Generative Reactor Design

Gwendolyn J. Chee

January 22, 2020

1 Generative Design

Generative design is an exploratory design method that autonomously generates optimal designs by iteratively varying design geometry to meet user-defined performance metrics [11, 16]. The user defines the design parameters and the generative design software helps the user create many solutions simultaneously [5]. This sometimes results in unanticipated unique solutions, that would have been difficult to discover using traditional methods [5]. Generative design varies the parameters of the problem definition [14]. At each iteration step, the design is evaluated on the performance metrics. Based on the results, the generative design algorithm changes the interval allowed for each design geometry variable, refining design constraints (problem definition) and moving towards designs that best meets performance metrics.

There is confusion to how generative design differs from other shape optimization tools. Generative design is more than topology optimization which has been around since 1988 [6]. Topology optimization requires the user to start with a complete design and the software improves it by removing material, it answers the fundamental engineering question of how to place material in a domain space to obtain best structural performance [21]. Generative design does not require the user to start with a complete design, but instead a few design constraints. Table 1 summarizes the differences between generative design and traditional design optimization tools.

Table 1: Comparison of generative design and traditional design optimization tools [4].

Point of Comparison	**Generative Design**	**Traditional Design Optimization Tools**
Initial input	A few design constraints	Complete design
No. of outcomes	Generates a wide set of designs that meets design constraints	Identifies one unique design that meets constraints
Solving strategies	Uses multiple strategies to solve design problem	Mainly uses topology optimization solutions

1.1 State of Work

Generative design is used in the following industries: automotive [9], aerospace [8], architecture, product design etc. For the automotive and aerospace industries, the goal of generative design is to decrease the weight of the vehicle while ensuring each part can continue to withstand the stresses and strains that are put on it within a safety factor. These industries can rely on commerical softwares such as Autodesk Fusion360 [5] and SolidWorks Topology [13] to produce their generative designs, since these softwares have structural simulation-based generative design technology.

1.2 Genetic Algorithms

Multi-objective design problems inevitably require a trade off between desirable attributes [8, 22]. In nuclear reactor design there are many trade offs, one example is the trade-off between neutron economy and fuel enrichment. A reactor design must have sufficient neutron economy to ensure criticality, but must also have a low fuel enrichment to reduce proliferation risk. Conflicting objectives means that there is no one perfect solution, but a set of equally optimal solutions [8]. Multi-objective problems are difficult to optimize, such problems cannot be handled by classical optimization methods such as gradient methods, because only the local optimum will be found [18]. Evolutionary algorithms have proven to be successful methods to optimize multi-objective problems [11] as they can find a solution near the global optimum [18]. The most popular evolutionary algorithms used to solve multi-objective problems are genetic algorithms [8, 11]. Genetic algorithms imitate natural selection to evolve solutions by (1) maintaining a population of solutions, (2) allowing fitter solutions reproduce, and (3) letting lesser fit solutions die off, resulting in final solutions that are better than the previous generations [18]. Figure 1 depicts the iterative process of using a genetic algorithm to solve a problem. The key part of this process is defining evaluation and termination criteria.

Two types of genetic algorithms are typically used to solve shape optimization problems: parametric and cell. Parametric genetic algorithms optimize complex shapes by varying parametric variables to meet the desired design performance [24]. Examples of parametric variables are: diameter of a sphere, twist angle of a cylinder, etc. Optimization of aerodynamic configurations [15], and truss and bridge structures [17] used parametric genetic algorithms. Cell genetic algorithms represent the shape of an object to be optimized in small subdivided rectangular domains (pixels in 2D, voxels in 3D) [18]. Figure 2 shows an example of cell representation. Cell genetic algorithms are advantageous compared to parametric genetic algorithms since the initial structure and topology of the object need not be created and is instead developed through the iterative optimization process [18]. Both parametric and cell genetic algorithms will be explored for the generative reactor design problem.

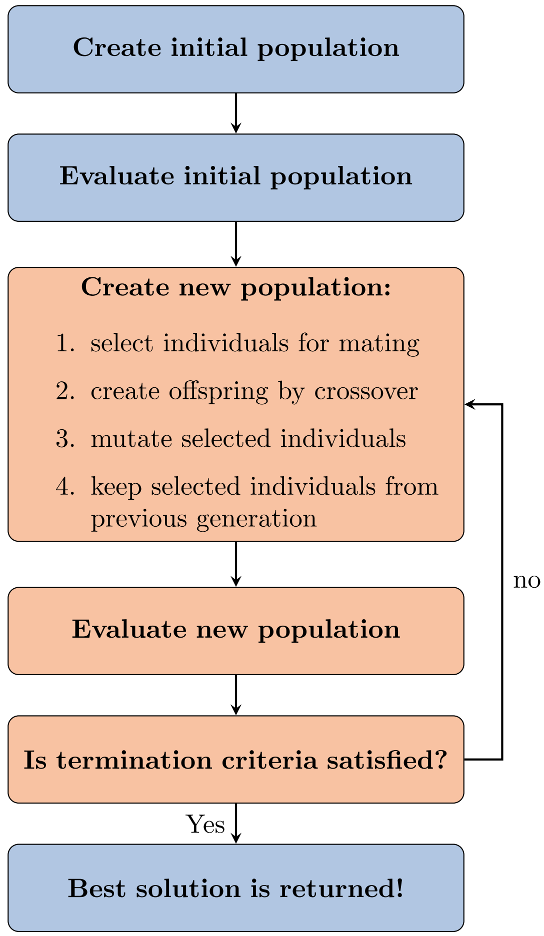

Figure 1: Process of solving a problem with genetic algorithm [18]. When a population does not meet termination criteria, a new generation is created. This occurs iteratively till the termination criteria is met.

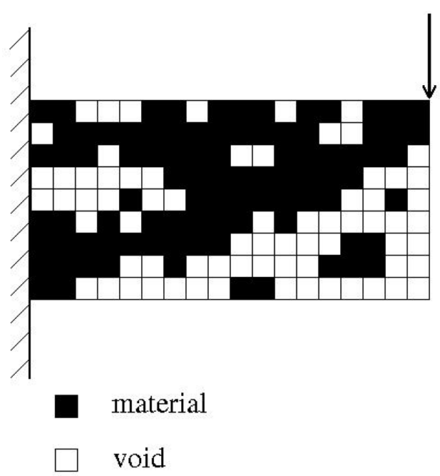

Figure 2: Cell representation in 2D [18].

2 Generative Reactor Design

For nuclear reactors, generative design is constrained by variables such as mass or volume of fuel, fuel enrichment, effectiveness of heat transfer, effective neutron multiplication factor, etc. Nuclear reactors not only experience physical forces, but also require evaluation of the neutronics of the system, therefore generative design of a nuclear reactor cannot make use of tools such as Autodesk Fusion360 or SolidWorks Topology. Therefore, a framework that couples well-developed advanced genetic algorithms with well-supported monte-carlo particle transport codes (Serpent [12], MCNP [25], etc.) and thermal hydraulics codes (RELAP7 [3] etc.) must be created to successfully produce generative reactor designs.

2.1 Workflow

Figure 3 depicts a framework for leveraging genetic algorithms to design nuclear reactors. This is a general framework that does not specify algorithms or analytical softwares. Instead, it provides placeholders for algorithm and software types, so that a user may choose the types of algorithms and softwares they want to use in their mission towards using generative design to design a nuclear reactor. Table 2 lists algorithm and software types that could be used in the framework.

Table 2: Example algorithm and software types to populate generative reactor design framework (Fig 3).

Framework Component	**Examples**
Genetic algorithm	parametric, cell [18]
Computer-aided design software	Trelis [1], FreeCAD [10], SolidWorks [13], grasshopper3d [20], GenerativeComponents [2]
Neutron transport code	Serpent [12], MCNP [25], SCALE [7], OpenMC [19]
Thermal hydraulics code	RELAP5-3D [23], RELAP7 [3], TRACE [26]
User-defined metrics/criteria	keff, heat transfer rate, fuel enrichment, mass of fuel

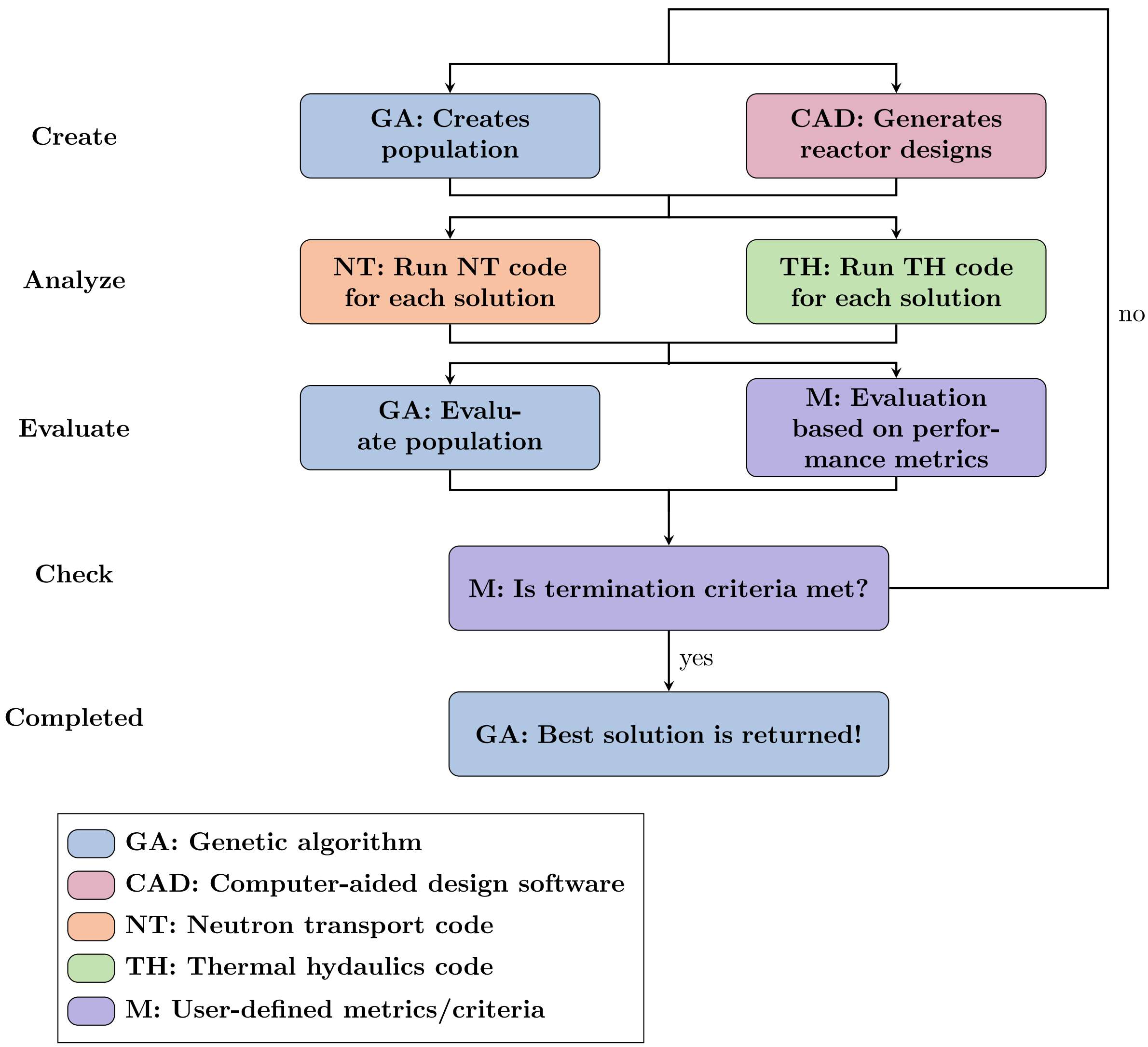

Figure 3: Generative reactor design framework. Each component of the framework is user-selected.

References

[1] Trelis (Version 16.5), 2018.

[2] ROBERT Aish. Bentley's Generative Components: a Design Tool for Exploratory Architecture. *Bentley Systems Inc*, 2003.

[3] David Andrs, Ray Berry, Derek Gaston, Richard Martineau, John Peterson, Hongbin Zhang, Haihua Zhao, and Ling Zou. Relap-7 level 2 milestone report: Demonstration of a steady state single phase pwr simlulation with relap-7. Technical report, Technical report, Idaho National Laboratory, 2012.

[4] Autodesk. Fusion 360 Generative Design, 2020.

[5] Autodesk. Autodesk Fusion 360, 2020.

[6] Martin Philip Bendse and Noboru Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, November 1988.

[7] J. A. Bucholz. SCALE: A modular code system for performing standardized computer analyses for licensing evaluation. NUREG NUREG/CR-0200-Vol.2-Bk.2; ORNL/NUREG/CSD-2-Vol.2-Bk.2 ON: DE82013370; TRN: 82-013156, Oak Ridge National Lab., TN (USA), 1982.

[8] Jonathan Byrne, Philip Cardiff, Anthony Brabazon, and Michael ONeill. Evolving parametric aircraft models for design exploration and optimisation. *Neurocomputing*, 142:39–47, October 2014.

[9] Raymond Deplazes. Autodesk Collaborates With Volkswagen Group on Generative Design in Electric Showcase Vehicle, July 2019.

[10] Brad Falck, Daniel Falck, and Brad Collette. *Freecad [How-To]*. Packt Publishing Ltd, 2012.

[11] Sivam Krish. A practical generative design method. *Computer-Aided Design*, 43(1):88–100, January 2011.

[12] Jaakko Leppanen, Maria Pusa, Tuomas Viitanen, Ville Valtavirta, and Toni Kaltiaisenaho. The Serpent Monte Carlo code: Status, development and applications in 2013. *Annals of Nuclear Energy*, 82:142–150, August 2014.

[13] Matt Lombard. *SolidWorks 2007 Bible*, volume 529. John Wiley & Sons, 2008.

[14] Justin Matejka, Michael Glueck, Erin Bradner, Ali Hashemi, Tovi Grossman, and George Fitzmaurice. Dream Lens: Exploration and Visualization of Large-Scale Generative Design Datasets. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pages 1–12, Montreal QC, Canada, 2018. ACM Press.

[15] Raino AE Mkinen, Jacques Priaux, and Jari Toivanen. Multidisciplinary shape optimization in aerodynamics and electromagnetics using genetic algorithms. *International Journal for Numerical Methods in Fluids*, 30(2):149–159, 1999.

[16] Sangeun Oh, Yongsu Jung, Seongsin Kim, Ikjin Lee, and Namwoo Kang. Deep Generative Design: Integration of Topology Optimization and Generative Models. *Journal of Mechanical Design*, 141(11), November 2019.

[17] A. M. Raich and J. Ghaboussi. Evolving structural design solutions using an implicit redundant genetic algorithm. *Structural and Multidisciplinary Optimization*, 20(3):222–231, 2000.

[18] Gbor Renner and Anik Ekrt. Genetic algorithms in computer aided design. *Computer-aided design*, 35(8):709–726, 2003.

[19] Paul K. Romano and Benoit Forget. The OpenMC Monte Carlo particle transport code. *Annals of Nuclear Energy*, 51:274–281, January 2013.

[20] David Rutten. *Grasshopper3D*. 2015.

[21] Ole Sigmund and Kurt Maute. Topology optimization approaches: A comparative review. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055, December 2013.

[22] Herbert A. Simon. *The sciences of the artificial*. MIT press, 2019.

[23] G. Strydom, A. S. Epiney, A. Alfonsi, and C. Rabiti. Comparison of the PHISICS/RELAP5-3d Ring and Block Model Results for Phase I of the OECD/NEA MHTGR-350 Benchmark. *Nuclear Technology*, 193(1):15–35, January 2016.

[24] Peter Von Buelow. ParaGen: Performative Exploration of generative systems. *Journal of the International Association for Shell and Spatial Structures*, 53(4):271–284, 2012.

[25] C. J. Werner, J. S. Bull, C. J. Solomon, F. B. Brown, G. W. McKinney, M. E. Rising, D. A. Dixon, R. L. Martz, H. G. Hughes, and L. J. Cox. MCNP6. 2 release notes. *Tech. Rep. LA-UR-18-20808, Los Alamos National Laboratory, Los Alamos, NM*, 2018.

[26] Y. Xu, T. Downar, A. Ward, T. Kozlowski, and K. Ivanov. Multi-physics coupled code reactor analysis with the US NRC code system TRACE/PARCS. Technical report, American Nuclear Society, 555 North Kensington Avenue, La Grange Park, IL 60526 (United States), 2006.