



# Cellular geometric semantic genetic programming

Lorenzo Bonin<sup>1</sup> · Luigi Rovito<sup>1</sup> · Andrea De Lorenzo<sup>2</sup> · Luca Manzoni<sup>1</sup>

Received: 31 May 2023 / Revised: 6 December 2023 / Accepted: 3 January 2024  
© The Author(s) 2024

## Abstract

Among the different variants of Genetic Programming (GP), Geometric Semantic GP (GSGP) has proved to be both efficient and effective in finding good solutions. The fact that the operators of GSGP operate on the *semantics* of the individuals in a clear way provides guarantees on the way the search is performed. GSGP is not, however, free from limitations like the premature convergence of the population to a small—and possibly sub-optimal—area of the search space. One reason for this issue could be the fact that good individuals can quickly “spread” in the population suppressing the emergence of competition. To mitigate this problem, we impose a cellular automata (CA) inspired communication topology over GSGP. In CAs a collection of agents (as finite state automata) are positioned in a  $n$ -dimensional periodic grid and communicates only locally with the automata in their neighbourhoods. Similarly, we assign a location to each individual on an  $n$ -dimensional grid and the entire evolution for an individual will happen locally by considering, for each individual, only the individuals in its neighbourhood. Specifically, we present an algorithm in which, for each generation, a subset of the neighbourhood of each individual is sampled and the selection for the given cell in the grid is performed by extracting the two best individuals of this subset, which are employed as parents for the Geometric Semantic Crossover. We compare this *cellular GSGP* (cGSGP) approach with standard GSGP on eight regression problems, showing that it can provide better solutions than GSGP. Moreover, by analyzing convergence rates, we show that the improvement is observable regardless of the number of executed generations. As a side effect, we additionally show that combining a small-neighbourhood-based cellular spatial structure with GSGP helps in producing smaller solutions. Finally, we measure the spatial autocorrelation of the population by adopting the Moran’s  $I$  coefficient to provide an overview of the diversity, showing that our cellular spatial structure helps in providing better diversity during the early stages of the evolution.

**Keywords** Geometric semantic genetic programming · Genetic programming · Cellular genetic programming

---

Lorenzo Bonin and Luigi Rovito have contributed equally to this work.

---

Extended author information available on the last page of the article

## 1 Introduction

Genetic Programming (GP) [1] is a prominent technique in Evolutionary Computation (EC). It consists of evolving programs, typically represented as trees, to solve specific supervised learning problems. Among the many variants of GP, Geometric Semantic GP (GSGP) [2] has received widespread interest since its introduction in 2012, showing its ability to outperform standard GP [3–5].

Unlike traditional GP, where the functions space is explored with search operators that act on the syntactic representations of individuals without taking into account the effects on the semantics, GSGP considers the space of the semantic representation of programs and each syntactic manipulation derives from the need to obtain a specific effect on the semantics. In this case, semantics can be generally defined as the behaviour of a program when it is executed, which can be approximated as the vector having as elements all outputs generated on the training data [2].

Despite its advantages over standard GP, GSGP still suffers from some limitations, as mentioned by Moraglio et al [6]. In particular, while the fitness landscape induced by GSGP is unimodal, the fact that not all points (programs) in that landscape can be represented leads to the emergence of local optima or “holes” in the landscape, where the evolution might get stuck. Another effect is that GSGP might promote convergence towards individuals that dominate others in the early generations. In other words, the number of generations needed for one individual to fully propagate, i.e., the *takeover time*, is small. This can lead to being able to move in the fitness landscape only via mutation, due to the convex hull of the population—the only area of the space that can be explored via crossover—having shrunk too much and not around the global optimum.

In this work, we propose a new form of GSGP, which takes inspiration from one of the oldest natural computing models, *cellular automata* (CA) [7–9], to force the interactions among individuals of a population to be local according to a given topology.

Recall that CA is a formal model comprising a grid of identical cells, or automata, each updating its internal state based on a local rule dependent solely on a fixed number of neighboring automata. In a similar fashion, all individuals of a population are positioned into a grid (and  $n$ -dimensional torus). To update the position  $i$  in the grid, there is a selection phase happening only among individuals in the neighbourhood of  $i$ , i.e., in a ball of a fixed radius; if crossover happens, it will also be only between two individuals selected in the same neighbourhood. Hence, all interactions are local according to the topology imposed on the population. One effect of this is that even a good individual is limited in “propagation speed” by this topology, allowing more time for other—possibly better—individuals to be generated. This should help the optimization process to keep a higher rate of diversity in the population during the early generations, allowing for a more complete and global exploration of the search space at the beginning of the evolution before it starts to converge to dominant individuals.

The resulting *cellular GSGP* (cGSGP) is compared, in this paper, with standard GSGP in multiple regression problems. A two-dimensional torus-based topology

and multiple neighbourhood sizes are tested, together with multiple ways of performing only local selections. The results show that, in fact, cGSGP provides an improvement compared to GSGP in terms of the quality of the solutions.

The paper is structured as follows: Section 2 provides an overview of previous CA-inspired evolutionary methods. Section 3 provides a formalisation and explanation of the proposed method. The experimental settings and the datasets used in the experiments are described in Sect. 4, while in Sects. 5 and 6 the results obtained are presented and discussed. Finally, Sect. 7 summarizes the main contributions of this work and draws some possible directions for future research.

## 2 Related works

The idea of imposing a certain spatial structure to the population of evolutionary algorithms was already explored in the literature [10–12], allowing each individual to interact only with a small subset of the population, i.e., its *neighbourhood*, usually specified by imposing a grid structure on the population, leading to the formation of semi-isolated groups of individuals having similar features.

In the case of classical genetic algorithm (GA) [13], the concept was introduced in the form of cellular Genetic Algorithm (cGA) [10–12]. cGA was used to improve optimisation performance for a well-known problem such as the Travelling Salesman Problem (TSP) [14], where it was combined with the Simulated Annealing algorithm. Extensive research on the impact of different neighbourhoods and topologies on the performance of cGA in various problem domains was performed in [10] and [12]. Alba et al. [10] also proposed a dynamic version of cGA, which adaptively adjusts the exploration/exploitation trade-off ratio during evolution.

Murata et al. [15] proposed a variation of GA for multiobjective optimisation with a cellular structure (C-MOGA) that was an extension of multiobjective GA (MOGA). In C-MOGA, the selection operator is applied in the neighbourhood of each cell. Nebro and coauthors [16] came up with another approach for multiobjective optimisation which takes inspiration from canonical cGA and results in an algorithm called MOCeLl.

Folino et al. [17] presented a cellular parallel implementation of GP endowed with a load-balancing mechanism that distributes the computational load among the processors equally, showing that the cellular approach can outperform both standard GP and the island model [18], where fully separate evolutions are carried out on four different subsets of the population. Authors in [19] and [20] showed how a spatial population structure in combination with a local elitist replacement scheme can effectively reduce the bloat phenomenon (i.e., the increase in average tree size without a corresponding increase in fitness) for GP, without compromising the performance.

In addition to GA and GP, several other evolutionary algorithms employed the cellular model to control the diffusion of solutions throughout the population [21–23], demonstrating how the idea of imposing a cellular structure is a concept applicable in general to populations of solutions independently from the algorithm working on them.

However, to the best of our knowledge, the use of a cellular structure has not been explored in the context of GSGP, an area that we plan to start tackling in this work.

Promoting diversity within a population can also be achieved through a process known as speciation. This concept is rooted in the observation that natural ecosystems consist of distinct physical spaces, or niches, each characterized by specific features. This diversity of niches facilitates the emergence of various species, with individuals of a species sharing common attributes. Within each niche, resources are constrained, leading individuals to compete for them.

Likewise, the population of an Evolutionary Algorithm can be strategically partitioned into separate niches comprising individuals with similar structural characteristics. This arrangement ensures that only individuals belonging to the same species can interact through crossovers.

Della Cioppa et al. [24] proposed an adaptive species discovery strategy which addresses some limitations of niching methods which rely on a priori knowledge about the fitness landscape. Authors in [25] took inspiration from the design principles of the NEAT algorithm [26] to leverage speciation to limit programs growth in GP. Through their algorithm, named neat-GP, complexity is built only when the problem requires it and the population is progressively divided into species containing individuals with similar size and shape. Juárez-Smith et al. [27] integrated neat-GP with a local search operator to improve the quality of solutions. Authors in [28] define a network distance metric to speciate a population of artificial gene regulatory networks, showing how speciation tends to facilitate diversity and keep small-sized individuals. Martins et al. [29] combined GA and speciation with a grid pattern recognition approach to reduce investment risk and increase profits. Additionally, they made an analysis of the differences between the speciated and the non-speciated approach. Wickman et al. [30] leveraged speciation to evolve a diverse population of policies in the context of Reinforcement Learning.

Even though there are several works about cellular EAs, it is not trivial to assess whether the cellular structure in EA is state-of-the-art or not. Several works in the literature have attempted to study the effect of selection pressure and sampling strategy in both standard EA [31, 32] and cellular-based EA [33, 34].

The cellular structure in EA can also have a significant impact on the *takeover time*, i.e., the number of generations required for a single best individual to fully propagate to the population with copies of itself. Clearly, a small takeover time implies a fast loss in the diversity of the population. In cellular-based EA the spatial structure itself provides a way of controlling takeover time and selection pressure, as shown, for example, in [34].

Therefore, considering that premature convergence to local sub-optimal solutions is one of the major limitations of GSGP, our intuition is that imposing a cellular structure may be beneficial in the context of GSGP. Such an approach may help increase the takeover time and avoid a fast loss in diversity.

### 3 Methods

In this section, we recall the basics of GSGP and then we will introduce cGSGP, focusing on how the evolution process must be changed to take into account the topology imposed on the population.

### 3.1 Geometric semantic genetic programming

Traditionally, Genetic Programming (GP) explores the space of solutions by means of operators acting on the syntax of programs, i.e., their formal representation, without any regard to the semantic effect of the operators on the individuals. In recent years, the integration of semantic awareness in GP has been shown to improve its performance in different scenarios, attracting the interest of the scientific community [35]. Although several definitions are possible, here the semantics of an individual is the vector of output values of a program on a set of inputs. Hence, we can map each individual into a point on the *semantic space*, a finite-dimensional vector space, usually  $\mathbb{R}^n$ .

GSGP, introduced for the first time by Moraglio et al. [2], defines operators that act directly on the semantics of individuals, the Geometric Semantic Operators (GSOs).

Specifically, the *Geometric Semantic Crossover* (GSC) takes the following form:

$$T_{XO} = R \cdot T_1 + (1 - R) \cdot T_2 \quad (1)$$

where  $T_1, T_2 : \mathbb{R}^n \rightarrow \mathbb{R}$  are the two parents, and  $R$  is a random real function with outputs in the interval  $[0, 1]$ , usually represented by a random tree. Since, for each input  $x \in \mathbb{R}^n$ ,  $T_{XO}(x)$  is a linear combination of the two parents' outputs, the GSC ensures that the offspring has semantics belonging to the segment joining the semantics of the two parents on the semantic space.

Similarly, the *Geometric Semantic Mutation* (GSM) is defined as follows:

$$T_M = T + m \cdot (R_1 - R_2) \quad (2)$$

where  $T : \mathbb{R}^n \rightarrow \mathbb{R}$  is the function being mutated,  $R_1, R_2$  are two random real functions with outputs in the interval  $[0, 1]$  and  $m$  is a parameter called mutation step. GSM can be considered as a “weak” perturbation of the individual since the new individual is contained in the hyper-sphere of radius  $m$  centred in the semantics of the parent in the semantic space.

Notice that the same geometric semantic operators of crossover and mutation that induce an unimodal fitness landscape, however, also increase the size of the individuals. For crossover, in particular, this increase is exponential in the number of generations, as already highlighted when GSGP was introduced [2]. Practical solutions to this problem have been proposed over the years [2, 36–40], to make this approach usable in real-world scenarios. The combination of the formal properties of GSGP and the presence of efficient implementations [41] allowed it to outperform standard GP on many symbolic regression tasks and real-world problems [5, 37].

### 3.2 Cellular GSGP

Standard GSGP does not impose any spatial structure on the population, allowing selection to be performed across the entire population and the crossover to happen between any two individuals.

In cGSGP a neighbourhood structure is imposed in a way inspired by CA [7–9]. Recall that CA is a formal model where finite automata are distributed according to a certain topology and each cell updates its own internal state following a local rule depending only on a certain number of neighbour automata.

Similarly, we distribute the individuals of a population on a toroidal grid. In this scenario, each individual belongs to a specific cell of this grid, hence to a spatial location, which is chosen randomly at the beginning. At this point, it is possible to introduce the concept of neighbourhood within our population. Specifically, we consider an individual to be in the neighbourhood of radius  $r$  of a given cell if it is contained in the hypercube with side  $2r + 1$  corresponding to the Moore neighbourhood of radius  $r$ . In the following the  $n$ -dimensional toroidal grid with neighbourhood of radius  $r \in \mathbb{N}$  will be denoted by  $\mathcal{T}_r^n$ . We choose to impose a toroidal configuration to the grid in order to have a complete neighbourhood for all cells: a hypercube has incomplete neighbourhoods for cells near the faces of the hypercube. Furthermore, notice that there are no “disconnected” neighbourhoods in  $\mathcal{T}_r^n$  when  $r > 0$ ; this means that information can travel between any two points of the grid, thus there are no isolated or independent sub-populations.

Once a neighbourhood structure has been defined, it is necessary to define how this structure is going to influence the selection process and the crossover operator<sup>1</sup>.

Let us now consider a position  $i$  in an  $n$ -dimensional toroidal grid with neighbourhood  $\mathcal{N}_i$ . For example, in two dimensions  $i = (i_1, i_2)$  and a neighbourhood of radius  $r$  it is  $\mathcal{N}_i = \{j = (j_1, j_2) \text{ such that } \|i - j\|_\infty \leq r\}$ , where  $\|x\|_\infty$  is the maximum norm  $\|x\| = \max_{1 \leq i \leq n} |x_i|$ . Moreover, each individual is contained in its own neighbourhood (Fig. 1).

For any given population each position  $i$  has an associated tree  $T_i$ , so we expect that when we need to replace a tree in position  $i$  that it will be derived from the trees in positions part of  $\mathcal{N}_i$  (its neighbourhood).

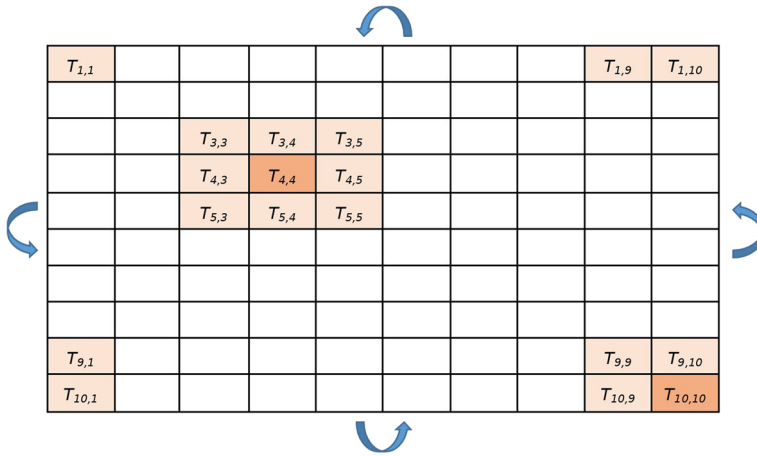
If crossover is not performed, it is sufficient to replace (and possibly mutate) the individual in position  $i$  with one selected locally (maybe itself). Furthermore, as long as each selection is performed in a local way, forcing crossover to be local is trivial: it is sufficient to perform crossover between two locally selected individuals. Hence, we will focus on how we can define *local* selection (Figs. 2, 3).

### 3.2.1 Selection strategies

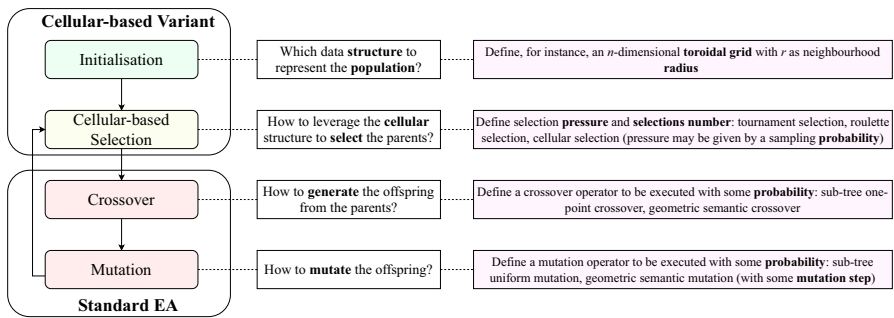
As a first strategy, we employ a rank-based selection method within local neighbourhoods (RKS), where the entire neighbourhood  $\mathcal{N}_i$  of position  $i$  is considered. In this case, the selection procedure returns the elements of  $\mathcal{N}_i$  ordered by fitness, from the best one to the worst one without any repetition. As a consequence, if no crossover is performed then the new tree in position  $i$  will be the best one in  $\mathcal{N}_i$ . If, instead, crossover is performed, then it will be between the two best individuals in  $\mathcal{N}_i$ .

The selection given by RKS is deterministic and, without the presence of only local interactions, would be too strict and would force premature convergence in a few generations. Local interactions avoid this problem (different neighbourhoods can

<sup>1</sup> Mutation, involving only one individual, is inherently local and needs no modifications.



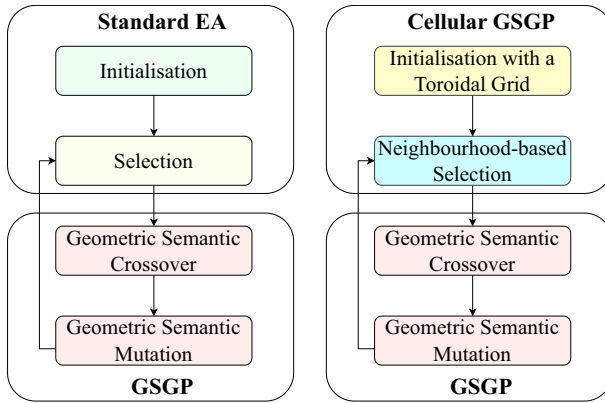
**Fig. 1** Example of a population with 100 individuals distributed according to  $\mathcal{T}_1^2$ . In the figure, two positions with the corresponding neighbourhoods are highlighted, following the toroidal structure when the position is close to the borders. Each cell contains the individual in that position, whose row index and column index are indicated as a subscript



**Fig. 2** Diagram describing a baseline cellular-based evolutionary algorithm. For each activity, the main design choice is highlighted, and an overview of the main hyper-parameters and variants is presented. Especially, we highlight in bold the hyper-parameters and the main building blocks that compose the design choices

have different best individuals), but we might still want to have a less “draconian” selection. To this end, as a second strategy, we employ a tournament-based selection method within local neighbourhoods ( $TRS_p$ ), where the tournament size is defined w.r.t. a real value  $p \in (0, 1]$  representing a proportion of the given neighbourhood.

The idea of  $TRS_p$  is that RKS is not applied on the entire neighbourhood of an individual  $i$ , but on a subset  $S \subseteq \mathcal{N}_i$ . This subset  $S$  is selected by iterating across the elements of  $\mathcal{N}_i$  and inserting them into  $S$  with probability  $p$ . In other words,  $S$  is a sampled subset of  $\mathcal{N}_i$  with probability  $p$ . Notice that now  $p$  can be used as a parameter to tune the selection pressure, with  $p = 1$  resulting in the most pressure as in the RKS selection strategy.



**Fig. 3** Diagram comparing a vanilla GSGP and our proposed cGSGP method. The figure highlights which are the components that differ between the two techniques

In Algorithm 1 we provide a pseudo-code for the  $TRS_p$  selection strategy, which is equivalent to RKS when  $p = 1$ .

**Algorithm 1** Selection strategy adopted within cGSGP (CellularSelectionGSGP)

---

**Input**

- $P$  population at the beginning of the current generation
- $p$  probability of inserting an individual into a tournament
- $r$  neighbourhood radius
- $n$  toroidal grid dimension

**Output**

- $P_S$  set of pairs containing the selected parents for each position in the grid

**Require:**  $0 < p \leq 1, p \in \mathbb{R}$

**Require:**  $r \in \mathbb{N}$

**Require:**  $n \in \mathbb{N}$

**Ensure:**  $P$  is an  $n$ -dimensional toroidal grid

```

P_S ← {}
for i ← 1 to |P| do
    S ← {}
    ▷ Extract the neighbourhood with radius r of the ith individual in P
    N_i ← Neighbourhood(P, i, r, n)
    for j ← 1 to |N_i| do
        ▷ Sample a random real number in [0, 1)
        if RandomUniform(0, 1) < p then
            S ← S ∪ {N_i[j]}
        end if
    end for
    ▷ Take the two individuals in S with the best fitness
    S_BestFirst, S_BestSecond ← SelectTwoBest(S)
    P_S ← P_S ∪ {(S_BestFirst, S_BestSecond)}
end for
return P_S
    
```

---



## 4 Experimental study

In the experimental phase, we try to provide answers to the following questions:

- RQ1** Is it possible to overcome the issue of converging too fast to small and possibly sub-optimal areas of the search space?
- RQ1** Can cGSGP exploit the presence of local communication to produce better solutions than GSGP?

To this end, we compare standard GSGP with cGSGP by using a specific grid topology with different neighbourhood sizes. In particular, we consider a two-dimensional toroidal grid with neighbourhood of radius 1, 2, and 3, denoted by  $\mathcal{T}_1^2$ ,  $\mathcal{T}_2^2$ , and  $\mathcal{T}_3^2$ , respectively.

Each method is tested on eight regression datasets that have already been used in GP-related problems [5, 37, 42, 43]. Table 1 summarizes the number of samples and the number of variables for each dataset. References are added to provide a link to the complete description of these datasets.

Each dataset has been partitioned in 100 train-test splits. Specifically, 70 % of the observations, selected at random with uniform probability, has been used to build the training set, while the remaining 30 % has been used to build the test set. We run for each method and for each dataset 100 repetitions with different seeds, one for each split (the KJ6 dataset is unique in that it is defined for a single split; therefore, we use the same split for all repetitions, with the seed influencing only the evolution).

The optimisation problem we aim to address is a single-objective minimisation problem in which the fitness is the Root Mean Squared Error (RMSE) [51] computed between the target and the predictions obtained by applying the discovered symbolic models on the training set. We check the performance on the test set to obtain information on the way the model behaves on unseen data.

**Table 1** The number of observations and the number of variables of tested datasets

Dataset	# Observations	# Variables
Vladislavleva14 (V14) [44]	6024	5
Keijzer6 (KJ6) [45, 46]	170	1
Airfoil (ARF) [47]	1502	5
Concrete (CNC) [3]	1029	8
Slump (SLM) [48]	102	9
Toxicity (TXC) [49]	234	626
Yacht (YCH) [50]	307	6
Parkinson (PRK) [4]	5875	18

**Table 2** Parameters adopted during the experimental phase

Parameter	Value
Max depth	6
Tree generation	Ramped half-and-half [1]
Crossover probability	0.90
Mutation probability	0.50
Mutation step	$\mathcal{U}(0, 1)$ for each mutation event
Operators	+, -, *, ÷ (protected)
Constants	100 $\sim \mathcal{U}(-100, 100)$ fixed for each run

The evolution parameters are fixed for all runs and described in Table 2. We test different pairs of population size and number of generations. The chosen combinations are described in Table 3.

We choose the different types of population sizes in order to obtain three different perfectly squared bi-dimensional toroidal grids whose dimensions are evenly distributed. Moreover, we set the number of generations in order to always obtain, approximately, the same number of fitness evaluations, and thus encourage a fairer comparison (since the total number of fitness evaluations is not a multiple of 900, we set the corresponding number of generations to the nearest rounded integer available).

We test each method by using: crossover and mutation together sequentially ( $\mathcal{E}_{cx+mut}$ ), only crossover ( $\mathcal{E}_{cx}$ ), and only mutation ( $\mathcal{E}_{mut}$ ). For the sake of simplicity, we define this variant as *exploration pipeline*.

Each method is compared with the others methods by using a Wilcoxon signed-rank test [52], and the corresponding  $p$ -values are collected and adjusted by using a Holm-Bonferroni correction [53] with  $\alpha = 0.05$  (this test is carried out only if a Kruskal-Wallis test [54] with  $\alpha = 0.05$  is passed for the involved methods and the given dataset).

To guarantee that the results obtained are due to the presence of a communication topology in cGSGP, GSGP was tested with tournament selection with different tournament sizes up to 12. When the results were statistically compared, no significant difference was found in the results. As a consequence, we present the results comparing cGSGP with GSGP with tournament selection of size 4.

All the code implementing GSGP and cGSGP used in the experiments is available in the following repository: <https://github.com/lurovi/CA-GSGP>.

## 5 Results

In this section, we present the results from our experimental phase and discuss the outcomes.

**Table 3** Combinations of population size and number of generations that are tested. For each combination, the side length of the squared toroidal grid is shown, along with the corresponding total number of fitness evaluations

Population size	Generations	Side length	Fitness evaluations
100	1000	10	100000
400	250	20	100000
900	111	30	99900

## 5.1 Statistical comparison with the baseline

We showcase the experimental results comparing GSGP and cGSGP, utilizing the combinations of topology and neighbourhood radius introduced earlier, and employing two distinct selection methods: RKS and  $\text{TRS}_{0,6}$ .

The comparisons for all the proposed combinations of parameters are shown in the following tables, in which the median RMSE values on the test sets are presented. Each table describes the outcome for a specific combination of population size and number of generations. Within the same table, for each dataset, all topologies, selection strategies, and variants of exploration pipeline are presented.

For a given dataset, cGSGP methods that exhibit statistically better performance than the methods in the table with the same selection strategy and exploration pipeline, including the corresponding GSGP, are marked with an asterisk (\*). Moreover, we highlight in bold the cGSGP methods that, taken individually, are meaningfully better, according to a Wilcoxon signed-rank test, than the GSGP baseline with the same exploration pipeline (Tables 4, 5, 6).

As a general observation for  $\mathcal{E}_{\text{cx+mut}}$ , in all datasets, cGSGP is able to perform better than GSGP, with the specific selection method (RKS or  $\text{TRS}_{0,6}$ ) not providing a significant difference. The only exception seems to be KJ6 for 100 as population size.

In case the population size is small, what influences the results tends to be the topology: at least a minimum size of the neighbourhood is needed, in fact,  $\mathcal{T}_1^2$  is the topology with the smallest neighbourhood (9 positions including the central one) and is usually unable to provide results which are better than GSGP in a statistically significant way.

Larger neighbourhoods instead tend to provide more consistent results. Notice that, while the best results are obtained with  $\mathcal{T}_3^2$ , a quite large neighbourhood, a smaller one, with radius 2, is sufficient to consistently perform better than GSGP. This could be useful to obtain good results while limiting the amount of local interaction between the individuals in the population. Moreover, it seems that by increasing the size of the population it becomes more likely that the cGSGP method can perform better than the baseline. This occasionally seems to happen even within a small neighbourhood radius. This result suggests that a large enough population size is probably necessary to take advantage of the cellular-based selection performed on the toroidal grid.

**Table 4** Table of the median best fitness for cGSGP on the test set and all considered topologies

Method	100 as population size				1000 generations			
	V14	KJ6	ARF	CNC	SLM	TXC	YCH	PRK
$\mathcal{E}_{\text{cx+mut}}$								
GSGP	1.47	1.72	2.6	6.5	4.38	2262.13	3.69	8.35
$\mathcal{T}_1^2$ w/RKS	<b>1.23</b>	1.88	2.53	6.47	4.29	<b>2274.38</b>	3.88	8.4
$\mathcal{T}_2^2$ w/RKS	<b>1.47</b>	3.08	<b>2.36</b>	<b>6.13</b>	<b>4.11</b>	<b>2194.0</b>	<b>3.32</b>	<b>8.16</b>
$\mathcal{T}_3^2$ w/RKS	<b>1.47</b>	3.84	<b>2.34 *</b>	<b>5.89 *</b>	4.13	<b>2174.52 *</b>	<b>3.22 *</b>	<b>8.05 *</b>
$\mathcal{T}_1^2$ w/TRS <sub>0.6</sub>	1.06	1.75	2.64	6.61	4.5	2317.46	4.09	8.5
$\mathcal{T}_2^2$ w/TRS <sub>0.6</sub>	1.31	2.48	<b>2.43</b>	<b>6.29</b>	<b>4.05</b>	<b>2243.8</b>	<b>3.44</b>	<b>8.22</b>
$\mathcal{T}_3^2$ w/TRS <sub>0.6</sub>	<b>1.57</b>	3.13	<b>2.34 *</b>	<b>6.01 *</b>	4.35	<b>2201.03 *</b>	<b>3.3 *</b>	<b>8.11 *</b>
$\mathcal{E}_{\text{cx}}$								
GSGP	0.75	2.77	38.76	16.14	9.62	2407.62	13.63	10.58
$\mathcal{T}_1^2$ w/RKS	0.41	2.56	39.45	17.24	10.65	2421.11	13.67	10.93
$\mathcal{T}_2^2$ w/RKS	<b>0.29</b>	2.83	38.76	20.79	11.5	2401.69	13.86	10.97
$\mathcal{T}_3^2$ w/RKS	<b>0.26 *</b>	2.84	38.74	27.19	12.09	2421.87	14.09	11.09
$\mathcal{T}_1^2$ w/TRS <sub>0.6</sub>	1.06	2.88	38.77	16.25	9.42	2384.49	13.4	10.51
$\mathcal{T}_2^2$ w/TRS <sub>0.6</sub>	<b>0.37</b>	2.57	38.65	17.61	11.25	2404.46	13.89	10.93
$\mathcal{T}_3^2$ w/TRS <sub>0.6</sub>	<b>0.28</b>	2.82	39.01	21.95	10.84	2415.91	13.95	10.89
$\mathcal{E}_{\text{mut}}$								
GSGP	1.48	2.08	2.66	6.53	3.8	2278.42	4.11	8.68
$\mathcal{T}_1^2$ w/RKS	2.17	2.48	<b>2.58</b>	<b>6.37</b>	<b>3.69</b>	<b>2244.66</b>	<b>3.93</b>	<b>8.62</b>
$\mathcal{T}_2^2$ w/RKS	2.44	3.57	<b>2.48</b>	<b>6.1</b>	3.66	<b>2195.85</b>	<b>3.42</b>	<b>8.33</b>
$\mathcal{T}_3^2$ w/RKS	2.78	4.62	<b>2.4 *</b>	<b>6.02 *</b>	3.81	<b>2172.03 *</b>	<b>3.26 *</b>	<b>8.27 *</b>
$\mathcal{T}_1^2$ w/TRS <sub>0.6</sub>	1.63	1.93	2.69	6.54	3.79	<b>2273.45</b>	4.15	8.7
$\mathcal{T}_2^2$ w/TRS <sub>0.6</sub>	2.54	2.83	<b>2.5</b>	<b>6.23</b>	3.72	<b>2214.63</b>	<b>3.59</b>	<b>8.39</b>
$\mathcal{T}_3^2$ w/TRS <sub>0.6</sub>	2.75	3.58	<b>2.44 *</b>	<b>6.06 *</b>	3.7	<b>2186.87</b>	<b>3.41 *</b>	<b>8.33 *</b>

By inspecting the results for  $\mathcal{E}_{\text{cx}}$  and  $\mathcal{E}_{\text{mut}}$ , the general observation that we can formulate is that the mutation operator alone is more beneficial than the crossover operator alone as regards our proposed approach. As a matter of fact, we can see that when  $\mathcal{E}_{\text{mut}}$  is employed, it seems to be more likely that our method performs better than the baseline, while there is less significant difference when

**Table 5** Table of the median best fitness for cGSGP on the test set and all considered topologies

Method	400 as population size				250 generations			
	V14	KJ6	ARF	CNC	SLM	TXC	YCH	PRK
$\mathcal{E}_{\text{cx+mut}}$								
GSGP	3.88	1.46	3.83	7.53	4.16	2375.11	5.85	9.07
$\mathcal{T}_1^2$ w/RKS	<b>2.16</b>	0.94	3.98	7.83	3.92	<b>2244.14</b>	6.44	9.26
$\mathcal{T}_2^2$ w/RKS	<b>0.93</b>	0.89	<b>3.28</b>	<b>7.12</b>	<b>3.83</b>	<b>2188.79</b>	<b>5.32</b>	<b>8.9</b>
$\mathcal{T}_3^2$ w/RKS	<b>0.89</b>	1.24	<b>2.9 *</b>	<b>6.87 *</b>	<b>3.87</b>	<b>2201.99</b>	<b>4.45 *</b>	<b>8.71 *</b>
$\mathcal{T}_1^2$ w/TRS <sub>0.6</sub>	9.43	1.3	4.53	8.05	4.11	2302.64	6.93	9.3
$\mathcal{T}_2^2$ w/TRS <sub>0.6</sub>	<b>1.02</b>	<b>1.04</b>	<b>3.38</b>	<b>7.23</b>	<b>3.75</b>	<b>2274.06</b>	<b>5.45</b>	<b>8.99</b>
$\mathcal{T}_3^2$ w/TRS <sub>0.6</sub>	<b>0.73 *</b>	1.04	<b>3.03 *</b>	<b>7.0 *</b>	<b>3.77</b>	<b>2228.65 *</b>	<b>4.85 *</b>	<b>8.77 *</b>
$\mathcal{E}_{\text{cx}}$								
GSGP	6.17	1.4	16.62	11.66	6.24	2372.28	11.42	10.04
$\mathcal{T}_1^2$ w/RKS	<b>1.84</b>	<b>0.85</b>	17.31	12.57	6.27	<b>2279.96</b>	11.71	10.3
$\mathcal{T}_2^2$ w/RKS	<b>0.31</b>	0.99	17.41	13.45	6.87	<b>2231.01</b>	12.0	10.35
$\mathcal{T}_3^2$ w/RKS	<b>0.28</b>	1.12	16.93	13.62	7.19	<b>2229.27</b>	12.1	10.33
$\mathcal{T}_1^2$ w/TRS <sub>0.6</sub>	4.19	1.32	17.24	11.7	6.5	<b>2290.9</b>	11.54	10.04
$\mathcal{T}_2^2$ w/TRS <sub>0.6</sub>	<b>0.37</b>	<b>0.97</b>	16.48	12.59	6.54	<b>2291.52</b>	11.84	10.3
$\mathcal{T}_3^2$ w/TRS <sub>0.6</sub>	<b>0.28 *</b>	1.16	16.77	13.22	6.83	<b>2258.57</b>	11.99	10.35
$\mathcal{E}_{\text{mut}}$								
GSGP	0.83	0.92	4.55	8.68	3.96	2232.81	6.65	9.44
$\mathcal{T}_1^2$ w/RKS	0.86	0.98	<b>4.46</b>	<b>8.6</b>	<b>3.82</b>	<b>2228.39</b>	6.86	9.47
$\mathcal{T}_2^2$ w/RKS	1.02	1.19	<b>3.47</b>	<b>7.6</b>	<b>3.8</b>	<b>2215.58</b>	<b>5.61</b>	<b>9.2</b>
$\mathcal{T}_3^2$ w/RKS	1.26	1.44	<b>3.12 *</b>	<b>7.1 *</b>	<b>3.73</b>	<b>2207.89</b>	<b>5.09 *</b>	<b>9.01 *</b>
$\mathcal{T}_1^2$ w/TRS <sub>0.6</sub>	0.82	0.92	5.02	8.96	3.88	2231.76	7.04	9.58
$\mathcal{T}_2^2$ w/TRS <sub>0.6</sub>	0.88	1.18	<b>3.67</b>	<b>7.77</b>	<b>3.71</b>	<b>2221.74</b>	<b>5.94</b>	<b>9.23</b>
$\mathcal{T}_3^2$ w/TRS <sub>0.6</sub>	1.08	1.35	<b>3.24 *</b>	<b>7.3 *</b>	<b>3.68</b>	<b>2212.48</b>	<b>5.14 *</b>	<b>9.12 *</b>

$\mathcal{E}_{\text{cx}}$  is used. This could suggest that the mutation operator is the one that benefits more from our proposed selection procedure.

The cGSGP with the  $\mathcal{E}_{\text{cx}}$  combination performs better than the baseline only for the synthetic datasets (V14 and KJ6) and for TXC. On the other hand, the cGSGP with the  $\mathcal{E}_{\text{mut}}$  combination performs better than the baseline for all the

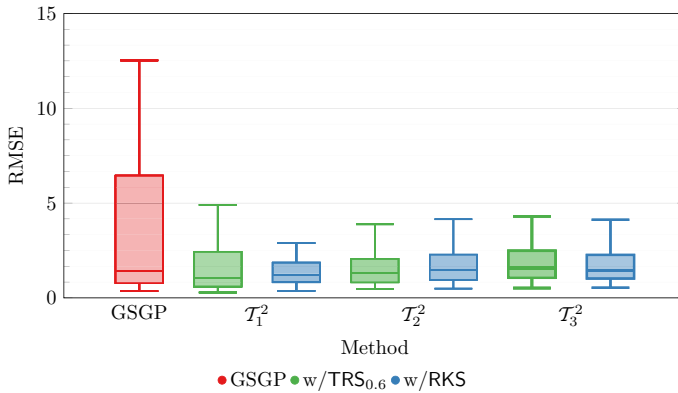
**Table 6** Table of the median best fitness for cGSGP on the test set and all considered topologies

Method	900 as population size				111 generations			
	V14	KJ6	ARF	CNC	SLM	TXC	YCH	PRK
$\mathcal{E}_{\text{cx+mut}}$								
GSGP	97.29	1.46	4.77	8.52	4.42	2431.04	7.01	9.41
$\mathcal{T}_1^2$ w/RKS	<b>8.07</b>	1.01	5.5	9.18	4.3	<b>2298.56</b>	7.88	9.6
$\mathcal{T}_2^2$ w/RKS	<b>0.78</b>	<b>0.69</b>	<b>4.32</b>	<b>8.41</b>	<b>4.11</b>	<b>2223.62</b>	<b>6.75</b>	<b>9.35</b>
$\mathcal{T}_3^2$ w/RKS	<b>0.63</b> *	<b>0.7</b>	<b>3.86</b> *	<b>8.05</b> *	<b>3.94</b> *	<b>2184.18</b> *	<b>6.11</b> *	<b>9.19</b> *
$\mathcal{T}_1^2$ w/TRS <sub>0.6</sub>	204.42	1.94	6.07	9.28	4.47	<b>2304.14</b>	8.11	9.64
$\mathcal{T}_2^2$ w/TRS <sub>0.6</sub>	<b>1.46</b>	<b>0.87</b>	<b>4.58</b>	8.49	4.24	<b>2206.71</b>	6.95	9.41
$\mathcal{T}_3^2$ w/TRS <sub>0.6</sub>	<b>0.8</b> *	<b>0.84</b>	<b>4.03</b> *	<b>8.11</b> *	<b>3.91</b> *	<b>2191.4</b> *	<b>6.41</b> *	<b>9.28</b> *
GSGP	25.91	1.49	8.06	10.56	5.44	2344.68	9.94	9.92
$\mathcal{T}_1^2$ w/RKS	<b>1.8</b>	<b>0.86</b>	10.26	11.14	5.9	<b>2258.13</b>	10.67	10.05
$\mathcal{T}_2^2$ w/RKS	<b>0.37</b>	<b>0.84</b>	10.08	12.04	6.04	<b>2198.97</b>	10.86	10.11
$\mathcal{T}_3^2$ w/RKS	<b>0.28</b> *	<b>0.71</b>	9.77	12.4	5.96	<b>2205.84</b>	10.97	10.15
$\mathcal{T}_1^2$ w/TRS <sub>0.6</sub>	132.09	1.43	9.25	10.57	5.61	<b>2317.25</b>	10.14	9.98
$\mathcal{T}_2^2$ w/TRS <sub>0.6</sub>	<b>0.96</b>	<b>0.78</b>	9.28	11.19	5.86	<b>2253.37</b>	10.63	10.07
$\mathcal{T}_3^2$ w/TRS <sub>0.6</sub>	<b>0.34</b> *	<b>0.73</b>	9.36	11.93	5.85	<b>2200.64</b>	10.58	10.1
$\mathcal{E}_{\text{mut}}$								
GSGP	0.58	0.83	8.56	11.35	4.38	2195.01	7.78	9.78
$\mathcal{T}_1^2$ w/RKS	0.64	0.84	8.83	<b>11.26</b>	4.3	<b>2193.37</b>	7.95	9.81
$\mathcal{T}_2^2$ w/RKS	0.8	0.76	<b>6.5</b>	<b>9.7</b>	<b>3.97</b>	<b>2190.31</b>	<b>7.15</b>	<b>9.62</b>
$\mathcal{T}_3^2$ w/RKS	0.89	0.75	<b>5.25</b> *	<b>8.94</b> *	<b>3.71</b> *	<b>2188.36</b>	<b>6.5</b> *	<b>9.48</b> *
$\mathcal{T}_1^2$ w/TRS <sub>0.6</sub>	0.6	0.78	9.68	11.81	4.52	2195.13	8.1	9.89
$\mathcal{T}_2^2$ w/TRS <sub>0.6</sub>	0.68	0.78	<b>7.01</b>	<b>10.13</b>	<b>3.89</b>	<b>2192.04</b>	<b>7.26</b>	<b>9.68</b>
$\mathcal{T}_3^2$ w/TRS <sub>0.6</sub>	0.66	0.85	<b>5.72</b> *	<b>9.29</b> *	<b>3.69</b> *	<b>2189.93</b>	<b>6.73</b> *	<b>9.55</b> *

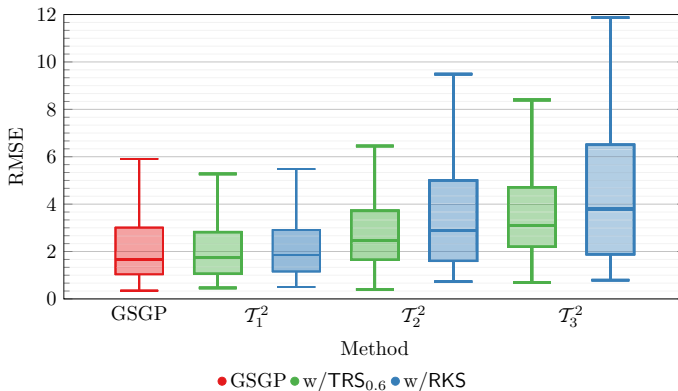
datasets except for the synthetic ones. In general, the vanilla combination  $\mathcal{E}_{\text{cx+mut}}$  still seems to be the one that provides more consistent results.

## 5.2 Best RMSE distribution

The detailed results for each dataset are presented as box-plots of the RMSE on the test set of the best individuals in the last generation. Especially, we present



**Fig. 4** Test results on the V14 dataset. The boxplot shows the distribution of RMSE at the last generations

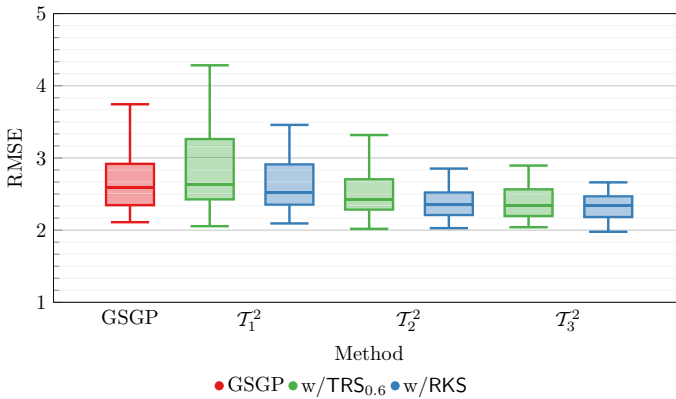


**Fig. 5** Test results on the KJ6 dataset. The boxplot shows the distribution of RMSE at the last generations

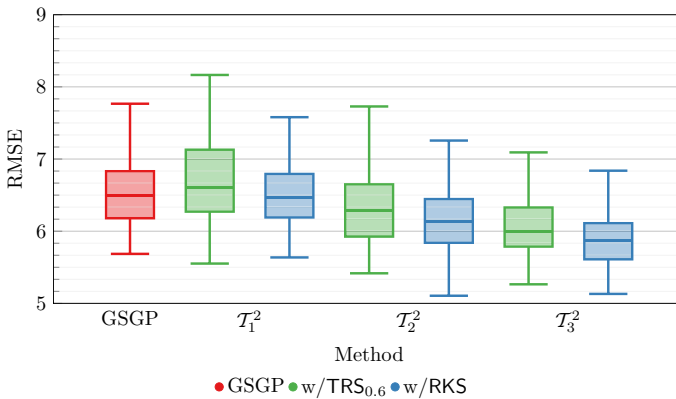
the results for 100 as population size and 1000 generations, with  $\mathcal{E}_{\text{cx+mut}}$  as exploration pipeline.

Figure 4 presents the results for the *Vladislavleva14* (V14) synthetic dataset. The plot shows that cGSGP method is able to provide better results than the baseline for this specific dataset, which benefits from an exploration based on local interactions of neighbourhoods. Particularly, the RKS strategy seems the one that consistently performs better than the baseline.

Figure 5 presents the results for the *Keijzer6* (KJ6) synthetic dataset. In this case, the cGSGP fails to outperform the baseline, meaning that the proposed selection strategies are not beneficial for this specific type of problem when the population size is small. Moreover, it seems that results from cGSGP tend to even become worse if we set a larger radius with the RKS strategy.



**Fig. 6** Test results on the ARF dataset. The boxplot shows the distribution of RMSE in the last generations

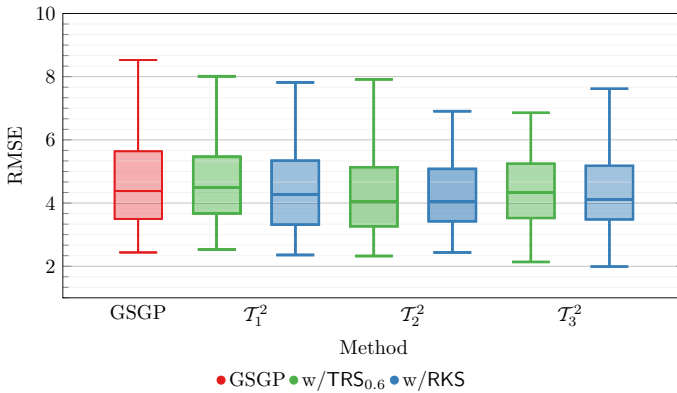


**Fig. 7** Test results on the CNC dataset. The boxplot shows the distribution of RMSE at the last generations

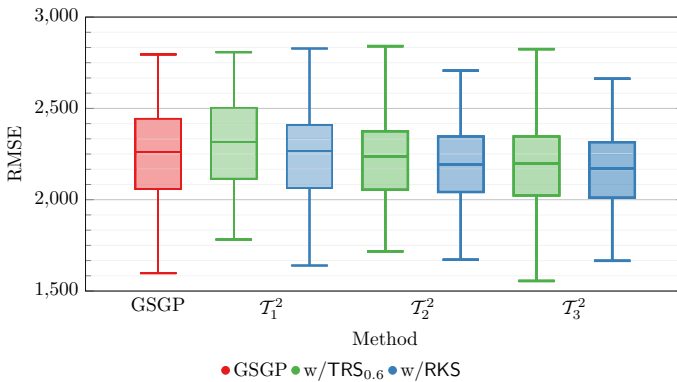
Figure 6 illustrates the results for the *Airfoil* (ARF) dataset. As it is possible to observe, all variants of cGSGP perform better than GSGP, with the two-dimensional torus  $T_1^2$  being not statistically better than GSGP independently from the selection method used. It is also possible to observe that the use of TRS<sub>0.6</sub> produces lower errors, but the difference is only a slight one. The largest neighbourhood, i.e.,  $T_3^2$ , is able to produce results that are better than all other cGSGP methods independently from the selection procedure employed.

The results for the *Concrete* (CNC) dataset presented in Figure 7 show a pattern where different topologies influence the quality of the results. While for  $T_1^2$  the results remain not statistically better than GSGP, it is possible to observe that an increase in the size of neighbourhood corresponds to an improvement in the quality of the solutions independently from the selection method employed.





**Fig. 8** Test results on the SLM dataset. The boxplot shows the distribution of RMSE at the last generations

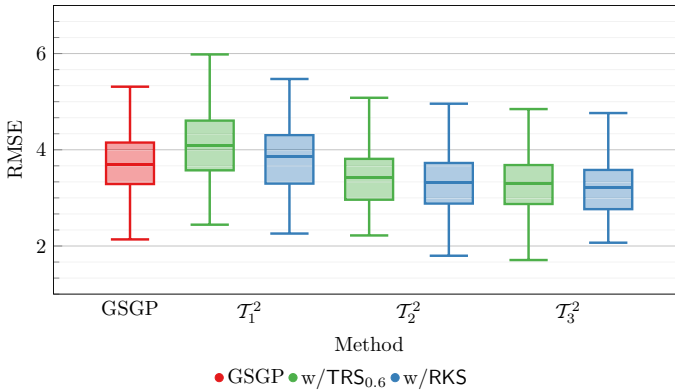


**Fig. 9** Test results on the TXC dataset. The boxplot shows the distribution of RMSE at the last generations

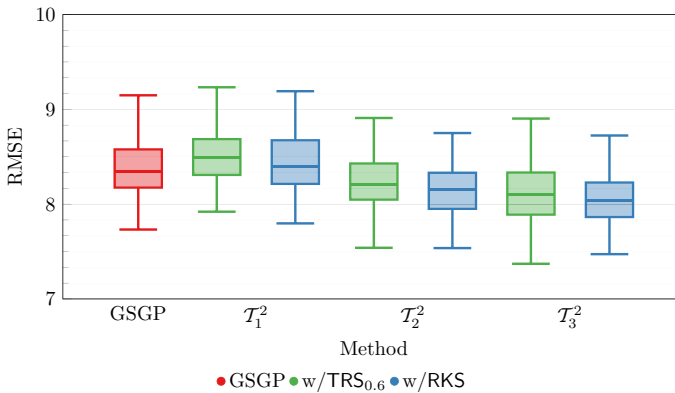
Similarly to the CNC dataset, also for the *Slump* (SLM) dataset, whose results are presented in Figure 8, the performances of GSGP and cGSGP are comparable. In fact, for two different topologies, the performances are not better than GSGP in a statistically significant way.

Figure 9 presents the results for the *Toxicity* (TXC) dataset. As it is possible to observe, there is still a slight increase in the quality of the solution with the increase in the size of the neighbourhood, with the  $T_3^2$  providing the best results also compared to other cGSGP configurations.

The results for the *Yacht* (YCH) dataset are presented in Figure 10 confirming the trend observed in the other datasets. A local-only interaction helps and is able to produce better results than GSGP, with an additional increase in the solutions' quality for larger neighbourhoods.



**Fig. 10** Test results on the YCH dataset. The boxplot shows the distribution of RMSE at the last generations

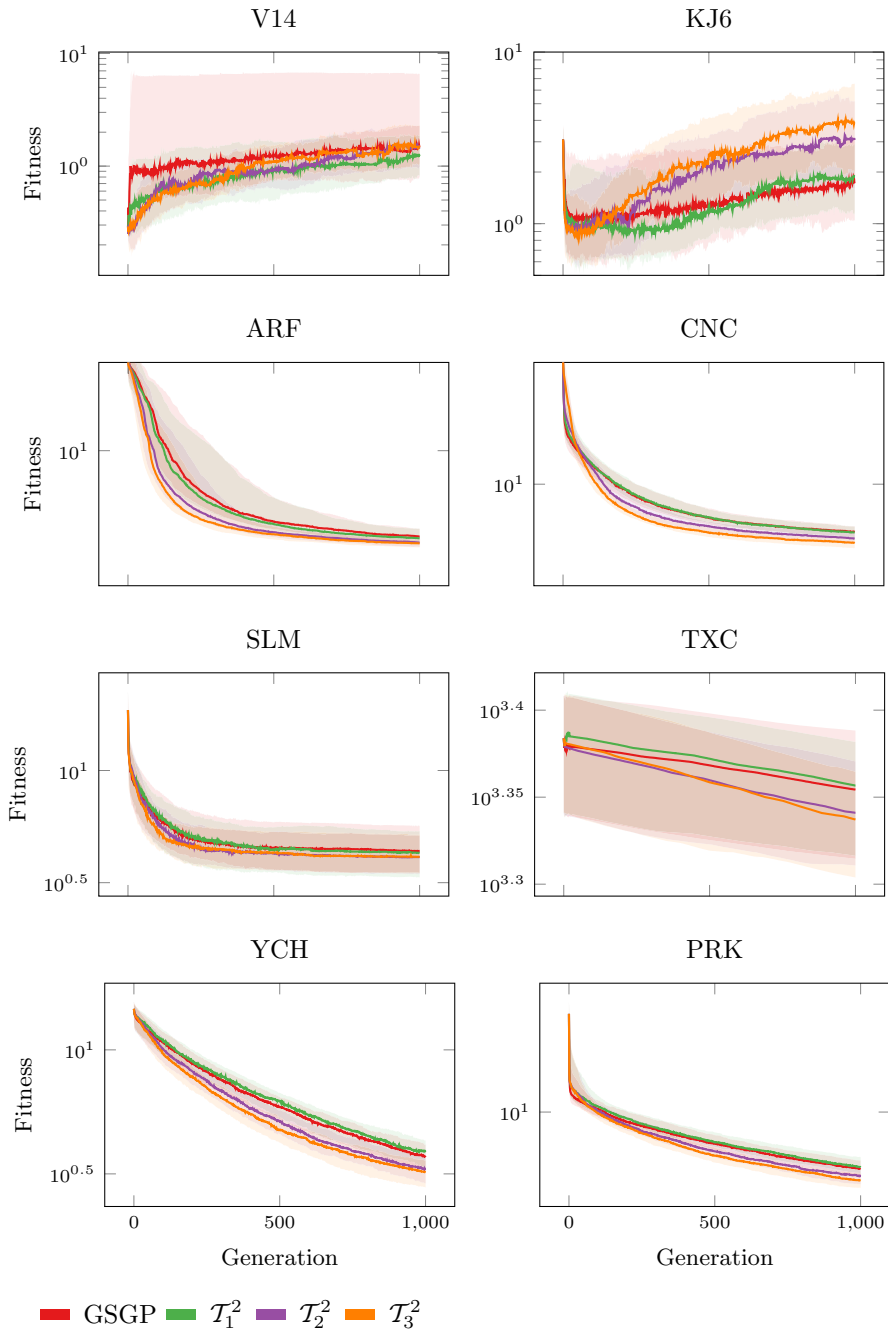


**Fig. 11** Test results on the PRK dataset. The boxplot shows the distribution of RMSE at the last generations

Similarly to many of the other datasets, also for the *Parkinson* (PRK) dataset, whose results are presented in Figure 11, we can observe that, regardless of the selection strategy employed, a cGSGP-based method with a sufficiently large radius is necessary to perform better than GSGP.

### 5.3 Convergence rate analysis

In this section, we analyze the convergence rate of our proposed techniques. For illustrative purposes, the convergence rates of GSGP and cGSGP are depicted in Figure 12. In these plots, both  $\mathcal{E}_{\text{cx+mut}}$  and RKS are employed for a population size of 100 spanning 1000 generations. Specifically, for every generation, we illustrate the fitness evaluated on the test set of the best individual of the corresponding



**Fig. 12** Trend of test RMSE of the best individual. The plot is based on  $\mathcal{E}_{cx+mut}$  and RKS strategy, with 100 as population size and 1000 generations

generation; the shaded area denotes the inter-quartile range. It is evident that, due to the enforcement of elitism, this individual is also the best one discovered during the course of the evolution up to that point.

Line-plots indicate a negative trend for the two synthetic datasets for both baseline and cGSGP, particularly prominent for KJ6. In both scenarios, the test fitness seems to grow with the increasing number of generations. This is especially noticeable for  $\mathcal{T}_2^2$  and  $\mathcal{T}_3^2$  in KJ6. In contrast, within the V14 dataset, the proposed approach consistently outperforms the baseline. However, it appears to plateau after the final generation, suggesting potential over-fitting across all the methods and both synthetic datasets.

For all other datasets, we consistently observe, for all involved methods, either a constantly decreasing trend (TXC, YCH, and PRK) or an abrupt drop in fitness during the initial generations, leading to a plateau in the subsequent ones (ARF, CNC, and SLM). In these situations, GSGP and  $\mathcal{T}_1^2$  display similar behaviors. Yet, cGSGP methods with an extended radius tend to exhibit a more pronounced downward trend.

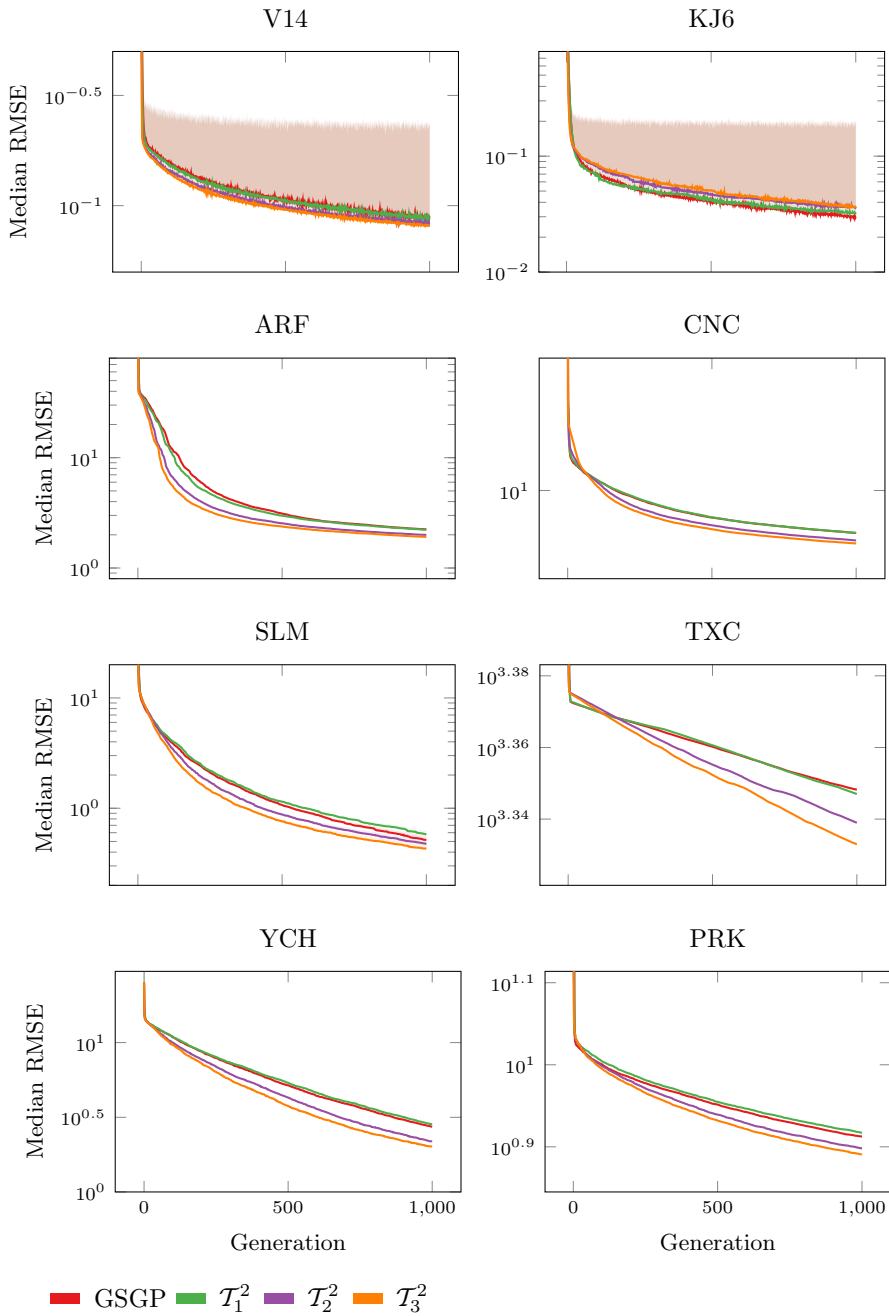
The insights gained from this analysis reinforce our findings from Sect. 5.1. Here, it seems that the evolutionary process gains from a cellular-inspired selection procedure, especially with a sizeable radius. This approach enables the optimization to derive superior results compared to the standard GSGP. Moreover, based on the convergence rates, this benefit appears to hold true regardless of the number of executed generations.

#### 5.4 Population fitness distribution

After assessing the performance of the proposed methods across the generations as regards the best-discovered solutions, we analyze the fitness distribution on the train set of the entire population. To this end, we show the median train fitness in Figure 13, where  $\mathcal{E}_{cx+mut}$  and RKS are employed for a population size of 100 and 1000 generations. Specifically, for each generation, we plot the median of all the fitness values evaluated on the train set by the individuals in the population of the generation at hand (the shaded area represents the inter-quartile range).

For almost all the tested datasets, the median RMSE consistently decreases. This is also true for the two synthetic datasets, which, according to the convergence rates analysis, suffer from over-fitting-related issues. As regards the real-world datasets, the median RMSE on the train set confirms the trend observed in the line-plots of the previous section, with TXC that exhibits a decreasing trend with a negative concavity for  $\mathcal{T}_2^2$  and  $\mathcal{T}_3^2$ . Finally, similarly to what we have already seen in the previous section, we can observe a plateau after a few generations in ARF and CNC.

An analysis based on the train fitness distribution of the population should reveal some insights into the diversity preservation mechanisms that play along with the proposed techniques. However, as we can see, except for the synthetic datasets and occasionally for SLM, there is little or no variability in the train fitness values. Hence, this type of analysis does not reveal useful information on the diversity of the population, which is what we are going to try analyzing in deeper detail in the next section.



**Fig. 13** Trend of median train RMSE of the population. The plot is based on  $\mathcal{E}_{\text{cx+mut}}$  and RKS strategy, with 100 as population size and 1000 generations

## 5.5 Semantic diversity analysis

Evaluating the diversity of the population across the generations is a challenging task, especially in the context of GSGP, where the analysis of the syntactical structure of trees may not be feasible for multiple reasons: (1) if two individuals are *syntactically* different, then any geometric crossover and mutation resulting from them will always be different, (2) more importantly, a tree-representation of a GSGP individual is exponential in size. Hence, in this context, we try to analyze the diversity of the population by introducing a measure that accounts for the *semantic* distance of individuals.

Each individual evaluated on the train set produces a semantic vector, which contains all the train predictions of the individual. It is then possible to compute the Global Moran's I (denoted as  $I$ ) [55, 56] on the semantic vectors of a given population and use it to measure diversity.  $I$  is a measure of spatial autocorrelation [57], assessing similarity or dissimilarity of values located in neighbouring locations and formally defined as

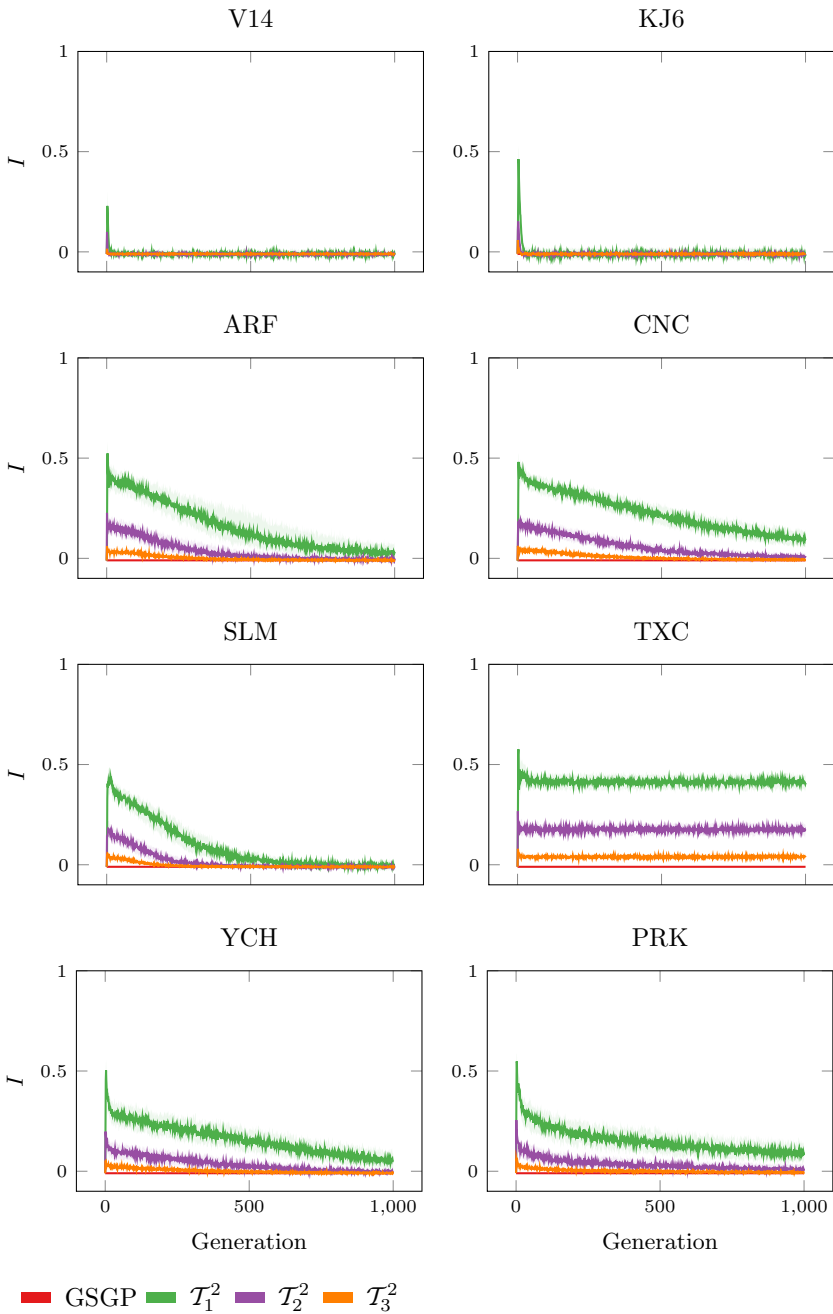
$$I = \frac{N}{W} \frac{\sum_{i=1}^N \sum_{j=1}^N \mathbf{w}_{ij} (\mathbf{x}^i - \bar{\mathbf{x}})(\mathbf{x}^j - \bar{\mathbf{x}})}{\sum_{i=1}^N (\mathbf{x}^i - \bar{\mathbf{x}})^2}$$

where  $N$  is the population size,  $\mathbf{w}_{ij} \in \mathbb{R}$  is the value located at the  $i$ -th row and the  $j$ -th column of the matrix  $\mathbf{w} \in \mathbb{R}^{N \times N}$ ,  $\mathbf{w}_{ii} = 0$  for  $1 \leq i \leq N$ ,  $W = \sum_{i=1}^N \sum_{j=1}^N \mathbf{w}_{ij}$ ,  $\mathbf{x}^i$  is a representation of the  $i$ -th individual in the population as a semantic vector, and  $\bar{\mathbf{x}}$  is the mean of the representations of all the individuals in the population. In our case,  $\mathbf{w}_{ij} = 1$  if  $i \neq j$  and the  $j$ -th individual belongs to the neighbourhood of the  $i$ -th individual (in vanilla GSGP the entire population is a single neighbourhood), otherwise  $\mathbf{w}_{ij} = 0$ .

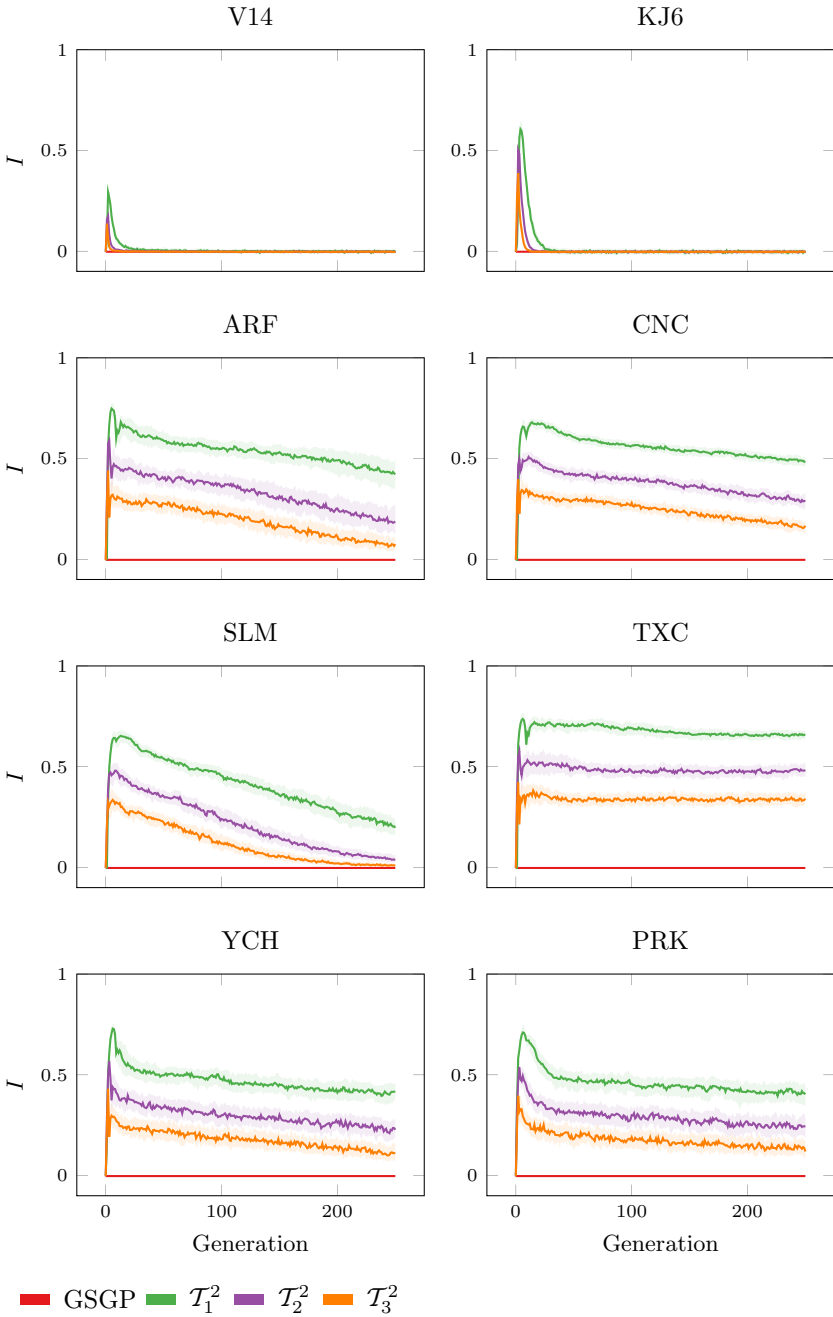
$I$  ranges in the interval  $[-1, 1]$ , where values close to 1 indicate positive spatial autocorrelation, while values close to -1 suggest the presence of negative autocorrelation. Values near 0 highlight a spatial pattern not different from a random phenomenon. A high value means that similar semantic content is spatially clustered in the population, hence preserving diversity between different clusters, while a value near 0 suggests a spatial arrangement of semantic vectors which is not different from a random one.

We show the trend of  $I$  in the following line-plots, where both  $\mathcal{E}_{\text{cx+mut}}$  and RKS are employed; the shaded area denotes the inter-quartile range. In this type of analysis, we test all three combinations of population size and number of generations to better show the relationship between population size and diversity of cGSGP. We remark that considering  $I$  is one possible diversity measure and necessarily represents only a partial view on the topic of semantic diversity for GSGP.

Figure 14 shows the  $I$  trend for 100 as population size. For V14 and KJ6 datasets we observe a flattening towards 0 after the very first generations for the different cGSGP methods, while the baseline is always near 0. In all the other cases, it is evident how the cGSGP methods exhibit higher diversity than the baseline, which consistently shows  $I$  values always close to 0, for a high number of generations. The

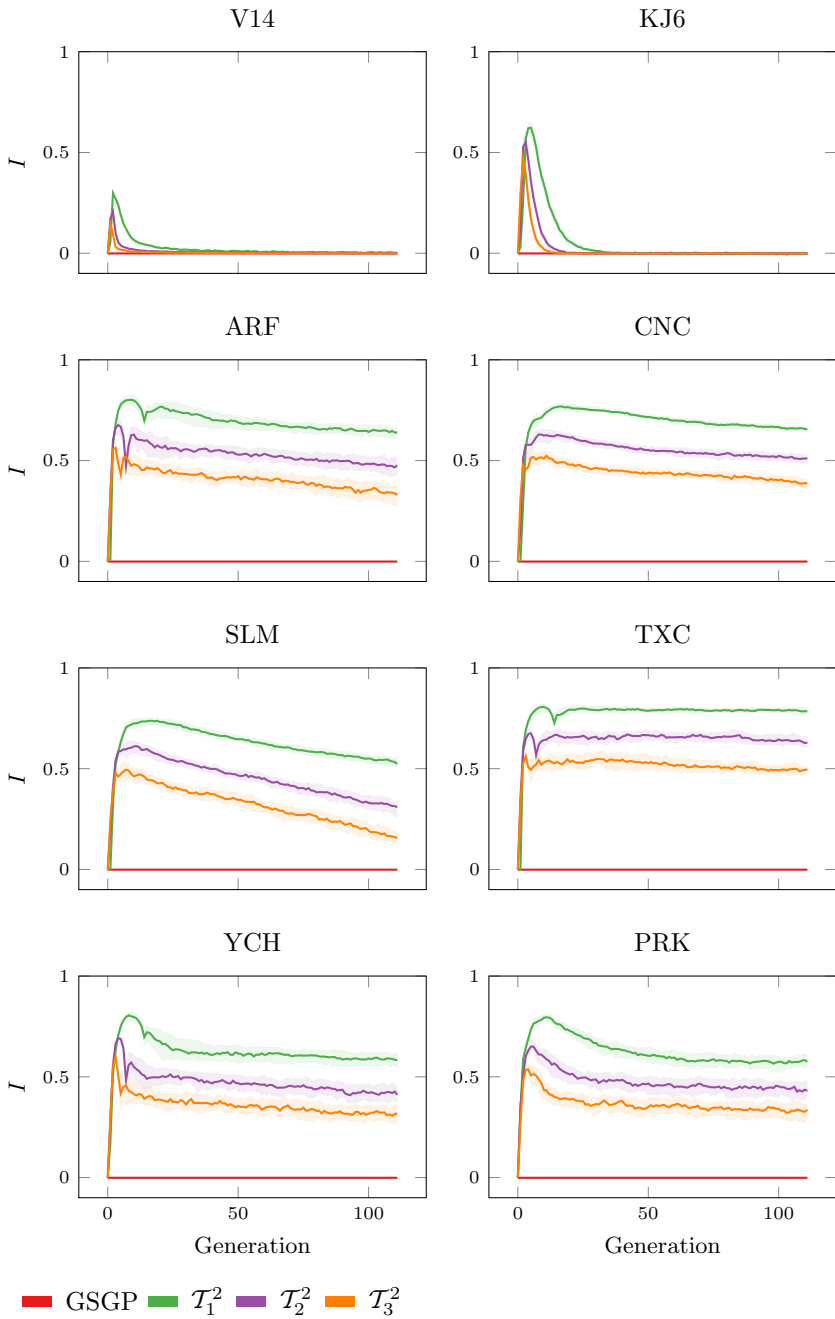


**Fig. 14** Trend of  $I$ . The plot is based on  $\mathcal{E}_{cx+mut}$  and RKS strategy, with 100 as population size and 1000 generations

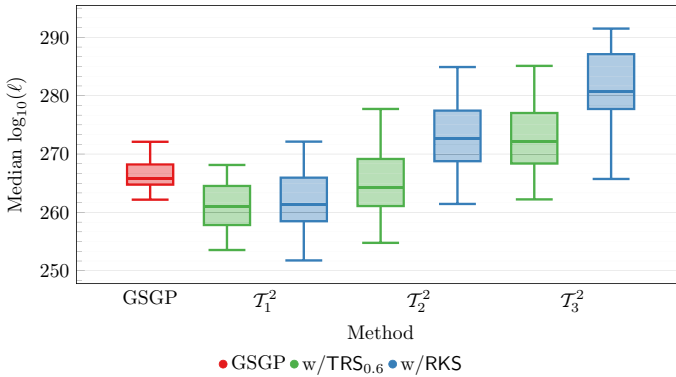


**Fig. 15** Trend of  $I$ . The plot is based on  $\mathcal{E}_{cx+mut}$  and RKS strategy, with 400 as population size and 250 generations





**Fig. 16** Trend of  $I$ . The plot is based on  $\mathcal{E}_{cx+mut}$  and RKS strategy, with 900 as population size and 111 generations



**Fig. 17** Distribution of the median  $\log_{10}(\ell)$  computed on the population of the last generation (100 as population size for 1000 generations) for all the tested datasets and repetitions

diversity decreases as the size of the neighborhood radius increases, with  $\mathcal{T}_1^2$  always showing the highest  $I$ . For all the different cGSGP methods, diversity decreases as generations progress, except for TXC, where it remains almost constant after an initial transitory phase.

Figure 15 shows the  $I$  trend for 400 as population size. As before, for V14 and KJ6, we can observe a fast convergence towards 0 for all cGSGP methods, while standard GSGP exhibits null  $I$  for the entire evolution, just as it happens for all the other datasets.  $\mathcal{T}_1^2$  is still the method with the highest diversity for all the different cases and the trend of diversity over generations is consistent with what was seen above, with TXC maintaining  $I$  values that are almost constant for all methods.

Figure 16 shows  $I$  trend for 900 as population size. Even here, V14 and KJ6 show trajectories which quickly flatten towards 0 for all cGSGP methods. Diversity is always null for the baseline for all datasets. Also, the rest of the analysis is entirely analogous to the previous cases. TXC shows nearly constant trends for all methods, while all the other datasets exhibit a decrease in diversity over generations for all methods. Once again, we can observe a clear increase in diversity as the neighborhood radius decreases.

Notice that  $\mathcal{T}_1^2$  shows the highest diversity in all the cases: local interactions are forced to happen only within a small neighbourhood. This means that the spread of good individuals across the generations will be slower than that obtained with a larger radius, resulting in a higher  $I$ .

In conclusion, the population size does not seem to affect the trend of diversity for the different methods, since we can observe analogous results for all the different sizes. On the contrary, the size of the neighbourhood seems to have a central role in the formation of spatial clusters preserving semantic diversity of individuals.

## 5.6 Solution size analysis

Finally, we provide an analysis on the size of the solutions. As we have already stated in Sect. 3.1, the solution size always increases at every generation, because of the way the GSGP operators are defined. In this section, we try to analyze whether imposing a CA-inspired communication topology over GSGP leads to smaller solutions. This type of analysis is mainly justified by previous research works in which it was demonstrated that a spatial structure may help to reduce program growth [19, 20].

In a tree-based GP context, a common property that measures the solution size is the number of nodes (denoted as  $\ell$ ). In GSGP, this value can easily become extremely large, increasing exponentially with the number of generations. For this reason, in the following, we report the base-10 logarithm in the number of nodes with a focus on its median value in the population.

As expected, the number of nodes increases exponentially w.r.t. the number of generations. In Figure 17 we show the distribution of the median  $\log_{10}(\ell)$  in the last generation for all tested problems and for all repetitions.

Based on the plot,  $\mathcal{T}_1^2$  tends to discover smaller solutions than GSGP, while  $\mathcal{T}_2^2$  and  $\mathcal{T}_3^2$  tend to discover larger solutions than GSGP (except for  $\mathcal{T}_2^2$  with TRS<sub>0.6</sub>). This suggests that imposing a cellular spatial structure characterized by a small neighbourhood actually helps in limiting the program growth by some orders of magnitude. However, differently from other methods, when a cellular structure is imposed on GSGP, there is no clear reduction in the solution sizes for all neighbourhoods. Additionally, by comparing this outcome with the previous analysis on the convergence rates (Sec. 5.3), the increase in median individual size is not hindering the quality of the solutions generated.

## 6 Discussion

Finally, based on the outcome of the experimental session, we try to summarize the main findings to provide proper answers to our research questions:

**RQ1** Is it possible to overcome the issue of converging too fast to small and possibly sub-optimal areas of the search space?

The issue of premature convergence is probably related to the lack of any inherent mechanism of diversity preservation in GSGP. To reduce this problem we imposed a spatial structure to the population, thus allowing local-only communication. This structure can limit the takeover time, allowing the preservation of diversity. Clearly, there is a trade-off between the amount of local communication and the semantics of the individuals that can be generated using only the individuals in a limited neighbourhood. In particular, for any semantic crossover the results are bound to

be in the convex hull of the possible parents, a smaller neighbourhood restricts the part of the space where new individuals can be generated, thus limiting the possible semantics that can be obtained. Hence, new semantics might require more time to be generated, with a decrease in the speed of exploration.

Experimentally, we observed that a small neighbourhood radius provides better diversity among different neighbourhoods at the cost of slowing down the exploration. By increasing the radius, the exploration gets faster at the cost of losing overall diversity. Therefore, a reasonable trade-off should be found to obtain both good diversity and a reasonably fast exploration.

By analyzing the Moran's I coefficient, we observe that cGSGP can provide diversity among different local neighbourhoods in the early stages of the evolution, enabling a more complete exploration of the search space before dominant solutions propagate to the entire population. This implies that a spatial structure such as a cellular-based one is beneficial also in the context of GSGP.

**RQ2** Can cGSGP exploit the presence of local communication to produce better solutions than GSGP?

In the majority of cases, cGSGP-based methods can produce better solutions than GSGP. This is especially true for cGSGP-based methods with two or three as radius applied to real-world datasets. Hence, we assume that our neighbourhood-based selection strategy that exploits local communication is beneficial for the improvement of GSGP. Furthermore, by analyzing convergence rates, this seems to hold regardless of the actual number of generations executed.

## 7 Conclusion

In this paper, we introduce the cellular GSGP (cGSGP), a way to endow a communication topology on GSGP that allows to improve the performance compared to standard GSGP. This seems to be because by leveraging a spatial structure within GSGP we are able to increase the takeover time and reduce the corresponding loss of diversity, limiting GSGP operators to act only on individuals belonging to the same neighbourhood. In particular, in the early stage of the evolution, the population is characterized by low intra-neighbourhood and high inter-neighbourhood diversity, as measured via Moran's I coefficient. While not explored in this work, a cellular structure can be useful from a scalability point of view: local communications allow for easy distribution of the computation among multiple nodes.

In terms of future directions of research, the influence of topology is an important aspect to study. Furthermore, the effect of which local selection to use appears to be important, so this aspect requires further studies. Additional studies should also be performed to understand the propagation of information inside the population to obtain some insights into the interplay between topology and evolution. Finally, it would be interesting to implement cGSGP in a distributed way, taking advantage of the local-only communications that are part of the methods.

**Authors' contributions** The method design is by LM, the experiments were performed by LR and LB and the code implementation was performed by LR and LB. LR and LB contributed to writing the first draft of the manuscript; LM and AdL contributed to revising the manuscript. LM and AdL provided supervision during all phases of the project.

**Funding** Open access funding provided by Università degli Studi di Trieste within the CRUI-CARE Agreement.

## Declarations

**Conflict of interest** The authors have no financial or non-financial interests to disclose.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. J.R. Koza, Genetic programming as a means for programming computers by natural selection. *Stat. Comput.* **4**, 87–112 (1994)
2. A. Moraglio, K. Krawiec, C. Johnson, Geometric semantic genetic programming, vol. 7491 (2012), pp. 21–31. [https://doi.org/10.1007/978-3-642-32937-1\\_3](https://doi.org/10.1007/978-3-642-32937-1_3)
3. M. Castelli, L. Vanneschi, S. Silva, Prediction of high performance concrete strength using genetic programming with geometric semantic genetic operators. *Expert Syst. Appl.* **40**(17), 6856–6862 (2013)
4. M. Castelli, L. Vanneschi, S. Silva, Prediction of the unified Parkinson's disease rating scale assessment using a genetic programming system with geometric semantic genetic operators. *Expert Syst. Appl.* **41**(10), 4608–4616 (2014)
5. Vanneschi, L., Silva, S., Castelli, M., Manzoni, L.: Geometric semantic genetic programming for real life applications, in *Genetic Programming Theory and Practice XI* (2014), pp. 191–209
6. T.P. Pawlak, B. Wieloch, K. Krawiec, Review and comparative analysis of geometric semantic crossovers. *Genet. Program Evolvable Mach.* **16**, 351–386 (2015)
7. J.V. Neumann, Theory of self-reproducing automata. *Math Comput* **21**, 745 (1966)
8. E.F. Codd, *Cellular Automata* (Academic Press, Cambridge, 1968)
9. P. Sarkar, A brief history of cellular automata. *ACM Comput Surv (CSUR)* **32**(1), 80–107 (2000)
10. E. Alba, B. Dorransoro, The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. *IEEE Trans. Evol. Comput.* **9**(2), 126–142 (2005)
11. E. Alba, B. Dorransoro, *Introduction to Cellular Genetic Algorithms* (Springer, Boston, 2008), pp.3–20. [https://doi.org/10.1007/978-0-387-77610-1\\_1](https://doi.org/10.1007/978-0-387-77610-1_1)

12. C. Salto, E. Alba, Cellular genetic algorithms: Understanding the behavior of using neighborhoods. *Appl. Artif. Intell.* **33**(10), 863–880 (2019)
13. J.H. Holland, Genetic algorithms and the optimal allocation of trials. *SIAM J. Comput.* **2**(2), 88–105 (1973)
14. Y. Deng, J. Xiong, Q. Wang, A hybrid cellular genetic algorithm for the traveling salesman problem. *Math. Probl. Eng.* **2021**, 1–16 (2021)
15. T. Murata, M. Gen, Cellular genetic algorithm for multi-objective optimization, in *Proc. of the 4th Asian Fuzzy System Symposium* (Citeseer, 2002), pp. 538–542
16. A.J. Nebro, J.J. Durillo, F. Luna, B. Dorronsoro, E. Alba, Mocell: a cellular genetic algorithm for multiobjective optimization. *Int. J. Intell. Syst.* **24**(7), 726–746 (2009)
17. G. Folino, C. Pizzuti, G. Spezzano, A scalable cellular implementation of parallel genetic programming. *IEEE Trans. Evol. Comput.* **7**(1), 37–53 (2003)
18. W.N. Martin, Island (migration) models: evolutionary algorithms based on punctuated equilibria, in *Handbook of Evolutionary Computation* (1997)
19. P.A. Whigham, G. Dick, Implicitly controlling bloat in genetic programming. *IEEE Trans. Evol. Comput.* **14**(2), 173–190 (2009)
20. G. Dick, P.A. Whigham, Controlling bloat through parsimonious elitist replacement and spatial structure, in *European Conference on Genetic Programming* (Springer, 2013), pp. 13–24
21. Y. Shi, H. Liu, L. Gao, G. Zhang, Cellular particle swarm optimization. *Inf. Sci.* **181**(20), 4460–4493 (2011)
22. M.A. Al-Betar, A.T. Khader, M.A. Awadallah, M.H. Alawan, B. Zaqabeh, Cellular harmony search for optimization problems. *J. Appl. Math.* **2013**, 61–80 (2013)
23. M. Zhang, N. Tian, V. Palade, Z. Ji, Y. Wang, Cellular artificial bee colony algorithm with gaussian distribution. *Inf. Sci.* **462**, 374–401 (2018)
24. A. Della Cioppa, A. Marcelli, P. Napoli, Speciation in evolutionary algorithms: adaptive species discovery, in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation* (2011), pp. 1053–1060
25. L. Trujillo, L. Muñoz, E. Galván-López, S. Silva, neat genetic programming: controlling bloat naturally. *Inf. Sci.* **333**, 21–43 (2016)
26. K.O. Stanley, R. Miikkulainen, Evolving neural networks through augmenting topologies. *Evol. Comput.* **10**(2), 99–127 (2002)
27. P. Juárez-Smith, L. Trujillo, M. García-Valdez, F. Vega, F. Chávez, Local search in speciation-based bloat control for genetic programming. *Genet. Program Evolvable Mach.* **20**, 351–384 (2019)
28. S. Cussat-Blanc, K. Harrington, J. Pollack, Gene regulatory network evolution through augmenting topologies. *IEEE Trans. Evol. Comput.* **19**(6), 823–837 (2015)
29. T.M. Martins, R.F. Neves, Applying genetic algorithms with speciation for optimization of grid template pattern detection in financial markets. *Expert Syst. Appl.* **147**, 113191 (2020)
30. R. Wickman, B. Poudel, T.M. Villarreal, X. Zhang, W. Li, Efficient quality-diversity optimization through diverse quality species, in *Proceedings of the Companion Conference on Genetic and Evolutionary Computation* (2023), pp. 699–702
31. D.E. Goldberg, K. Deb, A comparative analysis of selection schemes used in genetic algorithms, in *Foundations of Genetic Algorithms*, vol. 1 (Elsevier, 1991), pp. 69–93
32. H. Xie, M. Zhang, Impacts of sampling strategies in tournament selection for genetic programming. *Soft. Comput.* **16**, 615–633 (2012)
33. J. Sarma, K. De Jong, An analysis of the effects of neighborhood size and shape on local selection algorithms, in *International Conference on Parallel Problem Solving From Nature* (Springer, 1996), pp. 236–244
34. M. Giacobini, M. Tomassini, A.G. Tettamanzi, E. Alba, Selection intensity in cellular evolutionary algorithms for regular lattices. *IEEE Trans. Evol. Comput.* **9**(5), 489–505 (2005)
35. L. Vanneschi, M. Castelli, S. Silva, A survey of semantic methods in genetic programming. *Genet. Program Evolvable Mach.* **15**, 195–214 (2014)
36. M. Castelli, S. Silva, L. Vanneschi, A c++ framework for geometric semantic genetic programming. *Genet. Program Evolvable Mach.* **16**, 73–81 (2015)
37. L. Vanneschi, M. Castelli, L. Manzoni, S. Silva, A new implementation of geometric semantic gp and its application to problems in pharmacokinetics, in *Genetic Programming: 16th European Conference, EuroGP 2013, Vienna, Austria, April 3-5, 2013. Proceedings 16* (Springer, 2013), pp. 205–216

38. M. Castelli, L. Vanneschi, A. Popovič, Controlling individuals growth in semantic genetic programming through elitist replacement. *Comput. Intell. Neurosci.* **2016**, 42–42 (2016)
39. J.F.B. Martins, L.O.V. Oliveira, L.F. Miranda, F. Casadei, G.L. Pappa, Solving the exponential growth of symbolic regression trees in geometric semantic genetic programming, in *Proceedings of the Genetic and Evolutionary Computation Conference* (2018), pp. 1151–1158
40. D. Koga, K. Ohnishi, Non-generational geometric semantic genetic programming, in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)* (IEEE, 2021), pp. 1–7
41. L. Trujillo, J.M.M. Contreras, D.E. Hernandez, M. Castelli, J.J. Tapia, GSGP-CUDA—A CUDA framework for geometric semantic genetic programming. *SoftwareX* **18**, 101085 (2022)
42. M. Castelli, L. Manzoni, L. Vanneschi, S. Silva, A. Popovič, Self-tuning geometric semantic genetic programming. *Genet. Program Evolvable Mach.* **17**, 55–74 (2016)
43. D.R. White, J. McDermott, M. Castelli, L. Manzoni, B.W. Goldman, G. Kronberger, W. Jaśkowski, U.-M. O’Reilly, S. Luke, Better GP benchmarks: community survey results and proposals. *Genet. Program Evolvable Mach.* **14**, 3–29 (2013)
44. E.J. Vladislavleva, G.F. Smits, D. Hertog, Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. *IEEE Trans. Evol. Comput.* **13**(2), 333–349 (2009). <https://doi.org/10.1109/TEVC.2008.926486>
45. Keijzer, M.: Improving symbolic regression with interval arithmetic and linear scaling, in *European Conference on Genetic Programming*, (Springer, 2003), pp. 70–82
46. M. Streeter, L.A. Becker, Automated discovery of numerical approximation formulae via genetic programming. *Genet. Program Evolvable Mach.* **4**, 255–286 (2003)
47. T.F. Brooks, D.S. Pope, M.A. Marcolini, Airfoil self-noise and prediction. Technical report (1989)
48. L.-C. Yeh, Simulation of concrete slump using neural networks. *Proc. Inst. Civ. Eng. Constr. Mater.* **162**(1), 11–18 (2009)
49. F. Archetti, S. Lanzani, E. Messina, L. Vanneschi, Genetic programming for computational pharmacokinetics in drug discovery and development. *Genet. Program Evolvable Mach.* **8**, 413–432 (2007)
50. I. Ortigosa, R. Lopez, J. Garcia, A neural networks approach to residuary resistance of sailing yachts prediction, in *Proceedings of the International Conference on Marine Engineering MARINE*, vol. 2007 (2007), p. 250
51. J.S. Armstrong, F. Collopy, Error measures for generalizing about forecasting methods: empirical comparisons. *Int. J. Forecast.* **8**(1), 69–80 (1992)
52. F. Wilcoxon, *Individual comparisons by ranking methods* (Springer, New York, 1992)
53. S. Holm, A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* **6**, 65–70 (1979)
54. W.H. Kruskal, W.A. Wallis, Use of ranks in one-criterion variance analysis. *J. Am. Stat. Assoc.* **47**(260), 583–621 (1952)
55. P.A. Moran, Notes on continuous stochastic phenomena. *Biometrika* **37**(1/2), 17–23 (1950)
56. H. Li, C.A. Calder, N. Cressie, Beyond moran’s i: testing for spatial dependence based on the spatial autoregressive model. *Geogr. Anal.* **39**(4), 357–375 (2007)
57. L. Anselin, S. Rey, Properties of tests for spatial dependence in linear regression models. *Geogr. Anal.* **23**(2), 112–131 (1991)

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Authors and Affiliations

Lorenzo Bonin<sup>1</sup> · Luigi Rovito<sup>1</sup> · Andrea De Lorenzo<sup>2</sup> · Luca Manzoni<sup>1</sup>

✉ Luca Manzoni  
lmanzoni@units.it

Lorenzo Bonin  
lorenzo.bonin@phd.units.it

Luigi Rovito  
luigi.rovito@phd.units.it

Andrea De Lorenzo  
andrea.delorenzo@units.it

- <sup>1</sup> Dipartimento di Matematica e Geoscienze, Università degli Studi di Trieste, Via Alfonso Valerio 12/1, 34127 Trieste, TS, Italy
- <sup>2</sup> Dipartimento di Ingegneria e Architettura, Università degli Studi di Trieste, Via Alfonso Valerio 6/1, 34127 Trieste, TS, Italy