Building meta-optimization: A study on the reuse of previous simulation data to reduce computational costs

Lucas Camilotti¹, Roberto Zanetti Freire¹, Nathan Mendes²

¹Pontifical Catholic University of Parana (PUCPR), Industrial and Systems Engineering Graduate Program (PPGEPS), Curitiba, Brazil
²Pontifical Catholic University of Parana (PUCPR), Mechanical Engineering Graduate Program (PPGEM), Curitiba, Brazil

Abstract
The use of simulations in building parameter optimization has become an established methodology in the past decades. Although different techniques employ varied numbers of total simulations, these are performed in isolation, with little to no interaction between them. In this context, we explored the possibility of re-utilizing previous data to speed-up future simulations. The Domus software was assumed as a simulation tool, and a case study considering the view factor data, which represents how the long-wave radiation heat transfer is calculated using a geometrical factor, was considered. This study presents a strategy to reuse the view factor calculation during building design considering the input vector changes of the optimization method. Results showed a considerable reduction of the optimization time. Finally, the methodology showed no drawbacks, with no improvements in the worst scenario.

Key Innovations
- Re-utilization of simulation data.
- Speed up of the optimization process.

Practical Implications
The proposed approach presented in this study may vary in terms of results and applicability depending on the simulation software utilized and the optimization problem tackled.

Introduction
The increasing energy consumption around the world has become a considerable challenge, with not only the overall consumption more than doubling since the 70s but with future predictions showing a similar pattern (IEA, 2019). Confronted with this scenario, researches have been performed to find ways to mitigate this problem. In the building sector, building simulations are considered to estimate and reduce the energy demand during the design and retrofit stage considering programs capable of replicating the physical phenomena inside and outside a building that affects heat and moisture transfer through building materials. Considering the advancements of hardware in recent decades, the use of simulation tools has become more accessible, and today, more than 400 different options exist (Attia et al., 2013).

The possibility of calculating and predicting the building hygrothermal behavior through simulations, variables such as energy usage, temperature, relative humidity, and thermal comfort has allowed the method to be coupled with optimization problems. By using physical/occupational characteristics of the building and the environment as parameters, an optimization problem can be formulated, with the simulated variables acting as optimization objectives (Nguyen et al., 2014). Many techniques have been applied to solve these kinds of problems, such as direct optimization methods (Bamdad et al., 2017), meta-heuristics (Delgarn et al., 2016) and artificial neural networks (Ilbeigi et al., 2020), all of them requiring a certain of simulations to be performed.

As simulations can reach high levels of complexity, especially for complex building geometries, they tend to take a considerable amount of time to finish and, is usually the interest of the designer to perform as few simulations as possible. In this context, techniques such as parallelism (Labib and Baltazar, 2020; Yang et al., 2014; Zhang, 2009) and surrogate models (Barnes and McArthur, 2020; Li et al., 2017) have shown some degree of success. During these, simulations are treated in an isolated form, a factor that can facilitate the process since they can be performed independently of outside factors. However, this can also be a drawback since it completely disregards data previously generated data that could be potentially reused to avoid performing repeated calculations.

In this work, we tackle this possibility and validate the proposed methodology using the Domus simulation software, re-utilizing the view factor calculation. The view factor represents how the long-wave radiation heat transfer is calculated using a geometrical factor. This calculation procedure is performed before the simulation using analytical and numerical methods depending on the building geometry, a calculation procedure that could be time-consuming. The
optimization of geometrical variables, e.g., window-to-wall ratio (WWR), indicates that the view factor must be recalculated after each modification.

Based on the previously mentioned problem involving optimization and building simulation, this work presents an innovative strategy to reduce simulation time through the reuse of previous simulation data. This new methodology can potentially provide another way to further reduce the computational time of costly parametric building optimizations, and even be used alongside other approaches, such as parallelization and surrogate models, as they can be applied independently.

The following sections present a brief overview of simulations, building optimization, and the view factor calculation to be reused. We then define our case study, along with the two scenarios in which the experiment is run: sequential and parallel. Finally, we present the results and give a brief discussion on the methodology, along with possible future steps to be taken.

**Literature Review**

**View factor**

A building simulation is a complex process comprised of several sequential steps, such as hygrothermal interactions between the building envelope and both inside and outside environments, ventilation, thermal loads caused by people and equipment to name a few (Hensen and Lamberts, 2012). Among these phenomena, radiation plays an important role in the heat exchange among the facades of a thermal zone, and a way to estimate these interactions is using the view factor value (Gupta et al., 2017).

The view factor calculation is a process used to determine the influence of long-wave radiation from one facade to another, where the goal of the view factor calculation is to determine how much of each facade is visible to the other (Figure 1). Considering two surfaces, \( i \) and \( j \), and their normals \( (n_i \) and \( n_j) \), the view factor calculation utilizes the line-of-sight \( r \) between infinitesimal elements in the surfaces, and their angles of incidence \( (\theta_i \) and \( \theta_j) \) (Figure 1).

Therefore, it is a process entirely dependent on the geometry of the building (Hensen and Lamberts, 2012), and techniques to calculate the view factor range from analytical formulations to finite element methods (Augusto et al., 2007).

**Building design optimization**

A multi-objective optimization problem can be generally described as the search for the input vector \( \theta \) that minimizes the vector of functions \( J \) (Miettinen, 2008). In mathematical terms, it can be expressed as presented in Equation (1).

\[
\theta^* = \arg \min_{\theta \in \mathbb{R}^n} J(\theta) \rightarrow J(\theta) = [J_1(\theta), ..., J_m(\theta)]. \quad (1)
\]

The input vector \( \theta \) may as well be under a set of constraints, usually denoting physical limitations, when the problem addresses the building design. Additionally, it’s often impossible to optimize all objectives simultaneously, and a compromise has to be reached. This differentiates the methodology from mono-objective optimization in the sense that more than one solution for the problem will be available and the selection of which one best satisfies the problem is up to the designer. This set of solutions is denominated Pareto set, while its image on the objective space can be defined as the Pareto front (Miettinen, 2008). One example of the previously mentioned definitions can be verified in Figure 2.

In the context of building design optimization, input variables usually represent physical characteristics of the building (Bre and Fachinotti, 2017), occupational factors (Delgarm et al., 2016), and climate-related data (Chai et al., 2020), while the results of the simulation, represent the energy usage predictions (Gagnon et al., 2019), thermal comfort indicators (Delgarm et al., 2016), and life-cycle costs (Carreras et al., 2016), those considered the objectives to be optimized.
View factor reuse

In the Domus software, one way to calculate the view factor is considering the finite element method (Augusto et al., 2007), where triangulation is used to subdivide the building’s facades into smaller areas, as shown in Figure 3. The view factor is then calculated on an element-to-element basis, from one surface to another. Although smaller areas for the triangles can be used to achieve more precise results, this approach considerably extends the calculation time, which is performed before the whole-building energy simulation. Once finished, Domus updates the building file by inserting the calculated view factor, and the simulation process starts.

While it may not seem more than a necessary step during an isolated simulation, it becomes a concern to consider during an optimization process comprised of multiple simulations. Once not all design variables of a problem will necessarily cause modifications in the building geometry, the same view factor value could be assumed for different design vectors. This signifies that the view factor for the same geometry could potentially be calculated multiple times, wasting time and computational resources, a point that could be explored to optimize the optimization procedures.

By first considering the complete vector of the n design variables of the problem, see $\mathbf{I}$ in Equation (2), a subset $\mathbf{I}_v$ can be constructed, containing the k variables that will impact the overall geometry of the building, and by consequence, the view factor value (3).

$$\mathbf{I} = [I_1, I_2, ..., I_n] \quad (2)$$

$$\mathbf{I}_v = [I_1, I_2, ..., I_k] \rightarrow \mathbf{I}_v \subseteq \mathbf{I} \quad (3)$$

This will be highly dependent on the variables of the problem, with $\mathbf{I}_v = \emptyset$ on the best-case scenario, signifying that no inputs alter the view factor and it only needs to be calculated once; and with $\mathbf{I}_v = \mathbf{I}$ on the worst case, signifying that all variables will impact it and a calculation will always be necessary.

If variables of $\mathbf{I}_v$ are discrete (continuous variables can be discretized), there will be a finite number of combinations $\theta_v$, that will result in a value for the view factor. As simulations are performed, the calculated view factor can be extracted from the building file, mapped to their respective $\theta_v$, and stored in a database. Before new simulations are performed, the $\theta_v$ can be queried, and if it exists, this view factor can be reused by inserting it into the building file, skipping the calculation process, and going straight to the simulation.

Building project assumed as a case study

To validate the methodology, we applied the proposed strategy to a building design optimization problem running under different scenarios and compared the result with the same problem without the view factor reuse strategy. We utilized an adapted version of a building design optimization study which was presented in Camilotti and Freire (2020). This adaption considers small modifications on the design variables and objectives, aiming for better fitting of the evaluated criteria, as described in the sequence.

Figure 4 presents the building geometry, a common type of residential building found in the southern region of Brazil, more specifically in the city of Curitiba. The city is located at 925 meters above sea level, at a latitude of 25.43° South and longitude of 49.27° West, and is known for its temperate climate and well-defined seasons throughout the year. An overview of the thermal zones is shown in Figure 5, and a description for each one in Table 1. The building is 2.35 m in height. The building contains heat-generation equipment in most zones, but no artificial climatization system. The respective loads and consumption for each zone.
Both internal and external walls share the same construction pattern of ceramic brick with a concrete finish on both sides, while the roof is made of fiber-cement tiles, with a mixed tile finish in the interior. Insulation is usually not considered for these types of buildings, and thus, it possesses a high heat transfer rate (Table 3). The building windows (except for the ones in the bathrooms) are 1 m² and are opened in the intervals described in Table 4.

The presence of these many windows prevents Domus from utilizing an analytical approach for the view factor calculation, which it is capable of for sufficiently simple geometries. Additionally, even though not the case for this study, other real-world cases may also present other complications, such as non-orthogonal geometry. This factor justifies the application and evaluation of the data re-utilization method proposed in this paper.

While the three first variables are continuous, the two window-wall ratio variables were discretized, considering increments of 0.05. That was done to limit the number of possible view factor configurations to a finite amount. If continuous variables were used, there would an infinite number of possible combinations, making impossible the construction of a finite database of view factor values.

### Optimization objectives

Since no artificial ventilation, heating, or cooling system was considered in the building, the monthly energy use tends to be nearly constant and is thus not considered as an objective. The thermal and air quality, however, are still important factors that impact the daily stay of the occupants, and thus are considered.

The first optimization objective measures thermal comfort and utilizes the absolute value of the Predicted Mean Vote (PMV) thermal comfort scale proposed by Fanger (1972). The second objective measures indoor air quality and utilizes the negative value of a scale proposed by Fang (1998).

In both cases, thermal comfort $C_z$ and indoor air quality $Q_z$ are sampled at the zone $z$ for each time-step $s$ of the simulation, and then averaged considering all four zones of interest, as formulated in Equations 4 and 5. The goal is to minimize both objectives.

\[
J_1(\theta) = \frac{1}{4S} \sum_{s=1}^{S} \sum_{z=1}^{[1,2,5,6]} |C_z(\theta, s)| \quad (4)
\]

\[
J_2(\theta) = -\frac{1}{4S} \sum_{s=1}^{S} \sum_{z=1}^{[1,2,5,6]} Q_z(\theta, s) \quad (5)
\]
Algorithm and parameterization

The NSGA-II (Deb et al., 2002) algorithm was chosen to carry out the optimization processes. It is a population-based evolutionary meta-heuristic with its roots in the process of evolution and natural selection and considered a well-established multi-objective optimization algorithm in the field of parametric optimization, including building design (Brownlee and Wright, 2015; Chaturvedi et al., 2020; Delgarm et al., 2016).

The parameterization for the algorithm was kept the same as in the original study (Camilotti and Freire, 2020), which includes 100 iterations and ten times the number of design variables as population size (Table 7).

Table 7: Parameterization of the NSGA-II algorithm.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of iterations</td>
<td>100</td>
</tr>
<tr>
<td>Population size</td>
<td>50</td>
</tr>
<tr>
<td>Mutation power</td>
<td>20</td>
</tr>
<tr>
<td>Crossover power</td>
<td>20</td>
</tr>
</tbody>
</table>

Experiments

To better evaluate the efficacy of the methodology under a different scenario, the optimization was carried out in both sequential and parallel scenarios. In theory, as the number of parallel simulations increases, so does the chance of simulating a building with the same view factor, which decreases the efficacy of the approach. For our parallel case, we choose a total of 8 parallel simulations. To account for the stochastic nature of the optimization algorithm utilized, 11 independent runs were carried out for both experiments, resulting in a total of 22 runs.

Even considering that the design space of the optimization problem will generate a wide range of possible input combinations, these should not alter considerably the final simulation time itself. A set of 100 random inputs is simulated to construct an average of the time taken for a simulation to complete with the view factor calculation. This value is then used to create an estimation of how much time it would take to finish all the optimization runs without the view factor re-utilization.

Finally, the optimization with view factor reuse is carried out and measured. The times are then compared to the theoretical estimations, and the percentage improvement over the default method is presented.

Results

This section presents the results in terms of simulation time considering and not considering the view factor reuse strategy presented in this study.

Simulation time estimation

After performing 100 simulations with randomized inputs, the average simulation and optimization times are calculated and shown in Table 8. The estimation for the parallel experiment is the theoretical ideal case, where no time is lost with solution scheduling and idle simulators.

Table 8: Theoretical simulation and optimization time without re-utilizing view factor data.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>One simulation</td>
<td>(121 ± 3) s</td>
</tr>
<tr>
<td>Sequential optimization</td>
<td>(168.05 ± 0.06) h</td>
</tr>
<tr>
<td>Parallel optimization</td>
<td>(21.01 ± 0.06) h</td>
</tr>
</tbody>
</table>

Simulation time with view factor reuse

When reusing view factor data, the sequential simulation time could be drastically decreased, up to 66.67% in the best-case scenario. For the parallel case, although many repeated view factor calculations were performed, an improvement of 63.68% could be achieved in the best-case scenario.

Table 9: Total optimization time (in hours) with view factor re-utilization.

<table>
<thead>
<tr>
<th>Context</th>
<th>Min.</th>
<th>Mean</th>
<th>Max.</th>
<th>Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>56.01</td>
<td>56.45</td>
<td>56.90</td>
<td>0.41</td>
</tr>
<tr>
<td>Parallel</td>
<td>7.63</td>
<td>7.81</td>
<td>8.09</td>
<td>0.20</td>
</tr>
</tbody>
</table>

Discussion

Results showed a decrease in overall optimization time in all experiments. From the perspective of both objectives, when simulations with and without view factor reuse were considered, no significant changes could be observed. This was expected, as all runs used the same algorithm and parameterization, and the approach is only capable of reducing optimization time. For our chosen problem specifically, the improvements in simulation time were significant due to the considerable time that the view factor calculation took during each simulation.

Dependency in number of simulations

Although the time saves for sequential and parallel optimizations were similar, that was likely caused due to the small number of possible view factor configurations (89) in comparison to the number of performed simulations (5,000). However, if the numbers were similar, the negative impacts of the repeated calculations made during the parallel optimization would be much more expressive.

The results can be interpreted as a trade-off between parallelism and data reuse, as the more simultaneous simulations are performed, the less effective the reuse approach becomes. Although it can still improve optimization time if used along with parallel simulations. The approach is a lot more suited to a sequential environment when there is a lack of computational resources.
Nearly no disadvantages

It is important to note, however, that no run reusing the view factor data took longer than the ones not making use of the approach. This points to a method that did not provide any drawbacks, and that can be potentially applied to any problem without causing negative impacts on the optimization time. Additionally, as the approach does not impact the evaluation of the objective function, the results of the optimization process are not compromised.

Development of an automated way

The process of implementation can pose a significant challenge, especially for those without a background in software development and experience on the simulation tool works internally. For our case, we implemented this feature on an automated specialized optimization tool for the Domus simulation software. This is done to remove the implementation burden from the user, as well as other barriers found in building parametric optimization methodology (Camilotti, 2021).

From a more general standpoint, the same process could potentially be applied to other specialized optimization tools found in the literature. While generalized optimization tools provide a wide range of possibilities and potential to be used in any category of problem, specialized tools are designed to tackle a specific subset (Attia et al., 2013). As specialized tools rely on a single simulation software, they may take full advantage of that, and exploit program-specific details to benefit the overarching optimization process.

This poses a potential scenario for implementors of optimization tools to explore techniques and approaches, such as the one proposed in this paper, that works through simulation data reuse.

Conclusion

This paper explored the possibility of re-utilization of previously calculated data to accelerate future simulations during an optimization process involving building simulation and the Domus tool. The Domus software view factor calculation was defined as the focal to validate the proposed approach. As the view factor is solely dependent on the building geometry, not all variables of an optimization problem would impact the view factor calculation procedure, and the same view factor from previous simulations could be re-utilized depending on the given design variable vector.

Results showed that a decrease in simulation time can be achieved, especially in a sequential execution scenario, where not enough resources are available for simulations. Furthermore, the approach showed no negative impacts in simulation time for any of the runs, showing that the methodology, aside from implementation complexity, showed no weaknesses.

Since this study was also limited to discrete variables due to the necessity of building a finite database of view factor values, a possible next step could make use of a surrogate approach. In this way, a continuous model could be created for the view factor as simulations are performed, instead of a discrete mapping scheme.

It should be noted, that the results are entirely dependent on several factors, such as the simulation software utilized, the optimization algorithm, and the problem being solved (building characteristics and design variables), to name a few. In this context, more research could not only prove useful to adapt other tools for this kind of methodology, but also motivate the development of more elaborate integration mechanisms for optimization tools to work in tandem with simulation software and make full use of its features.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. The authors thank the Polytechnic School of the Pontifical Catholic University of Parana (PUCPR) for providing the necessary materials and equipment for the development of this research.

References


Chai, J., P. Huang, and Y. Sun (2020). Differential evolution - based system design optimization for net zero energy buildings under climate change. Sustainable Cities and Society 55, 44.


