lu. Linear-time succinct encodings of planar graphs via canonical orderings. *SIAM Journal on Discrete Mathematics*, 12(3):317–325, 1999.
- [He et al.(2000)He, Kao, and Lu] Xin He, Ming-Yang Kao, and Hsueh-I Lu. A fast general methodology for information-theoretically optimal encodings of graphs. *SIAM Journal on Computing*, 30(3):838–846, 2000.
- [Blelloch and Farzan(2010)] Guy E Blelloch and Arash Farzan. Succinct representations of separable graphs. In *Combinatorial Pattern Matching: 21st Annual Symposium, CPM 2010, New York, NY, USA, June 21-23, 2010. Proceedings 21*, pages 138–150. Springer, 2010.
- [Blandford et al.(2003)Blandford, Blelloch, and Kash] Daniel K. Blandford, Guy E. Blelloch, and Ian A. Kash. Compact representations of separable graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA ’03*, page 679–688, USA, 2003. Society for Industrial and Applied Mathematics. ISBN 0898715385.

- [Katebi et al.(2012)Katebi, Sakallah, and Markov] Hadi Katebi, Karem A. Sakallah, and Igor L. Markov. Graph symmetry detection and canonical labeling: Differences and synergies. In Andrei Voronkov, editor, *Turing-100 - The Alan Turing Centenary, Manchester, UK, June 22-25, 2012*, volume 10 of *EPiC Series in Computing*, pages 181–195. EasyChair, 2012. doi: 10.29007/gzcl.
- [Erdős and Evans(1989)] Paul Erdős and Anthony B Evans. Representations of graphs and orthogonal latin square graphs. *Journal of Graph Theory*, 13(5):593–595, 1989.
- [Gallian(2018)] Joseph A Gallian. A dynamic survey of graph labeling. *Electronic Journal of combinatorics*, 1(DynamicSurveys):DS6, 2018.
- [Claude and Navarro(2010)] Francisco Claude and Gonzalo Navarro. Fast and compact web graph representations. *ACM Transactions on the Web (TWEB)*, 4(4):1–31, 2010.
- [Boldi et al.(2011)Boldi, Rosa, Santini, and Vigna] Paolo Boldi, Marco Rosa, Massimo Santini, and Sebastiano Vigna. Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks. In *Proceedings of the 20th international conference on World Wide Web*, pages 587–596, 2011.
- [Buehrer and Chellapilla(2008)] Gregory Buehrer and Kumar Chellapilla. A scalable pattern mining approach to web graph compression with communities. In *Proceedings of the 2008 international conference on web search and data mining*, pages 95–106, 2008.
- [Brisaboa et al.(2014)Brisaboa, Ladra, and Navarro] Nieves R Brisaboa, Susana Ladra, and Gonzalo Navarro. Compact representation of web graphs with extended functionality. *Information Systems*, 39:152–174, 2014.
- [Kreher and Stinson(1999)] Donald L Kreher and Douglas R Stinson. Combinatorial algorithms: generation, enumeration, and search. *ACM SIGACT News*, 30(1):33–35, 1999.
- [Parque et al.(2014)Parque, Kobayashi, and Higashi] Victor Parque, Masakazu Kobayashi, and Masatake Higashi. Bijections for the numeric representation of labeled graphs. In *2014 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2014, San Diego, CA, USA, October 5-8, 2014*, pages 447–452. IEEE, 2014.
- [Parque and Miyashita(2017a)] Victor Parque and Tomoyuki Miyashita. On succinct representation of directed graphs. In *2017 IEEE International Conference on Big Data and Smart Computing, BigComp 2017, Jeju Island, South Korea, February 13-16, 2017*, pages 199–205. IEEE, 2017a.
- [Parque and Miyashita(2018a)] Victor Parque and Tomoyuki Miyashita. On graph representation with smallest numerical encoding. In *2018 IEEE 42nd Annual Computer Software and Applications Conference, COMPSAC 2018, Tokyo, Japan, 23-27 July 2018, Volume 1*, pages 817–818. IEEE Computer Society, 2018a. doi: 10.1109/COMPSAC.2018.00135.
- [Guo et al.(2015)Guo, Yang, Hsu, and Tsai] Shu-Mei Guo, Chin-Chang Yang, Pang-Han Hsu, and Jason S.-H. Tsai. Improving differential evolution with a successful-parent-selecting framework. *IEEE Transactions on Evolutionary Computation*, 19(5):717–730, 2015. doi: 10.1109/TEVC.2014.2375933.
- [Rahnamayan et al.(2008)Rahnamayan, Tizhoosh, and Salama] Shahryar Rahnamayan, Hamid R Tizhoosh, and Magdy MA Salama. Opposition-based differential evolution. *IEEE Transactions on Evolutionary computation*, 12(1):64–79, 2008.
- [Zhang and Sanderson(2009)] Jingqiao Zhang and Arthur C Sanderson. Jade: adaptive differential evolution with optional external archive. *IEEE Transactions on evolutionary computation*, 13(5):945–958, 2009.
- [Ghosh et al.(2012)Ghosh, Das, Roy, Minhazul Islam, and Suganthan] Saurav Ghosh, Swagatam Das, Subhrajit Roy, S.K. Minhazul Islam, and P.N. Suganthan. A differential covariance matrix adaptation evolutionary algorithm for real parameter optimization. *Information Sciences*, 182(1):199–219, 2012. ISSN 0020-0255. Nature-Inspired Collective Intelligence in Theory and Practice.
- [Storn and Price(1997)] Rainer Storn and Kenneth Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341, 1997.
- [Sutton et al.(2007)Sutton, Lunacek, and Whitley] Andrew M Sutton, Monte Lunacek, and L Darrell Whitley. Differential evolution and non-separability: using selective pressure to focus search. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1428–1435, 2007.
- [Segredo et al.(2018)Segredo, Lalla-Ruiz, and Hart] Eduardo Segredo, Eduardo Lalla-Ruiz, and Emma Hart. A novel similarity-based mutant generation strategy for differential evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 881–888, 2018.

- [Parque and Miyashita(2018b)] Victor Parque and Tomoyuki Miyashita. Numerical representation of modular graphs. In *2018 IEEE 42nd Annual Computer Software and Applications Conference, COMPSAC 2018, Tokyo, Japan, 23-27 July 2018, Volume 1*, pages 819–820. IEEE Computer Society, 2018b. doi: 10.1109/COMPSAC.2018.00136.
- [Parque and Miyashita(2017b)] Victor Parque and Tomoyuki Miyashita. On the numerical representation of labeled graphs with self-loops. In *29th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2017, Boston, MA, USA, November 6-8, 2017*, pages 342–349. IEEE Computer Society, 2017b. doi: 10.1109/ICTAI.2017.00061.
- [Parque(2021)] Victor Parque. On hybrid heuristics for steiner trees on the plane with obstacles. In *Evolutionary Computation in Combinatorial Optimization - 21st European Conference, EvoCOP 2021, Held as Part of EvoStar 2021, Virtual Event, April 7-9, 2021, Proceedings*, volume 12692 of *Lecture Notes in Computer Science*, pages 120–135. Springer, 2021. doi: 10.1007/978-3-030-72904-2_8.
- [Parque and Miyashita(2018c)] Victor Parque and Tomoyuki Miyashita. Obstacle-avoiding euclidean steiner trees by n-star bundles. In *IEEE 30th International Conference on Tools with Artificial Intelligence, ICTAI 2018, 5-7 November 2018, Volos, Greece*, pages 315–319. IEEE, 2018c. doi: 10.1109/ICTAI.2018.00057.