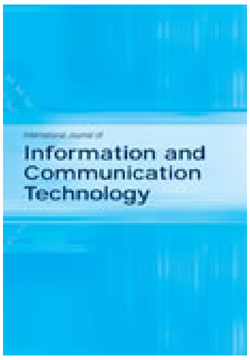


International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

Virtual landscape layout generation using physically constrained particle swarm optimisation

Yu Chen

DOI: 10.1504/IJICT.2025.10073894

Article History:

Received:	19 July 2025
Last revised:	01 September 2025
Accepted:	02 September 2025
Published online:	22 October 2025

Virtual landscape layout generation using physically constrained particle swarm optimisation

Yu Chen

Quanzhou Urban Planning and Design Group Co., Ltd.,
Quanzhou 362000, China
Email: y_chen2008@163.com

Abstract: The creation of virtual landscape layouts is becoming increasingly important for the application of technologies such as augmented reality (AR) and virtual reality (VR). In this paper, a method based on physical constraint particle swarm optimisation (PSO) is proposed to improve the optimisation and computational efficiency of virtual landscape layout generation. By adding physical constraints, the layout design can meet the requirements of real-world applications, and the computational efficiency and optimisation quality are greatly improved. Comparison results with other traditional algorithms show that the physical constraint PSO proposed in this study outperforms other algorithms in terms of compactness (0.92), objective function (0.85), constraint satisfaction (0.95) and feasibility (0.98). It also shows that it can be used in many different situations and is promising in producing virtual landscape layouts for the real world.

Keywords: physical constraints; particle swarm optimisation; PSO; virtual landscape layout generation.

Reference to this paper should be made as follows: Chen, Y. (2025) ‘Virtual landscape layout generation using physically constrained particle swarm optimisation’, *Int. J. Information and Communication Technology*, Vol. 26, No. 38, pp.59–74.

Biographical notes: Yu Chen received his Master’s degree from the Newcastle University in the UK in 2016. He is currently the Vice President of the Landscape Branch of Quanzhou Urban Planning and Design Group Co., Ltd. and a landscape architect. His research interests include landscape design and ecological restoration design.

1 Introduction

With the rapid development of information technology, virtual reality (VR) and augmented reality (AR) technologies are increasingly used in many fields. From entertainment and gaming to education and healthcare, from urban planning to military simulation, the integration of the virtual world with the real world has become particularly important (Al Qattan, 2019). In particular, how to build a high-quality, immersive virtual landscape and how to achieve a seamless integration of the virtual landscape with the real environment have become hot issues in today’s technology research and application development. As the core component of the virtual environment, the virtual landscape layout not only needs to be visually beautiful and functionally practical but also should conform to the physical laws of reality to ensure that there is no sense of incompatibility in the user interaction process. Therefore, the study of virtual landscape layout generation technology has become an important topic in the fields of computer graphics, artificial intelligence and human-computer interaction (Ma et al., 2022).

However, the generation of virtual landscape layout is not a simple task. It not only involves a large number of scene elements, but also needs to consider the spatial relationships, physical properties and dynamic interactions between these elements. Traditional landscape design often relies on manual operation, which is inefficient and difficult to adapt to the needs of complex, large-scale virtual environments (Korkut and Surer, 2023). Therefore, how to generate aesthetic and physical virtual landscape layouts through automated algorithms has become an urgent problem in academia and industry.

To this end, this paper proposes a virtual landscape layout generation method based on physical constraints particle swarm optimisation (PSO). These physical constraints include collision detection between objects, gravity effects, and the reasonableness of spatial layout, aiming at avoiding the layout failure or unachievability problems commonly found in traditional methods. Compared with the traditional PSO algorithms, the physical constraint-based PSO method can better balance the optimisation efficiency and the physical constraints, so as to ensure the quality of landscape layout while enhancing the practical application value of the algorithm.

2 Relevant work

2.1 Virtual landscape layout generation

Early virtual landscape generation methods mostly relied on manual design and manual layout (Jia, 2022). Although this method can ensure the artistry of landscape design, it is difficult to adapt to the generation needs of large-scale and complex scenes due to the limitations of manual operation. Therefore, algorithms for automated generation of virtual landscape layouts have gradually become the focus of research. This type of method can ensure the aesthetics and reasonableness of the layout to a certain extent, but its flexibility is poor, and it is difficult to adapt to diversified design needs, especially when facing complex environments, it is difficult to generate layouts with a high degree of freedom.

With the development of computational intelligence technology, especially the rise of optimisation algorithms, researchers have begun to focus on how to automatically generate virtual landscape layouts through algorithms. Genetic algorithm (GA) and evolutionary algorithms (EA), as the early optimisation methods widely used in this field, search for the best solution by simulating the process of natural selection and evolution (Slowik and Kwasnicka, 2020). GA continually optimises the quality of the layout through selection, crossover, and mutation, and is especially suitable for solving the multi-objective optimisation (MOO) problems with complex constraints. EA, on the other hand, is much more extensive, and includes a variety of optimisation algorithms such as evolutionary strategies (ES), differential evolution (DE) and many other variants (Molina et al., 2018). EA methods usually provide more diverse selection strategies and show better global search capabilities in some scenarios. Through these algorithms, researchers can search for better quality layout solutions in high-dimensional design spaces.

But these evolution-based algorithms still have some problems. GA and EA can help with the layout problem of virtual landscapes to some extent, but they do not always find the best solutions quickly. When faced with more complicated physical restrictions and design goals that are multidimensional, they tend to settle on local optimal solutions. So, figuring out how to add physical limitations to these optimisation approaches so that the layout not only looks good but also works in real life has become an important area of study.

Also, deep learning (DL) can make layouts that look realistic, but because there are no physical limits, the layouts it makes may not always work in the actual world, especially in VR and AR, where physical reasonableness is very crucial.

In the past few years, layout generation approaches based on reinforcement learning (RL) have also started to get more attention than older methods. RL mimics how smart bodies interact with their surroundings. It can also change the layout design on the fly through the reward system, which makes virtual landscapes more intelligent. RL approaches work best for solving layout problems in situations that change all the time (Kaven et al., 2024). They can update the layout scheme in real-time based on changes in the environment, which makes them very adaptable and flexible.

PSO is an optimisation technique that mimics how groups of people work together in nature. It is also used to make virtual landscapes. By mimicking how birds search for food or fish swim, PSO can successfully explore the design space and find a better solution. PSO has done a lot of good work on some virtual landscape layout generation, but when it comes to dealing with complex physical limitations, typical PSO algorithms frequently cannot handle physical conflicts in the layout. So, figuring out how to use PSO with physical restrictions to make layout creation more realistic has become a major area of research.

Even if a lot of progress has been made in creating virtual landscapes, there are still a lot of problems that need to be solved. First, present algorithms are still slow and hard to use when working with high-dimensional, complicated scenes. Second, figuring out how to use optimisation algorithms with physical constraints is still a big problem in current research. Also, DL approaches are better at generating quality, but they still do not handle physical rationality well enough. It is still vital for future study to figure out how to combine different methodologies, notably how to use the physics engine and optimisation algorithms together to make virtual landscape layouts that are more realistic, reasonable, and efficient.

2.2 Physical constraints

Physical constraints are limits in real-life problems that are set by the basic laws of nature. In domains like engineering, science, and computer modelling, physical constraints are commonly employed as an important part to make sure that a system or design solution can follow the rules and limits of reality (Rude et al., 2018). In many cases, physical constraints not only decide if a design is possible, but they also have a direct effect on how well it works in the end.

The basic rules of physics are what make up the heart of physical restrictions. Laws in mechanics, thermodynamics, electromagnetism, fluid dynamics, and other sciences are some of the most frequent physical limitations. For instance, the laws of thermodynamics that say energy cannot be created or destroyed, the laws of Newtonian physics that say force, acceleration, and mass are related, and the equations that describe how fluids move in fluid dynamics are all examples of the physical limits of a system in design and optimisation. These rules of physics provide limits on how the system can behave and what states it can be in. They also set limits on how far it can go.

In order to handle physical constraints in the optimisation process, some constraint handling techniques are usually employed. Some common methods are constraint penalty method, Lagrange multiplier method, etc. The basic idea of the constraint penalty method is to add a penalty term to the objective function that is related to the degree of constraint violation as a way of guiding the optimisation process to avoid regions where the constraints are not satisfied. The Lagrange multiplier method integrates the original constraints into the optimisation model by introducing additional variables, (i.e., Lagrange multipliers) so that these constraints are naturally satisfied during the solution process (Senkevich et al., 2021). These methods have been widely applied to solve complex optimisation tasks with physical constraints in several real-world problems such as engineering design, robot path planning, traffic scheduling, and so on.

Because of this, more and more researchers are looking into how to combine physical constraints with advanced optimisation techniques like GA, DL, and others (Bian and Priyadarshi, 2024). They want to be able to increase the system's overall performance while still making sure that the physical constraints are reasonable.

Physical constraints are not just limits in the design process; they are also important parts of optimisation and simulation. One of the most important problems in modern engineering design, computer simulation, and interdisciplinary optimisation is how to deal with these restrictions in a way that makes sure the final solution is possible and works well. As science and technology keep becoming better, the use of physical restrictions is growing and giving designers in many fields a lot of help.

2.3 Particle swarm optimisation

PSO is a group intelligence optimisation method that models how groups behave which was inspired by how groups of animals behave in nature, including how birds feed and fish swim notably how people work together and share knowledge (Khalid et al., 2023). PSO finds the best solution to a problem by simulating how particles might move around the solution space.

During the search process, the particle refers to two aspects of information: the optimal solutions it has found before, called individual experience; and the optimal solutions found by all particles in the whole group, called social experience. In addition, there is a factor known as inertia, which is used to control the extent to which the particle maintains its original state of motion and affects its ability to explore new regions.

By combining individual experience, social experience and inertia, the particle gradually adjusts its position and eventually hopes to converge to the global optimal solution (Duan et al., 2022). This method has been widely used in the solution of various optimisation problems.

PSO often stalls at local optima when tackling multi-modal or highly nonlinear problems. To mitigate this, researchers have integrated local search and other enhancements that strengthen both global exploration and local exploitation, thereby improving convergence speed and stability across diverse optimisation tasks.

Such a mechanism allows the particles to explore new regions to find better solutions, but also to use the existing information to accelerate the convergence speed and gradually approach the global optimal solution. Since PSO is an intelligent optimisation method based on group collaboration, the particles advance the search process together by sharing information, rather than searching for answers in isolation (Gad, 2022). This method not only enhances the global search capability but also avoids the problem of falling into local optimum.

Unlike traditional optimisation methods, (e.g., gradient descent) that need to compute derivatives, PSO does not need gradient information of the objective function, so it is more advantageous in dealing with high-dimensional, nonlinear or even discontinuous problems, and easier to be applied in complex real-world problems.

Despite the many advantages of PSO, certain challenges still exist, such as how to effectively avoid local optima and how to improve the convergence speed of the algorithm. In order to overcome these challenges, researchers have carried out a lot of optimisation and improvement of PSO by combining other optimisation methods such as GA, simulated annealing (SA) algorithm, etc. and proposed a hybrid optimisation algorithm to obtain better results in more complex optimisation problems (Elgamal et al., 2022).

As computing technology and algorithms have gotten better, PSO has been used in more areas as an optimisation technique. PSO has a lot of promises and might be used in a lot of different ways, both in research and in business.

3 Methodology

There are six essential parts to the virtual landscape layout generating approach based on physical constraint granular PSO that this study suggests. See Figure 1. First, the layout of the virtual landscape is defined so that it can be optimised later. Next, physical constraint modelling makes sure that the arrangement follows the rules of physics in the real world. The objective function and fitness evaluation then check the quality of the layout solution and help the optimisation process move further. The PSO algorithm looks for the best solution, and the particle position and velocity update module keeps changing the state of the particles to speed up convergence. Finally, the result visualisation and validation module shows the optimisation outcomes and checks them using 3D rendering or VR/AR methods.

3.1 Virtual landscape layout representation

When making a virtual landscape plan, the parts of the virtual landscape need to be shown in a way that makes sense first. A virtual landscape is a complicated system made up of many different landscape pieces, such mountains, rivers, houses, woods, and so on. The quality of the layout depends on the positions, shapes, and connections of these parts. We think of the virtual landscape layout as a high-dimensional space made up of many landscape elements. Each element's position can be shown by a vector, and its shape or other characteristics can be shown by other parameters (Buswell et al., 2018).

Let's say that the virtual landscape has n elements and that x_i can show the position of landscape element i:

$$x_i = (x_{i1}, x_{i2}, ..., x_{in}) \tag{1}$$

where x_{ij} denotes the coordinates of element i in the j^{th} dimension. Therefore, the whole layout can be represented as a variable X:

$$X = (x_1, x_2, ..., x_n) \tag{2}$$

where x_i is the position vector of the i^{th} element. Assuming that the distance between landscape elements i and j is d_{ij}, the relative position constraint between them can be expressed by the following equation:

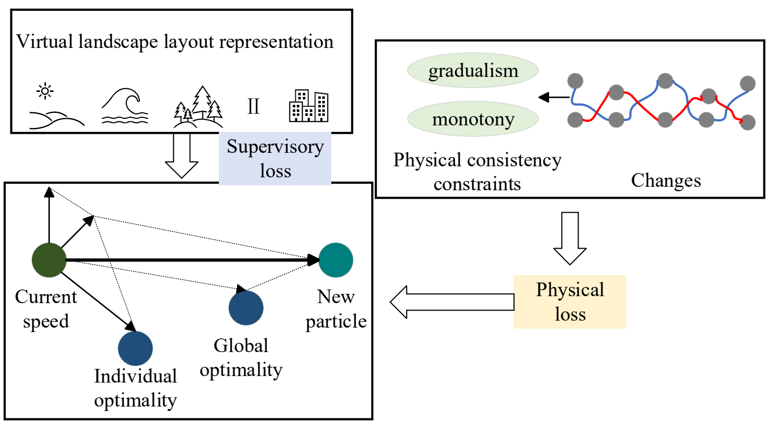
$$d_{ij} = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{in} - x_{jn})^2} \tag{3}$$

The distance d_{ij} between landscape elements should be greater than or equal to a predetermined minimum value $D_{\min}$, i.e.:

$$d_{ij} \geq D_{\min}, \quad \forall i, j \in \{1, 2, \ldots, n\} \tag{4}$$

The methods above can make sure that the elements in the virtual landscape layout are spread out in a way that makes sense and that they are in the right places.

Figure 1 Virtual landscape layout generation method (see online version for colours)



3.2 Physical constraint modelling

In virtual landscape layouts, elements not only have locations but also specific shapes and volumes, which make collision constraints another key factor (Yasin et al., 2020). Specifically, collision constraints are expressed through the following conditions:

$$S_i \cap S_j = \varnothing \tag{5}$$

This means that the spatial areas of any two landscape features must not cross each other and must be separate.

In addition, gravity constraints are a non-negligible factor in virtual landscape layout. The mass m_i of each element i and the gravitational acceleration g_i of the environment where it is located will determine the force F_i on the element. In particular, the following equation can be used to show the gravity constraint:

$$F_i = m_i \cdot g_i \leq F_{\max} \tag{6}$$

This limit makes sure that every part of the layout is within its physical carrying capability and follows the basic rules of how force works on things in nature.

In short, physical restrictions are what make virtual landscape layout generation possible. They make sure that the layout fits the design goals and is also realistic in the real world. These limits are what the optimisation process is based on. The PSO algorithm may then optimise within the limits of following the physical principles, which leads to stable and realistic virtual landscape layouts.

3.3 Objective function design and fitness evaluation

Objective function design and fitness assessment are important parts of the optimisation process for generating virtual landscape layouts. They help figure out how well the optimisation worked. The optimisation target is set by the objective function, and the fitness evaluation is used to rate how well the present layout solution is.

Usually, the objective function has both an aesthetic goal and a goal for physical rationality. The aesthetic goal is to make sure that the landscape features are arranged in a way that is both balanced and appealing, by looking at things like the distance and symmetric between them. To make sure that the layout is physically possible, physical reasonableness objectives include a number of limits, such as the minimum distance between pieces, collision limits, and gravity limits. For instance, if the distance limits are broken or collisions happen, penalty terms will be added to the objective function to make sure the optimisation algorithm does not get confused (Lemley and Casey, 2019). We can write the objective function as:

$$f(x) = w_1 \cdot f_{\text{aesthetic}}(x) + w_2 \cdot f_{\text{physical}}(x) \tag{7}$$

where $f_{\text{aesthetic}}(x)$ is the aesthetic objective, $f_{\text{physical}}(x)$ is the physical constraint goal, w_1 and w_2 are the appropriate weighting coefficients. We can find a balance between the importance of aesthetic and physical limitations by changing the weight coefficients.

The fitness evaluation is based on the value of the objective function. The closer the value of the objective function is to zero, the closer the solution is to being the best one. The evaluation method looks more than just how good something looks and how well it works; it also checks how well it meets physical limitations. Defaults cause fitness to go down, which helps the algorithm identify the best solution one step at a time.

In conclusion, the design of the objective function and the fitness assessment make it evident how to optimise the process of creating virtual landscape layouts. This helps the algorithm meet the design goals while also making sure that the layout is physically possible.

3.4 PSO algorithm implementation

The PSO method can be used to create virtual landscape layouts that solve the multi-objective problem by mimicking the collaborative search of particles in the solution space. The main principle behind PSO is to identify the ideal layout for the design by getting closer to the global optimal solution through the movement of the particle population (Elbes et al., 2019).

The PSO update rules are carried out through the following formula:

$$v_i^{k+1} = w \cdot v_i^k + c_1 \cdot r_1 \cdot \left(p_i - x_i^k\right) + c_2 \cdot r_2 \cdot \left(g - x_i^k\right) \tag{8}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{9}$$

where v_i^k and x_i^k are the speed and position of particle i at the k^{th} iteration, respectively; p_i and g are the best position of the particle itself and the best position of the whole population, respectively; r_1 and r_2 are random numbers; w is the inertia weight; c_1 and c_2 are the learning factors.

To balance the aesthetic and physical objectives, PSO updates the fitness assessment of the particles in the following way:

$$f_{\text{total}}(x) = f_{\text{aesthetic}}(x) + f_{\text{physical}}(x) + P(x) \tag{10}$$

where $f_{\text{aesthetic}}(x)$ denotes the aesthetic goal component, $f_{\text{physical}}(x)$ denotes the physical constraint component, and $P(x)$ is the penalty term associated with constraint violations.

3.5 Particle position and velocity updates

To help the particles stay away from areas that do not fit the physical limits throughout the search process, the following penalty function is used to figure out the distances between the particles and give them the right penalty:

$$P_{ij} = \theta\left(d_{\min} - d_{ij}\right) \quad \text{where } d_{ij} = \left\|x_i - x_j\right\| \tag{11}$$

where d_{ij} denotes the Euclidean distance between particle i and particle j, and $\theta(x)$ is a threshold function defined as:

$$\theta(x) = \begin{cases} 0 & \text{if } x \geq 0 \\ |x| & \text{if } x < 0 \end{cases} \tag{12}$$

When $d_{ij} < d_{\min}$, θ gives a positive value, which punishes particle i and particle j for colliding or becoming too close to one other. Adding the penalty term can stop particles from making layouts that are not physically possible throughout the updating process, and make sure that the particles stay within the compliance region.

Also, in real-world situations, the location update of particles must be limited by spatial bounds. We use boundary processing to make sure that the particles always stay within the effective region of the space (Tafuni et al., 2018). This keeps the particles from going beyond the effective spatial range. If a particle's position goes over the set boundary ranges $x_{\min}$ and $x_{\max}$, its position will be fixed to:

$$x_i = \min\left(\max\left(x_i, x_{\min}\right), x_{\max}\right) \tag{13}$$

This always keeps the particles in the effective solution space and stops the boundary crossing effect from happening throughout the search.

The PSO method can locate the best layout solution that fits the physical limitations in a stable and efficient way because of its dynamic search approach and the appropriate use of constraints. These mechanisms not only make the algorithm work better, but they also make sure that the virtual landscape production is of high quality and makes sense.

3.6 Result visualisation and validation

Result visualisation and validation are highly crucial parts of making a virtual landscape plan. Not only does it assist researchers see the optimisation results, but it also helps them judge the quality of the layouts that were made and the algorithms' performance. Since making virtual landscape layouts involves many goals (like aesthetics, physical limitations, spatial distribution, etc.), good visualisation tools and validation methods can show the pros and cons of the layout solutions, which can then be used to make them even better.

The first thing is that the result visualisation reveals the location, relative relationship, and distribution of the elements in the layout in a way that makes sense by graphically showing the optimised layout. There are frequently a lot of various kinds of objects in a virtual landscape layout, and how these elements are arranged in space has a big effect on how the final layout looks. So, we can easily see how each layout piece is arranged and where they are in relation to one other using 2D or 3D visualisation tools. We can use different colours, sizes, and shapes to tell different types of elements apart, which makes the layout easier to understand and more attractive.

For instance, 3D visualisation can demonstrate how layout elements are related to each other in three dimensions, which can assist figure out how far apart they are, how much they overlap, and how rational the layout is between them. Also, users may zoom in, rotate, and see intricacies in the layout through dynamic interactive visualisation to get an even better look at the layout (Liu, 2025). This visualisation not only shows how the layout looks, but it also shows possible problems with the layout, including too many people in one place, elements colliding, or space being used in an illogical way.

On the other hand, constraint verification makes assurance that the optimised arrangement can actually be built. The algorithm keeps an eye on the distances between layout pieces, overlaps, and relationships with borders as it is making the layout. The system lowers the fitness value of the solution through a penalty mechanism if the layout it creates breaks any physical rules, like the distance between elements being less than the minimum requirement or elements overlapping.

In summary, result visualisation and validation play a crucial role in the generation of virtual landscape layouts. Through effective visualisation methods, researchers can intuitively understand the optimisation process and outcomes of the layout. Meanwhile, quantitative validation methods ensure that the generated layouts not only meet design objectives but also satisfy all physical constraints.

4 Experiments and results

4.1 Experimental data

To test the virtual landscape layout generating approach based on physical constraint PSO suggested in this research, the authors create their own virtual landscape layout dataset. The dataset is made just for the job of creating and improving virtual landscape layouts. It includes information about layout elements, physical constraints, and objective functions. The dataset that the researchers built themselves makes sure that the data is accurate and useful, plus it gives them room to change things for experiments.

There are many types of layout items in the collection, like buildings, green spaces, roads, water bodies, and so on. The placement of each piece must obey certain physical rules. We create the dataset by simulating a 10 km × 10 km city with distinct functional blocks, like residential regions, commercial areas, parks, green spaces, and areas with public utilities. We also define physical limits, including the minimum distance between pieces and the distribution of the road network according to urban planning standards, to make sure the layout is fair.

Table 1 shows the full details of the dataset:

Table 1 Details of the custom-built dataset for virtual landscape layout generation

Data item	*Description*
Element types	Includes buildings, green spaces, roads, water bodies, and other layout elements, simulating various urban planning components.
Number of elements	The dataset contains a total of 1,000 layout elements (including buildings, green spaces, roads, etc.).
Layout area	The layout area is set to a 10 km × 10 km urban region, which contains different types of land plots and functional zones.
Element sizes	Building sizes range from 20 m × 20 m to 150 m × 150 m, while green spaces range from 100 m^2 to 5,000 m^2.
Physical constraints	A minimum spacing of 20 m is maintained between each layout element, ensuring no overlap.
Road network	Road spacing is set to 50 m, and the road network follows specific planning rules such as road width and traffic flow considerations.
Green space distribution	Green space occupies 30% of the layout, with reasonable spacing between buildings, roads, and other elements.
Objective function	The objective function combines aesthetic evaluations (such as layout compactness, visual appeal) with functional aspects (such as traffic accessibility and green space coverage).
Constraint conditions	In addition to the minimum spacing, elements cannot exceed the area boundaries, and the total area of each layout element must not exceed 50,000 m^2.
Data source	The dataset is generated via simulation, adjusting layout elements based on urban planning rules and standards.

This study can make sure that the layout aspects in the experiments work well with the physical limits and that the data can be easily changed to fit varied experimental needs because it built its own dataset.

4.2 Experimental setup

The computer's graphics processing unit (GPU) is an NVIDIA RTX 3060 with 12 GB of video memory, which is capable of accelerating parallel data computation. To ensure efficient data reading and writing during experiments, the computer is also equipped with 32 GB of RAM and a solid-state drive (SSD) (Do et al., 2021). These hardware configurations provide sufficient computational resources for the PSO algorithm to support efficient iterative computation on large-scale datasets, guaranteeing a smooth layout generation process.

In terms of software, the experimental environment adopts Windows 10 operating system. The programming language was chosen to be Python due to its powerful scientific computing libraries, especially NumPy, SciPy and Pandas, which provide important support in data processing and algorithm implementation (Castro et al., 2023).

The experiments use a self-built dataset comprising 1,000 layout elements within a city region; each element spans defined size limits and is subject to physical constraints and objective-function optimisation. In the experiment, the main parameters of the PSO algorithm include the number of particles, the maximum number of iterations, the learning factor, and the inertia weights, etc. which are reasonably set according to the experimental requirements. In this experiment, the number of particles is set to 50, and the maximum number of iterations is 1,000 to ensure that iterations can be sufficiently iterated so as to achieve the expected optimisation effect.

4.3 Experimental procedure

Experiment 1 aims to evaluate the optimisation effectiveness of the PSO algorithm with the introduction of physical constraints in the generation of virtual landscape layouts. For this purpose, it was compared with four common optimisation algorithms: the GA, the SA, the ant colony algorithm (ACO), and the enhanced particle swarm optimisation (EPSO) algorithm. Each of these algorithms represents a different optimisation idea, covering classical ES, local search methods, population intelligence methods and improved versions of PSO. The comparison provides a clearer understanding of the performance advantages of physically constrained PSO for landscape layout tasks.

In order to measure the performance of each algorithm, we mainly refer to four evaluation metrics: layout compactness, objective function value, physical constraint satisfaction, and overall feasibility. Among them, layout compactness is derived by calculating the ratio of the average distance between layout elements to the total area, which is used to reflect whether the distribution of elements is centralised and reasonable; the objective function value takes into account aesthetic and functionality, such as green space coverage and accessibility; physical constraint satisfaction reflects whether the layout complies with the actual constraints such as the minimum spacing and the non-overlapping of the elements, and the proportion of constraint violations is used as the assessment; and finally, feasibility is used to judge whether the layout falls entirely within the specified area and ensures that no overlapping of elements occurs, measured using the proportion of feasible solutions. All metrics are normalised to ensure fair comparisons between different algorithms.

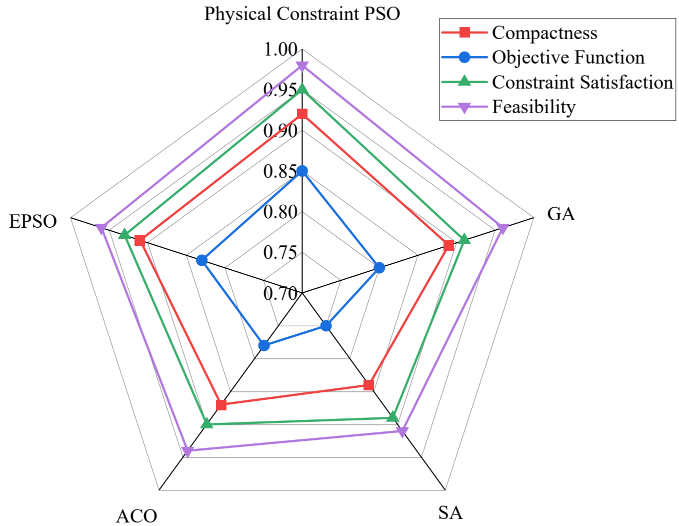
Figure 2 demonstrates the specific performance of physically constrained PSO with the other four algorithms on each of the above metrics.

In terms of the performance of layout compactness, physical constraint PSO scores 0.92, which is better than other algorithms. This indicates that it is more effective in controlling the spacing of the elements and improving the efficiency of space utilisation, and the overall layout is more compact and reasonable. In contrast, GA scored 0.89, which is a fair performance but slightly weaker in terms of layout density and spatial integration.

In terms of objective function values, the physically constrained PSO still leads with a score of 0.85. This indicates that the physically constrained PSO is more advantageous in dealing with multi-objective optimisation problems. While GA scored 0.80 and ACO

0.78, both of them are not as good as physical constraint PSO in this aspect in a comprehensive way.

Figure 2 Comparison of performance metrics for physical constraint PSO and other algorithms (see online version for colours)



In terms of physical constraint satisfaction, physical constraint PSO scored as high as 0.95, which shows its good ability to satisfy the practical requirements such as minimum spacing and avoiding the overlapping of elements, etc. GA and ACO scored 0.91 and 0.90, which can basically also satisfy most of the constraints, but with slightly less stability. SA has the lowest score of 0.89, indicating that it is more likely to violate constraints when dealing with physical constraints.

The physical constraint PSO scores 0.98 in terms of feasibility. This shows that the algorithm can make sure that all the parts of the plan satisfy the real space needs, stay inside the set region, and do not overlap. EPSO, on the other hand, got a score of 0.96, which is quite close to its performance. This means that it is likewise very feasible. Other algorithms, like SA, have a lower score of 0.91, which means that the layouts they make are not always spatially correct.

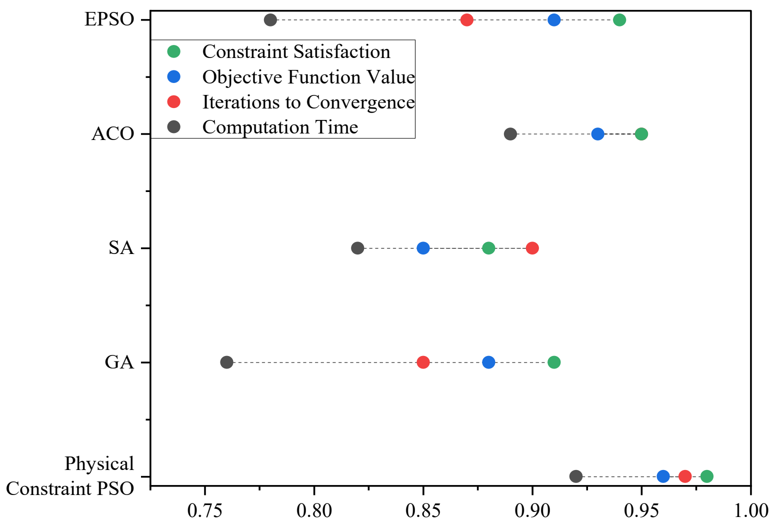
The goal of the second experiment is to see how well the physically restricted PSO algorithm works compared to other standard optimisation methods in terms of how quickly it converges and how efficiently it uses computing power. We picked GA, SA, ACO, and EPSO as the algorithms to compare to get a full picture of the pros and downsides of physically restricted PSO based on how well these algorithms work.

We look at four main metrics to see how well different algorithms optimise the identical virtual landscape layout generation problem: computation time, number of iterations to convergence, objective function value, and constraint satisfaction. All the measurements are normalised using standard values (between 0 and 1) to make sure that the experimental data is consistent and fair. This lets the results of different methods be compared using the same criterion. Computation time shows how long it takes to run the algorithm; shorter computation time means higher computation efficiency. The number of convergence iterations shows how many time steps the algorithm needs to converge;

fewer iterations to convergence means faster convergence. The objective function value shows how good the optimisation result is; a smaller value means a better optimisation effect. Finally, the constraint satisfaction shows how well the algorithm works under the given constraints. We use the objective function value to see how well the optimisation outcomes are; a lower value equals better optimisation. We use the constraint fulfilment to see how well the method works under the specified constraints. Normalising these metrics lets us compare the results of different algorithms on the same scale.

Figure 3 illustrates the normalised results of several algorithms on these four measures. The physically restricted PSO method was chosen as a benchmark, and the performance of the other algorithms was compared to it.

Figure 3 Comparison of computational efficiency and convergence speed (see online version for colours)



Physically constrained PSO outperforms other commonly used optimisation algorithms, including GA, SA, ACO and EPSO, in several key performance metrics.

In terms of computational efficiency, physically constrained PSO scores a normalised score of 0.92, which is the highest amongst all the algorithms, suggesting that it operates faster and consumes fewer resources. In contrast, GA scored 0.76, SA 0.82, ACO 0.89, and EPSO 0.78, indicating that all these algorithms are weaker than the physically constrained PSO in terms of execution efficiency.

In terms of convergence speed, the physically constrained PSO scored 0.97, which is significantly better than the other algorithms. GA and SA scored 0.85 and 0.90, respectively, and EPSO 0.87. ACO performs better with a score of 0.95 but is still slightly lower than physical constraint PSO.

In terms of objective function value, physical constraint PSO scores 0.96, showing a stronger comprehensive optimisation ability. ACO scores 0.93, EPSO and GA 0.91 and 0.88 respectively, while SA is the lowest at 0.85, reflecting the different algorithms' balance between aesthetics and functionality differences.

In terms of satisfying physical constraints, physical constraint PSO scores 0.98, which is significantly higher than ACO (0.95), EPSO (0.94), GA (0.91), and SA (0.88), reflecting its higher reliability in avoiding overlapping of elements and guaranteeing minimum spacing.

Overall, physically constrained PSO demonstrates strong advantages in terms of computational efficiency, convergence speed, optimisation quality and constraint handling, and is an effective optimisation method for virtual landscape layout generation tasks.

5 Conclusions

In this paper, a virtual landscape layout generation method based on physical constraint PSO is proposed to optimise the layout design by introducing physical constraints. The experimental part verifies the superiority of the physical constraint PSO algorithm in several indexes by comparing it with the traditional optimisation algorithm. The experimental results show that physical constraint PSO can not only complete the layout generation task more efficiently, but also satisfy the constraints better, showing a strong application prospect.

Although physically constrained PSO achieves good results in the experiments, its limitations cannot be ignored. First, the optimisation performance of physically restricted PSO may change depending on the size of the problem. When it comes to larger-scale virtual landscape layout challenges, the algorithm may take a lot longer to run and use a lot more memory. Second, even though we tested the benefits of physically limited PSO in a few methods, it might not be able to keep up its best performance in some situations, especially when the environment changes quickly. Also, the parameters for physically restricted PSO still need to be adjusted for each problem and figuring out how to do this automatically for different situations is a big topic that needs to be solved in the future.

There are several ways that future study can go. First, people are interested in how to make physically limited PSO work in more complicated dynamic situations. For instance, in VR or AR settings, the layout of the landscape will change based on how the user interacts with it. More study is needed on how to update the layout generation in real-time to keep up with these changes. Second, combining modern technologies like DL or RL with physically limited PSO algorithms in unknown contexts could make them much more adaptable and able to learn on their own. The algorithm can dynamically improve its own search strategy and parameter settings by adding an intelligent decision-making mechanism. This makes it better at addressing big issues.

Finally, as technology keeps becoming better, more and more people will want to create virtual landscape layouts. As the situation gets more complicated, the methods that are already in use may not work as well. The physically constrained PSO method is planned to be used and expanded more broadly in virtual landscape layout creation and other areas as a result of these efforts.

Declarations

All authors declare that they have no conflicts of interest.

References

Al Qattan, A. (2019) 'Virtual reality city as a model for the sustainability of urban life in the built environment', *International Journal of Current Engineering and Technology*, Vol. 9, No. 6, pp.810–820.

Bian, K. and Priyadarshi, R. (2024) 'Machine learning optimization techniques: a survey, classification, challenges, and future research issues', *Archives of Computational Methods in Engineering*, Vol. 31, No. 7, pp.4209–4233.

Buswell, R.A., De Silva, W.L., Jones, S.Z. and Dirrenberger, J. (2018) '3D printing using concrete extrusion: a roadmap for research', *Cement and Concrete Research*, Vol. 112, pp.37–49.

Castro, O., Bruneau, P., Sottet, J-S. and Torregrossa, D. (2023) 'Landscape of high-performance Python to develop data science and machine learning applications', *ACM Computing Surveys*, Vol. 56, No. 3, pp.1–30.

Do, J., Picoli, I.L., Lomet, D. and Bonnet, P. (2021) 'Better database cost/performance via batched I/O on programmable SSD', *The VLDB Journal*, Vol. 30, pp.403–424.

Duan, Y., Chen, N., Chang, L., Ni, Y., Kumar, S.S. and Zhang, P. (2022) 'CAPSO: chaos adaptive particle swarm optimization algorithm', *IEEE Access*, Vol. 10, pp.29393–29405.

Elbes, M., Alzubi, S., Kanan, T., Al-Fuqaha, A. and Hawashin, B. (2019) 'A survey on particle swarm optimization with emphasis on engineering and network applications', *Evolutionary Intelligence*, Vol. 12, pp.113–129.

Elgamal, Z., Sabri, A.Q.M., Tubishat, M., Tbaishat, D., Makhadmeh, S.N. and Alomari, O.A. (2022) 'Improved reptile search optimization algorithm using chaotic map and simulated annealing for feature selection in medical field', *IEEE Access*, Vol. 10, pp.51428–51446.

Gad, A.G. (2022) 'Particle swarm optimization algorithm and its applications: a systematic review', *Archives of Computational Methods in Engineering*, Vol. 29, No. 5, pp.2531–2561.

Jia, J. (2022) 'Computer-aided design method of parametric model for landscape planning', *Computer-Aided Design and Applications*, Vol. 19, pp.55–64.

Kaven, L., Huke, P., Göppert, A. and Schmitt, R.H. (2024) 'Multi agent reinforcement learning for online layout planning and scheduling in flexible assembly systems', *Journal of Intelligent Manufacturing*, Vol. 35, No. 8, pp.3917–3936.

Khalid, O.W., Isa, N.A.M. and Sakim, H.A.M. (2023) 'Emperor penguin optimizer: a comprehensive review based on state-of-the-art meta-heuristic algorithms', *Alexandria Engineering Journal*, Vol. 63, pp.487–526.

Korkut, E.H. and Surer, E. (2023) 'Visualization in virtual reality: a systematic review', *Virtual Reality*, Vol. 27, No. 2, pp.1447–1480.

Lemley, M.A. and Casey, B. (2019) 'Remedies for robots', *The University of Chicago Law Review*, Vol. 86, No. 5, pp.1311–1396.

Liu, Z. (2025) 'Evaluating digitalized visualization interfaces: Integrating visual design elements and analytic hierarchy process', *International Journal of Human-Computer Interaction*, Vol. 41, No. 9, pp.5731–5760.

Ma, B., Dong, Y., Liu, H. and Cao, Z. (2022) 'Soft multimedia assisted new energy productive landscape design based on environmental analysis and edge-driven artificial intelligence', *Soft Computing*, Vol. 26, No. 23, pp.12957–12967.

Molina, D., LaTorre, A. and Herrera, F. (2018) 'An insight into bio-inspired and evolutionary algorithms for global optimization: review, analysis, and lessons learnt over a decade of competitions', *Cognitive Computation*, Vol. 10, pp.517–544.

Rude, U., Willcox, K., McInnes, L.C. and Sterck, H.D. (2018) 'Research and education in computational science and engineering', *Siam Review*, Vol. 60, No. 3, pp.707–754.

Senkevich, S., Bolshev, V., Ilchenko, E., Chakrabarti, P., Jasiński, M., Leonowicz, Z. and Chaplygin, M. (2021) 'Elastic damping mechanism optimization by indefinite lagrange multipliers', *IEEE Access*, Vol. 9, pp.71784–71804.

Slowik, A. and Kwasnicka, H. (2020) 'Evolutionary algorithms and their applications to engineering problems', *Neural Computing and Applications*, Vol. 32, pp.12363–12379.

Tafuni, A., Domínguez, J., Vacondio, R. and Crespo, A. (2018) 'A versatile algorithm for the treatment of open boundary conditions in smoothed particle hydrodynamics GPU models', *Computer Methods in Applied Mechanics and Engineering*, Vol. 342, pp.604–624.

Yasin, J.N., Mohamed, S.A., Haghbayan, M-H., Heikkonen, J., Tenhunen, H. and Plosila, J. (2020) 'Unmanned aerial vehicles (uavs): collision avoidance systems and approaches', *IEEE Access*, Vol. 8, pp.105139–105155.