Optimizing Neighborhood-Scale Walkability

Andrew J. Sonta, S.M.ASCE[1]; and Rishee K. Jain, Ph.D., A.M.ASCE[2]

[1]Urban Informatics Lab, Dept. of Civil and Environmental Engineering, Stanford Univ., 473 Via Ortega, Room 269B, Stanford, CA 94107. E-mail: asonta@stanford.edu
[2]Urban Informatics Lab, Dept. of Civil and Environmental Engineering, Stanford Univ., 473 Via Ortega, Room 269A, Stanford, CA 94107. E-mail: rishee.jain@stanford.edu

ABSTRACT

Many designers and researchers have grappled with the problem of optimally locating buildings and use types in a neighborhood-scale development. But little work has used data-driven optimization to aid in creating urban design schemes. The paradigm of single-use Euclidian zoning has heavily impacted the way our neighborhoods, cities, and suburbs are designed, resulting in the physical separation of uses. However, as we grapple with emerging issues of environmental and social sustainability in cities, there is a pressing need to consider alternative urban designs that require less dependence on personal automobiles and that foster healthier cities. In this paper, we develop a methodology for (1) automatically assessing the walkability of neighborhoods by adopting a common walkability metric and (2) optimizing the layout of buildings and amenities across a known grid in order to maximize the walkability metric. We apply this methodology to a case study of the Potrero Hill neighborhood in San Francisco, California. We find that, in comparison to the existing layout that can be characterized by Euclidian-style separation of uses, the optimized layout suggests distributing amenities across the street network, resulting in a two-fold increase in walkability. This tool and analysis have the potential to provide computational and data-driven support for urban designers and researchers hoping to understand and improve the walkability of urban spaces.

INTRODUCTION

The design and planning of urban spaces has a long and storied history, with ideas about the best use of urban space dating to Ancient Rome. Some of the earliest plans for cities—including Paris, London, and Washington, D.C.—were created by master-builders or architects with the backing of government. Today, almost all cities implement some form of urban planning vis-à-vis rules about building form, use, and location (Best 2016).

Single-use zoning, also known as Euclidian zoning—in which cities are divided into areas with specific rules for building height, use, and density—became possible and prevalent after the landmark case *Village of Euclid v. Amber Realty Co.* in 1926 (Wickersham 2000). In the period following World War II, the physical separation of functional uses in cities became both feasible and desirable due to increased rates of property ownership and use of the personal automobile (Best 2016). Even in dense cities, single-use zoning replaced existing mixed-use developments (Jacobs 1961). However, recent environmental and social concerns (*e.g.,* the public and planetary health consequences of automobile pollution) have led urbanists, local governments, and city planners to rethink rigid Euclidian rules. One important reason is that developments with a mix of uses reduce residents' dependence on personal vehicles. Aside from the obvious environmental implications, urbanists such as Jane Jacobs (1961) have argued that increased use of sidewalks and reduced dependence on cars create vibrant, socially resilient communities. As a result, the study and desirability of walkable communities have increased greatly in recent years.

Recently, we have also seen a vast surge in urban data resources and computing power. Given these resources, researchers now have a unique opportunity to put these concepts of ideal urban form to the test. This dual paradigm of evolving urban planning concepts and maturing cyber- physical analysis has the potential to validate or entirely upend the consensus of what makes a city effective. As a result, there is a pressing need to explore how computing tools such as optimization can augment current decision-making processes around zoning and rule-making in urban areas. Given the complexity of city planning—which includes street and path layouts, building geometries, and use types—various approaches must be explored. In this paper, we develop a methodology for maximizing the walkability of a neighborhood-scale development by choosing the layout of buildings in an existing street grid, given the number of buildings, each building's prescribed use, and possible lots for placing each building. In a case study, we compare the existing layout of a neighborhood in San Francisco, CA with an optimized layout that distributes key urban amenities quite differently.

BACKGROUND

Recent urban design research has identified the concept of walkability as a key metric in addressing environmental and social sustainability concerns in cities. Porta and Renne (2005) include interconnectedness and accessibility of the street network as a critical component of their tool for assessing the sustainability of urban form. Furthermore, they argue that in addition to these street network characteristics, the community must colocate a diversity of land uses so that multiple uses can be accessed by walking.

Some studies have used heuristic algorithms to optimize the walkability of neighborhood-sized developments. These heuristics produce best-practice guidelines for walkable communities built on architectural and urban design expert knowledge (Southworth 2005). While these guidelines can be important and effective tools for urban designers in their planning work, they lack an objective score that can be automatically calculated and applied quickly to various design alternatives. Exploiting automated computational tools can greatly expand the solution space and reveal previously overlooked options.

A few recent studies have explored the notion of optimizing physical layouts of structures in real-world environments. Razavialavi and Abourizk (2017), for example, developed a genetic algorithm framework for optimizing layout on a construction site. Rakha and Reinhart (2012) developed a generative modeling platform that assesses different parametric urban massing forms for walkability. They adopt the walkability scoring system discussed in Carr et al. (2010) as the metric for optimization, and they utilize a genetic algorithm to optimize walkability by placing different uses. This previous quantitative optimization work, while valuable in advancing the role of computing in assessing urban form, has not been applied to evaluate the performance of existing urban areas. Furthermore, the implications of the walkability optimization results have not been fully explored, especially in their relationship to conventional wisdom about effective urban design.

METHODOLOGY

The purpose of the methodology outlined in this section is to maximize a quantitative walkability metric of a neighborhood-scale development given constraints about the number and possible locations for each building type. The methods we outline here can be used to compare optimized layout of buildings and amenities with alternative designs, including those created through heuristics or those that already exist in cities.

Our approach follows a procedure with three main steps:

1. *Problem definition*—define the walkability objective function and how it is measured, and define the solution space (*i.e.*, possible locations of buildings) as well as the constraints (*i.e.*, number of each building type available).
2. *Generate random designs*—develop a routine for creating a population of randomly generated designs, which are defined by the locations of each building type.
3. *Optimize design*—assess designs, create a new set of designs based on the best performers, and repeat until convergence.

Problem Definition

In order to accomplish the ultimate goal of maximizing walkability, we first need a walkability metric and a set of variables that can be changed to vary this metric. In this paper, we adopt the metric discussed in Rakha and Reinhart (2012) and hereafter refer to it as the *Street Score*. The Street Score is a value between 0 and 100, and it is calculated for one residential unit at a time. In its most general form, the Street Score (S) is calculated as the sum of walking distance scores between each residential unit and a prescribed number of different amenities (*e.g.*, parks, restaurants, grocery) as follows:

$$S = \frac{1}{A}\left(\sum_{a=1}^{A} \mathbf{w}_a \cdot \mathbf{s}_a\right)100$$

where the vector $\mathbf{w}_a$ is the weighting vector for amenity a and the vector $\mathbf{s}_a$ is the distance score vector for amenity a (defined below). The vectors $\mathbf{w}$ and $\mathbf{s}$ can have different sizes for each amenity, but the size of $\mathbf{w}_a$ is always equal to the size of $\mathbf{s}_a$. This difference in vector sizes is simply a function of the fact that the implementation of the Street Score metric can specify different numbers of each amenity to consider in the scoring (*e.g.*, 2 coffee shops vs. 20 restaurants). The distance score is calculated as a function of the walking distance (x) from the residential unit to the amenity under consideration. This walking distance must be defined according to the street grid (*e.g.*, in a perpendicular north-south, east-west grid, the distance would be the L1 norm, or the "Manhattan" walking distance). For instance i of amenity a, the distance score is calculated as follows:

$$s_{a,i} = \begin{cases} 1 & x < d_1 \\ 1-[0.9/(d_2-d_1)](x-d_1) & d_1 \le x < d_2 \\ 0.1-[0.1/(d_3-d_2)](x-d_2) & d_2 \le x < d_3 \\ d & x > d_3 \end{cases}$$

for walking distance x, where d_1, d_2, and d_3 are set by the user. The result is a score, based on the distance, that is scaled between 0 and 1.

The design variables for the problem are the locations of buildings and amenities. The categories of buildings/amenities can be set according to the individual problem, but it is important to note that the initial work by Rakha and Reinhart used residential units, restaurants, generic shops, coffee shops, bookstores, banks, grocery stores, parks, schools, and entertainment venues. In this initial work, we simplify the design space by defining specific lots at which different buildings of different sizes can be placed.

Optimization

Given a calculable objective function (Street Score) and design variables (locations of buildings/amenities), the next step is to perform optimization. We utilize a genetic algorithm, as these have been shown in previous work to be effective in optimizing physical layouts with large solution spaces (Rakha and Reinhart 2012; Razavialavi and Abourizk 2017). Each step in the genetic algorithm requires creating a routine specific to this specific problem setting. These subroutines are outlined in this subsection. To initialize the population, we must be able to create random designs. Given a street grid with possible lots as well as a building stock with numbers of available building/amenity types, we can randomly assign each building type to a lot. For implementation, it can be simplest to randomly assign larger amenities—that may take up multiple lots—first, working from largest amenities to smallest. Once an initial population is created, a Street Score can be calculated for each neighborhood design. To adapt the Street Score methodology from a single residential unit to an entire neighborhood, we randomly sample residential units from a neighborhood, calculate the Street Score for each, and average the results. Given Street Scores calculated for all neighborhood designs, we can select parents that will help us create future generations. Different selection criteria can be utilized, including truncation selection, tournament selection, and roulette wheel selection, as discussed in Kochenderfer (2018).

Once parents are selected, crossover and mutation must be implemented to create new generations. The process for crossover is detailed in Algorithm 1. The notion is to randomly choose the location of each building/amenity from the parents' locations for that building/amenity. First, all non-residential buildings are selected from the parents and placed, and then the residential units are filled in randomly. The concept of simulated annealing can be incorporated into the overall genetic algorithm through modification of simple crossover (and mutation), as discussed in Adler (1993). In simulated annealing crossover, a child is created from two parents, and it is always accepted if it performs better than the parents. If it performs worse than its parents, it is accepted with a probability that shrinks over generations. Formally, the child is accepted with the following probability:

$$\begin{cases} 1 & \Delta S \leq 0 \\ min\left(e^{-\Delta S/t},1\right) & \Delta S > 0 \end{cases}$$

Algorithm 1 Crossover

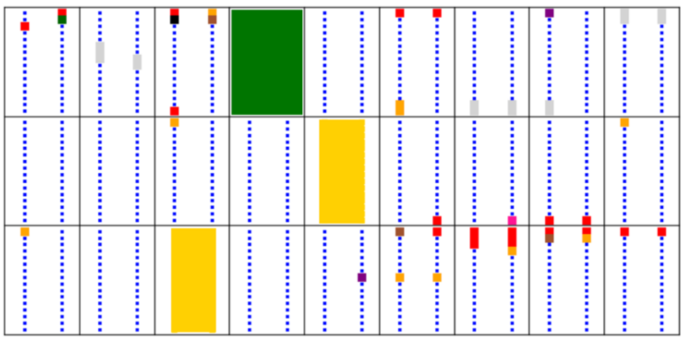
```
function CROSSOVER(parent1, parent2)
    child is empty
    for all buildings in parents do
        choose random location for building from parents
        child's building location ← random location
    if child's Street Score > best of parent's Street Score then
        return child
    else
        return child with probability according to annealing schedule
```

Algorithm 2 Mutation

```
if random number < mutation probability then
    for all non-residential buildings to mutate do
        swap residential location with non-residential location
        if mutated Street Score > unmutated Street Score then
            return mutated individual
        else
            return mutated individual with probability according
to annealing schedule
```

where ΔS is the difference between the child's Street Score and the best of the parents' Street Scores, and t is a temperature value that decreases according to an *exponential annealing schedule*, which makes use of the following decay factor: $t^{(k+1)} = \gamma t^{(k)}$, where γ is a user-defined parameter.

The algorithm for mutation is shown in Algorithm 2. Mutation is only performed on a child with probability given as a parameter in the overall genetic algorithm. When it is performed, a given number of randomly chosen non-residential buildings/amenities are swapped with residential buildings/amenities. Mutation is implemented in this way because the relative locations of residential units and non-residential amenities are the key drivers in the Street Score function. Simulated annealing can also be applied to mutation, using the original individual and the mutated individual as the candidates for acceptance. Crossover and mutation are used to create new generations of neighborhood designs. In the overall algorithm, we track the best performing individuals to determine the overall most walkable neighborhood design.



Amenity	Number	Weight Vector
Park	1	[6]
School	2	[1]
Office	15	[0]
Shopping	10	[0.5, 0.45, 0.4, 0.35, 0.3]
Coffee	2	[1.25, 0.75]
Entertainment	2	[1]
Bank	1	[1]
Bookstore	1	[1]
Grocery	1	[3]
Restaurant	20	[0.75, 0.45, 0.25, 0.25, 0.225, 0.225, 0..225, 0.225, 0.2, 0.2]
Residential	700	N/A

Figure 1. Potrero Hill existing layout with amenities and their weight vectors.

CASE STUDY: POTRERO HILL, SAN FRANCISCO, CALIFORNIA

To evaluate the proposed optimization methodology and test it on a real-world urban area, we apply it to an existing neighborhood-scale urban design in the Potrero Hill area of San Francisco, California. The grid we consider in this case study is 9 blocks by 3 blocks and roughly 320,000 m^2 in area. Figure 1 shows the abstracted study space, the amenities that are present in the design space, and weight vector associated with each amenity (as described above). These amenities are found and categorized through a manual audit of the space using Google Maps. The categorizations and weights in this study largely follow those in Rakha and Reinhart's previous work, which were chosen based on their analysis of both importance and the need for choice (as lengths of the weight vectors indicate how many of each amenity are considered in the calculation of the Street Scores). We increased the weight for the park amenity given its prominence in the existing design. Furthermore, consistent with Rakha and Reinhart, we did not consider offices to be an amenity.

To convert the physical layout to an abstract layout with appropriate dimensions and with lots for placing buildings and amenities, we used the osmnx package (Boeing 2017) for Python. We made certain assumptions in order to simplify the abstract representation of the neighborhood. Based on our assessment of the study area, we assume that, on average, there are 32 possible lots in each block. We also assume that the park and the schools each occupy one full block—where a full block is defined as the lots entirely contained by four intersections. Furthermore, we assume that all other building types each occupy one lot. It is important to note that this last assumption could be easily changed such that different building/amenity types take up different numbers of lots and/or partial lots to reflect multi-use development. For this study, however, we aimed to keep the abstract neighborhood representation as simple as possible in order to focus on optimization and interpretation.

For calculating the Street Score, we need to set values for the parameters d_1, d_2, and d_3

(discussed above). Given the geometry of the neighborhood, we set $d_1 = 50m$, $d_2 = 300m$, and $d_3 = 1600m$. It is important to note that these values are smaller than they are in Rakha and Reinhart's initial work. We choose smaller values because the physical distances in our case study are much smaller than those in the Rakha and Reinhart study, and therefore it would be relatively difficult to achieve a perfect Street Score. For the existing urban layout, we calculate the Street Score to be 31.8.

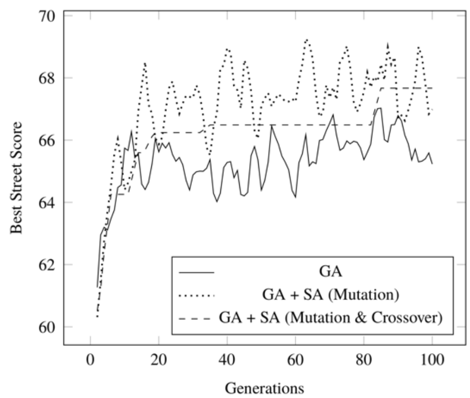


Figure 2. Comparison of implementations with varying degrees of simulated annealing.

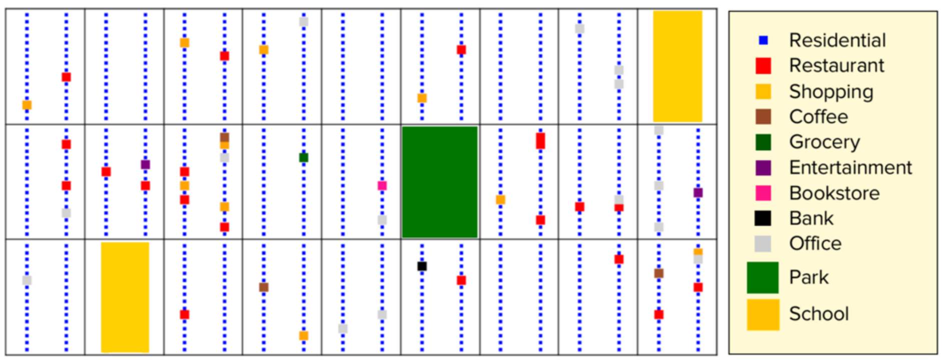


Figure 3. Optimized neighborhood layout.

In order to optimize this layout, we execute the genetic algorithm outlined above. The first step in this algorithm is to generate an initial population. We first generate a random population of neighborhood designs and assess their Street Scores. The random design routine first chooses random blocks for placing the park and schools, since these amenities take up full blocks. It then chooses random lots for placing all other amenities, and finally it fills up the remaining lots with residential units. After generating a population of 1,000 individuals, we calculate the Street Score for each. The resulting distribution has a mean of 52.1 and a standard deviation of 3.4.

We implement a version of truncation selection in the genetic algorithm to bias toward the better performing individuals. We first sort the individuals by decreasing Street Score (since we are maximizing). We then choose from the best performing individuals, but we also ensure that a randomly chosen set of the remaining population is incorporated in the selected group in order to protect against local minima. The mutation and crossover routines are implemented as discussed in the Methodology section. On a small population, we test three versions of the genetic algorithm, each with different levels of use of the concept of simulated annealing. In the baseline case, we do not include simulated annealing, but we test two other cases: one in which simulated annealing is incorporated into mutation, and another in which simulated annealing is incorporated into both mutation and crossover. When simulated annealing is incorporated, we use the exponential annealing schedule with $\gamma = 3/4$. The results from this test are shown in Figure 2 (where GA represents 'genetic algorithm' and SA represents 'simulated annealing'). As we can see, the genetic algorithm with simulated annealing incorporated into mutation performs the best.

Once deciding that simulated annealing should only be applied to the mutation step, we execute the genetic algorithm with the following parameters: 1,000 designs points in a single population, 100 generations, 5% probability of mutation, 500 parents, 4 children per parent pair, and initial annealing temperature of 10. The optimization convergence is shown in Figure 2. The best performing individual found after all generations are scored has a Street Score of 68.7. This is a little more than a two-fold increase from the existing layout (which was 31.8) and a roughly 32% increase over the random layouts. The final optimized layout is shown in Figure 3.

DISCUSSION AND CONCLUSIONS

Results of our analysis indicate that the average Street Score for the randomly generated layouts is significantly higher than the Street Score for the existing neighborhood layout. Perhaps even more surprising, the existing layout's score is about 6 standard deviations below the randomly generated layouts' mean score. It is important to note here that the existing layout score could start to approach the random layout score if the parameters d_1, d_2, and d_3 are increased. However, this finding still suggests a significant difference between the random and existing layouts. This is partially explained by the fact that the existing layout is reminiscent of the planning notion of Euclidian zoning. In the existing layout, the shop and restaurant uses are generally clustered in the lower right hand side of the grid. This clustering negatively impacts the Street Scores for any residential units located relatively far from the cluster (in our case, the houses on the upper left). Additionally, the grocery amenity in the existing layout is located all the way in the upper left corner of the grid, having a similar effect on the scores for residential units on the bottom right.

The optimization routine seems to converge around a maximum about 32% higher than the random layout. This optimized layout (as seen in Figure 3) has a much more dispersed layout of amenities. Importantly, the park and grocery amenities are located quite centrally in the grid. In addition, the schools are distributed on the left and the right, and the restaurants and shops tend to be distributed evenly across the entire grid. This makes intuitive sense: the more distributed amenities are, the higher chance that all residential units will be proximate to at least one of each amenity—questioning the benefits of Euclidean zoning for walkability in urban neighborhoods.

The main limitations in this work result from the various assumptions that were involved in setting some of the scoring parameters, including the distance parameters and the weighting parameters. Future work should consider which parameter values are most appropriate for different problem settings. However, while assumptions had to be made, the results still suggest important differences between the existing Euclidian-style layout and the more mixed layout suggested through optimization. Another limitation of this work is that the optimization and analysis were solely focused on an existing neighborhood layout in a real neighborhood. While the optimization could be easily applied to a neighborhood designed from scratch, a few things would need to be known before optimization: the street grid, the number of each building/amenity, and the possible lots for building/amenity placement. Future work should consider the problem of co-optimizing the street grid and possible locations along with building placement, as this might provide further insights into the optimal design for the urban fabric.

Finally, the findings from this analysis should not be the sole input when designing a new layout or assessing the performance of an existing layout. To be sure, there are metrics other than walkability that should be seriously considered when designing an urban space, such as proximity of amenities to transit stops, public health effects, or expected economic activity. The relative colocation of a polluting factory with a grocery store may increase walkability, but it could have dire consequences for public health. Similarly, it may improve walkability to distribute amenities across a given area, but for canonical economic reasons such as those first suggested in Hotelling's law (Hotelling 1929), it may be more economically profitable for two similar businesses to be located near each other. While the work presented in this paper cannot provide a sole rationale for designing a neighborhood one way versus another, it can provide helpful input for urban designers, engineers, and city governments in considering new layouts or evaluating existing ones.

REFERENCES

Adler, D. (1993). "Genetic algorithms and simulated annealing: A marriage proposal." *IEEE International Conference on Neural Networks - Conference Proceedings*, 1104–1109.

Best, R. E. (2016). "Modeling, Optimization, and Decision Support for Integrated Urban and Infrastructure Planning." Stanford University.

Boeing, G. (2017). "OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks." *Computers, Environment and Urban Systems*, Pergamon, 65, 126–139.

Carr, L. J., Dunsiger, S. I., and Marcus, B. H. (2010). "Walk score™ as a global estimate of neighborhood walkability." *American journal of preventive medicine*, Elsevier, 39(5), 460–463.

Hotelling, H. (1929). "Stability in Competition." *The Economic Journal*, 39(153), 41.

Jacobs, J. (1961). *The Death and Life of Great American Cities*. *New York*.

Kochenderfer, M. J., and Wheeler, T. A. (2018). *Algorithms for optimization*. MIT Press.

Porta, S., and Renne, J. L. (2005). "Linking urban design to sustainability: formal indicators of social urban sustainability field research in Perth, Western Australia." *Urban Design International*, 10(1), 51–64.

Rakha, T., and Reinhart, C. F. (2012). "Generative Urban Modeling: A design work flow for walkability-optimized cities." *SimBuild2012 5th National Conference of IBPSA-USA*, 1–8.

Razavialavi, S., and Abourizk, S. M. (2017). "Site Layout and Construction Plan Optimization Using an Integrated Genetic Algorithm Simulation Framework." *Journal of Computing in Civil Engineering*, 31, 04017011 (1-10).

Southworth, M. (2005). "Designing the walkable city." *Journal of urban planning and development*, American Society of Civil Engineers, 131(4), 246–257.

Wickersham, J. (2000). "Jane Jacob's Critique of Zoning: From Euclid to Portland and Beyond." *BC Envtl. Aff. L. Rev.*, HeinOnline, 28, 547.