A Constraint-Driven Framework for AIGC-Assisted Landscape Scheme Development and Optimisation

Xiaobin Li

Liaoning Communication University, Shenyang 110000, China

Abstract

With the rapid iteration of AI technologies, their application in landscape architecture has become increasingly common, shifting design practice from experience-based manual drawing toward a collaborative model of data-assisted generation and value judgment. However, scheme reasoning in landscape architecture must consider multiple constraints, including codes and standards, site conditions, ecology, engineering, cost, and maintenance. Current AI tools cannot adequately handle these constraints; their outputs are often visually usable but difficult to implement. This paper takes constraints as the organizing logic and proposes a "three-tier constraints–three-stage–dual-loop" human–AI collaborative framework for scheme reasoning. The framework consists of constraint coding in the early stage, generative scheme reasoning in the middle stage, and scheme evaluation and iteration in the late stage, supported by a dual-loop mechanism combining AI iteration and human decision-making. An exploratory practice involving a residential children's play area preliminarily indicates that constraint coding can improve the usability and evaluability of AI outputs. Generative AI is suitable for rapid multi-style variant generation and imagery reasoning under constrained conditions, whereas engineering responsibility and design direction judgment should remain with designers.

Keywords

Generative AI, landscape architecture, constraint-driven design, scheme reasoning, human–AI collaboration.

1. Introduction

Generative AI has entered the schematic design phase of landscape architecture, with applications mainly concentrated in image generation, scheme variants, and imagery expression. However, landscape scheme reasoning is not unconditional free creation; it must balance multiple constraints such as site conditions, design positioning, construction cost, and future maintenance, including setback lines, grading, sunlight and drainage, fire safety, accessibility, and plant adaptability. Designers can currently use AI to rapidly generate large numbers of design images, but these images cannot be treated as real design schemes because they fail to respond to and balance the many site constraints. Common problems include distorted scale, non-compliance with design codes, and unsuitable plants for the local environment. The reason is that these constraints are not systematically classified and collected in the generation process, and existing prompting methods cannot provide sufficient effective information.Wang et al. in a study on auxiliary generation of rural landscape plans, similarly pointed out that although general AIGC tools can quickly produce visually complete drawings, real site information is easily weakened, resulting in discontinuous road and water network topology and misaligned red-line boundaries. The root cause is that site constraints are not continuously transmitted through explicit interfaces in the generation chain[1]. In scheme reasoning, there is still a lack of mechanisms for constraint identification, layering, translation, evaluation, and iteration.

To better support AI-assisted landscape scheme reasoning, this paper proposes three research questions: first, how can key constraints in the landscape schematic design phase be layered and coded; second, what tasks can generative AI currently perform in scheme reasoning, and where are its boundaries; third, how can an operational human–AI collaborative framework be constructed to meet the practical needs of scheme reasoning. Existing studies have begun to explore human–AI collaborative workflows, but most focus on a single stage or a single toolchain and have not yet formed a coherent framework organized by constraints [2].

Combining an exploratory practice involving a residential children's play area, this paper argues that under a constraint mechanism, scheme reasoning can be precisely generated under constrained conditions, generated schemes can be compared and evaluated according to constraints, and the evaluation process can be traceable. The paper develops a three-tier constraint system-hard, soft, and tacit constraints-and their AI translation methods, and constructs a "three-stage–dual-loop" framework for scheme reasoning.

2. Constraint Layers and AI Translation

2.1. Logic of Constraint-driven Design

The essence of scheme reasoning is to search for different combinations that balance various factors within a constraint space. Constraints are the skeleton that organizes design decisions. For AI, the degree to which constraints can be operationalized directly determines output quality. Only constraints that can be clearly expressed, stably controlled, and effectively evaluated can enable AI to generate meaningful variations within their scope; constraints that cannot be translated must enter the human judgment process. Therefore, the purpose of constraint layering is not taxonomic classification, but the establishment of a correspondence between translation and evaluation.

2.2. Three-tier Constraint System

Based on the practical characteristics of the landscape schematic design phase, this paper divides constraints into three tiers: hard constraints, soft constraints, and tacit constraints.

Hard constraints refer to constraints with regulatory force or non-negotiable engineering requirements, including codes, industry standards, and local regulations, such as setback lines, grading, sunlight and drainage, fire safety, accessibility, and plant adaptability. Hard constraints are characterized by clear boundaries, quantifiable management, and mandatory compliance.

Soft constraints refer to subjective evaluations of scheme quality, project positioning, and artistic expression that influence the scheme but have flexible space, such as functional proportion, circulation organization, design style, and client preferences. Soft constraints can be weighed and compromised, but such compromise affects scheme quality.

Tacit constraints refer to constraints that cannot be directly expressed in drawings but affect project implementation, including local material supply, plant adaptability and seasonal performance, procurement preferences, and maintenance capacity. Tacit constraints require industry experience; they are not directly usable clues and cannot be explicitly reflected in task briefs or codes, yet they are important reasons determining whether a scheme is adopted.

2.3. AI Translation and Evaluation of Constraints

The three tiers of constraints have different translation and evaluation methods. Hard constraints are generally translated into negative prompts, control drawings, and rule tables, and are evaluated mainly through compliance checks of "satisfied/not satisfied." Soft constraints are translated into prompt weights, reference images, zoning diagrams, and style models, and are evaluated through designers' subjective ratings and indicator comparisons. Tacit constraints are difficult to translate directly into generative inputs and should be

translated into expert rule bases, case libraries, and manual review checklists, with evaluation relying on professional experience. Table 1 presents the correspondence.

Table 1. Constraint layers, AI translation, and evaluation methods (residential children's play area)

Constraint tier	Typical constraints	AI translation methods	Evaluation methods
Hard constraints	Red-line setback; fire access lanes and high-rise firefighting surfaces; sunlight duration; accessibility; fall height and impact-attenuating surfacing; drainage slope; non-toxic and thorn-free plants	Negative prompts; control drawings (setback lines, fire access lines, sunlight analysis); rule tables; prohibited-item lists	Compliance check: satisfied/not satisfied, item-by-item checklist
Soft constraints	Age zoning (0–3, 3–6, 6–12 years); supervision sightlines; caregiver seating areas; diversity of play types; circulation continuity; color and character	Prompt weights; reference images; zoning diagrams; style models	Designer subjective rating (high/medium/low) + indicator comparison (zoning area ratio, sightline coverage)
Tacit constraints	Property management and maintenance capacity; equipment procurement preferences; construction habits; local plant adaptability; cost control; long-term maintenance cost	Expert rule bases; case libraries; manual review checklists (difficult to translate directly into generation)	Professional experience judgment; multidisciplinary review; confirmation by operations and maintenance parties

3. Capabilities and Limits of Generative AI

With rapid annual iteration, generative AI can increasingly perform tasks in the landscape schematic design phase. At present, AI can analyze text content and extract textual requirements and key points of design briefs as prompts; generate conceptual imagery, master-plan concept drawings, and node renderings; use image-to-image, sketch/line-drawing/CAD drawing control to constrain scheme form; use style models to control drawing style; batch-generate drawing variants and compare parameters in workflows; and use large language models to organize constraint lists, generate evaluation tables, and draft scheme narratives. For residential children's play areas, for example, AI already has discussable output capacity in zoning imagery, equipment combination atmosphere, multi-style variants, and caregiver seating scenes.

However, AI still has tasks it cannot adequately perform, such as final review of mandatory codes and legal responsibility; adaptation of schemes to site scale, grading, and drainage; plant adaptation to local environments and accuracy of seasonal performance; conflict handling and coordination with other disciplines; structural and constructional accuracy; cost expectation and scheme adaptation; and stable and controllable spatial logic and implementability. For children's play areas, spatial qualities such as safety clearances between equipment, actual

equipment heights, and the balance between sunlight and shade cannot be accurately represented by generated images.

Accordingly, this paper summarizes the capability–boundary matrix of AI in the schematic design phase in Table 2. The reasonable positioning of AI is a tool for rapid reasoning and variant comparison under constraints, rather than a subject of scheme decision-making [3]. Designers shift from design operators to constraint definers, output evaluators, and scheme integrators. This table lays the foundation for the credibility and operability of the subsequent framework.

Table 2. Capability–boundary matrix of AI in the schematic design phase

Capability dimension	Currently achievable	Currently difficult	Suggested human–AI division
Requirement translation	Text-to-prompt; draft constraint lists	Identification of tacit constraints	AI drafts; designer verifies
Concept generation	Imagery; master-plan concepts; node atmosphere	Stable spatial logic	AI generates; designer selects
Multi-scheme variants	Batch variants; style comparison	Engineering feasibility of variants	AI batches; designer compares
Constraint control	Control drawings; negative prompts; rule tables	Final code review; safety clearance verification	AI assists; designer reviews
Evaluation feedback	Evaluation table generation; preliminary constraint satisfaction assessment	Engineering judgment; maintenance judgment	AI preliminary assessment; human final assessment
Engineering implementation	Scheme narrative drafting	Grading; drainage; construction; cost	Human-led

A review of generative AI in architecture and engineering also points out that precisely mapping vague natural-language descriptions into compliant geometric constraints remains a core challenge, and that engineering feasibility and code compliance cannot be directly guaranteed at the image level [4].

4. Constraint-Driven Scheme Reasoning Framework

Combining the landscape design process with AI translation of constraints, this paper divides the scheme reasoning framework into three stages.

Stage A: Early constraint coding. The task is constraint identification, layering, and translation. In this stage, a constraint list (layered into hard/soft/tacit), prompts and negative prompts, control condition drawings (setback, fire access, sunlight, and zoning diagrams), rule tables, and evaluation indicators are produced according to the three-tier constraint system.

Stage B: Middle generative scheme reasoning. The task is batch scheme generation and multi-scheme variants after constraint inputs: control condition drawings constrain the form of

generated drawings, prompt weights express the priority of soft constraints, and designers conduct manual selection and local refinement.

Stage C: Late scheme evaluation and iteration. The task is to evaluate constraint satisfaction, review engineering feasibility, and make designer decisions. Evaluation results are fed back to the constraint coding in Stage A to form an iteration.

The framework contains two loops: an AI rapid loop and a human decision loop. The AI rapid loop occurs in Stage B, addressing variant quantity and design direction through "generate–select–regenerate." The human decision loop connects Stage C and Stage A, addressing whether constraints are accurate and whether schemes are implementable through "constraint calibration–engineering judgment–scheme integration." The division is that AI rapidly explores possibilities under given constraints, while humans judge whether the constraints themselves are correct and whether outputs are acceptable.

The "three-tier constraints–three-stage–dual-loop" human–AI collaborative framework enables AI outputs to be evaluated with evidence, the evaluation process to be traceable, and generated results to be iteratively updated, thereby upgrading aimless AI drawing generation to constraint-driven scheme reasoning.

5. Exploratory Practice: Residential Children's Play Area

The exploratory case is a children's play area in a residential cluster. The site must satisfy hard constraints such as fire safety, sunlight duration, and accessibility; accommodate soft constraints such as community character, age zoning, and caregiver seating; and is affected by tacit constraints such as property operation capacity, cost control, and equipment procurement preferences. In scheme reasoning, a constraint list was first established according to Table 1; constraint coding was completed according to design needs; fire access, sunlight, grading, and zoning control drawings were prepared; and scheme variants with constraint-driven prompts were batch-generated through a workflow to determine the design direction. Finally, designers checked item by item according to the evaluation methods in Table 1, recorded omitted constraints, rework counts, and scheme discussability, and fed evaluation results and optimization directions back to the constraint list for a new round of iteration. This process was repeated until a scheme was collaboratively reasoned out.

The exploratory practice preliminarily shows that variants generated with constraint-driven prompts were clearly superior to those without constraints in zoning rationality, supervision sightlines, and safety clearance awareness; omitted constraints were reduced, and the proportion of schemes entering discussion increased. However, grading, drainage, equipment safety clearances, and cost control still required manual review, and the translation of tacit constraints remained highly dependent on professional experience. This is consistent with existing findings that engineering-experience knowledge is difficult for general generative models to capture stably [5]. The results do not support the judgment that "AI can automatically complete scheme reasoning," but they support the framework proposition that "constraint coding can improve the usability of AI outputs." The practice also exposed two problems: control drawing production is time-consuming, and the completeness of the constraint list depends on project experience; setting prompt weights for soft constraints involves trial-and-error costs. These two issues constitute directions for subsequent workflow optimization.

6. Conclusion

Taking constraints as the organizing logic, this paper proposes a three-tier constraint system and a "three-stage–dual-loop" human–AI collaborative framework for generative AI application in the landscape schematic design phase, and conducts an exploratory practice involving a

residential children's play area. The conclusions are as follows. First, AI application in the schematic design phase should be organized by constraints rather than styles, and the quality of constraint translation determines output usability. Second, hard constraints should be translated into control drawings and rule tables, soft constraints into prompt weights and diagrams, and tacit constraints should explicitly enter the human decision loop. Third, generative AI currently excels at variant generation and imagery reasoning under constraints, while engineering judgment and responsibility decisions should remain with designers. This conclusion echoes Zhou and Zhang's judgment that AI prompts designers to shift from executors to guides, integrators, and judges: the significance of the constraint-driven framework lies in translating this role shift into operational process division rather than remaining at the level of principle. Future research will improve the constraint library and evaluation indicators, conduct multi-project validation, and promote workflow templating.

References

[1] Wang, S. J., Tian, J. W., Wang, G. Q., et al. (2026). Site constraint transfer and multi-model collaboration in auxiliary generation of rural landscape plans. Landscape Architecture. https://doi.org/10.3724/j.fjyl.LA20260407. (in Chinese)

[2] Wei, C., Du, Y., Huang, X., et al. (2026). A cross-stage human–GenAI collaborative workflow for landscape architecture design: a pocket park feasibility study. Buildings, 16(19), 3819. https://doi.org/10.3390/buildings16193819

[3] Zhu, Z. B., Song, Y. F., Yue, B. R., et al. (2026). Three roles of AI in knowledge production in landscape planning and design practice: servant, opponent, and partner. Chinese Landscape Architecture, 2026(3). (in Chinese)

[4] Generative AI-driven architectural and structural engineering designs: a review. (2026). In IEEE.

[5] Shao, Y. H., Lin, P. Y., & Liu, S. (2026). Context modeling and application boundaries of generative artificial intelligence in landscape architecture design assistance. Journal of Chinese Urban Forestry, 24(2), 103–109. (in Chinese)