Bull. Aust. Math. Soc. **113** (2026), 1–9
doi:10.1017/S0004972725100178

THREE ALGORITHMIC APPROACHES TO THE GENERAL POSITION PROBLEM

ZAHRA HAMED-LABBAFIAN, NARJES SABEGHI✉, MOSTAFA TAVAKOLI and SANDI KLAVŽAR

(Received 1 April 2025; accepted 6 May 2025; first published online 4 July 2025)

Abstract

If G is a graph, then $X \subseteq V(G)$ is a general position set if for every two vertices $v, u \in X$ and every shortest (u, v)-path P, no inner vertex of P lies in X. We propose three algorithms to compute a largest general position set in G: an integer linear programming algorithm, a genetic algorithm and a simulated annealing algorithm. These approaches are supported by examples from different areas of graph theory.

2020 *Mathematics subject classification*: primary 05C12; secondary 05C69, 05C76, 05C85.

Keywords and phrases: general position set, general position number, integer linear programming, genetic algorithm, simulated annealing algorithm.

1. Introduction

The general position problem in graph theory was independently introduced (at least) three times, which clearly demonstrates that the concept is of wide interest. The first time, it was investigated in the context of hypercubes [7] and later extended to the general case [9, 18]. The latter article has led to extensive research on the problem, and a recent review article [19] lists 126 references. The study of general position sets has extended to various types, such as Steiner position sets [6], edge general position sets [10], mutual-visibility sets [2], mobile position sets [5], monophonic position sets [13], vertex position sets [12] and lower general position sets [3]. Among the recent achievements in the field, we would like to highlight [1, 4, 14–17].

We now define the problem. If $G = (V(G), E(G))$ is a simple, connected graph, then the set $X \subseteq V(G)$ is a *general position set* for the graph G if for every two vertices $v, u \in X$ and every shortest (u, v)-path P, no inner vertex of P lies in X. We are interested in the size of a largest general position set of G which is called the *general position number* of G and denoted by $\mathrm{gp}(G)$.

Sandi Klavžar was supported by the Slovenian Research and Innovation Agency (ARIS) under the grants P1-0297, N1-0355 and N1-0285.

© The Author(s), 2025. Published by Cambridge University Press on behalf of Australian Mathematical Publishing Association Inc.

https://doi.org/10.1017/S0004972725100178 Published online by Cambridge University Press

The *General Position Set* problem for a given graph G and a positive integer $k \leq |V(G)|$ asks whether there exists a general position set S for G such that $|S| \geq k$. It was already shown in the seminal paper [9] that the *General Position Set* problem is NP-complete for arbitrary graphs. The closely related *Lower General Position* problem was proved to be NP-complete in [3], and the *Monophonic Position Set* problem was demonstrated to be NP-hard in [13]. Interestingly, it is not clear whether the *Monophonic Position Set* problem is NP-complete for general graphs.

Since determining the general position number of a graph is difficult in general, it is reasonable to approach it in different algorithmic ways. Korže and Vesel [8] described a reduction from the problem of finding a general position set in a graph to the satisfiability problem and applied the approach to the case of general position sets of hypercubes. However, we are not aware of any systematic algorithmic approach to the *General Position Problem*. Therefore, in this note, we propose three different algorithms for approaching the problem. These approaches are integer linear programming (ILP), genetic algorithm (GA) and simulated annealing (SA), and are described in the next section. In Section 3, these approaches are tested on several large graphs from different areas of graph theory.

2. Three approaches

In this section, we present in respective subsections the three approaches to compute the general position number as announced in the introduction.

2.1. Integer linear programming model. Let G be a graph with vertex set $V(G) = \{v_1, \ldots, v_{n(G)}\}$, where $n(G)$ denotes the order of G. To find a general position set of G with the maximum cardinality, for a given set $X \subseteq V(G)$, define binary variables x_j, $j \in [n(G)]$, which constitute a characteristic vector of X, that is,

$$x_j = \begin{cases} 1 & \text{if } v_j \in X, \\ 0 & \text{otherwise.} \end{cases}$$

Then, the objective function is to maximise $\sum_{j=1}^{n} x_j$. For each two vertices v_i and v_j, let $I_G(v_i, v_j)$ be the set of vertices different from v_i and v_j which lie on at least one shortest (v_i, v_j)-path. Therefore, if $v_i, v_j \in X$, that is, if $x_i + x_j = 2$, then we must have $\sum_{x_\ell \in I_G(v_i,v_j)} x_\ell = 0$, which can be considered as the inequality:

$$\sum_{x_\ell \in I(v_i,v_j)} x_\ell \leq 0.$$

Using integer modelling techniques, we model this constraint as

$$\sum_{v_\ell \in I(v_i,v_j)} x_\ell + M(x_i + x_j - 1) \leq M, \quad i, j \in [n(G)],$$

https://doi.org/10.1017/S0004972725100178 Published online by Cambridge University Press

where M is an upper bound for the constraint $\sum_{v_\ell \in I(v_i,v_j)} x_\ell \leqslant 0$. In this case, we can select $M = n(G)$. The ILP for our problem is thus:

$$\max \sum_{j=1}^{n} x_j$$

such that

$$\sum_{v_\ell \in I(v_i,v_j)} x_\ell + n(G)(x_i + x_j) \leq 2n(G), \quad i,j \in [n(G)],\ x_i \in \{0,1\},\ i \in [n(G)].$$

2.2. Genetic algorithm. In this subsection, we present a Genetic Algorithm (GA for short) to find the largest general position sets. GA is a population-based method and is well suited for the general position problem because it efficiently explores large search spaces while maintaining feasible solutions through its crossover and mutation operators.

Before implementing the GA, it is necessary to define the stopping criterion, the fitness function and the genetic operators specific to this problem. Each instance of a solution S for a graph G is represented as a binary (characteristic) vector of size $n(G)$, that is, if the vertex i is included in the solution, then $S(i) = 1$, otherwise, $S(i) = 0$. The stopping criterion is defined as the maximum number of iterations. The objective of the algorithm is to obtain a solution that satisfies the general position constraints while maximising the number of selected vertices. The fitness function for a solution S is defined as

$$\text{Fitness}(S) = \sum S - M \cdot f,$$

where $\sum S$ represents the total number of selected vertices and f denotes the number of violations of the general position condition. Specifically, for every pair of vertices u and v with $S(u) = 1$ and $S(v) = 1$, if there exists a vertex w on a shortest (u, v)-path with $S(w) = 1$, then f is increased. The penalty term $M \cdot f$ ensures that solutions violating the general position condition are penalised proportionally to the number of violations, where M is a large constant balancing the trade-off between maximising the number of selected vertices and maintaining feasibility. The initial population is generated randomly, ensuring diversity in the search space.

For parent selection, we use a random selection method. The offspring generation process involves combining two parent solutions: the new solution instance is a binary vector, where a vertex is included if it appears in at least one of the parents.

The mutation operator selects two vertices randomly and swaps their values.

The GA presented in Algorithm 1 outlines the general structure of the approach for solving the general position problem, in terms of the various operators which have been explained above.

2.3. Simulated annealing. Simulated Annealing (SA for short) is a single-solution point-based metaheuristic algorithm inspired by the annealing process in metallurgy. Before implementing SA for the general position problem, it is essential to define the

https://doi.org/10.1017/S0004972725100178 Published online by Cambridge University Press

Algorithm 1 GA for the general position problem.

Require: Graph G, maximum number of iterations *Maxit*, and initial population of size n_p.

Ensure: a general position set and gp(G).

1: Create feasible initial populations with size n_p and evaluate them using the fitness function.
2: $count = 1$
3: **while** $count \leq Maxit$ **do**
4: Select parents randomly.
5: Generate and evaluate offspring using the crossover operator.
6: Generate and evaluate mutated populations using the mutation operator.
7: Merge the initial population, offspring and mutated populations, and sort them based on the fitness function (new population).
8: Truncate the new population and generate a new population with size n_p.
9: $count = count + 1$
10: **end while**
11: Return the individual with the best fitness value as the best solution.

stopping criterion, the neighbourhood generation mechanism and the fitness function. The algorithm begins with an initial solution S_0, which is a vector of zeros and ones representing the inclusion or exclusion of vertices in the solution set. The quality of a solution is measured by the fitness function, Fitness(S), which is defined in the same way as for the GA.

The acceptance probability of a new solution is determined by the Metropolis criterion, which depends on the current temperature and the change in the objective value. The temperature gradually decreases according to a predefined cooling schedule. In our implementation of the Simulated Annealing algorithm, we employed a geometric cooling schedule, where the temperature is updated as

$$T_{k+1} \leftarrow T_k * \rho,$$

where ρ is the cooling rate and we start with an initial temperature, T_0. The algorithm terminates once the maximum number of iterations or a minimum temperature is reached.

The SA is formally presented in Algorithm 2.

3. Experimental results

We have performed our experiments on sets of examples. The first examples come from mathematical chemistry, the second examples deal with hypercubes, while the last examples come from algebraic graph theory. The implementation code for the algorithms is available upon request from the authors.

https://doi.org/10.1017/S0004972725100178 Published online by Cambridge University Press

Algorithm 2 Simulated Annealing (SA) algorithm for the general position problem.

Require: Graph G, maximum number of iterations $Maxit$, initial Temperature T_0 and the rate of temperature reduction ρ.

Ensure: A general position set of G and $\mathrm{gp}(G)$.

Create a feasible initial solution S.
$best\ solution \leftarrow S$
$best\ fitness \leftarrow S_{Fit}$ ($S_{Fit} = \mathrm{Fitness}(S)$)
$count \leftarrow 1$
$T \leftarrow T_0$
while $count \leq Maxit$ **do**
  Generate and evaluate neighbours of S.
  Let $best\ neighbour$ be the neighbour with the best fitness value, $(best\ neighbour)_{Fit}$.
  $S \leftarrow best\ neighbour$, $S_{Fit} \leftarrow (best\ neighbour)_{Fit}$
  if $S_{Fit} > best\ fitness$ **then**
    $best\ solution \leftarrow S$
    $best\ fitness \leftarrow S_{Fit}(\mathrm{Fitness}(S))$
  else
    Generate a random number r.
    if $\exp(-S_{Fit})/T > r$ **then**
      $best\ solution \leftarrow S$
      $best\ fitness \leftarrow S_{Fit}(\mathrm{Fitness}(S))$
    end if
  end if
  if $k \geq cooling\ time$ **then**
    $T \leftarrow T * \rho$
    $k \leftarrow 0$
  end if
  $k \leftarrow k + 1$
  $count \leftarrow count + 1$
end while
Return the $best\ solution$ with the best fitness value as the best solution.

From the area of chemical graph theory, we have considered the so-called middle-fullerene graphs C_{42}, C_{44}, C_{46} and C_{48} from [11] (see Figures 1 and 2).

The maximum general position sets of C_{42} and C_{44}, as indicated in Figure 1 with solid dots, were obtained by solving the corresponding ILP model. In the case of the middle-fullerene graphs C_{46} and C_{48}, their examination was carried out using all three algorithms. The results are reported in the top two rows of Table 1 and the concrete largest general position sets obtained can be seen in Figure 2.

Let Q_n denote the n-dimensional hypercube. It is known ever since [7] that it is intrinsically difficult to determine $\mathrm{gp}(Q_n)$ in general. The small values

https://doi.org/10.1017/S0004972725100178 Published online by Cambridge University Press

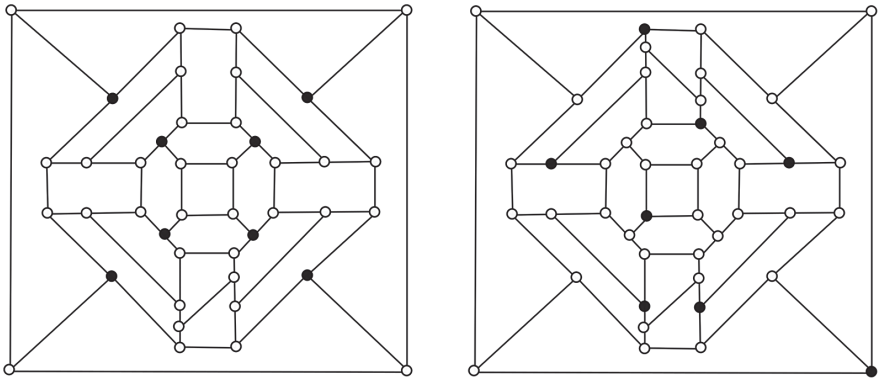


FIGURE 1. Maximum general position sets for C_{42} and C_{44}.

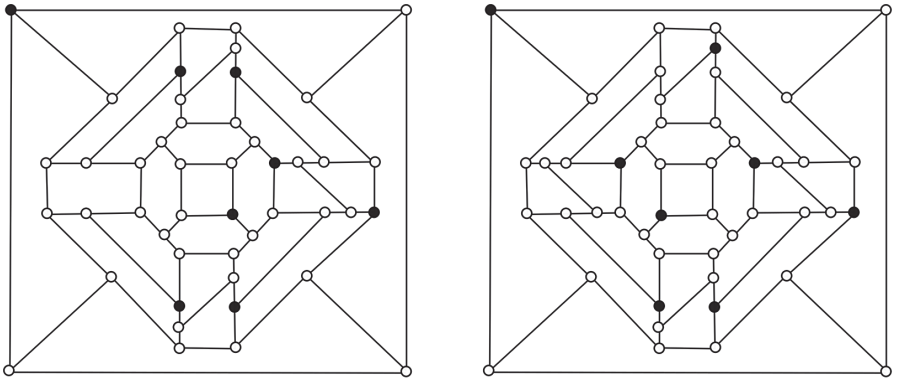


FIGURE 2. Maximum general position sets for C_{46} and C_{48}.

$\mathrm{gp}(Q_1) = \mathrm{gp}(Q_2) = 2$ and $\mathrm{gp}(Q_3) = 4$ can be determined by hand, while the values $\mathrm{gp}(Q_4) = 5$, $\mathrm{gp}(Q_5) = 6$, $\mathrm{gp}(Q_6) = 8$ and $\mathrm{gp}(Q_7) = 9$ were reported in [8, Table 1]. In this note, we have verified these values by our three algorithms. The computational results are presented in the middle of Table 1.

For the last set of examples, we have considered three examples from algebraic graph theory, the Cayley graphs $\mathrm{Cay}(\mathbb{Z}_9, \{1, 3, 6, 8\})$, $\mathrm{Cay}(\mathbb{Z}_9, \{1, 2, 3, 6, 7, 8\})$ and $\mathrm{Cay}(\mathbb{Z}_{20}, \{1, 3, 17, 19\})$. The computational results are summarised in the last three rows of Table 1.

As can be seen from Table 1, SA was the fastest in six cases and GA in four cases. Both algorithms generally seem to be faster than ILP, which has never been the fastest.

4. Conclusion and future directions

In this paper, we have proposed and implemented two meta-heuristic algorithms, Genetic Algorithm (GA) and Simulated Annealing (SA), to tackle the general position

https://doi.org/10.1017/S0004972725100178 Published online by Cambridge University Press

TABLE 1. The general position number of some graphs and running times of the ILP, GA and SA.

G	$n(G)$	$\mathrm{gp}_{\mathrm{ILP}}$ $\mathrm{gp}_{\mathrm{GA}}$ $\mathrm{gp}_{\mathrm{SA}}$	ILP time	GA time	SA time
C_{46}	46	8	73.10	29.72 $n_p = 50$ $Maxit = 5000$	22.32 $T_0 = 10$ $Maxit = 500$
C_{48}	48	8	71.46	38.34 $n_p = 50$ $Maxit = 5000$	11.95 $T_0 = 10$ $Maxit = 500$
Q_3	8	4	0.03	0.05 $n_p = 10$ $Maxit = 100$	0.010 $T_0 = 10$ $Maxit = 10$
Q_4	16	5	0.38	0.22 $n_p = 20$ $Maxit = 200$	0.075 $T_0 = 10$ $Maxit = 10$
Q_5	32	6	11.52	1.84 $n_p = 20$ $Maxit = 400$	1.93 $T_0 = 10$ $Maxit = 50$
Q_6	64	8	359.01	82.11 $n_p = 50$ $Maxit = 4500$	144.985 $T_0 = 10$ $Maxit = 500$
Q_7	128	9	3617.39	2060.06 $n_p = 50$ $Maxit = 8000$	2574.79 $T_0 = 10$ $Maxit = 100$
$\mathrm{Cay}(\mathbb{Z}_9, \{1, 3, 6, 8\})$	9	4	1.22	0.06 $n_p = 10$ $Maxit = 50$	0.015 $T_0 = 10$ $Maxit = 10$
$\mathrm{Cay}(\mathbb{Z}_9, \{1, 2, 3, 6, 7, 8\})$	9	4	0.41	0.01 $n_p = 10$ $Maxit = 50$	0.017 $T_0 = 10$ $Maxit = 10$
$\mathrm{Cay}(\mathbb{Z}_{20}, \{1, 3, 17, 19\})$	20	7	0.85	0.54 $n_p = 20$ $Maxit = 250$	0.286 $T_0 = 10$ $Maxit = 50$

problem. While both methods show promising results for various graphs, there is potential for exploring other meta-heuristic approaches that offer a comparable trade-off between solution quality and computation time. For instance, algorithms such as Ant Colony Optimisation or Particle Swarm Optimisation may enhance the search capabilities, especially when coupled with high-performance computing

https://doi.org/10.1017/S0004972725100178 Published online by Cambridge University Press

resources. These approaches could be particularly useful for instances where even linear programming models fail due to the size and complexity of the input graph.

References

[1] J. Araujo, M. C. Dourado, F. Protti and R. Sampaio, 'The iteration time and the general position number in graph convexities', *Appl. Math. Comput.* **487** (2025), Article no. 129084.

[2] G. Di Stefano, 'Mutual visibility in graphs', *Appl. Math. Comput.* **419** (2022), Article no. 126850.

[3] G. Di Stefano, S. Klavžar, A. Krishnakumar, J. Tuite and I. G. Yero, 'Lower general position numbers in graphs', *Discuss. Math. Graph Theory* **45** (2025), 509–531.

[4] V. Iršič, S. Klavžar, G. Rus and J. Tuite, 'General position polynomials', *Results Math.* **79** (2024), Article no. 110.

[5] S. Klavžar, A. Krishnakumar, J. Tuite and I. G. Yero, 'Traversing a graph in general position', *Bull. Aust. Math. Soc.* **108** (2023), 1–13.

[6] S. Klavžar, D. Kuziak, I. Peterin and I. G. Yero, 'A Steiner general position problem in graph theory', *Comput. Appl. Math.* **40** (2021), 1–15.

[7] J. Körner, 'On the extremal combinatorics of the Hamming space', *J. Combin. Theory Ser. A* **71** (1995), 112–126.

[8] D. Korže and A. Vesel, 'General position sets in two families of Cartesian product graphs', *Mediterr. J. Math.* **20** (2023), Article no. 203.

[9] P. Manuel and S. Klavžar, 'A general position problem in graph theory', *Bull. Aust. Math. Soc.* **98** (2018) 177–187.

[10] P. Manuel, R. Prabha and S. Klavžar, 'The edge general position problem', *Bull. Malays. Math. Sci. Soc.* **45** (2022), 2997–3009.

[11] A. I. Melker and M. A. Krupina, 'Geometric modeling of midi-fullerene growth from C_{32} to C_{60}', *St. Petersbg. State Polytech. Univ. J.: Phys. Math.* **3** (2017), 22–28.

[12] M. Thankachy, S. V. Ullas Chandran, J. Tuite, E. Thomas, G. Di Stefano and G. Erskine, 'On the vertex position number of graphs', *Discuss. Math. Graph Theory* **44** (2024), 1169–1188.

[13] E. J. Thomas, S. V. Ullas Chandran, J. Tuite and G. Di Stefano, 'On monophonic position sets in graphs', *Discrete Appl. Math.* **354** (2024), 72–82.

[14] E. J. Thomas, S. V. Ullas Chandran, J. Tuite and G. Di Stefano, 'On the general position number of Mycielskian graphs', *Discrete Appl. Math.* **353** (2024), 29–43.

[15] J. Tian and S. Klavžar, 'Variety of general position problems in graphs', *Bull. Malays. Math. Sci. Soc.* **48** (2025), Article no. 5.

[16] J. Tian and K. Xu, 'On the general position number of the k-th power graphs', *Quaest. Math.* **47** (2024), 2215–2230.

[17] J. Tuite, E. Thomas and S. V. Ullas Chandran, 'On some extremal position problems for graphs,' *Ars Math. Contemp.* **25** (2025), Article no. P1.09.

[18] S. V. Ullas Chandran and G. Jaya Parthasarathy, 'The geodesic irredundant sets in graphs', *Int. J. Math. Comb.* **4** (2016), 135–143.

[19] S. V. Ullas Chandran, S. Klavžar and J. Tuite, 'The general position problem: a survey', Preprint, 2025, arXiv:2501.19385 [math.CO].

ZAHRA HAMED-LABBAFIAN, Department of Applied Mathematics,
Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran
e-mail: hamedlabbafianzahra@gmail.com

NARJES SABEGHI, Department of Mathematics,
Faculty of Basic Sciences, Velayat University, Iranshar, Iran
e-mail: n.sabeghi@velayat.ac.ir

https://doi.org/10.1017/S0004972725100178 Published online by Cambridge University Press

MOSTAFA TAVAKOLI, Department of Applied Mathematics,
Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran
e-mail: m_tavakoli@um.ac.ir

SANDI KLAVŽAR, Faculty of Mathematics and Physics,
University of Ljubljana, Ljubljana, Slovenia;
Institute of Mathematics, Physics and Mechanics, Ljubljana, Slovenia;
and
Faculty of Natural Sciences and Mathematics, University of Maribor, Maribor, Slovenia
e-mail: sandi.klavzar@fmf.uni-lj.si

https://doi.org/10.1017/S0004972725100178 Published online by Cambridge University Press