Journal of Iranian Architecture & Urbanism
(JIAU)
Homepage: https://www.isau.ir/

ORIGINAL RESEARCH PAPER

Optimization of spatial organization in architectural plan design using particle swarm optimization algorithm*

Maryam Sadeghian [1] , **Akram Hosseini** [2,**]

[1] *M.A. in Architecture, Faculty of Architecture and Urbanism, Ferdowsi University of Mashhad, Mashhad, Iran.*
[2] *Assistant Professor, Faculty of Architecture and Urbanism, Ferdowsi University of Mashhad, Mashhad, Iran.*

ARTICLE INFO

Article History:

Received	2020/03/05
Revised	2020/09/01
Accepted	2021/04/07
Available Online	2022/09/22

Keywords:

Optimization
Spatial Organization Design
Architecture Plan
Particle Swarm Optimization Algorithm

Use your device to scan and read the article online

Number of References
73

Number of Figures
9

Number of Tables
1

Extended ABSTRACT

BACKGROUND AND OBJECTIVES: Today, due to the wide range of variables affecting architectural design, the computer is used as a tool in interaction with the design process to find optimal and high-performance solutions. It is necessary to investigate these methods due to their limitations and the complexity of space planning regarding the effective number of parameters. Improving the design quality and construction of architectural works is a common concern in developing countries, a step that can promote contemporary Iranian architecture. Furthermore, one of the design and construction characteristics in these developing countries is the tendency to improve the quality of architecture by using new technologies.

METHODS: Space planning in architecture is one of the most practical and complex issues in architectural design and is considered one of the most challenging issues in recent research. The use of new technology-dependent methods in design, especially with an emphasis on using evolutionary algorithms as a solution, has been considered in the present research. Therefore, this study investigates the application and use of these algorithms as a solution for design optimization. The research questions of this study are:
- Concerning the application of optimization algorithms in space planning design, which algorithms have been used as basic or complementary algorithms? How frequently have the Swarm Intelligence Algorithms, especially particle swarm optimization algorithms, been used in this regard?
- What are the possibilities and limitations in spatial organization design in architecture using genetic algorithms compared to the particle swarm optimization algorithm as the two main evolutionary algorithms?
- What is the implementation process and application of the particle swarm optimization algorithm in spatial organization design in architecture?
In order to answer the research questions, 35 types of research that have used optimization algorithms in architectural spatial planning design are collected. Then, the content analysis method was used to extract all the variables used in architectural plan optimization. In the same way, the basic algorithm and the complementary algorithms, if any, were extracted. Based on the study of specific sources regarding the evolutionary optimization algorithm from the available bibliographic resources and the analysis of planning requirements and architectural space design, the genetic algorithm and swarm particle optimization algorithm were explained in designing the architectural spatial organization. The speed and quality of these two algorithms in investigating the research problem have been scrutinized based on software capacities in algorithm implementation and possibilities and limitations in using analytical methods for designing architectural plans. After explaining the objectives and numerical criteria, the particle swarm optimization algorithm using Microsoft Visual Studio programming software and .NET programming platform in C #, with WinodowsForm graphical user interface, was used to monitor the algorithm developing process and its results further.

FINDINGS: This research introduces the features of metaheuristic algorithms and presents various optimization algorithms, including deterministic, heuristic, and

© 2022, JIAU. All rights reserved.

https://dx.doi.org/10.30475/ISAU.2021.222373.1370

OPEN ACCESS

* This article is derived from the first author`s master thesis entitled "Design development of high-rise apartment complex of Mashhad based on evolutionary optimization in design process ", supervised by the second authors, at Ferdowsi University of Mashhad.

** Corresponding Author:
Email: *akram.hosseini@um.ac.ir*
Phone: +98(51)38805421

Extended ABSTRACT

metaheuristic algorithms. Also, the application of optimization algorithms in architecture is explained. The background of the evolutionary optimization algorithms and swarm particle optimization in the architectural plan design were also examined. After comparing the performance of the genetic algorithm and particle swarm optimization algorithm in optimizing the spatial organization, the swarm optimization algorithm structure was introduced. Finally, the application of this algorithm in plan design was studied and explained. Accordingly, plan design algorithms were classified into three stages: In the first stage, the goals, criteria, and constraints affecting the architectural spatial organization were determined, and they were classified into (1) The primary criteria and constraints, (2) The designers' criteria and constraints, and (3) The contacts' (client and users) criteria and constraints. In the numeric stage, the criteria of land boundary, list of spaces, permissible aspect ratios, permissible dimensions, total area, space interference, space adjacency, daylighting, and verification of the spaces were quantified. Then, the algorithm is determined based on the particle swarm optimization algorithm in two steps. Finally, the implementation platform of the algorithm is determined.

CONCLUSION: The difficulty of working with programming languages, software skills, and the software complexity due to inadequacy to combine various scientific fields has made using programming languages uncommon for designers to control the design parameters. This study compared the performance of two genetic algorithms and particle swarm optimization as the representatives of the two main groups of evolutionary algorithms in a base problem. It was shown that the particle swarm optimization algorithm converges faster and has a higher quality to optimize the plan regarding the parameters affecting the plan design. Implementation of an operational solution to optimize the spatial organization of the plan with emphasis on the affecting parameters in the formation of architectural plans was proposed in a three-step process using this algorithm. The problem objectives were examined in three related areas. The quantification process and the final model implementation were completed based on the particle swarm optimization algorithm in the .NET programming platform, along with a graphical interface as user interfaces for architects to understand the implementation process better. In addition to achieving optimal plans, future research interests in this field were also introduced. Regarding the specific entity of architecture, the existing computer software is insufficient for implementing the algorithm, providing visual and operational efficiency, and needs to be developed and customized for broader application in various fields of design.

HIGHLIGHTS:

- After comparison of two genetic algorithms and particle swarm optimization as representatives of the two main groups of evolutionary algorithms in a base problem in the field of space system design, particle swarm optimization algorithm was selected due to faster convergence and higher quality.
- Implementation of practical solution to optimize the spatial system, was done in 3 sepes including 1- defining problem goals, criteria and constraints, 2- numericalize and quantifying criteria and constraints and 3- determining fitness function and implementing it.
- The objectives of the spatial system optimization problem were studied in three general fields of Primary, Designers and Contacts Criteria and constraints. To achieve a better dominance and understanding of the process, the final model was implemented based on the particle swarm optimization algorithm that was integrated with a graphical user interface in the .NET programming platform.

ACKNOWLEDGMENTS:

This research did not receive any specific grant from funding agencies in the public, commercial, or not-forprofit sectors.

CONFLICT OF INTEREST:

The authors declared no conflicts of interest.

COPYRIGHTS

©2022 The author(s). This is an open access article distributed under the terms of the Creative Commons Attribution (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, as long as the original authors and source are cited. No permission is required from the authors or the publishers. (https://creativecommons.org/licenses/by/4.0/).

HOW TO CITE THIS ARTICLE

Sadeghian, M.; Hosseini, A., (2022). Optimization of spatial organization in architectural plan design using particle swarm optimization algorithm. *Journal of Iranian Architecture & Urbanism*., 13(1): 19-38.

https://dx.doi.org/10.30475/ISAU.2021.222373.1370

https://www.isau.ir/article_129458.html

نشریه علمی معماری و شهرسازی ایران
(JIAU)
https://www.isau.ir/

مقاله علمی- پژوهشی

بهینه‌سازی نظام فضایی در طراحی پلان معماری با استفاده از الگوریتم بهینه‌سازی انبوه ذرات*

مریم صادقیان [۱]، اکرم حسینی [۲و**]

۱. کارشناسی ارشد معماری، دانشکده معماری و شهرسازی، دانشگاه فردوسی مشهد، مشهد، ایران.
۲. استادیار، دانشکده معماری و شهرسازی، دانشگاه فردوسی مشهد، مشهد، ایران.

مشخصات مقاله

تاریخ ارسال ۱۳۹۸/۱۲/۱۵
تاریخ بازنگری ۱۳۹۹/۰۶/۱۱
تاریخ پذیرش ۱۴۰۰/۰۱/۱۸
تاریخ انتشار آنلاین ۱۴۰۱/۰۶/۳۱

واژگان کلیدی

بهینه‌سازی
طراحی نظام فضایی
پلان معماری
الگوریتم بهینه‌سازی انبوه ذرات

چکیده

امروزه با توجه به گستردگی متغیرهای تاثیرگذار بر طراحی معماری، از کامپیوتر به عنوان ابزاری در تعامل با فرایند طراحی جهت یافتن راه‌حل‌های بهینه و با کارایی بالا استفاده می‌گردد. با توجه به محدودیت های موجود در استفاده از این روش‌ها برای طراحان و مطرح بودن طراحی نظام فضایی پلان به عنوان یکی از پرسابقه‌ترین پرچالش‌ترین مسائل طراحی از نظر تعداد پارامترهای تاثیرگذار، در این تحقیق پس از بررسی انتشاراتی که در آن‌ها طراحی نظام فضایی معماری با استفاده از الگوریتم‌های بهینه‌سازی انجام شده است به شیوه تحلیل محتوای متن و مقایسه دو الگوریتم ژنتیک و بهینه‌سازی انبوه ذرات به عنوان نمایندگان دو گروه از الگوریتم‌های تکاملی در یک مساله‌ی مبنا، الگوریتم بهینه‌سازی انبوه ذرات به دلیل همگرایی سریع‌تر و کیفیت بالاتر انتخاب گردید. پس از معرفی و بررسی امکانات، محدودیت‌ها و نحوه‌ی کاربرد آن در طراحی نظام فضایی پلان، پیاده‌سازی راهکار عملیاتی آن برای بهینه‌سازی نظام فضایی شامل ۱- تعیین اهداف مساله، معیارها و محدودیت‌ها، ۲- عددی‌سازی و کمی‌کردن معیارها و محدودیت‌ها و ۳- تعیین تابع برازندگی و پیاده‌سازی آن در پلتفرم برنامه‌نویسی .NET و به زبان C# همراه با محیط گرافیکی به‌عنوان رابط کاربری جهت تسلط و درک بیشتر کاربران از فرآیند انجام گردید.

نکات شاخص

- پس از مقایسه دو الگوریتم ژنتیک و بهینه‌سازی انبوه ذرات به عنوان نمایندگان دو گروه از الگوریتم‌های تکاملی در یک مساله‌ی مبنا در حوزه نظام فضایی، الگوریتم بهینه‌سازی انبوه ذرات به دلیل همگرایی سریع‌تر و کیفیت بالاتر انتخاب گردید.
- پیاده‌سازی راهکارعملیاتی برای بهینه‌سازی نظام فضایی در ۳ مرحله ۱- تعیین اهداف مساله، معیارها و محدودیت‌ها، ۲- عددی‌سازی و کمی‌کردن معیارها و محدودیت‌ها و ۳- تعیین تابع برازندگی و پیاده‌سازی.
- اهداف مساله بهینه‌سازی نظام فضایی در سه حوزه کلی معیارها و محدودیت‌های اولیه، طراح و مخاطبین بررسی گردید و مدل نهایی‌شده بر اساس الگوریتم بهینه‌سازی انبوه ذرات در پلتفرم برنامه‌نویسی .NET همراه با یک محیط گرافیکی به عنوان رابط کاربری جهت تسلط و درک بیشتر از فرآیند پیاده‌سازی شد.

نحوه ارجاع به مقاله

صادقیان، مریم و حسینی، اکرم. (۱۴۰۱). بهینه‌سازی نظام فضایی در طراحی پلان معماری با استفاده از الگوریتم بهینه‌سازی انبوه ذرات، *نشریه علمی معماری و شهرسازی ایران*، ۱۳(۱)، ۱۹-۳۸.

* این مقاله برگرفته از پایان‌نامه کارشناسی ارشد نویسنده نخست با عنوان «توسعه طرح مجموعه آپارتمان‌های مرتفع مشهد مبتنی بر روش‌های بهینه‌سازی تکاملی در فرایند طراحی» می‌باشد که به راهنمایی نویسنده دوم در دانشگاه فردوسی مشهد انجام گرفته است.

** نویسنده مسئول
تلفن: ۰۰۹۸۵۱۳۸۸۰۵۴۲۱
پست الکترونیک: akram.hosseini@um.ac.ir

مقدمه

ارتقای کیفیت در طراحی و ساخت آثار معماری دغدغه‌ای مشترک در میان کشورهای در حال توسعه است؛ گامی که می‌تواند موجبات اعتلای معماری معاصر ایران را نیز فراهم آورد. یک از مشخصه‌های کیفیت طراحی و ساخت در معماری این کشورها، گرایش‌هایی است که با استفاده از فناوری‌های نوین به دنبال ارتقاء کیفیت در معماری می‌باشند (Mahdavinejad, 2014).

پس از گسترده‌شدن استفاده از فناوری‌هایی چون کامپیوتر در معماری جهت افزایش سرعت و دقت محاسبات، بهینه‌سازی طرح‌های معماری نیز به یکی از مباحث مورد توجه محققین تبدیل شد. بهینه‌سازی روشی برای جستجوی بهترین راه‌حل در یک شرایط خاص است. هدف بسیاری از مسائل بهینه‌سازی عملی، همانند مسائل نظری جستجویی برای بهترین پیکربندی از متغیرها برای دستیابی به اهداف مساله مورد نظر می‌باشد (Blum & Roli, 2003, 296). همراهی این روش با تکنیک تکامل به‌عنوان یکی از روش‌های کارا در زمینه‌ی بهینه سازی، در سال‌های اخیر، به دلیل وابستگی به سیستم‌های اجتماعی و طبیعی، به یک زمینه تحقیق جذاب تبدیل شده است. این تکنیک با تکیه بر تکامل بیولوژیک، از روش‌هایی مانند تولید مثل، جهش، بازسازی و انتخاب طبیعی برای تولید راه‌حل‌های مناسب استفاده می‌کند (Juneja & Na-gar, 2016, 1). روش‌ها یا الگوریتم‌های بهینه‌سازی که بر پایه تکنیک تکامل شکل می گیرند، به افراد در یافتن پاسخ‌های بهینه یا نزدیک به بهینه کمک می‌کنند و طراحان را قادر می‌سازد که انتخاب‌های آگاهانه‌تری در فرآیند طراحی داشته باشند.

هم‌زمان با استفاده‌ی روزافزون الگوریتم‌های تکاملی در علوم مهندسی و همین‌طور حوزه‌های مختلف معماری و ساخت، مساله‌ی برنامه‌ریزی فضاهای معماری به‌عنوان یکی از پیچیده‌ترین مسائل در طراحی معماری برای یافتن مکان‌های مناسب برای مجموعه‌ای از اجزای فضا با توجه به معیارهای خاص طراحی، تعریف شده است؛ به گونه ای که نه تنها الزامات طراحی را برآورده می‌کند، بلکه کیفیت معماری را مطابق با ترجیحات طراحی به حداکثر رسانده و زیبایی‌شناسی و قابلیت استفاده را نیز برآورده می سازد (Dutta & Sarthak, 2011, 312).

در معماری، سامان‌دهی فضاها به معنای روابط فضاهای مختلف در کنار یکدیگر، جهت پاسخ گویی به عملکردها و کارکردهای مورد نیاز می باشد؛ به گونه‌ای که این نوع سامان‌دهی در بهترین حالت نسبت به دیگر انواع سامان‌دهی قرار گیرد (Caldas & Nor-ford, 2002). هدف از طرح این مسائل که تحت عنوان مساله تخصیص فضا در طراحی به کمک کامپیوتر نیز شناخته می‌شوند، ارائه‌ی الگوریتمی است که بتواند یک طرح یا چینش فضایی را بر اساس محدویت‌های توپولوژیک و هندسی پیشنهاد دهد. این محدودیت‌ها می‌توانند از سمت طراح تحت تاثیر عوامل عینی و ذهنی مختلفی قرار گیرند (Rahbar et al., 2019).

در این تحقیق، ضمن معرفی الگوریتم بهینه‌سازی انبوه ذرات در زمینه‌ی بهینه‌سازی، مزیت‌ها و محدودیت‌های کاربرد آن به‌عنوان یکی از الگوریتم‌های نسبتاً جدید و کارا در معماری تبیین می‌گردد. همچنین راهکار عملیاتی پیاده‌سازی این الگوریتم، جهت بهینه‌سازی نظام فضایی در پلان، در یک روش سه مرحله‌ای در بهینه‌سازی پلان یک واحد مسکونی تک‌خوابه ارائه می‌گردد.

مبانی نظری

الگوریتم‌های بهینه سازی

در تعریفی کلی، الگوریتم‌های بهینه‌سازی شامل سه دسته الگوریتم‌های معین[۱]، اکتشافی[۲] و فرااکتشافی[۳] می‌باشند. زمانی که روند پیشرفت الگوریتم بدون هیچ تصادف و احتمالی باشد و الگوریتم از یک مقدار اولیه شروع و به مقدار مشخصی برسد، این روش یک الگوریتم معین نامیده می‌شود (Yang & Karamanoglu, 2013, 6). یک الگوریتم اکتشافی اطلاعات مربوط به سیستم را جمع‌آوری می‌کند، راه‌حل‌های تصادفی را تست کرده و برای تولید راه‌حل بعدی تصمیم گیری می‌کند. بنابراین، این روش‌به ماهیت مساله وابسته می‌باشد (Datta et al., 2019, 3). لذا این الگوریتم برای یافتن راه‌حل نزدیک به بهینه طراحی شده است (Hansen et al., 2010, 96). الگوریتم‌های فرااکتشافی، اکتشاف‌ها و تابع هدف را بدون وابستگی به ساختار مسئله به صورت کارآمد ترکیب می‌کنند (Datta et al., 2019, 3). آن‌ها در مقام راهنمای جستجو، استراتژی‌های سطح بالا برای یافتن فضاهای جستجو با استفاده از روش‌های مختلف هستند (Blum & Roli, 2003, 271).

در منابع مختلف تقسیم‌بندی‌های مختلفی برای الگوریتم‌های فرااکتشافی موجود می‌باشد. یک از آن‌ها تقسیم‌بندی بر اساس تعداد عامل‌های تاثیرگذار در فرایند بهینه‌سازی می‌باشد. مطابق با آن می‌توان الگوریتم‌های فرااکتشافی را به سه دسته گسترده تقسیم نمود.

روش‌های خط سیر[۴] یا تک نقطه‌ای که روش‌هایی هستند که با یک پاسخ واحد سروکار دارند. مانند الگوریتم تبرید شبیه‌سازی شده (SA)[۵]؛ روش‌های مبتنی بر جمعیت[۶] که با مجموعه‌ای از پاسخ‌های مختلف در فرایند بهینه‌سازی سروکار دارند که شامل الگوریتم‌هایی مانند ژنتیک(GA)[۷]، بهینه سازی کلونی مورچه‌ها(ACO)[۸] و بهینه‌سازی ذرات (PSO)[۹] می‌باشد و در انتها روش‌های ترکیبی[۱۰] نیز ترکیب روش‌های جستجوی تک نقطه‌ای و مبتنی بر جمعیت می‌باشد (Baghel et al., 2012).

علاوه بر آن سه گروه اصلی از الگوریتم‌های فرااکتشافی را بر اساس ماهیت پیدایش می‌توان به صورت زیر معرفی نمود.

الگوریتم‌های محاسبات تکاملی (EC)[11]

روش‌های محاسبات تکاملی با تاریخی غنی، یک روش کلی برای حل مسائل بهینه‌سازی هستند که تابع هدف را عموما بدون استفاده از بینش عمیق‌تر و با توجه به خواص ریاضی آن، به شیوه انتزاعی و کارآمد مورد استفاده قرار می‌دهد. این تکنیک‌ها می‌توانند مسائل غیر محدب، غیرخطی و چندبعدی را با محدودیت‌های خطی یا غیرخطی با متغیرهای پیوسته یا گسسته حل‌کنند(Cuevas et al., 2019, 8).

الگوریتم ژنتیک به عنوان پایه‌ای‌ترین و شناخته‌شده‌ترین الگوریتم این گروه که از دهه‌ی ۶۰ میلادی به بعد به‌عنوان روشی تصادفی برای حل مسائل بهینه‌سازی و جستجو استفاده می‌شود، شامل عملیاتی بر روی یک جمعیت از راه‌حل های ممکن است. با توجه به نظریه تکامل داروین، با استفاده مکرر از این روش، گونه‌های اولیه به گونه های جدید تغییر پیدا کرده و فقط آن‌هایی که به‌عنوان راه‌حل مناسب‌ترند، باقی می‌مانند (Fasoulaki, 2007). الگوریتم‌های برنامه‌نویسی ژنتیک (GP)[12]، استراتژی تکاملی (ES)[13] و تکامل تفاضلی(DE)[14] از دیگر الگوریتم‌های محاسبات تکاملی می‌باشد.

الگوریتم‌های هوش جمعی (SI)[15]

رویکرد الگوریتم هوش جمعی الهام گرفته از درک علمی انسان از سیستم‌های زیست‌شناختی، طبیعی یا اجتماعی است که در برخی از سطوح انتزاع، می‌توانند به عنوان فرآیندهای بهینه‌سازی مطرح گردند (Cuevas et al., 2019, 6). هوش جمعی بخشی از محاسبات تکاملی است که رفتار اجتماعی جمعی از سیستم‌های غیرمتمرکز و خودسازمان‌یافته‌ی طبیعی یا مصنوعی را بررسی می‌کند (Zhang, Wang, & Ji, 2015, 1). این گونه روش‌ها که مبتنی بر تعاملات و رفتار اجتماعی دسته‌های پرندگان و ماهی‌ها و شامل انواع مختلفی می‌باشند، بر ظرفیت این گروه‌ها برای هوشمندانه عمل کردن به عنوان یک کل تاکید می‌نمایند.

الگوریتم بهینه‌سازی انبوه ذرات نوعی سیستم بیولوژیکی است که در آن رفتارهای جمعی افراد در تعامل با یکدیگر و محیط آن‌ها، فرآیند بهینه‌سازی را تشکیل می‌دهند. آنها از حرکت جمعی پرندگان و ماهی‌ها الهام گرفته و شبیه‌سازی‌شان به واسطه مدل‌سازی «رفتار اجتماعی انسان» انجام می‌شود (Juneja & Nagar, 2016, 2).

بهینه‌سازی کلونی مورچه‌ها یک روش فرااکتشافی است که از به جای ماندن آثار ماده‌ای به نام فِرمون[16] در مسیر برخی از گونه‌های مورچه‌ها، الهام گرفته شده است. مورچه‌ها از فِرمون، به‌عنوان وسیله‌ای ارتباطی جهت تشخیص مسیر صحیح استفاده می‌کنند (Dorigo & Stützle, 2019, 311).

الگوریتم‌های مرتبط با فیزیک[17]

این روش‌ها تکنیک‌های بهینه‌سازی الهام‌گرفته از فیزیک می‌باشند. الگوریتم تبرید شبیه‌سازی‌شده متداول‌ترین ابزار بهینه‌سازی در این گروه، الهام گرفته از یکی از مراحل ذوب فلزات می باشد (Datta, Roy, & Davim, 2019, 4)، تشکیل رودخانه پویا[18] و چکه‌های آب هوشمند[19] از دیگر الگوریتم‌های این گروه می‌باشند.

کاربرد الگوریتم‌های بهینه‌سازی در معماری

امروزه با توجه به گستردگی متغیرهای تاثیرگذار بر طراحی، از کامپیوتر به عنوان ابزاری مولد استفاده می‌گردد تا فضای جستجو را برای یافتن راه‌حل با کارایی بالا برای پاسخ مساله، به گونه‌ای که تمامی حالات در مسیر طراحی دیده شود، بررسی کند. در این روش کامپیوتر به صورت اتوماتیک، گزینه‌های ممکن را تولید و ارزیابی می‌کند و به طراح پاسخ‌های بهینه یا نزدیک به بهینه را ارائه می‌دهد (Caldas & Norford, 2002, 173).

یکی از اهداف به کار بردن کامپیوتر، هوش مصنوعی و الگوریتم‌ها در فرآیند طراحی سپردن کارهای محاسباتی همراه با تکرار زیاد به آن می‌باشد. تعریف ورودی برای کارهای کمی کامپیوترها و کارهای کیفی که نیاز به ذهن پیچیده بشر دارد، بر عهده طراح می‌باشد. هدف نه جایگزین کردن نوآوری معمار با بهره‌وری کامپیوتر، بلکه استفاده از ابزار مفید و کاربردی کامپیوتر برای کارهای تکراری در برنامه‌های پیچیده طراحی می‌باشد(Rodrigues et al., 2013, 1).

پیشرفت‌ها در ابزارهای محاسباتی طراحی در پیوند با هوش مصنوعی به امکاناتی منتهی گشته که کامپیوترها می‌توانند کاملاً با فرایند طراحی تعامل داشته باشند (Caldas & Norford, 2002, 173). اگرچه در کنار تمام قدرتی که ابزارهای دیجیتالی در کنار تنوع در اختیار طراحی قرار می‌دهند، این طراح است که به مدیریت ابزار و جهت‌دهی به آن می‌پردازد. بنابراین فرض این‌که ماشین‌ها به‌جای انسان طراحی می‌کنند منتفی خواهد بود (Mahdav-inejad & Refalian, 2014). الگوریتم‌های بهینه‌سازی فرااکتشافی این پتانسیل را دارند تا دید عمیق‌تری را نسبت به فضای طراحی به کاربر داده و با حفظ یک مقدار متعادل از همگرایی و واگرایی منجر به طرحی خلاقانه گردند (Dino & Üçoluk, 2017, 1).

الگوریتم ژنتیک، به‌عنوان یک روش محاسباتی مبتنی بر اساس اصول تکامل، به معماری معرفی شده‌است تا مشکلات فرم و عملکرد در پروژه‌های معماری را سامان‌دهی کند (Fasoulaki, 2007) و به‌طور ویژه در بهینه‌سازی طراحی پلان‌های طبقات

(Michalek & Papalambros, 2002)، سایت‌پلان‌ها (Finucane et al., 2006)، بهینه‌سازی‌های طراحی نما (Caldas & Norford, 1999)، بهینه‌سازی سازه فرم‌های ساختمانی (Papavlou & Turner, 2009) و در چند طرح مفهومی (Soddu, 2005) استفاده گردید (Li, 2012, 392). با وجود این‌که اکثر الگوریتم‌های مبتنی بر محاسبات تکاملی در ابزارهای بهینه‌سازی توسط معماران مورد استفاده قرار می‌گیرند(Cichocka et al., 2017, 154) الگوریتم‌های بهینه‌سازی ژنتیک و انبوه ذرات به الگوریتم‌های موثر و کارآمدتری برای حل مسائل بهینه‌سازی پیچیده تبدیل شده‌اند (Juneja & Nagar, 2016, 1).

روش تحقیق

همزمان با افزایش دغدغه ارتقای کیفیت طراحی و ساخت در ایران و جهان، گستردگی متغیرهای تاثیرگذار بر طراحی می‌تواند کنترل عوامل مختلف را برای طراح با مشکل روبرو سازد. ابزارهای کامپیوتری نیز علی‌رغم پیشرفت‌های فراوان بیشتر به عنوان ابزارهای ترسیم و ارائه‌ی کار در بین طراحان شناخته‌شده و ایفای نقش می‌کنند.

- در تحقیقات و پروژه‌های حوزه معماری چه روش‌ها و ابزارهایی جهت کنترل پارامترهای مختلف و در نتیجه بهینه‌ساختن طرح پیشنهاد گردیده‌است؟

- علی رغم پیشرفت روزافزون ابزارها و نرم‌افزارهای کامپیوتری، دلیل رایج نبودن این ابزاها در بین طراحان این حوزه جهت کنترل پارامترهای موثر بر طراحی چه عواملی می‌باشد؟

مساله‌ی برنامه‌ریزی فضاهای معماری به‌عنوان یکی از پرکاربردترین و در عین حال پیچیده‌ترین مسائل طراحی معماری از پرچالش‌ترین مسائل در تحقیقات اخیر بوده‌است، استفاده از روش‌های نوین و وابسته به فناوری در طراحی به ویژه با تاکید بر استفاده از الگوریتم‌های تکاملی به عنوان راهکار در تحقیقات اخیر مورد توجه می باشد، بررسی کاربرد و چگونگی استفاده از این الگوریتم‌ها به‌عنوان راهکاری جهت بهینه‌سازی از دیگر سوالات این تحقیق می‌باشد. دیگر سوالاتی که این پژوهش به دنبال پاسخ به آن است در ادامه آورده شده است.

- در تحقیقاتی با موضوع کابرد الگوریتم‌های بهینه‌سازی در طراحی نظام فضایی چه الگوریتم‌هایی به‌عنوان الگوریتم پایه یا مکمل کاربرد داشته‌اند و میزان استفاده از الگوریتم‌های هوش جمعی و به‌ویژه الگوریتم بهینه‌سازی انبوه ذرات چگونه می‌باشد؟

- چه امکانات و محدودیت‌هایی در طراحی نظام فضایی معماری با استفاده از الگوریتم ژنتیک در مقایسه با الگوریتم بهینه‌سازی انبوه ذرات به عنوان نمایندگان اصلی دو گروه از الگوریتم‌های تکاملی وجود دارد؟

- فرایند پیاده‌سازی و کابرد الگوریتم بهینه‌سازی انبوه ذرات در طراحی نظام فضایی معماری چگونه می‌باشد؟

جهت پاسخ گویی به سوالات تحقیق، مطابق با جدول (۱)، تعداد ۳۵ پژوهش که در زمینه‌ی استفاده از الگوریتم‌های بهینه‌سازی در طراحی نظام فضایی معماری انجام شده‌اند، گردآوری و سپس به شیوه تحلیل محتوای متن تمامی متغیرهایی که بهینه‌سازی پلان معماری بر اساس آن‌ها انجام شده‌است، استخراج و تدوین گردید. با همین شیوه الگوریتم پایه و در صورت وجود الگوریتم مکمل مورد استفاده نیز استخراج شد.

بر اساس مطالعه منابع تخصصی در خصوص الگوریتم بهینه‌سازی تکاملی از منابع کتابشناختی موجود و هم چنین مطالعه پیش‌نیازهای برنامه‌ریزی و طراحی فضای معماری و نحوه به کاربردن الگوریتم ژنتیک و الگوریتم بهینه‌سازی انبوه ذرات در طراحی نظام فضایی معماری تدوین و تبیین گردید و پس از مقایسه سرعت و کیفیت این دو الگوریتم در پاسخ به مساله‌ی مبنای طراحی‌شده، با توجه قابلیت و ماهیت نرم افزارهای مورد نیاز برای پیاده‌سازی الگوریتم، امکانات و محدودیت‌های روش مذکور در طراحی پلان‌های معماری به شیوه تحلیلی بررسی و پس از تبیین اهداف مساله و عددی‌سازی معیارها، پیاده‌سازی یک نرم‌افزار مبتنی بر الگوریتم بهینه‌سازی انبوه ذرات با استفاده از نرم‌افزار برنامه‌نویسی Microsoft Visual Studio و پلتفرم برنامه‌نویسی.NET و به زبان C#، همراه با محیط گرافیکی Winodows Form به‌عنوان رابط کاربری، جهت تسلط و درک بیشتر از روند پیشرفت الگوریتم و نتایج به‌دست‌آمده، انجام گردید.

پیشینه کاربرد الگوریتم‌های بهینه‌سازی تکاملی و بهینه‌سازی انبوه ذرات در طراحی پلان معماری

روش‌های نخستین استفاده از الگوریتم‌های تکاملی در معماری، در سال ۱۹۹۵ توسط جان فریزر[۲۰] معرفی گردید. نتیجه پژوهش‌های وی، منتهی به ایده‌ی طراحی فضای پاسخگو به نیازهای مختلف بود و در بیشتر موارد فعالیت‌های وی با ساختار خودساز مدل مطالعاتی در محیطی پویا انجام می‌گرفت که منجر به رویکردی با عنوان فرایند طراحی پارامتریک در مسیر کنترل پارامترهای مؤثر بر طراحی شد (Mardomi et al., 2015). اما در چند دهه‌ی اخیر بسیاری از محققین بر روی موضوع تعریف الگوریتم جهت بهینه‌سازی چیدمان فضاها کار کرده‌اند. اولین تلاش‌ها به دهه‌ی ۱۹۶۰ باز می‌گردد که لوین[۲۱] کتابی را برای یافتن چیدمان بهینه با استفاده از گراف‌ها ارائه داد و پس از آن گریسون[۲۲] در دهه‌ی ۱۹۷۰ از گراف‌ها برای ارائه‌ی پلان طبقات و برنامه‌ریزی فضاها استفاده کرد (Rahbar et al., 2019).

علاوه بر نظریه‌ی گراف‌ها استفاده از الگوریتم‌های تکاملی برای طراحی نظام فضایی در معماری در سال‌های ۱۹۹۰ آغاز شد و در سال‌های اخیر محبوبیت زیادی کسب کرده است (Dutta & Sarthak, 2011).

Table 1. Presentation of last researches data in the field of spatial system optimization in architecture

Author	Year	Spatial system optimization variables											Optimization methods			
		Shape and dimensions of the site	Distance of site components	Location of masses and spaces	Floor leveling	shape of the building boundry	Building orientation	Shape and dimensions of spaces	spaces adjacency	Spaces distance	Location and dimensions of openings	Furniture arrangement	Evolutionary Computation Algorithm	Swarm Itelligence Algorithm	physics-based Algorithm	Other Methods
Schnier and Gero	1996					*		*					GA			
Rosenman	1997							*					GA			
Gero and Kazakov	1997					*	*	*					GA			
Jagielski and Gero	1997					*	*	*					GP			
Jo and Gero	1998					*	*	*					GA			
Bentley	1998					*	*		*				GA			
Elezkurtaj and Franck	1999							*	*		*		GA			
Michalek and Papalambros	2002					*		*	*		*		GA		SA	SQP
Virirakis	2003							*	*		*	*	GP			
Wetter and Wright	2004				*								GA			
Bausys and Pankrašovaite	2005							*	*				GA			
Makris, et al.	2006					*		*			*		GA			
Wang et al.	2006					*							GA			
Homayouni	2007							*	*				GA			
Doulgerakis	2007					*	*	*	*		*		GA			
Banerjee et al.	2008					*		*	*				GA			
Inoue and Takagi	2008					*		*	*				GA			VD
Wong and Cha	2009								*				GA			
Thakur et al.	2010							*	*		*		GA			DA
Knecht and Koenig	2010					*		*	*				GA			K-DT
Flack	2011					*	*	*	*		*		GA- GP			
Poblete and de la	2011					*		*					GA	PSO		VD
Rodrigues et al.	2013					*		*	*	*			IVCGA			
Talbourdet et al.	2013					*		*	*	*			ES			SHC
Conceição et al.	2015		*										GA			
Su and Yan	2015									*			GA	PSO		
Wang and Malkawi	2015					*	*						GA			
Sönmez	2015					*	*	*	*				GA			
Cubukcuoglu et al.	2016							*	*		*	*	GA			
Papadaki and Chassiakos	2016	*	*	*									GA			
Nagy et al.	2017							*	*		*		MOGA			
Dino and Üçoluk	2017					*	*						GA			
Bahrehmand et al.	2017							*	*		*		GA			
RazaviAlavi and AbouRizk	2017	*	*	*									GA			
Kumar et al.	2017					*		*	*				GA			

تحقیقات متعددی با هدف مرور و بررسی آثاری که در آن‌ها از الگوریتم‌های تکاملی جهت بهینه‌سازی نظام فضایی پلان مورد استفاده قرار گرفته، ارائه شده است. مطالعه لیگت[۲۳] در سال ۲۰۰۰ با عنوان"طرح‌بندی خودکار تجهیزات: گذشته، حال و آینده" از اولین مطالعاتی است که طی آن تحقیقاتی با رویکرد استفاده از الگوریتم‌های محاسبات تکاملی برای مساله طراحی نظام فضایی معماری بررسی گردید. به دلیل نوپا بودن این رویکرد در این مطالعه تنها سه تحقیق مورد بررسی قرار گرفت (Liggett, 2000).

مطالعه دیگری که با عنوان "بررسی رویکردهای محاسباتی برای طرح‌بندی نظام فضایی (۲۰۰۰-۱۹۶۵)" در سال ۲۰۰۶ توسط همایونی انجام شد، رویکردهای مختلف محاسباتی اتخاذ شده برای چیدمان فضا را مطرح نمود (Homayouni, 2007).

در سال ۲۰۰۹ ژن و لین[۲۴] در تحقیقی با عنوان "روش‌های تکاملی برای اتوماسیون" به بررسی کاربرد محاسبات تکاملی در طراحی نظام های فضایی در معماری پرداختند (Gen & Lin, 2009).

در سال ۲۰۱۱ در تحقیقی با عنوان "مروری بر برنامه‌ریزی فضاهای معماری با استفاده از روش‌های محاسبات تکاملی" داتا و سارتک[۲۵]، ۱۶ تحقیق را با موضوع کاربرد الگوریتم های محاسبات تکاملی در طراحی پلان فضاهای معماری مورد بررسی قرار داده و نتایج گوناگونی مانند استفاده حداکثری از الگوریتم ژنتیک و تجاری نبودن تحقیقات را ارائه نمودند. آن‌ها اعتقاد داشتند روش‌های تکاملی با مدل‌های بهبود یافته‌تر می‌توانند در آینده به طور مؤثر برای بهینه‌سازی نظام فضایی در معماری به کار روند (Dutta & Sarthak, 2011, 320).

در سال ۲۰۱۳ ردریگیس[۲۶] و همکاران در مقاله‌ای با عنوان "استراتژی تکاملی ارتقا یافته با یک روش جستجوی محلی برای مساله‌ی تخصیص فضا در معماری"، ۶ رویکرد مختلف بر اساس نوع مساله‌ی تخصیص فضا مطرح کردند. این رویکردها شامل تخصیص ناحیه، تقسیم بندی ناحیه، تخصیص فضا، سلسله مراتب ساخت، اکتشاف مفهومی و تطبیق طراحی می‌باشد (Rodrigues et al., 2013).

در سال ۲۰۱۵ کالکستو و چلانی[۲۷] در تحقیقی با عنوان "مروری بر ادبیات بهینه‌سازی برنامه‌ریزی نظام فضایی با استفاده ازیکی از روش‌های الگوریتم تکاملی: ۱۹۹۲- ۲۰۱۴" به این نتیجه دست یافتند که توسعه محاسبات در سال‌های اخیر تأثیر چشمگیری بر توسعه مسائل مربوط به نظام‌های فضایی در معماری نداشته و الگوریتم‌های جدیدی برای بهینه‌کردن زمان پردازش نیاز است تا تعداد عناصر موجود در چیدمان فضاها افزایش یابد (Calixto & Celani, 2015, 669).

در سال ۲۰۱۸ تیانشن دو و همکاران در تحقیقی با عنوان "مروری بر تولید خودکار طرح‌های فضایی معماری همراه با بهینه‌سازی عملکرد انرژی" به بررسی ۱۵ مقاله جهت یافتن روش‌های خودکار تولید، همراه با بهینه‌سازی انرژی پرداختند (Du et al., 2018, 860).

مقایسه و بررسی تحقیقاتی که از روش‌های تکاملی جهت بهینه‌سازی نظام فضایی استفاده کرده اند (جدول ۱)، نشان می‌دهد:

۱- متغیرهای طراحی از طیف وسیعی شامل شکل‌گیری مرز سایت، جهت‌گیری پلان، سطح‌بندی طبقات، هندسه، مکان‌یابی فضاها، محل قرارگیری بازشوها و مبلمان فضاها برخوردار می‌باشند.

۲- هم راستا با نتیجه تحقیق ردریگز و همکاران ، (Rodrigues et al., 2013) هیچ یک از روش‌ها، همه‌ی متغیرها را به طور هم‌زمان در معرض بهینه‌سازی قرار نداده‌اند.

۳- بهینه‌سازی نظام فضایی، امکان ادغام با دیگر اهداف معماری را دارد؛ به گونه ای که در ۸ مورد از تحقیقات، بهینه‌سازی نظام فضایی همراه با مباحث بهینه‌سازی انرژی، در ۲ تحقیق همراه با بهینه‌سازی فرم و در یک تحقیق با توجه به هر یک از مباحث مدیریت پروژه و توده-فضاهای شهری انجام شده است.

۴- علی‌رغم پژوهش‌های متعدد با موضوع استفاده از الگوریتم‌های محاسبات تکاملی در طراحی نظام‌های فضایی، الگوریتم ژنتیک کاربرد گسترده‌تری در بهینه‌سازی نظام فضایی داشته است. به گونه‌ای که در ۳۳ مورد از تحقیقات، از الگوریتم ژنتیک و در ۳ مورد دیگر از روش‌های برنامه‌نویسی ژنتیک و استراتژی تکاملی استفاده شده است.

۵- علی‌رغم کارآمدی الگوریتم بهینه‌سازی انبوه ذرات برای برنامه‌ریزی در محیط‌های فضایی پویا و پیچیده (Nandanwar et al., 2016, 82) و با وجود برخورداری از سرعت و کیفیت بالای پاسخ گویی به مساله طراحی، در هیچ یک از تحقیقات، از این الگوریتم یا دیگر الگوریتم‌های هوش جمعی به تنهایی و به‌عنوان الگوریتم اصلی جهت بهینه‌سازی نظام فضایی استفاده نگردیده است. این در حالی است که مطالعات ژانگ[۲۹] و همکاران در سال ۲۰۱۵ نشان می‌دهد تعداد انتشارات هر سال مربوط به الگوریتم بهینه‌سازی انبوه ذرات به عنوان پایه و اساس بسیاری از تغییرات الگوریتم‌های هوش جمعی، بالاتر از سایر الگوریتم‌های مبتنی بر هوش جمعی می‌باشد (Cichocka et al., 2017, 155) و با وجود کاربرد گسترده آن‌ها در زمینه‌های مهندسی برق، سیستم‌های کنترل خودکار، نظریه ارتباطات، تحقیقات عملیاتی، مهندسی مکانیک، سوخت و انرژی، پزشکی، شیمی و زیست‌شناسی (Zhang et al., 2015)؛ از حدود سال ۲۰۰۴ تنها در تحقیقات حوزه انرژی و طراحی سازه ساختمان مورد استفاده قرارگرفته است.

مقایسه عملکرد الگوریتم ژنتیک و الگوریتم بهینه‌سازی انبوه ذرات در بهینه‌سازی نظام فضایی

سابقه‌ی زیاد و تعدد استفاده از الگوریتم ژنتیک به عنوان نماینده الگوریتم‌های محاسبات تکاملی در تحقیقات پیشین، سبب گردید تا این الگوریتم به یکی از گزینه‌های مورد توجه نگارندگان تبدیل گردد و در مقایسه با آن نتایج قابل توجه الگوریتم بهینه‌سازی انبوه ذرات در حوزه‌هایی به غیر از طراحی فضایی که به مقایسه آن با الگوریتم ژنتیک پرداخته بودند؛ این الگوریتم را در ارائه‌ی روشی جهت بهینه‌سازی نظام فضایی مورد توجه قرار داد. از نتیجه این تحقیقات می‌توان به همگرایی سریع‌تر و نتیجه‌ی بهتر الگوریتم بهینه‌سازی انبوه ذرات(Chiu et al., 2012) در کاهش افت مسیر در ارتباطات بی

سیم شبکه‌های شهری در تحقیق چو[۳۰] و همکاران و نتیجه‌ای مشابه با آن در مقایسه عملکرد افزونه طراحی شده بر مبنای الگوریتم بهینه‌سازی انبوه ذرات (Silvereye) در مقایسه با سایر افزونه‌ها در یک مساله بهینه‌سازی مرتبط با سازه توسط سیچاکا[۳۱] و همکاران اشاره کرد (Cichocka et al., 2017).

از این رو نگارندگان با طراحی یک مساله‌ی مبنا با پارامترهای کمتر نسبت به موضوع تحقیق، عملکرد این دو الگوریتم را در فرایند بهینه‌سازی نظام فضایی مورد مقایسه قرار دادند. مساله‌ی تعریف شده شامل بهینه‌سازی ابعاد و مکانیابی فضاهای یک پلان با ابعاد ۱۲ در ۸ متر مربع و تعداد ۸ فضا می‌باشد که مطابق با تصویر عملکرد هر یک از دو الگوریتم پیاده‌سازی شده ژنتیک و بهینه‌سازی انبوه ذرات در بهینه سازی این مساله مورد سنجش و مقایسه قرار گرفت (شکل ۱).

نمودار میزان پیشرفت بهینه‌سازی در تابع برازندگی هر یک در طی ۳۰،۰۰۰ تکرار[۳۲] و به صورت میانگینی از ۴۰ مرتبه اجرای هر یک از دو الگوریتم مطابق با شکل ۲ ارائه می‌گردد. این نمودار نشان می‌دهد اگر چه در تکرارهای اولیه هر دو الگوریتم همگام با یکدیگر پیش رفته‌اند اما با زیاد شدن تکرارها و در گام‌های نهایی، الگوریتم بهینه‌سازی انبوه ذرات سرعت بیشتری در همگرایی و همین‌طور توانایی بیشتری در کاهش عدد تابع برازندگی و بهبود نتایج داشته است.

با توجه به نتایج مقایسه‌ی این دو الگوریتم و همچنین گرایش دیگر حوزه‌های مهندسی و ساختمان به آن‌ها در سال‌های اخیر، پیاده‌سازی الگوریتم بهینه‌سازی انبوه ذرات به‌عنوان الگوریتم پایه جهت بهینه‌سازی نظام فضایی برای یک پلان تک‌خوابه مسکونی با محدودیت‌ها و معیارهای تعریف‌شده (در

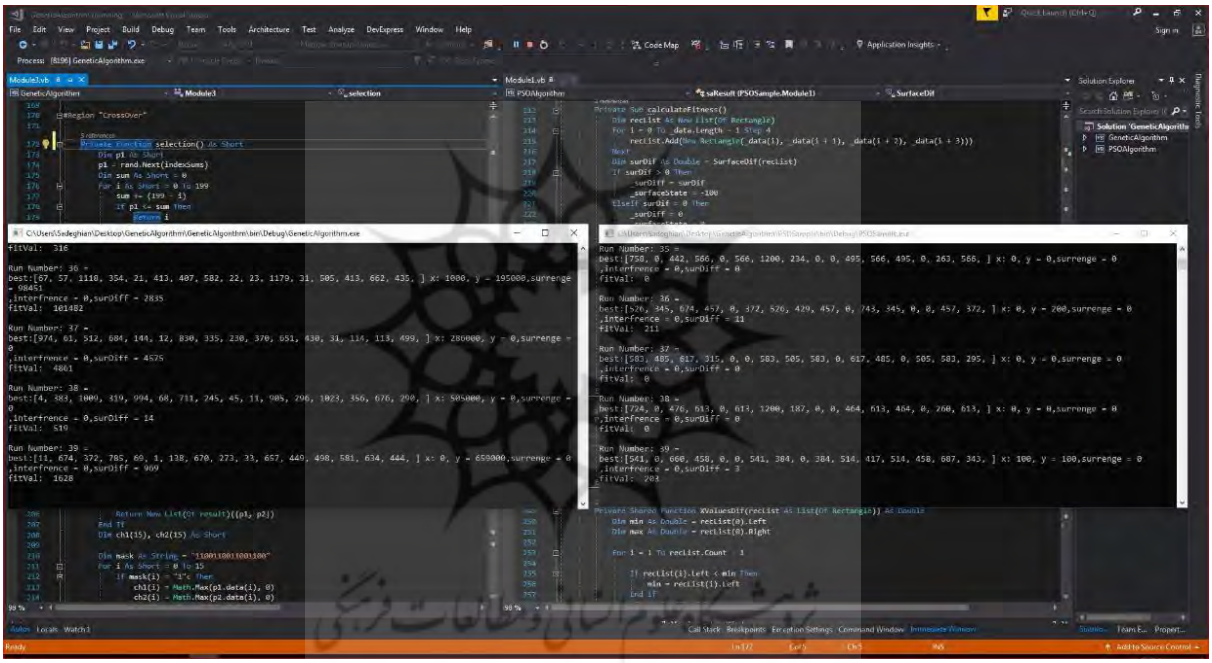

Fig.1. Implementation and comparison of genetic algorithm and particle swarm optimization algorithm in the base problem

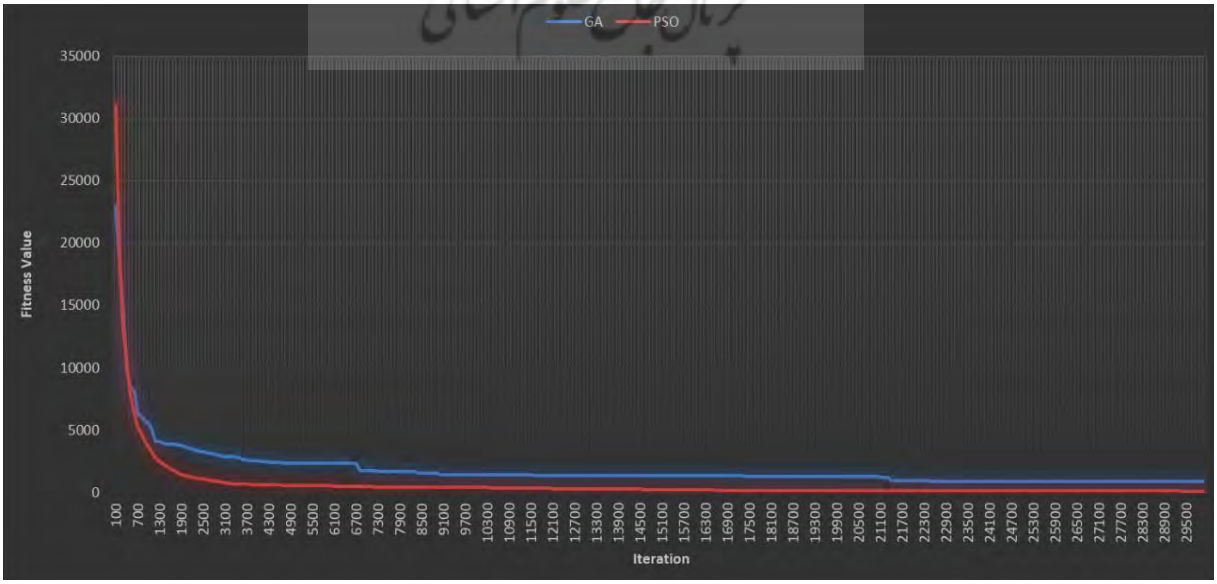

Fig.2. Comparison diagram of the performance of genetic algorithm and particle swarm optimization algorithm in the base problem

ادامه معرفی خواهند شد)، انتخاب گردید و مراحل پیاده‌سازی آن با کمک زبان برنامه‌نویسی C# همراه با یک رابط گرافیکی جهت درک بیشتر از مراحل بهینه‌سازی، در یک عملیات ۳ مرحله‌ای ارائه‌گردید.

معرفی ساختار الگوریتم بهینه‌سازی انبوه ذرات

در سال ۱۹۹۵ دکتر راسل ابرهارت[۳۳] که یک مهندس الکترونیک بود، همراه با دکتر جیمز کندی[۳۴] به‌عنوان روانشناس علوم اجتماعی، یک روش بهینه‌سازی تصادفی که بعدها روش بهینه‌سازی انبوه ذرات نام گرفت، ابداع کردند. این روش که الهام گرفته شده از زندگی طبیعی گله‌ها می‌باشد، از روش مشابه با آن‌ها، برای یافتن بهترین راه‌حل مناسب برای مساله در فضای جستجو استفاده می کند؛ روشی مانند پیدا کردن محل مناسب توسط پرندگان در یک دسته یا حشرات در یک گروه.

برخلاف دیگر الگوریتم‌های تکاملی، الگوریتم بهینه‌سازی انبوه ذرات، خط مشی بقای مطلوب را اجرا نمی‌کند و هیچ عملیات انتخابی در آن وجود ندارد، در نتیجه تمام ذرات در طول فرآیند جستجو حفظ می‌شوند. ذرات، شرایط خود را تحت تاثیر سه عامل تغییر می دهند: حفظ پایداری خود، بهترین موقعیت مطلوب شخصی و بهترین موقعیت مطلوب در گروه (Juneja & Nagar, 2016).

تابع برازندگی[۳۵] برای اندازه گیری عملکرد یک راه حل، که شامل میزان دستیابی به چند هدف مختلف می‌باشد، استفاده می‌شود. این تابع، کیفیت یک راه حل را نسبت به بقیه جمعیت مورد سنجش قرار می‌دهد(Liu et al., 2019, 6).

بهترین مقدار تابع برازندگی که هر ذره تا این مرحله به آن دست‌یافته و در حافظه ذخیره شده است را بهینه ذره می‌گویند و با "pBest" نمایش داده می‌شود. بهترین مقدار تابع برازندگی در مجموعه ذرات در حال تکامل را بهینه‌ی کلی می‌گویند که با "gBest" نشان داده می‌شود.

سرعت حرکت ذره[۳۶] به چگونگی (میزان و جهت) حرکتی که متغیرهای هر ذره باید انجام دهند تا به بهینه‌ی ذره یا بهینه‌ی کلی نزدیک شده و پاسخ‌های مناسب‌تر را جستجو کنند، می‌گویند. در الگوریتم بهینه‌سازی انبوه ذرات، هر ذره دارای موقعیت و سرعت خاص خود می‌باشد که در ابتدا به صورت تصادفی برای هر یک از ذرات تعیین می‌گردد. در هر تکرار، موقعیت و سرعت هر ذره با ارزیابی تابع هدف در مساله، به روزرسانی می‌گردد(Li et al., 2018). روند کلی جستجوی پاسخ در الگوریتم بهینه‌سازی انبوه ذرات در شکل ۳ نمایش‌داده شده است.

تعدادی از پاسخ‌ها، به‌صورت تصادفی تولید شده و به‌عنوان ذرات اولیه برای شروع الگوریتم، در نظرگرفته می‌شوند. تعداد این ذرات با توجه به نوع پروژه، متغیر در نظر گرفته می‌شوند. علی‌رغم

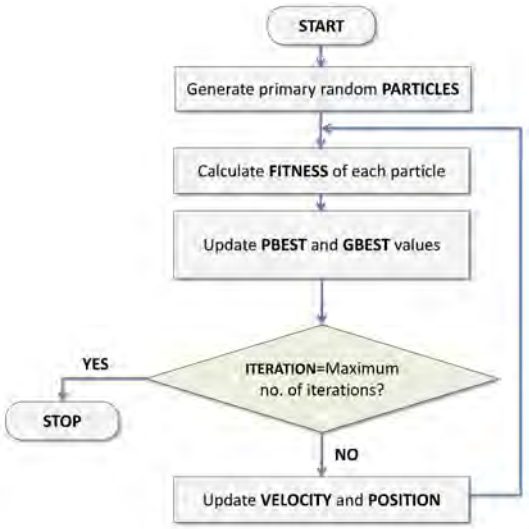

Fig.3. Flowchart of PSO algorithm
(Juneja & Nagar, 2016, 3)

این که در تحقیقات بر پایه‌ی الگوریتم بهینه‌سازی انبوه ذرات، هیچ توصیه‌ای در مورد تعداد ذرات نشده است، اکثر محققان از ۱۰ تا ۵۰ ذره استفاده کرده‌اند (Sowmya & Mp, 2013, 324). تعداد ذرات می‌تواند با توجه به تجربه، آزمایشات طراح الگوریتم و نوع مسئله متغیر باشد. بر اساس نظر شی[۳۷] و راسل ابرهارت، اندازه جمعیت به ندرت تاثیری در عملکرد روش الگوریتم دارد (Shi & Eberhart, 1998).

درهر تکرار از جستجوی الگوریتم، مقادیر بهترین pBest و gBest محاسبه می‌گردند. در صورت یافتن پاسخ بهتر، میزان سرعت مناسب برای هر ذره جهت جابجایی (تغییر متغیرها) ذرات تعیین می‌گردد. در هر تکرار، هر ذره با توجه به سرعت تعیین شده برای رسیدن به پاسخ بهتر، تغییر می‌کند. این جستجو تا دست‌یابی به نتیجه‌ی مناسب (مقدار تعیین شده برای تابع برازندگی) تکرار می‌گردد.

کاربرد الگوریتم بهینه‌سازی انبوه ذرات در طراحی پلان

کارلن[۳۸] در کتاب "مبانی برنامه‌ریزی فضا"، برنامه‌ریزی فضا را شامل فعالیت‌های زیر می‌داند (Karlen, 2009):

۱. داده‌های ورودی: نیاز کاربر، داده‌های پایه، داده‌های متنی و استانداردهای مختلف.

۲. پیش‌پردازش داده‌ها: جمع‌بندی عوامل کمی تأییدشده و مفید شامل (مساحت، اندازه تجهیزات، نقاط ارتباطی و غیره).

۳. تحقیق درمورد ناشناخته‌ها: جمع‌آوری اطلاعات موردی درباره‌ی چیزهای مختلف.

۴. تجزیه و تحلیل داده‌ها: کشف روابط مختلف شامل (روابط متقابل کار، منطقه‌بندی عمومی و خصوصی و غیره).

۵. تفسیر و نموداری‌کردن داده‌ها (Dutta & Sarthak, 2011, 315-316)

در تلاش برای بهینه‌سازی مسائل معماری توسط الگوریتم‌های تکاملی پیچیدگی‌های فراوانی وجود دارد، تبدیل مسائل معماری به مسائل عددی، کنترل مقیاس فضای جست‌وجو به وسیله‌ی کدگذاری، نحوه‌ی ارزشیابی و چگونگی تولید پاسخ‌های جدید در هر نسل (Golabchi et al., 2012) و همین‌طور چگونگی استفاده از الگوریتم‌های تکاملی موضوعات مهم در بهینه‌سازی مسائل معماری با استفاده از این الگوریتم‌ها می‌باشد (Li L. , 2012, 39). بر این اساس نحوه‌ی کاربرد الگوریتم بهینه‌سازی انبوه ذرات در طراحی پلان در ۳ مرحله بیان می‌گردد:

۱. تعیین اهداف مساله، معیارها و محدودیت‌های تاثیرگذار

۲. عددی‌سازی و کمی‌کردن معیارها و محدودیت‌ها

۳. تعیین تابع برازندگی و پیاده‌سازی الگوریتم

۱. تعیین اهداف، معیارها و محدودیت های تاثیرگذار بر سازمان فضایی معماری

پیش از این‌که معمار وارد طراحی بنا شود با یک برنامه روبروست که مولفه‌های آن به‌کمک قوانین با هم رابطه برقرار می‌کنند. این برنامه‌ها معمولا عبارتند از خواسته‌های کارفرما، آیین‌نامه‌های جاری، قوانین زیبایی‌شناسی و ایستایی، درک طراح و خواسته‌های او. هرکدام از این‌ها ثابت‌هایی هستند که بایستی بدون تغییر و به‌اجبار وارد سیستم شوند. بعضی از آن‌ها اجبارهای بیرونی هستند و بعضی اجبارهای شخصی طراح است که قیدهای طرح را تعریف می‌کنند (Mahdavinejad & Re-falian,2014). لاوسون[۳۹]، نظرات کارفرما، نیازهای استفاده کننده، الزامات قانون‌گذار و رویکرد طراح را به عنوان مولدهای اصلی مسئله‌ی طراحی مطرح‌کرده و توضیح می‌دهد که هر یک، با محدودیت‌هایی که بر مسئله اعمال می‌کنند، به نوعی به تعریف مسئله کمک می‌کنند (Lawson, 1980). بر این اساس، معیارها و محدودیت‌های مساله طراحی نظام فضایی پلان، جهت حصول بهترین حالت ترکیب فضاها، در سه دسته به الگوریتم معرفی می‌گردند (شکل ۴).

معیارها و محدودیت‌های اولیه: این دسته دربرگیرنده ورودی های اصلی مساله و شامل محدودیت‌های اولیه طرح مانند الزامات قانونی و مرز زمین می‌باشد.

معیارها و محدودیت‌های طراح: دسته‌ای از معیارها، قیدهایی هستند که طراح متناسب با استانداردهای طراحی و یا تجربه و تخصص خود به عنوان ورودی‌های مساله هنگام پیاده‌سازی بر الگوریتم اعمال می‌نماید تا سرعت و کیفیت دستیابی به بهترین پاسخ برای الگوریتم میسر گردد. مانند: فضاهایی که باید طراحی و جانمایی شوند؛ تناسبات فضاها که نقش مهمی در قابل قبول بودن فضاهای پیشنهادی الگوریتم دارد؛ میزان دست‌یابی به مساحت کل با جمع فضاها یا اختلاف مساحت کل زمین با جمع مساحت فضاهای مختلف که منجر به تداخل یا ایجاد فاصله بین فضاها در شکل‌گیری طرح کلی پلان می‌گردد و این که میزان تداخل مجاز کل فضاها با یکدیگر چه میزان می‌باشد.

معیارها و محدودیت‌های مخاطبین (کارفرما و استفاده‌کنندگان): گروه دیگر، معیارهایی هستند که طراح جهت درک صحیح مساله طراحی با دریافت نیازهای استفاده کنندگان یا نظرات کارفرما که معمولاً با روش‌های طراحی معماری آشنا نیستند و صرفا نقطه نظرات خود را انتقال می‌دهند، بر الگوریتم اعمال می نماید. رعایت این معیارها نقش به سزایی در رضایت این گروه از طرح نهایی خواهد داشت. به‌عنوان نمونه افزودن فضاهایی که لزوم وجود آن ها به واسطه تجربه مخاطب در فضاهای مشابه، شکل گرفته و همینطور حذف برخی فضاها که می‌تواند به تدقیق لیست فضاهای قابل طراحی کمک نماید. مخاطب اطلاعات ارزشمندی را به طراح در موضوعاتی نظیر ابعاد و اندازه فضاها، هم جواری آن‌ها، اولویت های نورگیری یا جهت یابی مناسب برای هرفضا از طریق مقایسه با فضاهای مشابه ارائه می‌دهد.

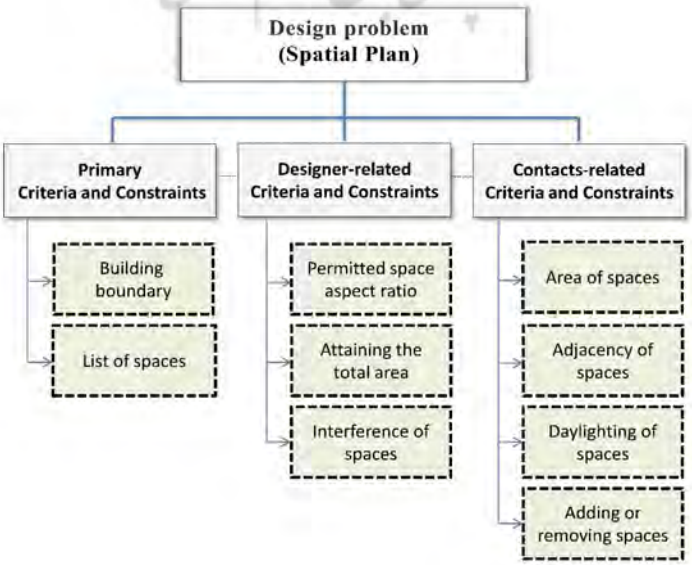

Fig.4. Problem goals, criteria, and limitations affecting spatial plan design

۲. عددی‌سازی و کمی‌کردن معیارها

هر یک از ریزموضوع‌ها در دسته‌های فوق، جهت استفاده در الگوریتم پیاده‌سازی شده بر پایه بهینه‌سازی انبوه ذرات، نیاز به عددی‌سازی دارد. معیارهای طراحی پلان به صورت زیر به مقادیر کمی تبدیل می‌گردند:

مرز زمین

در بحث داده‌های ورودی، به جز در رویکردهای الهام گرفته از اصول ریخت‌شناسی زیستی[۴۰]، با هر فضا به صورت یک شکل راست گوشه (مستطیل) رفتار می‌شود و استفاده از اشکال غیرمحدب (به‌عنوان مثال فضاهای L شکل) در تحقیقات به ندرت مشاهده می‌گردد (Rodrigues et al., 2013, 5). تعیین مرز زمین ($Boundry_i$)، به معنای تعیین طول و عرض مستطیل محاط کننده فضاها می‌باشد (رابطه ۱). فضای داخل این مستطیل می‌تواند مستطیل شکل یا ترکیبی از چند مستطیل باشد. همان‌گونه که در شکل ۵ مشخص شده است، در مرزهای «U و L» شکل و به طور کلی ترکیبی از مستطیل‌ها به عنوان محدوده‌ی زمین، می‌بایست کلیت زمین در یک مستطیل، محیط شده و فضاهای باقی مانده، به عنوان فضاهایی با موقعیت و اندازه ثابت، در پلان در نظر گرفته شوند.

$$Boundry_i = \{X_i, Y_i, Width_i, Height_i\} \quad (1)$$

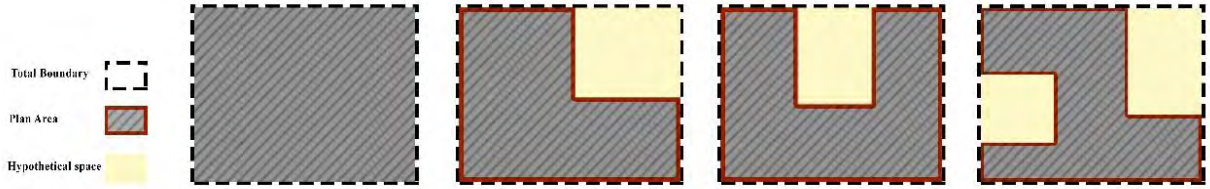

Fig.5. Determining the total boundary in planes with different shapes

لیست فضاها

همانگونه که در رابطه ۲ مشخص شده است، هر یک از فضاها ($Space_i$)، به صورت آرایه‌هایی با ۴ متغیر مربوط به مختصات محل قرارگیری و طول و عرض در الگوریتم، قابل تعریف و پیاده‌سازی هستند. مجموعه‌ای از این آرایه‌ها در مرز کلی زمین، می‌تواند یک پاسخ برای مساله یعنی مشخصات پلان باشد.

$$Space_i = \{x_i, y_i, width_i, height_i\} \quad (2)$$

تناسبات مجاز فضاها

در تعیین تناسبات فضاها (Asp_i) هر طراح بر اساس استانداردهای مورد نظر و تجربه خود، یک بازه a تا 1/a را به عنوان بازه مجاز تناسبات در نظر می‌گیرد ($minAsp_i$, $maxAsp_i$) که به فضاها قابلیت تغییر و جابجایی طول و عرض یا جهت‌گیری در پلان را می‌دهد. در صورت تخطی هر فضا از این میزان تناسبات، جریمه‌ای برای الگوریتم در نظر گرفته می‌شود. میزان جریمه اعمال شده، باعث نزدیک کردن پاسخ به تناسبات مجاز می‌گردد. نحوه‌ی کمی‌سازی این تناسبات همراه با پارامتر ابعاد مجاز فضاها درادامه بیان می‌گردد.

ابعاد مجاز فضاها

جهت اجتناب از پایین آمدن بازدهی الگوریتم و کوچک‌شدن فضای جستجو، مساحت دقیق فضاها تعیین نمی‌گردد و اغلب مساحت‌ها به صورت حداقل و حداکثر مجاز ($minS_i$, $maxS_i$) تعیین می‌شوند. این مساحت‌ها (S_i) در تکرارهای مختلف در آن بازه قابل تغییر می‌باشد. این ابعاد می‌توانند از صفر تا مساحت کل زمین برای هر فضا در نظر گرفته شوند. میزان جریمه یا اختلاف بازه مساحتی ($\Delta sRange_i$) هر فضا مطابق رابطه ۴ و با ضریب جریمه‌های (Y_1,Y_2) برای اختلاف مساحت فضاها و تناسبات آن‌ها سنجیده می‌شود و همانطور که در رابطه‌ی۳ و ۴ مشاهده می‌شود، مجموع این جریمه‌ها در فضاهای مختلف، اختلاف بازه مساحت ($\Delta sRange_{Total}$) در تابع برازندگی را تعیین می‌نماید.

$$\Delta sRange_{Total} = \sum_{i=1}^{n} \Delta sRange_i \quad (3)$$

$$\Delta sRange_i = \begin{cases} (minS_c - S_a) * Y_1 & if\,(S_a < minS_c) \\ (S_a - maxS_c) * Y_1 & if\,(S_a > maxS_c) \\ (Asp_i - maxAsp_i) * Y_2 & if\,(minS_c < S_a < maxS_c \ \wedge \ Asp_i > maxAsp_i\ ) \\ (minAsp_i - Asp_i) * Y_2 & if\,(minS_c < S_a < maxS_c \ \wedge \ Asp_i < minAsp_i) \\ 0 & if\,(minS_c < S_a < maxS_c \ \wedge \ minAsp_i < Asp_i < maxAsp_i) \end{cases} \quad (4)$$

دستیابی به مساحت کل

جهت پوشش محدوده‌ی زمین و دستیابی به مساحت کل، باید اختلاف مساحت کل مرز زمین ($S_{Boundry}$) با جمع مساحت فضاهای مختلف ($S_{Space(i)}$) سنجیده شده (رابطه ۵) و میزان حساسیت طراح به پوشش مرزهای زمین مشخص گردد. این مقدار ($\Delta s_{allowed}$) می‌تواند صفر تعیین گردد. همانگونه که در رابطه ۶ مشاهده می‌شود. اختلاف مساحت‌ها ($\Delta s_{Difference}$) با مساحت مجاز، مقدار اختلاف مساحت

کل (Δs_{Total}) در تابع برازندگی را مشخص می‌نماید. در برخی مسائل نیز جهت اغماض از برخی اختلاف‌ها و در نهایت امکان کشف راه‌حل‌های بیشتر، می‌تواند کمی بیشتر از صفر در نظر گرفته شود.

$$s_{Diffrence} = \left|S_{Bounry} - \sum_{i=1}^{n} S_{Space(i)}\right| \quad (5)$$

$$s_{Total} = \left|\Delta s_{Diffrence} - \Delta s_{allowed}\right| \quad (6)$$

میزان تداخل فضاها

میزان تداخل فضاها ($Interfrence_{(i,j)}$) به اختصاص بخشی از مرز کل زمین به دو فضا گفته می‌شود که در هر کدام از تکرارهای الگوریتم، در مجموعه فضاهای یک ذره قابل سنجش می‌باشد. این مقدار بسته به نظر طراح باید صفر و یا نزدیک به صفر در نظر گرفته شود. ایجاد تداخل به شرط رسیدن دقیق به مساحت کل زمین، باعث ایجاد فاصله بین فضاها می‌گردد. مجموع تداخل‌های بین فضاها پارامتر تداخل کل ($Interfrence_{Total}$) در تابع برازندگی را مقداردهی می‌نماید. در رابطه ۷ و ۸ به نحوه‌ی محاسبه تداخل فضاها و کل تداخل در هر تکرار در الگوریتم پیاده‌سازی‌شده اشاره شده است.

$$Interfrence_{(i,j)} = max(0,(min(Right_i,Right_j) - max(Left_i,Left_j)) * \max(0,(\min(Bottom_i,Bottom_j) - \max(Top_i,Top_j)) . \quad (7)$$

$$Interfrence_{Total} = \sum_{i=1}^{n}\sum_{j=i+1}^{n} Interfrence_{(i,j)} . \quad (8)$$

هم‌جواری فضاها

در مسئله حل روابط پلان معماری، بهترین ساختار برای مدل‌سازی و کدگذاری حالت‌های مختلف قرارگیری فضاها در کنار هم، استفاده از نظریه گراف‌ها و ماتریس مجاورت آن‌هاست (Golabchi et al., 2012, 119). جهت تعیین همجواری‌های بهینه، طراح با کمک گرفتن از ماتریس‌های همجواری، بازه‌های مجاز را در ماتریس اعمال و به صورت آرایه‌های سه‌بعدی به الگوریتم معرفی می‌کند. برای تعیین میزان جریمه اختلاف همجواری ($\Delta Adjacency_{Total}$) (رابطه ۹) در تابع برازندگی

$$\begin{matrix} if\ hasEssential = 0 \\ if\ hasEssential = 1 \end{matrix} \quad (9)$$

$$haseEssential = \wedge_{\{i,j|m_{(i,j)}=3\}} adjacency_{(i,j)} \quad \wedge \wedge_{\{i,j|m_{(i,j)}=-3\}} ! adjacency_{(i,j)} \quad (10)$$

برآیند همجواری‌های ($Adjacency_{(i,j)}$) ایجاد شده نسبت به میزان تعیین‌شده دوفضا ($m_{(i,j)}$) محاسبه می‌گردد.

اگر شرط همجواری دو فضا که به‌صورت قطع باید درکنار یکدیگر قرارگیرند یا بالعکس، ($m_{(i,j)} = \pm 3$) مطابق با رابطه ۱۰ رعایت نگردد (hasEssential=0)، جریمه‌ی سنگین که با (Y) در رابطه ۹ نشان داده شده است به عنوان اختلاف همجواری در تابع برازندگی محاسبه می‌شود تا آن گزینه در پیشنهادهای الگوریتم قرار نگیرد.

$$\Delta adjacency_{Total} = \begin{cases} Y \\ \sum_{i=1}^{n}\sum_{j=i+1}^{n} adjacency_{(i,j)} * m_{(i,j)} \end{cases}$$

نورگیری فضاها

همان‌طور که در رابطه ۱۱ و ۱۲ مشاهده می‌شود، برخورد با مساله نورگیری همچون مبحث همجواری با تعیین ماتریس نورگیری صورت می‌پذیرد، بدین صورت که جبهه‌های مختلف زمین به صورت مستطیل‌هایی با عرض حداقلی و طول جبهه ساختمان در نظر گرفته شده و همجواری فضاها با این جبهه‌ها به الگوریتم معرفی می‌گردد. یکسان‌سازی روش‌های برخورد با مسائل مختلف، به افزایش سرعت الگوریتم و کاهش پردازش‌ها کمک شایانی می‌نماید.

$$\begin{matrix} if\ hasEssential = 0 \\ if\ hasEssential = 1 \end{matrix} \quad (11)$$

$$haseEssential = \wedge_{\{i,k|m_{(i,k)}=3\}} lighting_{(i,k)} \quad \wedge \wedge_{\{i,k|m_{(i,k)}=-3\}} ! lighting_{(i,k)} . \quad (12)$$

تدقیق لیست فضاها

پس از دریافت نظرات مخاطبین در مورد لیست فضاهای پیشنهادی طراح و تصمیم نهایی توسط او، به هریک از ذرات، تعدادی فضا شامل یک آرایه ۴ متغیره افزوده یا کسر می‌گردد و در روند اجرای الگوریتم همانند فضاهای دیگر با آن‌ها رفتار می‌شود.

۳. پیاده‌سازی الگوریتم بر پایه الگوریتم بهینه‌سازی انبوه ذرات

در مساله پیاده‌سازی شده، هر ذره مطابق با رابطه ۱۳به مجموعه‌ای از فضاها با ۴ متغیر ذکرشده در محدوده زمین است که احتمال دارد به‌عنوان بهترین پاسخ مساله در نظر گرفته شود.

$$Particle_i = \{Space_1, Space_2, Space_3, \ldots . Space_n\} . \quad (13)$$

در پیاده‌سازی الگوریتم در نمونه مورد بررسی، تابع برازندگی با توجه به معیارها و محدودیت‌های تعیین‌شده، تشکیل می‌گردد و در هر تکرار، همه‌ی ذرات توسط آن مورد سنجش قرار می‌گیرند. جهت افزایش سرعت پیشرفت الگوریتم و بازدهی کیفی، پیاده‌سازی الگوریتم طبق شکل ۶، در دو گام انجام می‌شود. به این وسیله، فضای جستجوی الگوریتم برای یافتن پاسخ مناسب محدودتر شده و به اصطلاح از ایجاد مساله‌ای به نام مشقت بعد چندی[۴۱] جلوگیری می‌گردد. نفرین ابعاد

اغلب به عنوان یک مفهوم در جایی که فضای جستجوی مسئله به اندازه‌ای بزرگ است که پیدا کردن پاسخ مناسب از بین گزینه‌های محتمل غیرممکن گردد، مطرح می‌گردد. این اصطلاح برای اولین بار توسط بلمن[۴۲] در سال ۱۹۶۱ برای برآورد ترکیبی از توابع چندمتغیره استفاده شد (Beyer et al.,1999).

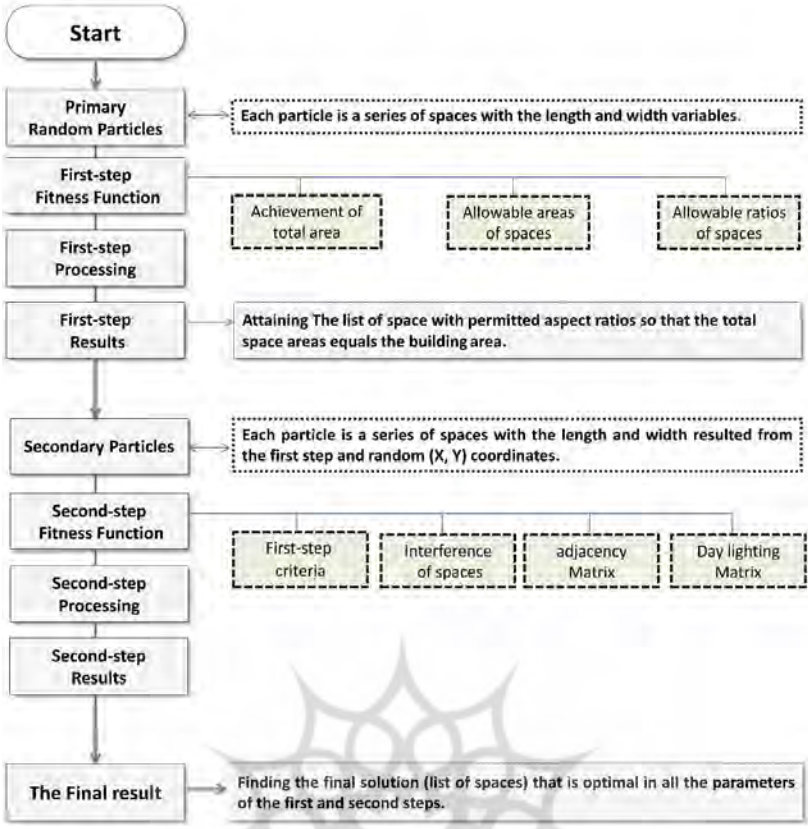

Fig.6. Flowchart of the implemented algorithm based on PSO algorithm for spatial plan optimization

گام اول

در این گام، تابع برازندگی ($F_1(x)$) تنها به بررسی پارامترهای مربوط به هندسه مانند میزان دستیابی به مساحت کل زمین (Δs_{Total}) و حفظ تناسبات و مساحت فضاها ($\Delta sRenge_{Total}$) می‌پردازد، در نتیجه هر یک از ذرات در این گام تنها دارای دو متغیر طول و عرض می‌باشد و متغیرهای مربوط به موقعیت مکانی در آن‌ها وجود ندارند. بهترین پاسخ یا ذره این بهینه‌سازی، شامل یک مجموعه از فضاها یا طول و عرض‌های حاصل از پیاده‌سازی الگوریتم می‌باشد که هنگام قرارگیری در تابع برازندگی به مقدار مینیم تعریف شده، دست یابد. وظیفه الگوریتم در این مرحله کنترل طول و عرض فضاها در هر ذره می‌باشد. این طول و عرض در گام دوم قابل تغییر و بهبود می‌باشد تا هم تنوع لازم حفظ‌شده و هم الگوریتم به پاسخ‌های مناسب‌تر دست یابد. با توجه به سهولت دستیابی به پاسخ در این مرحله برای الگوریتم، با تکرار متعدد الگوریتم در گام اول به مجموعه‌ای از پاسخ‌ها یا ذرات بهینه دست می‌یابیم که این مجموعه ذرات به عنوان داده‌های اولیه برای تولید ذرات تصادفی گام دوم و پیاده‌سازی الگوریتم آن مرحله به کار می‌رود. تابع برازندگی گام اول در رابطه ۱۴ و یک نمونه از نتایج الگوریتم، جهت تشکیل فضاهای پلان یک واحد تک‌خوابه مسکونی، با توجه به معیارهای تعریف‌شده، در شکل ۷ نشان داده شده است.

$$F_1(x) = \omega_1 \Delta s_{Total} + \omega_2 \Delta sRange_{Total} \qquad (14)$$

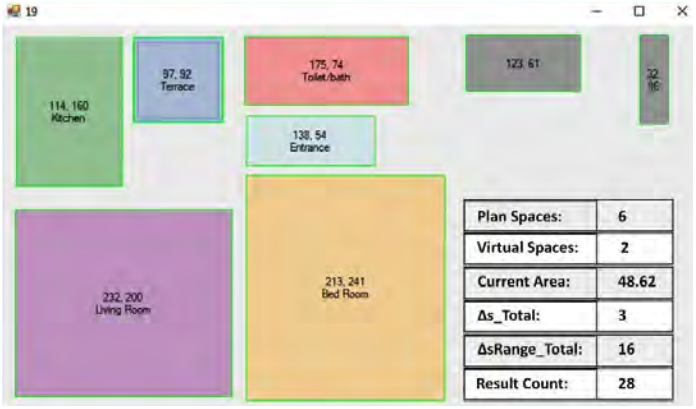

Fig.7. First step of algorithm implementation based on PSO algorithm

گام دوم

همانطور که در رابطه ۱۵ نشان‌داده شده است، در گام دوم، علاوه بر معیارهای تاثیرگذار در گام اول، به همه‌ی معیارها و محدودیت‌های دیگر در تعریف تابع برازندگی ($F_2(x)$) پرداخته می‌شود تا نتیجه نهایی بهترین پاسخ ممکن برای همه‌ی معیارها و محدودیت‌های تعریف شده که شامل معیارهای گام اول و همینطور سنجش میزان تداخل($Interfrence_{Total}$)، همجواری ($\Delta Adjacency_{Total}$) و قرارگیری در جبهه مناسب جهت نورگیری ($\Delta Lighting_{Total}$) است، باشد. مقادیر w_1 تا w_5 ضرایب اهمیت می‌باشند که قابلیت کنترل حساسیت الگوریتم به هر یک از پارامترهای تابع برازندگی را به طراح می‌دهد. این گام نیز بر پایه‌ی تعدادی از ذرات تصادفی شکل می‌گیرد. در این ذرات فضاها دارای ۴ مشخصه بوده که علاوه بر طول و عرض، مختصات محل قرارگیری آن‌ها نیز تغییر پیدا می‌کند. طول و عرض اولیه فضاها در ذرات به صورت تصادفی از طول و عرض پاسخ‌های بهینه‌ی گام قبل و مختصات آن‌ها نیز به صورت تصادفی از اعداد مجاز داخل مرز زمین انتخاب می‌گردد که کلیه این ابعاد نیز در تکرارهای الگوریتم همراه با حفظ مساحت مجاز تغییر پیدا کرده تا نتیجه مطلوب حاصل گردد.

میزان پیشرفت الگوریتم وابسته به میزان حداقلی است که برای تابع برازندگی تعریف شده است. شکل ۸، تولید پلان یک واحد تک‌خوابه مسکونی را بر اساس معیارهای تعریف شده، پس از ۷۷۰۰۰۰ تکرار در الگوریتم نشان می‌دهد.

$$F_2(x) = \omega_1 \Delta s_{Total} + \omega_2 \Delta sRange_{Total} + \omega_3 Interfrence_{Total} + \omega_4 \Delta adjacency_{Total} + \omega_5 \Delta lighting_{Total} . \quad (15)$$

Fig.8. Secound step of algorithm implementation based on PSO algorithm

بستر پیاده‌سازی الگوریتم

افزونه گرس‌هاپر[۴۳] در راینو[۴۴] و دینامو[۴۵] در رویت[۴۶] با قابلیت بصری بودن، به صورت پویا در حال توسعه هستند. امیدوارکننده‌ترین گزینه، ترکیب رویت با دینامو و پراجکت فرکتال[۴۷] است که در بستر مشترک اتودسک[۴۸] ایجاد شده و می‌تواند در آینده نزدیک، ترکیبی از تمام عملکردهای آن ها در قالب یک راه‌حل باشند. با وجود این که این افزونه‌ها دارای الگوریتم‌های پایه‌ای جهت بهینه‌سازی می‌باشند، لیکن امکان تغییر در ساختار الگوریتم‌های بهینه‌سازی جهت هماهنگی بیشتر با موضوع مساله مگر با استفاده از برنامه‌نویسی (افزودن اسکریپت[۴۹] به این افزونه‌ها) برقرار نمی‌باشد. با این وجود، صرفاً توسط گروه کوچکی از کاربران حرفه‌ای مورد استفاده قرار می‌گیرد زیرا که این امر نیازمند آشنایی با دانش و تجربه برنامه نویسی پیشرفته و آموزش استفاده از افزونه‌ها می‌باشد. علاوه بر آن، کار با زبان‌های برنامه‌نویسی، به دلیل بصری نبودن، اغلب شباهت خاصی با روند کاری معماری نداشته و باعث کاهش استفاده آن‌ها توسط معماران می‌گردد در نتیجه هنوز ابزار ساده‌ای برای طراحی محاسبات برای گروه گسترده‌ای از معماران وجود ندارد.

نیزتوک و پاول[۵۰] در تحقیقی که با موضوع قابلیت استفاده از ابزارهای معاصر برای طراحی محاسباتی در اهداف معماری در سال ۲۰۱۸ انجام دادند، بیان می‌کنند که راه‌حل‌های پیاده‌سازی الگوریتم‌های موجود در تحقیقات، اغلب فاقد رابط بصری بوده و از نظر عملکردی بسیار محدود هستند. آن‌ها تاکید می‌نمایند ابزارهای آینده باید بر اساس ماهیت خاص فرآیند طراحی، در تماس مداوم با کاربران

اصلی آنها یعنی معماران، توسعه داده شوند. تمایل به استفاده از ابزارها برای طراحی معماری محاسباتی نشان می‌دهد که توسعه این نوع ابزارها باید روی نیازهای این گروه متمرکز شود (Nisztuk & Paweł, 2018, 80-81).

در این تحقیق جهت دریافت معیارها و محدودیت‌های مورد نظر و همین طور نمایش روند کار و نتایج، پیاده‌سازی الگوریتم توسط نرم‌افزار برنامه‌نویسی Microsoft Visual Studio و در پلتفرم برنامه‌نویسی .NET. و با استفاده از زبان C# به‌صورت یک نرم‌افزار کاربردی با رابط گرافیکی Windows Form و در بستر ویندوز انجام گردید. هر چند این نرم‌افزار هنوز قابلیت ارائه به صورت یک محصول تجاری را ندارد، اما در صورتی‌که در اختیار طراح قرار گیرد، می‌تواند با حداقل آموزش، در راستای بهینه‌سازی نظام فضایی در طراحی پلان مورد استفاده قرار گیرد.

تصویر ۹ یک نمونه از مراحل بهینه‌سازی پلان برای مسئله‌ای تعریف شده می‌باشد که توسط الگوریتم پیاده‌سازی شده بر پایه بهینه‌سازی انبوه ذرات در نرم‌افزار پیشنهادی به‌دست آمده است.

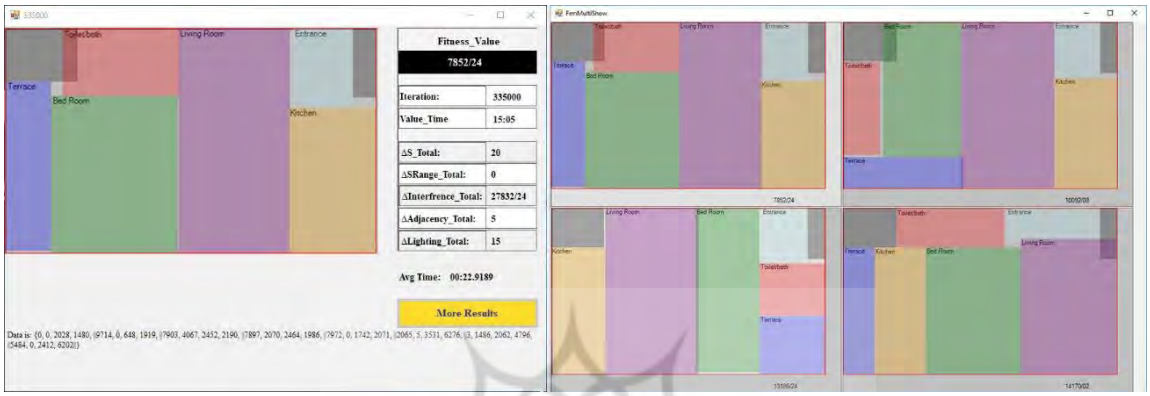

Fig.9. Graphical user interface of the implemented algorithm

نتیجه‌گیری

امروزه با گسترش پارامترهای تاثیرگذار برطراحی و دشواری کنترل این عوامل به صورت همزمان توسط طراح، استفاده از توان کامپیوترها و به‌طور ویژه الگوریتم‌های بهینه‌سازی به عنوان راهکار کاهش خطا در بسیاری از علوم از جمله معماری در حال گسترش می‌باشد. مساله طراحی نظام فضایی پلان به عنوان یکی از پرسابقه‌ترین و پرچالش‌ترین مسائل طراحی از نظر تعداد پارامترهای تاثیرگذار، همواره یکی از مهم‌ترین دغدغه‌های محققین این حوزه می‌باشد.

دشواری کارکردن با زبان‌های برنامه‌نویسی، نیاز به مهارت استفاده از نرم‌افزارها و پیچیدگی نرم‌افزارها به سبب طراحی شدن برای پاسخ‌گویی به مسائل حوزه‌های علمی مختلف، باعث رایج نبودن استفاده از آن‌ها به عنوان یک کمک طراح در کنترل پارامترهای تعریف شده توسط طراح می‌باشد. در این تحقیق پس از بررسی ۳۵ عدد از منابعی که در آن‌ها از الگوریتم‌های بهینه‌سازی جهت طراحی سازمان فضایی پلان‌های معماری استفاده شده‌بود؛ عملکرد دو الگوریتم ژنتیک و بهینه سازی انبوه ذرات به عنوان نمایندگان دو گروه اصلی از الگوریتم‌های تکاملی در یک مسئله‌ی مبنا مورد مقایسه قرار گرفت و الگوریتم بهینه‌سازی انبوه ذرات به دلیل همگرایی سریع‌تر و کیفیت بالاتر جهت بهینه‌سازی پلان با توجه به پارامترهای تاثیرگذار بر طراحی پلان انتخاب گردید.

پیاده‌سازی راهکاری عملیاتی جهت بهینه‌سازی نظام فضایی پلان با تاکید بر پارامترهای تاثیرگذار بر شکل‌گیری در طراحی پلان‌های معماری در یک فرایند ۳ مرحله ای با استفاده از این الگوریتم پیشنهاد گردید. ۱- تعیین اهداف مساله، معیارها و محدودیت‌ها ۲- عددی‌سازی و کمی‌کردن معیارها و محدودیت‌ها ۳- تعیین تابع برازندگی و پیاده‌سازی الگوریتم. اهداف مساله در سه حوزه کلی معیارها و محدودیت‌های اولیه، طراح و مخاطبین بررسی و فرایند عددی‌سازی و پیاده‌سازی مدل نهایی شده بر اساس الگوریتم بهینه‌سازی انبوه ذرات در پلتفرم برنامه‌نویسی .NET. همراه با یک محیط گرافیکی به عنوان رابط کاربری جهت تسلط و درک بیشتر معماران از فرآیند، پیاده‌سازی گردیدند؛ تا علاوه بر دستیابی به پلان‌های بهینه، مقدمه‌ای برای تحقیقات آتی در این زمینه فراهم گردد؛ چرا که به دلیل ماهیت خاص رشته معماری، نرم‌افزارهای مورد نیاز جهت پیاده‌سازی الگوریتم، کارآمدی بصری و اجرایی کافی نداشته و نیازمند توسعه و اختصاصی کردن جهت کاربرد وسیع‌تر در حوزه‌های مختلف طراحی می‌باشند.

پی نوشت

1. Deterministic Algorithm
2. Heuristic Algorithm
3. Metaheuristic Algorithm
4. Trajectory methods
5. Simulated Annealing: SA
6. Population based methods
7. Genetic Algorithm: GA
8. Ant Colony Optimization: ACO
9. Particle Swarm Optimization: PSO

10. Hybrid methods
11. Evolutionary Computation Algorithms: EC
12. Genetic Programming: GP
13. Evolution Strategies: ES
14. Differential Evolution: DE
15. Swarm Intelligence Algorithms: SI
16. Pheromone
17. Physics Based Algorithms
18. River Formation Dynamic
19. Intelligent Water Drops
20. John Frazer
21. Peter Hirsch Levin
22. John Grason
23. Robin S Liggett
24. Mitsuo Gen and Lin Lin
25. Kamlesh Dutta and Siddhant Sarthak
26. Eugenio Rodrigues
27. Victor Calixto and Gabriela Celani
28. Tiantian Du
29. Yudong Zhang
30. Chiu
31. Cichocka
32. Iteration
33. Russell C. Eberhart
34. James Kennedy
35. Fitness Function
36. Velocity
37. Yuhui Shi
38. Mark Karlen
39. Bryan Lawson
40. Biological morphology principles
41. Curse of Dimensionality
42. Richard Bellman
43. Grasshopper
44. Rhinoceros
45. Dynamo
46. Revit
47. Project Fractal
48. Autodesk
49. Script
50. Maciej Nisztuk and Myszkowski B Paweł

تشکر و قدردانی

موردی توسط نویسندگان گزارش نشده است.

تعارض منافع

نویسندگان اعلام می‌دارند که در انجام این پژوهش هیچ‌گونه تعارض منافعی برای ایشان وجود نداشته است.

تاییدیه‌های اخلاقی

نویسندگان متعهد می‌شوند که کلیهٔ اصول اخلاقی انتشار اثر علمی را براساس اصول اخلاقی COPE رعایت کرده‌اند و در صورت احراز هر یک از موارد تخطی از اصول اخلاقی، حتی پس از انتشار مقاله، حق حذف مقاله و پیگیری مورد را به مجله می‌دهند.

منابع مالی/ حمایت‌ها

موردی توسط نویسندگان گزارش نشده است.

مشارکت و مسئولیت نویسندگان

نویسندگان اعلام می‌دارند به‌طور مستقیم در مراحل انجام پژوهش و نگارش مقاله مشارکت فعال داشته و به طور برابر مسئولیت تمام محتویات و مطالب گفته‌شده در مقاله را می‌پذیرند.

References

1. Baghel, M., Agrawal, S., & Silakari, S. (2012). Survey of Metaheuristic Algorithms for Combinatorial Optimization. International Journal of Computer Applications, 58(19), 21 - 31.
2. Bahrehmand, A., Batard, T., Marques, R., Evans, A., & Blat, J. (2017). Optimizing layout using spatial quality metrics and user preferences. Graphical Models, 93, 25–38.
3. Banerjee, A., Quiroz, J. C., & Louis, S. J. (2008). A model of creative design using collaborative interactive genetic algorithm. Design Computing and Cognition, 8, 397-416.
4. Bausys, R., & Pankrašovaite, I. (2005, January). Optimization of architectural layout by the improved genetic algorithm. Civil Engineering and Management, 11(1), 13-21.
5. Bentley, P. J. (1998). The revolution of evolution in design: from coffee tables. Recent Advances in Soft Computing, (pp. 172–182).
6. Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is "Nearest Neighbor" Meaningful? In Proc. 7th Int. Conf. on Database Theory, (pp. 217–235).
7. Blum, C., & Roli, A. (2003). Metaheuristics in Combinatorial Optimization: Overview. ACM Computing Surveys, Vol. 35, No. 3, 268–308.
8. Caldas, L., & Norford, L. K. (1999). A Genetic Algorithm Tool for Design Optimization. ACADIA, 99.
9. Caldas, L., & Norford, L. K. (2002). A design optimization tool based on a genetic algorithm. Automation in Construction 11, 173–184.
10. Calixto, V., & Celani, G. (2015). A literature review for space planning optimization using an evolutionary algorithm approach: 1992-2014. XIX Congresso da Sociedade Ibero-americana de Gráfica Digital 2015, (pp. 662-671).
11. Chiu, C.-C., Cheng, Y.-T., & Chang, C.-W. (2012). Comparison of particle swarm optimization and genetic algorithm for the path loss reduction in an urban area. Applied Science and Engineering, 5(4), 371 - 380.
12. Cichocka, J. M., Migalska, A., Browne, W. N., & Rodriguez, E. (2017). SILVEREYE– the implementation of Particle Swarm Optimization algorithm in a design optimization tool. CAAD

Futures 2017, CCIS 724 (pp. 151–169). Springer Nature Singapore Pte Ltd.
13. Conceição António, C. A., Monteiro, J. B., & Afonso, C. F. (2014, December 5). Optimal topology of urban buildings for maximization of annual solar irradiation availability using a genetic algorithm. Applied Thermal Engineering, 73(1), 424-437.
14. Cubukcuoglu, C., Chatzikonstantinou, I., Ekici, B., & Sariyildiz, S. (2016). Multi-objective optimization through differential evolution for restaurant design. 2016 IEEE Congress on Evolutionary Computation (CEC) (pp. 2288 - 2295). Vancouver, BC, Canada: IEEE.
15. Cuevas, E., Barocio, E., Emilio, E., & Arturo, C. (2019). Introduction to Metaheuristics Methods. In Metaheuristics Algorithms in Power Systems (pp. 1 - 8). Springer Nature Switzerland AG 2019.
16. Datta, S., Roy, S., & Davim, J. (2019). Optimization Techniques: An Overview. In Optimization in Industry (pp. 1 - 13). Springer Nature Switzerland AG 2019.
17. Dino, I. G., & Üçoluk, G. (2017). Multiobjective Design Optimization of Building Space Layout, Energy, and Daylighting Performance. Journal of Computing in Civil Engineering, 31(5).
18. Dorigo, M., & Stützle, T. (2019). Ant Colony Optimization: Overview and Recent Advances. International Series in Operations Research & Management Science, 311 - 351.
19. Doulgerakis, A. (2007). Genetic programming + unfolding embryology in automated layout planning. Msc thesis, Bartlett School of Graduate Studies.
20. Du, T., Turrin, M., Jansen, S., & van den Dobbelsteen, A. (2018). A Review on Automatic Generation of Architectural Space Layouts with Energy Performance Optimization. 4thInternational Conference On Building Energy, Environmen, (pp. 856 - 861).
21. Dutta, K., & Sarthak, S. (2011). Architectural space planning. Artificial Intelligence Review December 2011, 311-321.
22. Elezkurtaj, T., & Franck, G. (1999, September 15-17). Genetic algorithms in support of creative. Architectural Computing from Turing to 2000, 645–651.
23. Fasoulaki, E. (2007). Genetic Algorithms in Architecture: A Necessity or a Trend? Generative Art Conference. Milan, Italy.
24. Finucane, E. L., Derix, C., & Coates, P. S. (2006). Evolving urban structures using computer optimization techniques. Generative Art.
25. Flack, R. W. (2011). Evolution of Architectural Floor Plans. European Conference on the Applications of Evolutionary Computation (pp. 313-322). Applications of Evolutionary Computation.
26. Gen, M., & Lin, L. (2009). Evolutionary Techniques for Automation. Springer Handbook of Automation, 487-502.
27. Gero, J. S., & Kazakov, V. A. (1997). Learning and re-using information in space layout planning problems using genetic engineering. Artificial Intelligence in Engineering, 12(3), 163–176.
28. Golabchi M., Garmaroudi A. A., Bastani H. (2012). Digital Architecture: CAD/CAM/CAE Technologies Application in Architecture, Tehran: University of Tehran.
29. Hansen, J. M., Raut, S., & Swami, S. (2010). Retail Shelf Allocation: A Comparative Analysis of Heuristic and Meta-Heuristic Approaches. Journal of Retailing 86 (1, 2010), 94–105.
30. Homayouni, H. (2007). A genetic algorithm approach to space layout planning. MSc thesis; University of Washington.
31. Homayouni, H. (2007). A Survey of Computational Approaches to Space Layout Planning (1965-2000). Computational Approaches to Space Layout Planning, 1 - 18.
32. Inoue, M., & Takagi, H. (2008). Layout Algorithm for an EC-based Room. IEEE Conference on Soft Computing in Industrial Applications (pp. 165–170). Muroran, Japan: IEEE.
33. Jagielski, R., & Gero, J. S. (1997). A genetic programming approach to the space. CAAD Futures, 875–884.
34. Jo, J. H., & Gero, J. S. (1998). Space layout planning using an evolutionary approach. Artificial Intelligence in Engineering, 12(3), 149-162.
35. Juneja, M., & Nagar, S. (2016). Particle swarm optimization algorithm and its parameters: A review. 2016 International Conference on Control, Computing, Communication and Materials (ICCCCM). 2016 IEEE.
36. Karlen, M. (2009). Space Planning Basics. Wiley, London 240.
37. Knecht, K., & Koenig, R. (2010). Generating floor plan layouts with k-d trees and evolutionary algorithms. GA2010 - 13th Generative Art Conference (pp. 238–253). Politecnico di Milano University, Italy: Domus Argenia Publisher.
38. Kumar, A., Dutta, K., Gupta, A., Badyal, S., & Rohan, D. (2017). Assisting an architect with alternative automated space layout designs using order crossover Genetic Algorithm in AutoCAD. 2017 International Conference on Advances in Mechanical Industrial, Automation and Management Systems (AMIAMS). 2017 IEEE.
39. Lawson B. (1980), How Designers Think, The Design Process Demystified. London: Architectural Press.
40. Li, L. (2012). The optimization of architectural shape based on Genetic Algorithm, Frontiers of Architectural Research. 392 - 399.
41. Li, Y., Chen, Y., Zhao, M., & Zha, X. (2018). Optimization of Planning Layout of Urban Building Based on Improved Logit and PSO Algorithms. Hindawi, Complexity, Volume 2018.
42. Liggett, R. S. (2000, March). Automated facilities layout: past, present and future. Automation in Construction, 9(2), 197-215.
43. Liu, J., Abbass, H. A., & Tan, K. C. (2019). Evolutionary Computation. In Evolutionary Com-

putation and Complex Networks (pp. 1-23). Springer Nature Switzerland AG 2019.
44. Mahdavinejad M., Refalian, G. (2014). Algorithmic Architecture: Data Processing and Advanced Quantitative Programming Methods in the Contemporary Architectural Design Process. Tehran: Islamic Azad University, Research Sciences Branch.
45. Mahdavinejad, M. (2014). Dilemma of Prosperity and Technology in Contemporary Architecture of Developing Countries. Naqsh-e Jahan, 4 (2), 36-46.
46. Makris, D., Ioannis, H., Georges, M., & Dimitri, P. (2006). MultiCAD - MOGA: A system for conceptual style design of buildings. International Conference on Computer Graphics and Artificial Intelligence: Conference Proceedings of the 9th, (pp. 73-84).
47. Mardomi, K., Soheilifard, M., Aghaazizi, M. (2015). Integration of Architecture & Structure in Optimizing Support's Location Using Genetic Algorithm Method (Case study: Cladding based on Iranian girih). Naqsh-e Jahan, 5(2), 65-75.
48. Michalek, J. J., & Papalambros, P. Y. (2002). Interactive design optimization of architectural layouts. Engineering Optimization, 34(5), 485–501.
49. Nagy, D., Lau, D., Locke, J., Stoddart, J., Villaggi, L., Wang, R., Benjamin, D. (2017). project Discover: An application of generative design for architectural space planning. Autodesk Research.
50. Nandanwar, M. K., Zadagaonkar, A. S., & Shukla, A. (2016, Jan - Feb). Path Planning through PSO Algorithm in Complex Environments. International Journal of Computer Science Trends and Technology (IJCST), 4(1).
51. Nisztuk, M., & Paweł, M. B. (2018). stability of contemporary tools for the computational design of architectural objects: Review, features evaluation and reflection. international Journal of Architectural Computing2018, Vol. 16(1), 58 –84.
52. Papadaki, I. N., & Chassiakos, A. P. (2016). Multi-objective construction site layout planning using genetic algorithms. Creative Construction Conference 2016, CCC 2016. 164, pp. 20 – 27. Procedia Engineering.
53. Papapavlou, A., & Turner, A. (2009). Structural evolution: A genetic algorithm method to generate structurally optimal Delaunay triangulated space frames for dynamic loads. 27th eCAADe Conference. Istanbul.
54. Poblete, B., & de la, C. I. (2011). Algoritmos genéticos como estrategia de diseño en arquitectura. Ph.D. thesis: Universitat Politècnica de Catalunya.
55. Rahbar, M., Mahdavinejad, M., Bemanian, M., Davaie Markazi, A. H., & Hovestadt, L. (2019). Generating Synthetic Space Allocation Probability Layouts Based on Trained Conditional-GANs. Applied Artificial Intelligence.
56. RazaviAlavi, S., & AbouRizk, S. (2017). Site Layout and Construction Plan Optimization Using an Integrated Genetic Algorithm Simulation Framework. Journal of Computing in Civil Engineering, 31(4), 04017011-1 - 04017011-10.
57. Rodrigues, E., Gaspa, A. R., & Gomes, A. (2013). Evolutionary strategy enhanced with a local search technique for the space allocation problem in architecture Part 1: Methodology. Computer-Aided Design.
58. Rosenman, M. A. (1997). The generation of form using an evolutionary. Evolutionary algorithms in engineering applications, 69–85.
59. Schnier, T., & Gero, J. S. (1996). Learning Genetic Representations as Alternative to Hand-Coded Shape Grammars. Artificial Intelligence in Design '96, 39–57.
60. Shi, Y., & Eberhart, R. C. (1998). A modified particle swarm optimizer. 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360). Anchorage, AK, USA, USA: IEEE.
61. Soddu, C. (2005). Visionary Variations of Milano.
62. Sönmez, N. O. (2015). Architectural Layout Evolution Through Similarity-Based Evaluation. international journal of architectural computing, 13(3+4), 271 – 279.
63. Sowmya, B., & Mp, S. R. (2013, August). Minimization of Floor Planning Area and Wire Length Interconnection Using Particle Swarm Optimization. International Journal of Emerging Technology and Advanced Engineering, 3(8), 321-330.
64. Su, Z., & Yan, W. (2015, November). A fast genetic algorithm for solving architectural design optimization problems. Artificial intelligence for engineering design analysis and manufacturing, 29(4), 457- 469.
65. Talbourdet, F., Michel, P., Andrieux, F., & Millet, J.-R. A knowledge-aid approach for designing high-performance buildings. Building Simulation, 6(4), 337–350.
66. Thakur, M. K., Kumari, M., & Das, M. (2010). Architectural layout planning using genetic algorithms. 3rd International Conference on Computer Science and Information Technology, pp. 5–11. IEEE.
67. Virirakis, L. (2003). GENETICA: A computer language that supports general formal expression with evolving data structures. Transactions on Evolutionary Computation, 7(5), 456–481.
68. Wang, B., & Malkawi, A. (2015). Genetic algorithm based building form optimization study for natural ventilation potential. 14th Conference of International Building Performance Simulation Association, (pp. 640 – 647). Hyderabad, India.
69. Wang, W., Rivard, H., & Zmeureanu, R. (2006). Floor shape optimization for green building design. Advanced Engineering Informatics, 20(4), 363-378.
70. Wetter, M., & Wright, J. (2004). A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based op-

timization. Building and Environment, 39(8), 989-999.
71. Wong, S. S., & Chan, K. C. (2009, September). EvoArch: An evolutionary algorithm for architectural layout design. Computer-Aided Design, 41(9), 649-667.
72. Yang, X.-S., & Karamanoglu, M. (2013). 1 - Swarm Intelligence and Bio-Inspired Computation: An Overview. Swarm Intelligence and Bio-Inspired Computation-Theory and Applications, 3-23.
73. Zhang, X., & Shiu, Y. Y. (2015, May). A directional mutation operator for differential evolution algorithms. Applied Soft Computing, 30, 529-548.

پژوهشگاه علوم انسانی و مطالعات فرهنگی
پرتال جامع علوم انسانی