Optimizing Urban Layout: A System Dynamics, Artificial Intelligence, and 3D Urban Information Model Approach for Active Management of City Complexity

Hanbin Wang [1], Ihab Hijazi [1,2], Andreas Donaubauer [1], Thomas H. Kolbe [1]

[1] Chair of Geoinformatics, Technical University of Munich, Arcisstr. 21, Munich, Germany - (hanbin.wang, ihab.hijazi, andreas.donaubauer, thomas.kolbe)@tum.de

[2] Urban Planning Engineering Department, An-Najah National University, Nablus, Palestine - (eehab@najah.edu)

Keywords: Artificial Intelligence, Genetic Algorithm, Models, System Dynamics, Urban Planning, CityGML

Abstract:

This study focuses on the integration of System Dynamics (SD), Artificial Intelligence (AI) technology, and 3D Urban Information Modeling (CityGML) in the field of urban planning. It aims to optimize urban layout to cope with rapid urban growth, emphasizing resource allocation through a combination of greedy strategies and AI methods. The core contribution of this paper is to provide spatial allocation of feedback from simulation techniques i.e. SD using the application of genetic algorithms (GA) based on a greedy strategy. The methodology was implemented using a fictitious case study in Berlin, where new students' dorms are required according to the growth in the number of students. The SD tool is used to simulate the growth over 5 years and determine the number of new dorms required. A genetic algorithm is used to optimize the planning objectives of allocating 16 new dorms in 186 possible locations. The algorithm effectively handles the feedback from SD and applies complex multi-objective optimization problems, addressing the challenge of accommodating a growing student population. It proposes strategic dormitory allocations that balance factors such as cost efficiency, campus accessibility, and proximity to public transportation systems. We use CityGML data to simulate and predict future urban transformation, providing dynamic and realistic representations for urban planning decisions and updating the original city models.

1. Introduction

Urban planning has played an important role in the development of modern cities. In recent decades the rapid pace of urbanization, in combination with population growth, the pressures of climate change and resource scarcity, and alongside the spur of innovation in technology and the forces of land economics have produced highly complex and interconnected urban systems. Urban planning can be defined as the science of order and coordination of the allocation of land use and the design detail and above all urban construction layout to encourage and enable the sustainable development of cities in long-term consideration (Levy, 2016). It is a scientific decision-making process for assessing the prospects of future goals and integrates different disciplines of resource allocation, socio-economic and environmental impacts in urban systems.

Historically, urban planning was reactive, responding to the immediate needs and challenges of urban changes (Netzel and Eber, 2003). However, it could be insufficient, potentially leading to operational inefficiencies and unsustainable outcomes under fast and complex uncertainty (Abd Elrahman and Asaad, 2021). System dynamics (SD) provides a mathematical framework for modeling nonlinear relationships in complex systems with time delays and multi-loop structures (Bala et al., 2017). Embedding SD models in urban planning therefore provides an insightful method for understanding nonlinearity and feedback loop structures in variable urban systems (Forrester, 1994). However, compared to other mathematical paradigms, SD was initially designed for non-spatial systems, while spatial elements are crucial in urban contexts. Therefore, it is important to integrate Geographic Information Systems (GIS) into SD models for spatiotemporal analyses of variables with spatial distributions, such as population density or pollution status. The combination of SD and GIS reveals intricate feedback relationships within the urban systems, including dynamic factors such as population growth (Whelan, 2001), traffic flow (Wang et al., 2008), and ecological simulation (Neuwirth et al., 2015).

Using system dynamics to model processes in the urban environment provides valuable feedback on urban growth, requiring strategic adjustments to urban configurations spatially, to respond to these changes. Some research focused on the spatial information of semantic city models such as the building structures, and the SD model was linked to the international standard CityGML (Hijazi et al., 2022). Xu and Coors (2012) conducted a sustainability assessment of urban housing by SD models, GIS, and 3D visualization. However, research has yet to look at spatial allocation for the results produced by such coupled models. In urban planning, decisions often have long-term consequences, and rational resource allocation is crucial to improve the efficiency and sustainable development of cities (Naess, 2001), which implies that there is a need to consider both long-term expectations and the prioritization of current needs, finding a reasonable optimization plan. Such heuristic decisions can be very complex or elusive for humans, thus, we hope to incorporate artificial intelligence (AI) techniques to help solving this problem. AI technique refers to the creation and usage of algorithms to construct dynamic computing environments that simulate the process of human intelligence (Winston, 1984). One of the approaches are Genetic Algorithms (GAs). GAs are search algorithms that draw on the principles of natural selection and genetics in the process of biological evolution to solve optimization problems (Holland, 1992). In this paper, we propose to integrate AI techniques with SD and GIS for spatial allocation of SD feedback regarding possible modification of the built environment to address gaps in urban allocation planning studies. This approach leverages the standardized information model of CityGML to enable efficient space allocation and powerful applications in a variety of urban scenarios, and to update the 3D city model with site allocation results. Thus, we contemplate an urbanization use case for planning purposes. The urbanization model is based on the research of urban simulation by Hijazi et al. (2022) and utilizes Berlin's CityGML data as input. For the

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-W5-2024-317-2024 | © Author(s) 2024. CC BY 4.0 License.

output of the urbanization model, this approach develops a GA framework for spatial allocation to address the optimization problem, facilitating rational resource allocation. Ultimately, the derived results are updated in the semantic urban model.
The structure of this paper is delineated as follows: after the introduction, section 2 provides the background on CityGML, System Dynamics in urban planning, and Genetic Algorithms. Section 3 presents the use case. Section 4 introduces detailed information on the framework, including the process of GA and the greedy strategy. Section 5 indicates the implementation and the result. Section 6 is the conclusion of the paper.

2. Background

2.1 CityGML

CityGML is an open standard model and XML-based format for representing, storing, and exchanging virtual 3D city and landscape models. It is an application schema for the Geography Markup Language (GML), developed by the Open Geospatial Consortium (OGC). This model defines the classes and relationships for the relevant topographic objects in cities and regional models concerning their geometrical, topological, semantical, and appearance properties, including buildings, bridges, tunnels, and other city facilities (Kolbe, 2009). It promotes the assimilation of urban geodata for a variety of applications for Smart Cities and Urban Digital Twins. In this study, we use CityGML data as the media data of SD, GA, and the representation on the 3D semantic city model, the framework is described in Section 4.3.

2.2 System Dynamics in Urban Planning

System Dynamics is a methodology and mathematical modeling technique for framing, understanding, and predicting complex systems. Initially developed in the 1950s, it was designed to augment corporate management insights into industrial dynamics (Forrester, 1994). SD facilitates comprehensive computer simulations of complex systems with adaptivity and temporal evolution. Moreover, it offers a way to link diverse disciplines, effectively bridging disparate academic realms (Forrester, 2009). Considering the variability and complexity of urban systems, it provides a structure for the interactions among various urban components, such as population, land use, economic growth, infrastructure development, and environmental factors. Urban simulations based on SD models serve as a basis and reference for sustainable urban design and policymaking (Kennedy et al., 2011). Planners can develop models to simulate the dynamic behaviors of urban systems under a variety of conditions, to predict the potential impacts and consequences of relevant policies, instead of reacting to the changes. Hijazi et al. (2022) proposed a novel modeling framework for coupling an urban system dynamics model using the modeling software Vensim with a semantic 3D city model, structured according to the international OGC standard CityGML.

2.3 Genetic Algorithms

Genetic Algorithms (GAs) are search algorithms that draw on the principles of natural selection and genetics in the process of biological evolution to solve optimization problems (Holland, 1992). They are based on Darwin's theory of natural selection, which means that the species best adapted to the environment will survive. In a GA, the various possibilities for solving a problem (called individuals) are encoded into a data structure, like a chromosome. A group of such individuals constitutes a population. Then, based on the concept of genetics, a new population is generated through a series of genetic operations, such as crossover and mutation in the population. Each individual has a score related to their fitness, and individuals with high fitness have a greater chance of being selected and passed on to the next generation. It can move randomly from one feasible plan to another, reducing the probability of getting stuck in a local optimum. Figure 1 shows the workflow of the GA. Until now, many new variants and improved versions of genetic algorithms have been widely used in machine learning, computer science, engineering design, operations research, etc. (Lambora et al., 2019). In urban planning, they were also used to solve various problems. Some studies focused on the site selection of public facilities, such as construction and demolition waste recycling plants (Liu et al., 2019), hospitals (Kaveh et al., 2020), and urban greening systems (Feng et al., 2022). Additionally, due to spatial heterogeneity, multiple land use types, and conflicts of interest among multiple stakeholders, some researchers (Porta et al., 2013; Li and Parrott, 2016) applied GA to multi-site land use allocation problems.

3. Use Case

Berlin is an international city known for its long history and vitality. As a center of culture and education, it is facing the challenges of rapid urbanization. This study draws on previous research on urban growth simulation (Hijazi et al., 2022) and attempts to apply the urban dynamics model to Berlin, utilizing AI technology for spatial allocation based on urban simulation results. We have set up a scenario to illustrate this concept: as more students are attracted to Berlin, the existing dormitories cannot meet the housing needs. This situation urgently requires the new dormitories to accommodate more students. Although reassignment of existing buildings as dormitories is also a feasible option, due to the uncertainty of whether the building meets the conditions, in this study we chose a more direct and simple method: selecting existing construction sites as potential locations. Our task is to forecast this growth and strategically determine the best locations for these new accommodations, ensuring they are convenient and meet student preferences, including proximity to campus, accessibility to public transportation, and construction costs. The goal is to continuously adapt and plan for future 5-year needs, ensuring that all students have access to comfortable and conveniently located housing. Figure 2(A) shows the distribution of 186 development plots in Berlin. The selection of potential sites is grounded in the standard land value data established by the Berlin Property Value Expert Committee as of January 1, 2023. This data presents the value per unit area of land and the type of land use, it is provided as a citable geographic information dataset available on the Berlin Geodata portal (Berlin.de, 2023). Figure 2(B) shows the land price of Berlin in different plots. As public facilities, the student dormitories are primarily for international or non-local students. Consequently, the location of these dormitories should ideally encompass as many university campuses as possible within a reasonable commuting distance. This arrangement is intended to facilitate the travel needs of students from various institutions. We selected the four 'main' universities mentioned in the Berlin Geodata Portal for the universities: ‘Around 100,000 people, about a fifth of them from all over the world, study in almost 700 courses at the four major universities in Berlin.’ (Berlin.de, 2023). These four universities are the Technical University of Berlin, the Humboldt University of Berlin, the University of Art of Berlin, and the Free University of Berlin. We chose the main campus of the four universities as the sites that the dormitory siting needs to cover, and the planning aims to make each dormitory cover as many schools as possible within the reachability range. Figure 2(C) shows the distribution of the main campus. Figure 2(D) shows the main subway station. The location of the dormitory should also consider the convenience of commuting. In this case, we intend to improve the accessibility

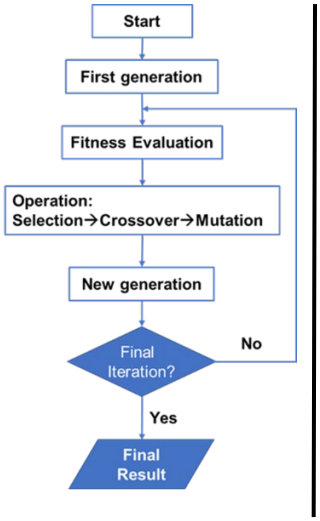


Figure 1 Workflow of GA.

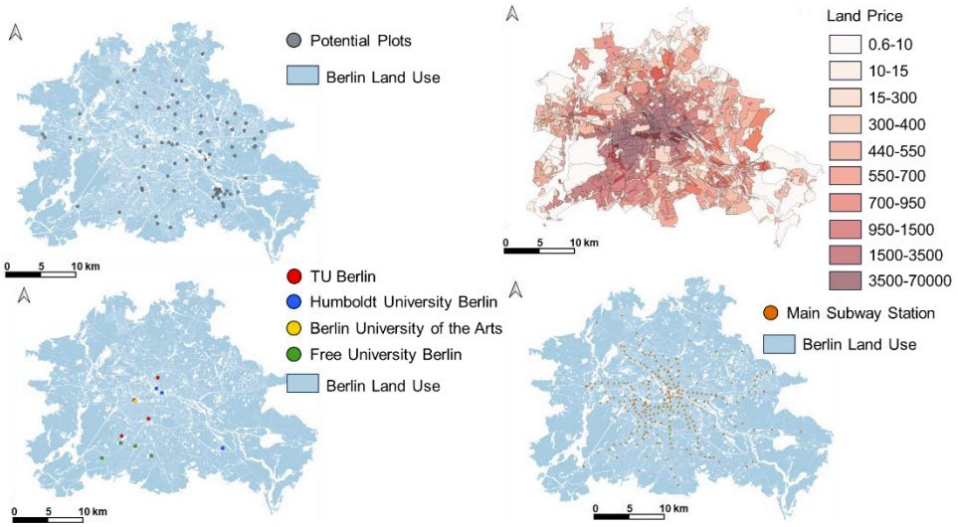


Figure 2 Distribution of: Potential Plots (A); Land Price (B); Main campus (C); Subway stations (D)

and include as many subway stations as possible around each dormitory. In the use case, we need the arrangement with a lower cost while meeting the maximum benefits of the above-mentioned needs for the resources. However, urban planning issues are often complex and it's not always feasible to consider all problems simultaneously. For instance, in this use case, we aim to minimize costs, however, the dormitory location is a long-term consideration, and the central objective is to serve the students' needs effectively. Thus, compared with the cost, commuting and the coverage of the school campus are more important. Thus, through a greedy strategy, we could consider the weight of these influencing factors and make maximum benefits in each selection round.

4. Spatial Allocation Using Genetic Algorithm

4.1 Framework

Urban planning involves a comprehensive understanding and forecasting of complex changes within urban environments. This can be achieved through an integrated SD model and 3D city model. This integration offers a robust methodology for interpreting spatial and temporal urban data. We extract the primary data source from CityGML data (number of existing dormitories, number of students, etc.), and it serves as an input to the SD model. For urban growth simulation, the expected prediction results of the SD model can provide a reference for urban planners to deal with new urban problems. The use case in this study can be considered as a multi-objective optimization problem, the space allocation of the dormitory should be weighed by different factors to find the optimal solution. Thus, in response to the feedback from the SD model, we employ AI techniques (GA) for decision-making and use a greedy strategy to account for the importance of different factors to meet the demands brought by urban growth. The result is a sequence of location numbers and will be linked through CityGML to change the semantic 3D city model to reflect the proposed planning decisions. Figure 3 shows the framework of the integration.

4.2 Encoding and Initialization

Building upon the framework outlined above, we make the assumption that 16 new dormitory buildings will be allocated among 186 plots, based on feedback from the SD model. To implement this allocation, possible planning solutions must be coded. In our coding scheme, each chromosome represents a potential allocation scheme for the dormitory buildings. Specifically, each chromosome consists of 16 segments, aligning with the assumed number of dormitory buildings. The value of each segment is an integer between 1 and 186, indicating the number of plots assigned to the corresponding dormitory building. Therefore, a chromosome is an array of integers of length 16, representing one plausible allocation scheme for a dormitory building (refer to Figure 4). During the initialization phase of the genetic algorithm, the first-generation population is formed with 100 randomly generated chromosomes. Each chromosome is created by randomly assigning values to its segments. This approach ensures that the initial population comprises 100 diverse and potential allocation schemes for the dormitory buildings.

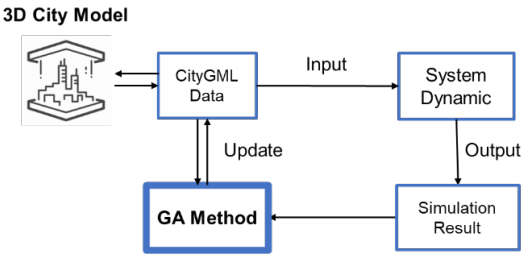


Figure 1 The integration framework of CityGML, SD, and GA.

Dormitory	1	2	3	4	5	·····	13	14	15	16
Plot	10	5	16	31	1	·····	20	17	46	13

Figure 2 Encoding and initialization diagram.

4.3 Fitness Function

4.3.1 Cost

As mentioned in section 3, the cost assessment is based on standard land value data established by the Berlin Committee of Experts on Real Estate Values (Berlin.de, 2023). To reduce the complexity of the planning problem, we assume that all dormitories have uniform specifications so that the cost of each dormitory is only related to the unit land value of the building area. Table 1 shows the area of the corresponding potential construction plots and their unit land value ($€/m^2$). For m dormitories, the total cost is the sum of the land values of the corresponding plots:

$$cost = \sum_{i=1}^{m} cost_i \quad (1)$$

Plot Number	District	Unit land value €/m^2
1	Spandau	150
2	Spandau	150
3	Treptow-Köpenick	180
…	…	…
185	Mitte	4000
186	Wilmersdorf	5000

Table 1 Standard land value of potential plots.

4.3.2 Coverage

In this study, one of the goals of optimizing the space allocation of student residences is to maximize the "coverage" of the residence halls to accommodate students from the four major universities. Each university has several campuses, so our goal is to ensure that these dormitories are distributed across as many campuses as possible to maximize student commuting convenience. We define "coverage" as the coverage of all campuses by dormitory distribution, which is obtained by calculating the distance from each campus to its nearest dormitory and then applying a decreasing coverage function. The output of this function is a coverage score, which reflects how well a dormitory covers a campus. For each campus, we select the 2 closest dormitories to it and calculate their coverage scores based on the distance. We then summed the coverage scores across all campuses as the "coverage". The advantage of this approach is that it ensures that each dormitory can house as many of the University's students as possible, rather than just serving the closest campus. This is because if all residence halls are too close to one campus, then their "coverage" will be reduced because other campuses will have lower coverage scores. Therefore, our approach can encourage a more even distribution of dorms, thereby improving commuting convenience for all students.

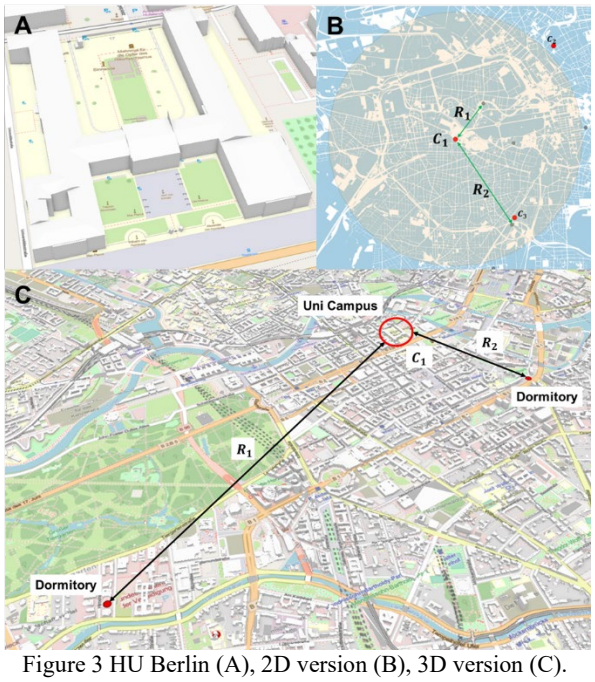


Figure 3 HU Berlin (A), 2D version (B), 3D version (C).

Figure 5 shows the campuses of the Humboldt University of Berlin. C_n indicates the campus center where n is the number of the campus, for the selection of the main campus is mentioned in section 3. R_i represents the distance from the nearest dormitory where i is the number of these dormitories. The coverage can be calculated as the function below:

$$Cov = \sum_{c=1}^{n} \sum_{i=1}^{2} \exp(-\alpha R_i) \quad (2)$$

In the coverage function, Cov is calculated by Euclidean distance, the unit is kilometer, and α is a positive constant used to control the decay speed. The coverage score will quickly decay to 0 as the distance increases. The decay rate of the coverage score can be controlled by adjusting the value of α. The advantage of this function is that it converts large distances into small scores and small distances into large scores and guarantees that all scores are between 0 and 1.

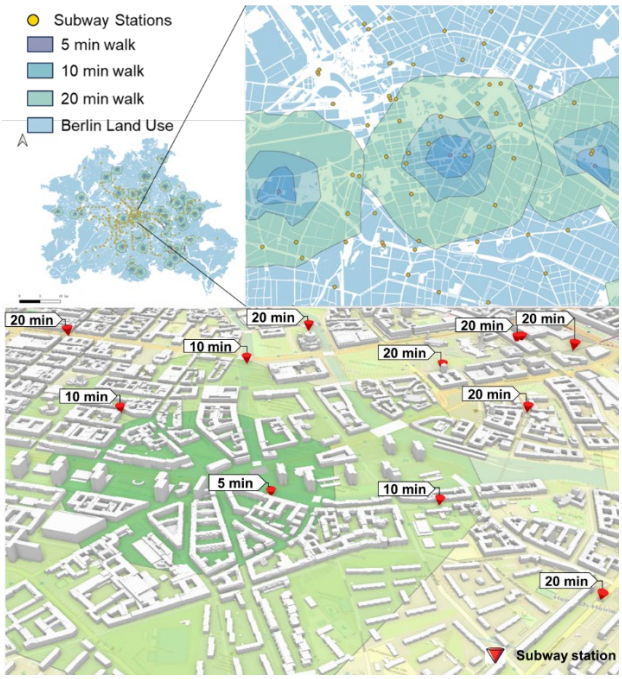


Figure 4 Isochrones and subway stations in Berlin.

4.3.3 Commuting

In the context of commuting convenience, we consider the accessibility of subway stations from dormitories and focus on the starting point of each dormitory. The goal is to have each dormitory within an acceptable distance of as many subway stations as possible. This allows students to choose the most convenient subway station based on their destination, thereby enhancing their commuting convenience. The accessibility of subway stations is intrinsically linked to the distribution of dormitories. A dormitory close to multiple subway stations will have high accessibility, and vice versa. However, due to varying traffic conditions and road structures at different locations, we generate isochrones based on walking time. Shorter walking time represents higher accessibility, it is a simplifying assumption to rank different solutions and needs validation in future research. These isochrones representing walking times allow us to quantify accessibility to subway stations more accurately. To further refine the measure of subway station accessibility from each dormitory, we introduce a scoring system. This system quantifies the convenience of each dormitory to its surrounding subway stations based on the isochrones. Subway stations within the same isochrone are considered to have the same accessibility. Thus, for dormitory d the fitness of commuting (Com_d) can be calculated by the function below:

$$Com_d = \sum_{i=1}^{N_d} S(t_i) \quad (3)$$

N_d is the number of subway stations near dormitory d, t_i is the time (minutes) from the $i-th$ subway station to dormitory d, and $S(t)$ is the score based on time t. In this scoring system, $S(t)$ equals 3 if $t \leq 5$, $S(t)$ equals 2 if $5 < t \leq 10$, $S(t)$ equals 1 if $10 < t \leq 20$. Overall commuting score Com is the sum of the score of each dormitory (Com_d).

$$Com = \sum_{d=1}^{16} Com_d \quad (4)$$

4.4 Genetic Operations

4.4.1 Crossover

In genetic algorithms, crossover is an operation that mimics the process of biological reproduction and genetic recombination to generate new candidate solutions. It involves the exchange of genetic information between two parental chromosomes to produce new offspring that combine the characteristics of both parents. This operation introduces diversity into the population and allows the exploration of new areas in the solution space. In our dormitory assignment problem, the crossover operation is implemented as follows: We first select two chromosomes from the population. These chromosomes represent two potential dormitory allocation scenarios. We then generate a random integer "n" in the range 2 to 16. For each fragment starting at position n in a chromosome, we swap the corresponding value between the two chromosomes. It should be noted that the dormitory building number on each chromosome is fixed. Therefore, the exchange operation is a mapping exchange between the same dormitory number in the two chromosomes. This ensures that each dormitory building is still allocated to a plot after the cross-operation, preserving the feasibility of the allocation scheme. Through this crossover operation, we generate new assignment schemes that combine elements of the two parent schemes, thereby enhancing the diversity and underlying quality of the solutions in our population. Figure 7 shows the step of crossing.

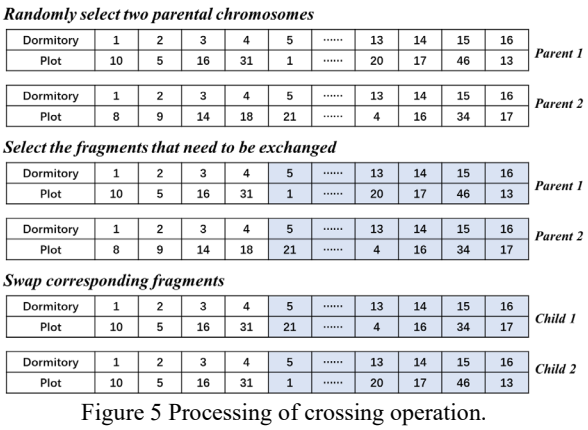


Figure 5 Processing of crossing operation.

4.4.2 Mutation

In the genetic algorithm, the mutation operation simulates the mutation phenomenon in biological inheritance, that is, the random change in the gene sequence can introduce new genes and increase the diversity of the population to explore new areas of the solution space. In our use case, the mutation operation is implemented as follows: First, a chromosome is selected from the population. This chromosome represents a possible dormitory allocation scheme. Then, generate a random integer m between 1 and 16. This integer m determines how many segments of the chromosome we want to mutate. Next, randomly select m fragments in the chromosome to mutate. For each selected segment, randomly change the number of the plot to which the dormitory building is located. This newly generated allocation scheme is based on the original with minor adjustments. Through the mutation operation, we can explore new regions of the solution space and find potential better solutions. At the same time, it can also prevent the research processing from falling into the local optimum, thus improving the searchability and efficiency of the genetic algorithm.

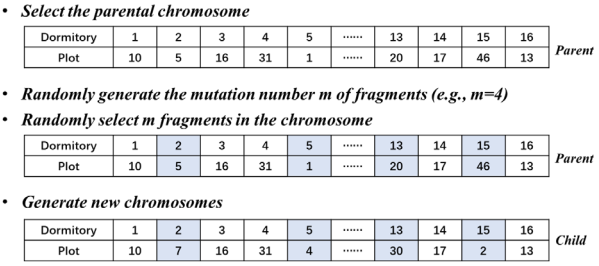


Figure 6 Processing of mutation operation.

4.5 Selection Method

In the GA, the selection operation is a key step, determining which chromosomes will be taken or disused to the next generation. The selection operation mimics the process of natural selection: individuals with better fitness have a greater chance of surviving and passing on their genes. In this work, the selection of an individual is based on the combination of roulette wheel selection and greedy strategy. This section presents their definitions and applications in this use case.

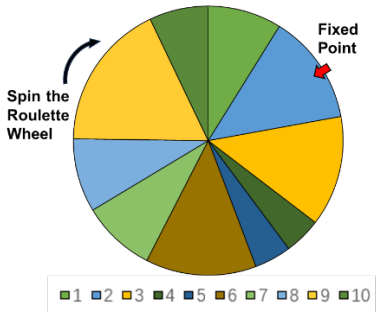
4.5.1 Roulette Wheel Selection

The roulette selection is a commonly used method for selecting individual strategies in genetic algorithms. The basic idea is that the probability of an individual being selected is proportional to its fitness value. Individuals with higher fitness have a greater probability of being selected, and vice versa. The selection probability $P(S_i)$ is calculated based on fitness.

$$P(S_i) = \frac{(S_i)}{\sum fit(S_i)} \quad (5)$$

$fit(S_i)$ refers to the fitness value of an individual, the probability of each individual $P(S_i)$ is equal to the quotient of their fitness and the sum of fitness of the whole population. Therefore, this method avoids only ensuring the spread of good genes, but also allows individuals with low fitness to have a chance to be selected, ensuring the diversity of the population, and helping the algorithm to jump out of the local optimal solution and find the global optimal solution. This process can be imagined as a roulette wheel, as shown in Figure 9, each individual occupies a part of the board, and the size of this part is proportional to its fitness value. Then, the roulette is rotated, and a stop position is generated at a 'fixed point', and the individuals corresponding to the stop at this point interval are selected. The process will be repeated until the wheel has selected enough individuals (Figure 9). In this process, individuals with high fitness will have a higher probability of being selected and may be selected several times. In this study, this method is defined as a ‘minimal’ selection unit, used to extract individuals from a population.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-W5-2024-317-2024 | © Author(s) 2024. CC BY 4.0 License.



Individual	Fitness Value
1	0.1
2	0.15
3	0.15
4	0.05
5	0.05
6	0.15
7	0.1
8	0.1
9	0.2
10	0.08

Figure 7 Roulette Wheel Selection.

4.5.2 Greedy Strategy

In traditional genetic algorithms, the selection strategy is usually based on a single fitness function, but in this use case, we want to optimize individuals on multiple indicators (coverage, commuting, and cost). In addition, as mentioned in the use case section, these three indicators have different consideration ratios, i.e., when making decisions, we prioritize coverage, then commutability, and finally cost. Thus, we propose a genetic strategy-based multi-stage selection algorithm. In this strategy, we divide the entire selection phase into three rounds. In the first round, individuals are selected according to the fitness of coverage, and the selected individuals will enter the second round, which is selected according to the fitness of commuting. Eventually, the selected individuals in the second round will enter the third round, and then carry out the final round of selection according to the fitness of the cost index (Figure 10). This greedy selection strategy can give priority to the most important indicators, and select the next indicator based on the selection results of the previous round.

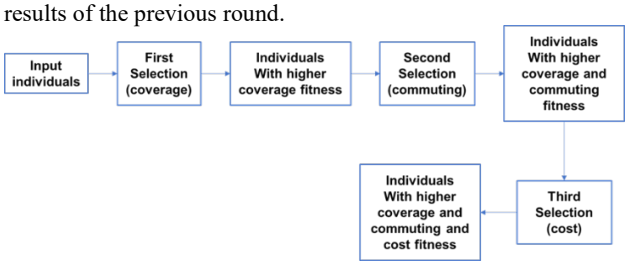


Figure 8 Workflow of Greedy Selection.

4.6 Elitist Genetic Algorithm (EGA)

To ensure that the fitness of the next generation population will not decrease, we will directly select the 20% chromosomes with the highest fitness of the previous generation to enter the next generation. At the same time, 80% of individuals will be generated from the crossed and mutated chromosomes. In this way, the number of chromosomes in each generation population remains stable. The elitist genetic algorithm (EGA) has been well developed such as NSGA-II and proven to be more efficient than traditional GA (Deb et al., 2002). In this study, we incorporate this elitist strategy into our algorithm to fit the needs of the use case. The core tenet of EGA is the preservation and reinforcement of elite individuals throughout the evolutionary cycle, ensuring a consistent trajectory towards the global or near-global optima. The algorithm operates as follows:

1. Initialization: Formulate an initial population of N individuals by the encoding regulations.
2. Termination Evaluation: If the stop conditions are met, conclude the algorithm, and otherwise to the next step.
3. Parent Selection: Separate N parents from the existing population through the fitness function.
4. Crossover Operations: Implement independent pairwise crossover procedures on the selected parents, leading to the birth of N offspring.
5. Mutation Operations: Select individuals among the offspring for mutation according to the mutation rate.
6. Population Fusion: Merge the original parent group and the newly formed offspring population, culminating in an augmented population size of 2N.
7. Elitist Selection: From this combined group of 2N entities, judiciously select N individuals based on a predetermined selection protocol.
8. Iteration: Back to step 2 to re-assess convergence and perpetuate the loop until the stop conditions are met.

Algorithm 1 Elitist GA (EGA)

Input: N (population size), $MAXGEN$ (maximum number of generations), $\mathcal{X}$ (crossover operator), $\mathcal{M}$ (mutation operator), $\mathcal{S}$ (selection operator)
Output: P (final population), x^* (best solution)
$P \leftarrow \text{InitializePopulation}(N)$
$f(P) \leftarrow \text{EvaluateFitness}(P)$
for $g = 1$ to $MAXGEN$ **do**
  $P_{best} \leftarrow \text{BestIndividual}(P, f(P))$
  $P_{parents} \leftarrow \mathcal{S}(P, f(P), N)$
  $P_{offspring} \leftarrow \mathcal{X}(P_{parents})$
  $P_{mutated} \leftarrow \mathcal{M}(P_{offspring})$
  $P' \leftarrow P \cup P_{mutated}$
  $P \leftarrow \mathcal{S}(P', f(P'), N)$
end for
$x^* \leftarrow \text{BestIndividual}(P, f(P))$
return P, x^*

Figure 9 Pseudocode of EGA.

5. Implementation

5.1 Urban Simulation of the Use Case in Berlin

Urban simulation of the use case is based on the concepts from a previous study about urban growth simulations in Munich (Hijazi et al., 2022). We use the bidirectional data exchange tools between system dynamics models and semantic 3D city model, it extracts CityGML data through the spatial ETL (Extract, Transform, Load) software package FME, and inputs it into the SD model to simulate and update the 3D city model. Figure 12 shows the data synchronization operation loop between the SD model and the CityGML data.

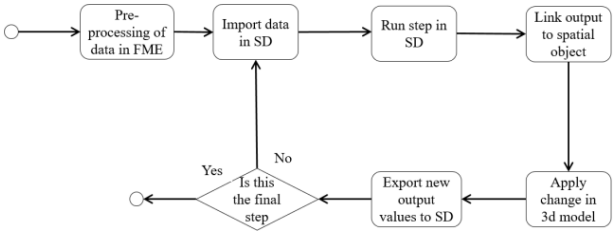


Figure 10 Diagram showing the interaction between Vensim and FME software (Hijazi et al., 2022).

The urban simulation of Berlin is based on the Vensim simulation software package (Figure 13). In this model, the level component is the number of students, it is impacted by the incoming and graduating students. The number of total dormitories is calculated by the number of students per dormitory and the number of students. Additionally, to make the model simple, we assume that the dormitories need to accommodate all students. Because of the limitation of the essential resources for urban growth, there's an imperative need to devise a strategic plan to optimize the utilization of these resources while ensuring sustainable development. Therefore, we plan dormitories in potential construction sites, considering three factors: cost, coverage of the university campus, and convenience of public

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-W5-2024-317-2024 | © Author(s) 2024. CC BY 4.0 License.

transportation around the dormitory (in this use case we consider the number of subway stations). Figure 16(A) shows one potential site (red area) in the Berlin mesh model view. Figure 14 shows the result, after 5 years, the number of students has increased by 30%, and 16 new dormitories are demanded to accommodate the growing number of students.

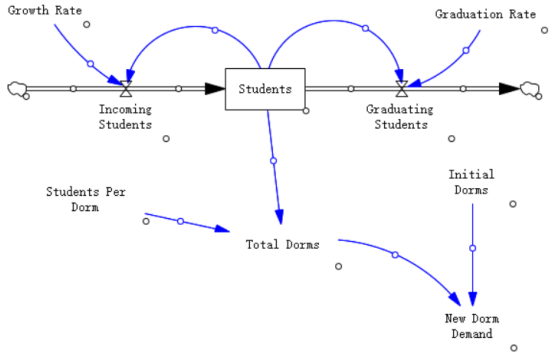


Figure 11 SD model of the use case in Berlin.

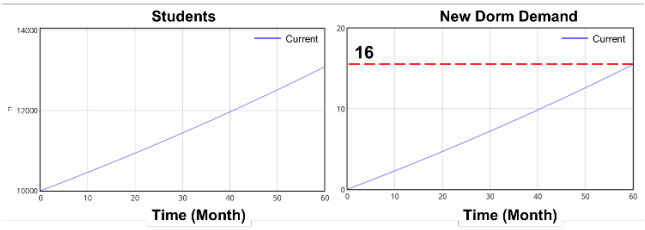


Figure 12 Results for students and new dormitories needed.

5.2 Algorithm implementation

As in the algorithm framework developed in section 4.3, we prioritize coverage, then commuting, and finally cost in the fitness function. Since we use distance to present coverage, thus the smaller the total distance, the better the coverage of the individual. Similarly, the smaller the cost present the higher the fitness. However, it is worth noting that the evaluation index for commuting convenience is the opposite: closer subway stations will get higher scores, so we want this value to be as large as possible. Figure 15 shows the changes in the fitness value indicators of coverage, commuting, and cost of individuals in the iterations of the greedy genetic algorithm.

In the figure of coverage with distance as fitness value, the decline of the average fitness value and the best fitness value indicates that the algorithm successfully found solutions that minimized the total distance and improved the coverage of the dormitory layout. This presents that within iterations the algorithm quickly identifies significantly better coverage strategies, which are then improved upon in subsequent generations. The figure for commuting shows an increasing trend in fitness values, with higher values indicating better solutions. This increase shows that the algorithm effectively found dormitory locations that are closer to subway stations in the second round. Fluctuations represent the algorithm's exploration process to find the easiest solution to implement. The figure of cost change reflects the third priority in the optimization process and exhibits greater fluctuations. Due to the greedy strategy, cost optimization depends on the results of previous rounds of coverage and commuting. These fluctuations indicate that while the algorithm strives to minimize costs, it is limited by the solutions generated by the first two objectives. Therefore, it does not consistently achieve the lowest cost due to trade-offs in coverage and commute improvements. This results in less stable progress in cost optimization, with occasional spikes where more cost-effective solutions are discovered but are not necessarily maintained due to the hierarchical nature of the optimization process. Overall, these graphs collectively demonstrate the ability of greedy genetic algorithms to improve urban planning solutions over time, reflecting initial rapid enhancements in coverage and commuting, followed by prioritization due to the influence of greedy strategies.

5.3 Updating the semantic 3D city model

For visualization, we use the results of the genetic algorithm as input and utilize the study of Hijazi et al. (2020) as a reference to link to the CityGML data, associate the plot information from the results with the coordinates in the 3D city model, and add new buildings at the selected locations. Figure 16 shows the process of updating the 3D city model of a part of Berlin. The red area in the mesh model represents a potential construction plot. In this use case, we only plan the location of the dormitory, not the size of the dormitory. Thus, in our experiments, all dormitories have the same shape, based on the existing average dormitory size. The picture on the left is the model in CityGML format and the picture right shows the updated model: the red building is a new dormitory added to the site.

6. Conclusion

This study demonstrates the effectiveness of our approach in optimizing urban layout in rapid urban growth by combining system dynamics, artificial intelligence, and geographic information systems. In particular, we used the SD model to simulate the demand for student dormitories in Berlin for 5 years and utilized the genetic algorithm based on a greedy strategy to solve this multi-objective optimization problem for student dormitory planning. We considered the balance between campus accessibility and public transportation proximity and cost in order according to priority. The algorithm shows significant improvements in dormitory layout coverage and location selection close to subway stations. Although there are fluctuations in cost minimization, this reflects the algorithm's process of making trade-offs between solutions. In addition, we applied the results to the 3D city model by transmitting to CityGML data and updating the 3D city model, focussing rather on the semantic model of CityGML than on automatically generating complex and realistic 3D building models. In future

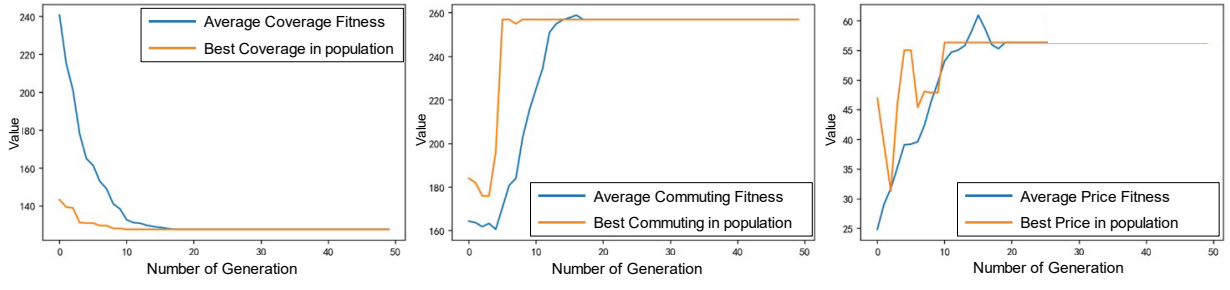


Figure 13 Changes of fitness value of individuals in the GA iterations: coverage (left), commuting (middle), and cost (right).

research, more emphasis can be placed on the building structure, which would allow for utilizing the site allocation results in downstream processes that have already been realized based on the CityGML standard, such as public participation, and 3D analysis and simulations, such as energy demand and solar potential estimation, impacts on microclimate, etc.

Since this study focuses on macro-planning of the location of the dormitories, it is inevitable that fitness function information (such as land value) comes from external data sources rather than CityGML data. Therefore, it is a limitation of our investigations that we have developed and evaluated our approach only using the specific context and the data available in Berlin. In future research, all basic information should be integrated including BIM (Beck et al., 2021), into the CityGML semantic model, for example, by creating a CityGML site allocation application domain extension (ADE) to formally describe the data demand required by our approach and thereby facilitate the transfer of the approach to another city.

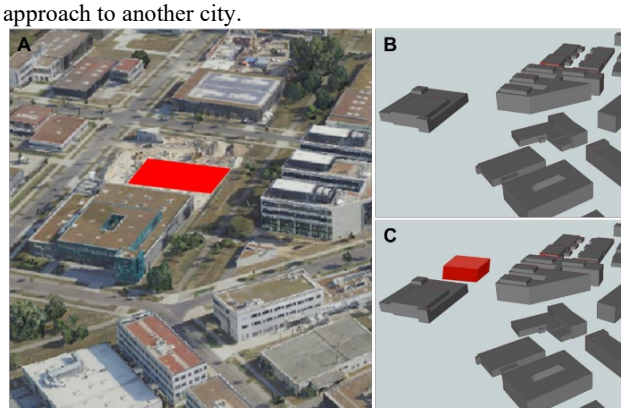


Figure 16 Mesh model, the red area is the construction site (A); CityGML model (B); Updated CityGML model, the new building is red (C).

References

Abd Elrahman, A. S., & Asaad, M., 2021. Urban design & urban planning: A critical analysis to the theoretical relationship gap. Ain Shams Engineering Journal, 12(1), 1163-1173.

Beck, Stefan F., Abualdenien, J., Hijazi, I., Borrmann, A., & Kolbe, T. H., 2021. Analyzing Contextual Linking of Heterogeneous Information Models from the Domains BIM and UIM. ISPRS International Journal of Geo-Information 10(12), 807-830.

Berlin.de. https://www.berlin.de/ (1 Jan 2023)

Bala, B. K., Arshad, F. M., & Noh, K. M., 2017. System dynamics. Modelling and Simulation, 274.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation, 6(2), 182-197.

Forrester, J. W., 1994. System dynamics, systems thinking, and soft OR. System dynamics review, 10(2-3), 245-256.

Forrester, J. W., 2009. Some basic concepts in system dynamics. Sloan School of Management, Massachusetts Institute of Technology. Cambridge, MA, USA. 1-17.

Feng, L., Mi, X., & Yuan, D., 2022. Optimal planning of urban greening system in response to urban microenvironments in a high-density city using genetic algorithm: A case study of Tianjin. Sustainable Cities and Society, 87, 104244.

Hijazi, I. H., Krauth, T., Donaubauer, A., & Kolbe, T, 2020. 3DCITYDB4BIM: A system architecture for linking bim server and 3d citydb for bim-gis-integration. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 4, 195-202.

Hijazi, I., Donaubauer, A., Hamm, A., Falkenstein, A., & Kolbe, T. H., 2022. Urban Growth Simulation Using Urban Dynamics and Citygml: A Use Case from the City of Munich. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 10, 97-104.

Holland, J. H., 1992. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press.

Kennedy, C., Pincetl, S., & Bunje, P., 2011. The study of urban metabolism and its applications to urban planning and design. Environmental pollution, 159(8-9), 1965-1973.

Kaveh, M., Kaveh, M., Mesgari, M. S., & Paland, R. S., 2020. Multiple criteria decision-making for hospital location-allocation based on improved genetic algorithm. Applied Geomatics, 12, 291-306.

Kolbe TH., 2009. Representing and exchanging 3D city models with CityGML. In 3D geo-information sciences (pp. 15-31). Berlin, Heidelberg: Springer Berlin Heidelberg.

Lambora, A., Gupta, K., & Chopra, K., 2019. Genetic algorithm-A literature review. In 2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon) (pp. 380-384). IEEE.

Liu, J., Xiao, Y., Wang, D., & Pang, Y., 2019. Optimization of site selection for construction and demolition waste recycling plant using genetic algorithm. Neural Computing and Applications, 31, 233-245.

Levy, J. M., 2016. Contemporary urban planning. Taylor & Francis.

Li, X., & Parrott, L., 2016. An improved Genetic Algorithm for spatial optimization of multi-objective and multi-site land use allocation. Computers, Environment and urban systems, 59, 184-194.

Neuwirth, C., Peck, A., & Simonović, S. P., 2015. Modeling structural change in spatial system dynamics: A Daisyworld example. Environmental Modelling & Software, 65, 30-40.

Netzel, D. M., & Eber, L., 2003. Shifting from reactive to proactive discipline in an urban school district: A change of focus through PBIS implementation. Journal of Positive Behavior Interventions, 5(2), 71-79.

Naess, P., 2001. Urban planning and sustainable development. European planning studies, 9(4), 503-524.

Porta, J., Parapar, J., Doallo, R., Rivera, F. F., Santé, I., & Crecente, R., 2013. High performance genetic algorithm for land use planning. Computers, environment and urban systems, 37, 45-58.

Wang, J., Lu, H., & Peng, H., 2008. System dynamics model of urban transportation system and its application. Journal of Transportation Systems engineering and information technology, 8(3), 83-89.

Whelan, J. G., 2001. Modeling Exercises Section 2, part of: MIT System Dynamics in Education Project. Massachusetts Institute of Technology (ed.), Cambridge, Massachusetts.

Winston, P. H., 1984. Artificial intelligence. Addison-Wesley Longman Publishing Co., Inc.

Xu, Z., & Coors, V. (2012). Combining system dynamics model, GIS and 3D visualization in sustainability assessment of urban residential development. Building and Environment, 47, 272-287.