NTNU
Norwegian University of
Science and Technology

Early-stage real estate development using nonlinear optimization

Nikolai Mathisen Bratsberg
Anne-Marie Mellbye

Industrial Economics and Technology Management
Submission date: June 2018
Supervisor: Kjetil Fagerholt, IØT
Co-supervisor: Lars Magnus Hvattum, IØT

Norwegian University of Science and Technology
Department of Industrial Economics and Technology Management

Preface

This thesis concludes our Master of Science degree in Applied Economics and Operations Research at the Department of Industrial Economics and Technology Management at the Norwegian University of Science and Technology. The course code is TIØ4905. The thesis is a continuation of the work done in our specialization project during the fall 2017.

We would like to thank our supervisors Professor Kjetil Fagerholt and Professor Lars Magnus Hvattum for their interest in supervising a thesis within an unfamiliar topic and their enthusiasm throughout the process. Their advice, guidance and feedback have been truly valuable. We would also like to express our gratitude to Håkon Kongelf for fruitful discussions and feedback. Lastly, we wish to thank Associate Professor Markus Grasmair for his eagerness to provide insights into the application of nonlinear optimization methods. The thesis is written in collaboration with an undisclosed company in the real estate industry.

Nikolai M. Bratsberg and Anne-Marie Mellbye

Trondheim, June 2018

Abstract

The process of planning the development of an urban site for housing engages many stakeholders and significant capital over an extended period of time. Without an accurate way of estimating the value of a site, real estate developers may initiate the process based on the wrong premises, increasing the risk for the company and its investors. Architects manually design site layout proposals, often uncertain whether they comply with governmental regulations, and of how much of the potential saleable area is lost in the trial-and-error process. Planning authorities cannot reliably determine whether one layout design is better than another. The combination of quantitatively complex regulations and objectives merits the study of applying optimization techniques in the planning process.

In this thesis, a comprehensive literature review reveals that little research has been done on the subject and that the solutions and results of related problems are not directly transferable. For this reason, a high-level model is developed for the problem of creating site layout designs, referred to as the *Site Development Problem* (SDP) in this thesis. The thesis is divided into two parts, where Part I develops a model for the SDP in two dimensions, while Part II extends this model into three dimensions. The model in Part I constitutes a single-objective, non-convex, nonlinear optimization problem, in which the objective is to maximize the total ground floor area of a fixed number of buildings within a site, subject to certain spatial constraints. The constraints are formulated using a distance that measures the feasibility or infeasibility of a solution. In Part II, the model incorporates heights of the buildings and the model becomes a single-objective, non-convex, mixed-integer nonlinear optimization problem, in which the objective is to maximize the aggregated volume of the buildings. The constraints are the same as in Part I, extended to include restrictions that involve the heights of the buildings.

Employing the method of Sequential Quadratic Programming (SQP) and MATLAB, the formulation is tested on six real sites in the Trondheim area in Part I. With the main purpose of studying the feasibility of using optimization to produce site layout designs, the placement of up to eight buildings is tested and discussed. Although there are no comparable results, visual evaluation of the output shows that the proposed formulation and solution method appear to produce a high degree of area utilization. The highest utilization obtained is 61.8 %. Four alternative definitions of the building variable set are proposed, but none obtain significantly better results than the original formulation. As the performance of SQP is highly dependent on the quality of the initial solutions provided to it, three strategies for constructing initial solutions are developed, one of which produces considerably better results than a random approach. In addition, a comparison is made between buildings restricted to a rectangular shape and the buildings with an angle used in the model formulation. The results show that allowing an angle along the length of the buildings considerably increases the utilization of the site.

In Part II, a heuristic is developed to solve the non-convex, mixed-integer nonlinear problem. The heuristic is compared to a naive approach in which the heights are optimized after the placement of the footprint areas of the buildings. The two solution approaches are tested on three real sites in the Trondheim area. The results show a gain in volume by including the heights of the buildings in the optimization process (i.e. using the heuristic) for the two smallest sites. For the largest site, a gain in volume is seen using the heuristic if there are considerable differences in the maximum allowable heights in the different zones of the site.

This thesis develops an optimization model in both two and three dimensions for the site layout design process on an urban site regulated for housing. The results show that there is significant potential in using optimization techniques to support the site layout design process. The solutions have a high utilization of the sites, and can guarantee that the spatial regulations are fulfilled. In addition, the proposed models and solution method extend existing packing research both in the type of shapes involved and in the included variables.

Sammendrag

Denne masteroppgaven omhandler planleggingsprosessen for utviklingen av en urban tomt regulert for boligutbygging. En slik prosess er kapitalkrevende og involverer mange ulike interessenter over en lang tidsperiode. Uten en nøyaktig metode for å estimere verdien av en tomt kan det være fare for at boligutbyggere går inn i prosessen på feil grunnlag, noe som øker risikoen for boligutbyggingsselskapet og deres investorer. Arkitekter designer utformingen av tomten manuelt, ofte uten garanti om at utformingen overholder gjeldende reguleringer for tomten. I tillegg kan en uviss mengde salgbart areal gå tapt ved bruk av denne fremgangsmåten basert på prøving og feiling. Dette kan også gå utover planleggingsmyndighetene, som ikke har en sikker metode for å verifisere om utformingen av en tomt er bedre enn en annen, og om de overholder reguleringsbestemmelsene. Kombinasjonen av kvantitative, komplekse reguleringer og ønskede egenskaper ved boligbygg legger grunnlaget for å bruke optimeringsteknikker i planleggingsprosessen av en tomt.

Et omfattende litteraturstudie viser at lite forskning har blitt gjort på området, og at løsningsmetoder og resultater for lignende problemer ikke er direkte overførbare. På grunn av dette introduseres en versjon av tomteutviklingsproblemet sammen med en overordnet modell i denne masteroppgaven. Oppgaven er delt inn i to deler, Del I og Del II. Del I utvikler en modell for problemet i to dimensjoner, mens Del II utvider problemet til tre dimensjoner. Modellen i Del I utgjør et ikke-konvekst, ikke-lineært optimeringsproblem med kun ett objektiv. Objektivet er å maksimere totalt grunnflateareal for et bestemt antall bygninger på en tomt, med visse restriksjoner knyttet til avstanden mellom bygningene. Restriksjonene er formulert ved å bruke et avstandsmål for gyldigheten til en løsning. I Del II inkluderes høydene på bygningene i modellen, slik at modellen utgjør et ikke-konvekst, ikke-lineært problem med både kontinuerlige variable og heltallsvariable. Problemet har fortsatt kun ett objektiv, nå utvidet til å maksimere det totale volumet til bygningene. Restriksjonene er de samme som i Del I, men utvidet for å inkludere betingelser som gjelder høydene på byggene.

Formuleringen av problemet er testet på seks ekte tomter i Trondheimsområdet ved bruk av sekvensiell kvadratisk programmering (SQP) og MATLAB i Del I. Hovedmålet med testingen er å undersøke gjennomførbarheten av å bruke optimering til å utvikle forslag til utforming av tomter. Plasseringen av opptil åtte bygninger er testet og diskutert. Siden det ikke finnes noe sammenligningsgrunnlag må resultatene vurderes visuelt. Fra dette viser resultatene at formuleringen og løsningsmetoden oppnår en høy grad av tomteutnyttelse. Den høyeste graden av utnyttelse oppnådd er på 61.8 %. Fire alternative variabelsett er foreslått, men ingen oppnår betraktelig bedre resultater enn det opprinnelige variabelsettet. Resultatene produsert av SQP er i høy grad avhenging av kvaliteten på startløsningene gitt inn til løsningsmetoden. Derfor er tre ulike strategier for å lage startløsninger utviklet, der en av dem produserer vesentlig bedre resultater enn en tilfeldig fremgangsmåte. I tillegg gjøres det en sammenligning mellom rektangulære og vinklede

bygg. Resultatene viser at det å tillate en vinkel langs lengden av bygningen øker utnyttelsen av tomten atskillig.

I Del II utvikles en heuristikk for å løse det ikke-konvekse, ikke-lineære problemet med kontinuerlige og heltallsvariable. Heuristikken sammenlignes med en naiv løsningsmetode der høydene blir optimert etter at plasseringen av fotavtrykket til bygningene er satt. De to løsningsmetodene sammenlignes og testes på tre ulike tomter i Trondheimsområdet. Resultatene viser økt totalt volum ved å inkludere høydene i optimeringsprosessen (det vil si, ved bruk av heuristikken) for de to minste tomtene. For den største tomten viser resultatene at heuristikken er best om det er store forskjeller på den maksimale høyden i de ulike høydesonene på tomten.

Denne masteroppgaven utvikler en optimeringsmodell i både to og tre dimensjoner for utformingen av en urban tomt regulert for boliger. Resultatene viser at det finnes stort potensial i å bruke optimeringsteknikker som støtte i denne prosessen. Løsningene har høy utnyttelsesgrad, og de kan garantere at reguleringer som innlemmes i modellen er overholdt. I tillegg utvider de foreslåtte modellene og løsningsmetoden eksisterende litteratur innen pakke- og kutteproblemer både med hensyn på hvilke former som inkluderes og hvilke variabler modellene bruker.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

More than one half of the world's population live in urban areas, and virtually all countries in the world are becoming increasingly urbanized (United Nations, 2015). This creates a need for new housing to be built at ever increasing levels of density — raising the risk of lower living quality and substandard housing. In an attempt to counter this, authorities strive to create more detailed regulations for urban areas, which significantly complicates the already complex process of early stage site development (Ratcliffe et al., 2009). In order to illustrate this, an introduction to early stage site development is given in the following.

Real estate developers continually screen sites to find potential profitable projects. If the developer decides to buy a property after an initial estimate of the project's potential value, a selection of architecture firms are often engaged to conduct more comprehensive *feasibility studies* of the project. A feasibility study includes designs of the proposed layouts and an examination of the legal implications (Lundevall, 2015). In this process, zoning rules produced by local municipalities are important to consider in order to correctly place the buildings and with that, estimate the site's potential value. The zoning rules for an area dictate its purpose, e.g. for housing, manufacturing or commercial use, and limitations on the placement and dimensions of buildings (Pressman, 2012). For sites that are regulated for housing, examples of zoning rules may be (Trondheim Kommune, 2015)

- *Total floor area within the site should not exceed X square meters*
- *The total height of the buildings should not exceed Y meters above leveled ground*
- *Orienting oneself in the housing area should be easy*
- *Roofs and facades should have color Z*

As seen in these examples, the rules may be of qualitative or quantitative nature. While some zoning regulations for residential areas exist to preserve a particular characteristic of an urban space, the majority are intended to sustain or improve safety, mobility, and general quality of living for the residents, by outdoor recreation areas, sufficient daylight and minimal noise pollution, to name a few (Pressman, 2012). Usually, the architecture firms create a selection of proposals of how to develop the site. An example of such a proposal is shown in Figure 1.1 by the architecture firm Alt.arkitektur. The developer then selects the most promising proposals as bases in the subsequent planning phase (Lundevall, 2015). Thus, in simple terms, the architect needs to design site layouts that meet or exceed the client's goals, while complying with regulations and laws that constrain the layout geometrically or functionally. The client's goal is often to maximize the saleable square meters in order to maximize the profit. With increasingly complex and detailed zoning regulations, the task of balancing this goal while acting in accordance with the zoning regulations can grow into a highly complicated design process.

Figure 1.1: An example of a feasibility study by Alt.arkitektur on a site in Kongsberg

To illustrate the sophisticated task of building layout design, an example of regulations and how they impact the design process is considered. In order to prevent the housing from becoming too dense, thus reducing outdoor space, a minimum amount of outdoor recreation area is required per square meter of saleable floor area. Moreover, it is usually required that the outdoor area is sufficiently sunlit, verified by measuring whether it is directly illuminated by the sun more than a certain amount of time on a specified date (e.g. spring equinox) (PBE, 2012). So, when an architect increases the size of a building in a feasibility study, more outdoor recreation area is required, but the shadow from the building is now larger, decreasing the previously approved outdoor area. The task of finding a good solution in terms of saleable floor area to meet the real estate developer's goal, while guaranteeing compliance with zoning rules, is clearly very difficult.

The complication of satisfying the zoning regulations escalates when one wishes to maximize or minimize some other quality in addition to saleable area. For instance, noise pollution is a purely negative quality, and a developer aims to minimize noise (e.g. from roads or railways) on building facades. Conversely, visible evening sun is in some countries a highly valuable quality and as such something that many developers want to maximize to increase the price of the housing units. Indeed, if the noise pollution and evening sun enter the site from the same direction, then the optimal orientation of the building is not evident. Creating site layout designs is an inherently multi-objective activity that demands the balancing of conflicting goals while adhering to complex, governmental regulations (Pressman, 2012).

The systems and processes involved in creating site layout designs are heavily influenced by tradition, generally relying on manually creating and defining structures and their properties according to the architect's knowledge and intuition. Software has been developed to support this manual flow, allowing the architects to create structures digitally (Pressman, 2012). However, with the main goal of supporting the existing work flow, the available plugins for the software provide mostly analytic services such as calculating the total saleable area, or extracting level of shading or facade sun at a particular time. The architect must still manually evaluate, at each step, how to improve a tentative solution in terms of the desired qualities without violating the regulations and constraints.

The partly manual, complicated method employed today makes early stage site development a resource-intensive task, and it is hard to determine whether the final results are feasible with respect to the zoning regulations. This also affects the planning authorities, which in turn are posed with a similar problem — they cannot reliably determine whether one plan for development has superior qualities relative to others. Thus, to achieve quality, the solution is often to reduce the housing density. While this indeed preserves quality, the effect on profitability can be dramatic. In addition, high density is generally an objective also for the planning authorities due to the rapid and increasing urbanization. If the problem is modeled as an optimization problem, it is possible to generate different solutions that are guaranteed to meet quantitative zoning regulations, while maximizing the density and other desirable qualities.

This thesis aims to introduce an optimization model that can incorporate certain zoning regulations while maximizing the saleable area. The problem, called the Site Development Problem (SDP) throughout the thesis, was first introduced as a part of Bratsberg and Mellbye (2017). In Bratsberg and Mellbye (2017), the problem is defined in two dimensions, the buildings are restricted to a rectangular shape, and the site is assumed to be convex. The model incorporates zoning regulations related to the required space between buildings, and the building ground floor area is used as a proxy for the maximization of saleable area. In this thesis, the Site Development Problem is extended in two parts. In Part I, the problem is still defined in two dimensions, but the buildings can be shaped as irregular hexagons, that is, polygons with six edges. In addition, the site may be non-convex. As in Bratsberg and Mellbye (2017), zoning regulations related to the space between buildings are considered, and the building ground floor area is maximized. In Part II, the optimization model introduced in Part I is extended into three dimensions. Zoning regulations related to the height of the buildings are incorporated. Specifically, the site is divided into different *height zones*, in which a maximum allowable height is dictated within each zone. In addition, the required space between buildings is, in reality, linked to the height of the buildings. This interrelationship is included in the extended model, leading to more complex constraints in three dimensions. The objective of the three-dimensional optimization model is to maximize the volume of the buildings and thus the saleable area. The buildings are still defined as polygons with six edges, while the site can be either convex or non-convex.

In Chapter 2 a literature review is conducted and relevant literature is examined. Next, Part I starts with Chapter 3 which introduces the problem of Part I. In Chapter 4, a mathematical model based on the problem description in Chapter 3 is presented. Further, the solution method used to solve the mathematical model in Part I is described in Chapter 5. Based on the mathematical model and the solution method, a computational study is conducted in Chapter 6. The structure of Part II is similar to that of Part I. First, the extended problem is introduced in Chapter 7. In Chapter 8, a mathematical model based on Chapter 7 is developed. The solution method used to solve the model is described in Chapter 9. The computational study for Part II is presented in Chapter 10. Lastly, Chapter 11 includes concluding remarks for both Part I and Part II, while Chapter 12 outlines possible future research areas and extensions to the problems considered in this thesis.

Chapter 2

Literature review

In this chapter, a wide and comprehensive literature search is conducted. Section 2.1 examines whether there exists any literature that applies optimization techniques to a problem similar to the Site Development Problem. In section 2.2, the *packing problem* is studied due to its similarities to the SDP. In short, the packing problem allocates a set of smaller objects to one or more larger objects to minimize the waste, or equivalently, to maximize the area or volume that the smaller objects cover. This is analogous to the SDP, which allocates buildings (smaller objects) to the site (the larger object) while maximizing the area or the volume of the buildings. Another problem considered in the literature review due to its apparent similarities to the SDP is the Facility Layout Problem discussed in Section 2.3. This problem type is comparable to the SDP because it concerns the allocation of facilities on a given area, like the SDP involves the allocation of buildings on a site. This literature review was first conducted as a part of Bratsberg and Mellbye (2017), and parts of it are revisited here. In addition, some extensions of the literature review are done in accordance with the extension of the SDP in this thesis compared to Bratsberg and Mellbye (2017).

Before searching for articles, a set of relevant keywords is identified. These keywords are used to search for relevant literature using the web search engine "Scopus". A more detailed description of how the literature search is conducted for each specific topic is given in the corresponding sections.

2.1 Optimization techniques applied to problems similar to the Site Development Problem

2.1.1 Material collection

In order to systematically explore literature that discusses the use of optimization for problems similar to the SDP, two groups of keywords are developed. The first group consists of words from the field of operations research, that is: "optimization", "minimize", "maximize" and "mathematical model". The second group consists of words related to the SDP. These are: "building site", "building placement", "building size", "building form", "building geometry", "apartment", "site development", "building footprint", "land allocation", "architectural design", "urban site", "building layout" and "spatial planning". All pairs of words from the first and the second group are combined in the search.

All the words in the search string must appear in either the title, abstract or keywords of the article for it to be considered further in the literature review. The initial search generated 981 articles. From these, only peer-reviewed journal articles written in English were considered. This left 508 articles. By removing articles within irrelevant subject areas such as medicine, chemistry and psychology, and articles with keywords related to these subject areas, the number of papers decreased to 174. From these, the title and in some cases the abstract of the papers were examined to determine the relevance of the articles. Papers that study features not relevant for the SDP, such as structural construction, durability of buildings and resource allocation, and papers that do not employ optimization techniques were filtered out. This left 62 papers. The abstract and in some cases the discussion part of the remaining papers were read to determine their relevance. The papers that only consider the optimization of small features of a building, for instance pipes or windows, were filtered out. In addition, papers that predominantly focus on simulation, and mostly treat the optimization techniques as "black box" methods, were excluded in this filtering step. Finally, eight papers were left for full review. These eight papers also include an article by He et al. (2015) that did not appear in the search results, but is included in the literature review because it offers valuable insights. The material collection process is illustrated in Figure 2.1.

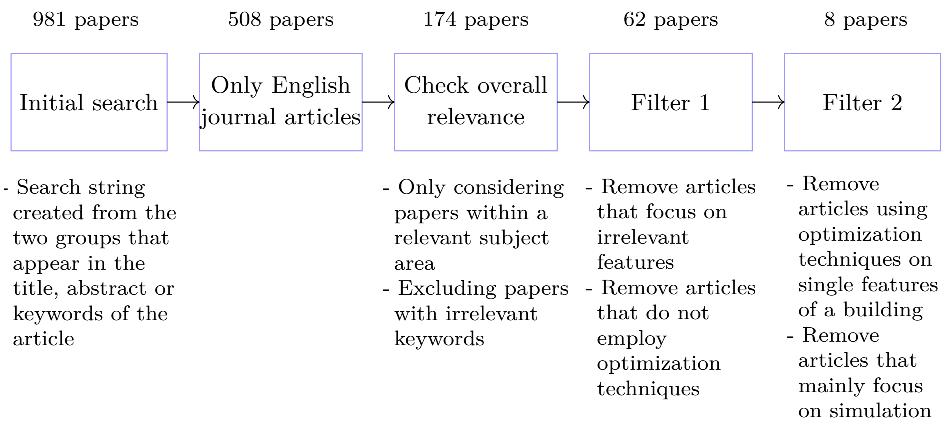


Figure 2.1: Material collection process for literature on similar problems to the SDP

2.1.2 Full review

The articles left for full review are systematized in Table 2.1. The SDP is included in the last row for comparison with the existing literature. The first column refers to the articles in Table 2.2. The second column states the objective of the problem considered in the article. The third column specifies the domain of the problem considered, whether it is one building, the floor layout of a building, or a single site, like it is for the SDP. Here, ”urban area” refers to a site that is not regulated for housing, or an area larger than a site, typically a problem concerning urban planning. Lastly, the fourth column states the solution method presented in the articles. Tian et al. (2015) review existing optimization tools for achieving energy efficient buildings, and do not study a particular problem. This article is therefore included with blank cells in Table 2.1, column three and four.

Table 2.1: The articles included in the full review systemized and compared to the SDP

Article	Objective	Domain	Solution method
1	Heat gain and loss	A building	Genetic algorithm
2	Optimal layout	Floor layout	FSQP
3	Sunlight	A site	Genetic Algoritm
4	View	Urban area	Evolutionary alg.
5	Saleable area	A site	Simulation + heuristic
6	Layout, energy and daylight	A building	Pareto-based search
7	Dispersion	Urban area	Genetic algorithm
8	Survey - optimization tools	-	-
SDP	Saleable area	A site	-

Table 2.2: Articles included in the full review

1	Jin and Jeong (2014)
2	Michalek et al. (2002)
3	Yi and Kim (2015)
4	Reinhard (2015)
5	He et al. (2015)
6	Dino and Üçoluk (2017)
7	Tong (2016)
8	Tian et al. (2015)

There is an on-going trend of incorporating optimization techniques into design processes to achieve energy efficient buildings (Tian et al., 2015). Two articles that consider the incorporation of optimization techniques to achieve energy efficient buildings are Jin and Jeong (2014) and Dino and Üçoluk (2017). Jin and Jeong (2014) recognize the inherent multi-objective nature of building design, and create a multi-objective model with the conflicting objectives of minimizing heat loss while maximizing heat gain and solar heat. A *genetic algorithm* is used to find an optimal building form with respect to achieving efficient energy performance. A genetic algorithm is inspired by the process of natural selection and uses this process to find good solutions. Dino and Üçoluk (2017) consider a multi-objective problem as well. They develop a multi-objective optimization tool for building design, and perform a *Pareto-based search* in the design space. A Pareto-based search looks for a Pareto set of the problem, that is, solutions where none of the objectives can be improved without degrading another (Dino and Üçoluk, 2017). The layout, orientation and building envelope are optimized for energy and daylight performance. Energy efficient buildings, and other aspects considered in some of the articles such as daylight and sunlight, are relevant to consider in an extended, multi-objective version of the SDP. However, both Dino and Üçoluk (2017) and Jin and Jeong (2014) perform the optimization on one single building, and do not consider the relative placement of buildings on a site, like the SDP does. Thus, the objective and also the domain of these articles are different from the SDP, as seen in Table 2.1.

There are, however, some authors that work with more than a single building and thus define the domain to be more similar to that of the SDP. Tong (2016) points to the fact that the relative distribution of buildings has a significant impact on our perception of the surroundings. Too much clustering or dispersion may both have negative effects. Thus, the author sets out to create an index that describes the distribution of buildings. Tong (2016) also incorporates governmental regulations in the constraints of the problem, similar to the SDP. However, both the regulations and the application of the index is primarily concerned with green spaces (classified as urban area in Table 2.1), and not a site regulated for housing like in the SDP. The regulations of green spaces compared to sites regulated for housing is quite different as the purpose of the two areas of land is not the same. Reinhard (2015) also considers an urban area, and uses an evolutionary multi-objective optimization approach to generate and evaluate a problem where the variables are the same as for the SDP, that is, the dimensions and placement of the buildings. However, the utilization is held constant while the view is maximized.

Natural, direct sunlight is often considered an important quality in urban housing and is part of common zoning regulations. The SDP includes zoning regulations concerned with daylight by requiring a certain amount of space between each building. Yi and Kim (2015) propose a genetic algorithm that minimizes the number of apartments that fail to meet the regulation requirements of minimum hours of sunlight in Korea. Similar to the SDP, Yi and Kim (2015) define the problem on a given site to be developed for housing, and consider the relative dispersion of the buildings. In addition, regulations related to the space between the buildings are incorporated in the model. However, the objectives of the SDP and the problem discussed in Yi and Kim (2015) are different, as the SDP only maximizes the ground floor area or the volume of the buildings, while Yi and Kim (2015) only consider sunlight.

Michalek et al. (2002) discuss another problem type that can be partly compared to the SDP. They develop an optimization model for the architectural floorplan layout design problem. The problem focuses on placing rooms inside a building. Because the components to be placed are rooms, they are subject to various constraints not applicable to the SDP, such as the need for windows, access halls, and doors. Nevertheless, the constraints that bound the area of the rooms the constraints that place the rooms inside the building, and those that ensure no overlap between rooms are highly relevant to the SDP. The mathematical model developed is a general nonlinear programming problem, and the authors propose to use Feasible Sequential Quadratic Programming to solve it. Sequential Quadratic Programming (SQP) is an iterative method of finding a ”search direction” and a ”step size” by solving quadratic subproblems of the nonlinear programming problem. Feasible Sequential Quadratic Programming is equivalent to SQP, except that once a feasible design is found, search directions are altered to maintain feasibility at every iteration (Michalek et al., 2002).

As illustrated in Table 2.1, there is only one article that defines the same objective and domain as the SDP. This is the article by He et al. (2015). They address the spatial aspects defining the rules for site development, exactly like the SDP that incorporates spatial zoning regulations. He et al. (2015) work with the problem in three dimensions, and consider individual geometry, spatial relationship with other buildings and site boundary constraints. This problem is therefore highly relevant and can provide useful insights. They solve it as a simulation optimization problem using a heuristic.

To summarize, little research has been done on problems similar to the SDP with optimization techniques applied. The literature search conducted left only eight articles for the full review, where only He et al. (2015) consider a problem with the same objective and domain. Still, the other articles are interesting to study for the single-objective SDP as it is currently defined, and also for an extended multi-objective problem.

2.2 Packing problems

The basic structure of packing problems can be described as follows: Two sets of elements are given, a set of large objects and a set of small objects. All or some of the smaller objects are grouped into one or more subsets. The subsets are assigned to the larger objects while maintaining the *geometrical conditions*, that is, all small objects must lie entirely within the larger ones and cannot overlap (Wäscher, Haußner, and Schumann, 2007). From this basic description of packing problems, it is clear that the SDP has some resemblance to this type of problems. In the SDP, the set of large objects consists only of one object, namely the site. The set of smaller objects are the buildings to be placed on the site. In addition, the geometrical conditions for the SDP and the packing problem are the same.

2.2.1 Material collection

As demonstrated in the typology by Dyckhoff (1990) and later Wäscher et al. (2007), there exists a large variation of packing problems. The keywords used in the literature search on packing problems are developed based on the typologies by Dyckhoff (1990) and Wäscher et al. (2007), using words with most similarities to the SDP. The set of keywords is divided into two groups. The first group consists of the words: ”packing”, ”cutting”, ”allocation”, ”placement” and ”nesting”. This group contains words that are often used synonymously with ”packing” in the literature on packing problems. The exception is the word ”cutting”. It is included in this group due to the duality between cutting and packing problems and their common basic structure. For example, packing smaller rectangles into a larger one may also be seen as *cutting* smaller rectangles from a larger one (Dyckhoff, 1990). The second group consists of words that specify the kind of packing problems that are relevant to compare to the SDP. These words are: ”irregular”, ”irregularly”, ”polygons”, ”rotations”, ”arbitrarily”, ”non-convex” and ”utilization”.

The search is conducted by combining all possible pairs of one keyword from the first group with one from the second group. To appear in the search result, the title of the paper has to contain the word included from the first group, while the word included from the second group has to appear in the title, abstract or keywords of the article. The initial search generated 11 818 papers. By limiting the search to include only journal articles written in English, 5 531 articles remained. To further narrow down the search, only the articles within the research fields of decision sciences, business management, computer sciences, engineering, and mathematics were considered. In addition, articles with irrelevant keywords were excluded. This left 372 articles for further consideration. In addition to the removal of duplicates, two criteria were developed to further narrow down the search. First, the paper has to address packing problems. Second, papers that cover packing of circular items are removed. When these criteria were applied, 50 papers remained. From these, papers that focus on solving problems with constraints irrelevant for the SDP, such as constraints related to the texture and material of objects to be used in cutting, or the selection of larger objects, were filtered out. This is because the SDP has only one large object (the site), and the selection process of the larger objects is therefore not relevant to consider. Finally, papers that discretize the problem were filtered out because the models in these articles are often Mixed-Integer Programming models (MIP-models). This gives little insight into how to model the continuous constraints in the SDP. Also, there is too little knowledge of typical solutions to the SDP to know the error in the objective function a discretization would yield, or to know the required resolution. In the end, 29 papers were kept for the full review. Figure 2.2 illustrates the material collection process.

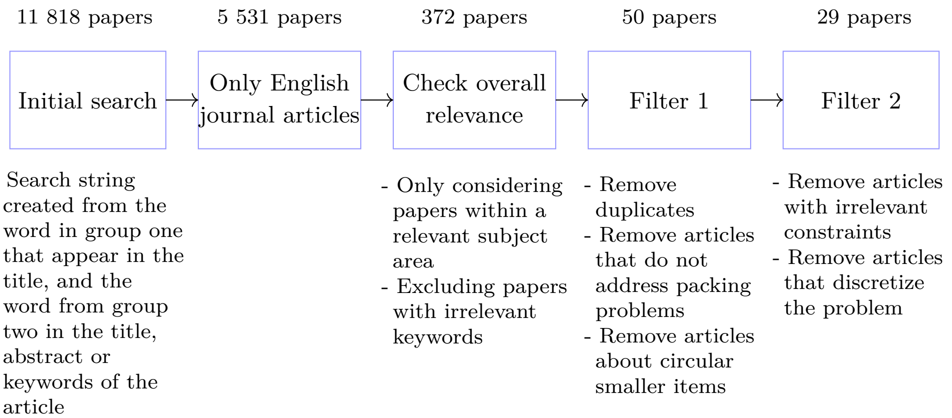


Figure 2.2: Material collection process for packing problems

2.2.2 Full review

The articles left for full review are systematized in Table 2.3, with one exception. Bennell et al. (2010) is not included in the table as they do not present a specific packing problem. Nevertheless, the article provides valuable insight into many of the concepts used in packing problems and is therefore included in the full review. Table 2.3 compares relevant properties of the SDP to the various packing problems the papers discuss. The numbers in the first column of the table correspond to the articles in Table 2.4.

In both the column "shape of large item" and "shape of small items", the description *irregular* is used. An irregular shape refers to a general, non-rectangular, non-circular shape that can be either convex or non-convex. Thus, the polygonal shape of the buildings and the convex or non-convex shape of the site in the SDP are classified as irregular. Further, the word *strip* is used to describe the shape of some of the large items. This refers to a rectangle with fixed width and variable length. The last two columns specify whether the authors introduce a new mathematical model or a new solution method. The abbreviations NLP, MIP, and MILP stand for Nonlinear Programming, Mixed-Integer Programming, and Mixed-Integer Linear Programming, respectively. The abbreviation SQP in the solution method column refers to Sequential Quadratic Programming. The terms *heuristic* and *metaheuristic* are used to describe certain solution methods. The strategy of a heuristic method is to find a set of rules that work well for the specific application design. These methods do not guarantee to be optimal but are accepted if they yield "good enough" results. The latter is also true for metaheuristics. However, metaheuristics are more general and less problem-dependent (Tay et al., 2002). The last four papers in Table 2.3 consider three-dimensional packing problems. The SDP is included in the last row for comparison with the literature in the full review.

Table 2.3: Comparison between the SDP and the articles left for full review

	Shape of large item	Shape of small items	Fixed dim. small items	Small item rotation	Model	Solution method
1	Rectangle	Irregular	Yes	Continuous	-	Metaheuristic
2	Rectangle	Irregular	Yes	Discrete	-	Meatheuristic
3	Strip	Irregular	Yes	No	-	Metaheuristic
4	Plane	Irregular	Yes	No	-	Algorithm
5	Strip	Irregular	Yes	No	-	Metaheuristic
6	Irregular	Irregular	Yes	Continuous	-	Metaheruristic
7	Convex	Rectangle	Yes	Discrete	NLP	-
8	Strip	Irregular	Yes	Discrete		Metaheuristic
9	Strip	Irregular	Yes	Discrete		Local search
10	Rectangle	Irregular	Yes	Continuous	-	SQP
11	Irregular	Irregular	Yes	Continuous		Metaheuristic
12	Convex	Rectangle	Yes	Discrete	-	Heuristic
13	Strip	Rectangle	Yes	Discrete	-	Heuristic
14	Strip	Irregular	Yes	Discrete	-	Metaheuristic
15	Strip	Irregular	Yes	Continuous		SQP
16	Strip	Rectangle	Yes	Discrete	-	Heuristic
17	Irregular	Irregular	Yes	Continuous	-	Cutting alg.
18	Rectangle	Irregular	Yes	Discrete	-	Heuristic
19	Strip	Convex	Yes	No	MILP	-
20	Strip	Irregular	Yes	No	MIP	-
21	Strip	Irregular	Yes	Discrete	-	Metaheuristic
22	Strip	Irregular	Yes	Discrete	-	Metaheuristic
23	Strip	Rectangle	Yes	No	-	Heurisitc
24	Rectangle	Rectangle	Area	No	-	Iterative merge
25	3D strip	Cubes	Yes	Discrete	-	Metaheuristic
26	Irregular	Irregular	Yes	No	-	Heuristic
27	3D strip	Box	Yes	Discrete	-	Heuristic
28	3D strip	Irregular	Yes	Discrete	-	Heuristic
SDP	Irregular	Irregular	No	Continuous	-	-

Table 2.4: Article overview for Table 2.3

1	Theodoracatos and Grimsley (1995)
2	Jakobs (1996)
3	Bennell and Dowsland (1999)
4	Watson and Tobias (1999)
5	Bennell and Dowsland (2001)
6	Tay et al. (2002)
7	Birgin et al. (2006)
8	Gomes and Oliveira (2006)
9	Imamichi et al. (2009)
10	Yu et al. (2009)
11	Martins and Tsuzuki (2010)
12	Cassioli and Locatelli (2011)
13	Leung and Zhang (2011)
14	Sato et al. (2012)
15	Yu et al. (2012)
16	He et al. (2013)
17	Dalalah et al. (2014)
18	Song and Bennell (2014)
19	Santoro and Lemos (2015)
20	Cherri et al. (2016)
21	Sato et al. (2016)
22	Pinheiro et al. (2016)
23	Wei et al. (2017)
24	Ji et al. (2017)
25	Szykman and Cagan (1995)
26	Egeblad (2009)
27	Allen et al. (2011)
28	Liu et al. (2015)

The shapes of the large and the smaller items

The second column of Table 2.3 states the shape of the large object in each problem. For the SDP, the shape of the large object is the shape of the site, which can be either convex or non-convex. In Table 2.3, only Birgin et al. (2006) and Cassioli and Locatelli (2011) consider a problem where the shape of the large object is convex. Birgin et al. (2006) develop a nonlinear programming model for the packing of rectangles in an arbitrary convex region. Cassioli and Locatelli (2011) propose a solution method for the same problem considered in Birgin et al. (2006). The proposed method is a heuristic based on iterated local search. Both packing problems in Birgin et al. (2006) and Cassioli and Locatelli (2011) are similar to the SDP in that the large object is convex. However the site in the SDP may also be non-convex, and the buildings are not restricted to be shaped as rectangles like the small objects of Birgin et al. (2006) and Cassioli and Locatelli (2011). Thus, there are various simplifications and techniques used by Birgin et al. (2006) and Cassioli and Locatelli (2011) that are not applicable to the SDP.

There are, however, three articles (Tay et al., 2002; Martins and Tsuzuki, 2010; Dalalah et al., 2014) that propose different solution methods to solve a packing problem where both the large and the small objects are irregular, like in the SDP. Both Tay et al. (2002) and Martins and Tsuzuki (2010) propose a metaheuristic to solve the problem. Tay et al. (2002) argue that a genetic algorithm is a suitable tool for the irregular packing problem due to a large number of possible arrangements of the smaller objects. Martins and Tsuzuki (2010), on the other hand, use *simulated annealing* to solve the same problem. This algorithm is a simulation of the recrystallization of atoms in metal during its annealing (gradually and controlled cooling). The observation of this process led to the development of simulated annealing, an algorithm that can skip local minima by exploring a direction that can locally deteriorate the objective function. The property of being able to ”escape” a local minimum is the reason why Martins and Tsuzuki (2010) argue that it is a suitable method, since local minima are frequent in the packing problem with irregular items. On the contrary to the metaheuristics proposed by Tay et al. (2002) and Martins and Tsuzuki (2010), Dalalah et al. (2014) develop a new two-dimensional cutting algorithm. The SDP is similar to all of these problems in that the shape of both the large object and the smaller objects can be either convex or non-convex.

Rotation

Another feature the SDP has in common with the packing problems considered in Tay et al. (2002), Martins and Tsuzuki (2010) and Dalalah et al. (2014), is continuous rotation of the small items. Theodoracatos and Grimsley (1995), Yu et al. (2009) and Yu et al. (2012) also solve a packing problem with continuous rotation. Theodoracatos and Grimsley (1995) suggest a solution method using simulated annealing. Yu et al. (2009) and Yu et al. (2012) use Sequential Quadratic Programming (SQP) to solve a packing problem based on a nonlinear programming formulation. They compute an overlap index to detect overlap, which is positive if the shapes overlap and negative if they do not. This overlap index is helpful when using SQP. Yu et al. (2012) use SQP as the local search strategy and then employ another strategy (also based on SQP) to escape local minima. These problems are similar to SDP in that they allow continuous rotation, and can provide useful and important insights since continuous rotations significantly complicate the problems. For instance, Cherri et al. (2016) develop MIP models for the Strip Packing Problem, but do not allow rotation as it will make the proposed models nonlinear.

The Strip Packing Problem

A significant part of the articles in Table 2.3 considers the large object to have a fixed width and variable length, as the term *Strip* in the second column indicates.

These problems belong to a class of problem known as Strip Packing Problems (Wäscher et al., 2007). The goal is to minimize the length of the large object by optimal packing/cutting of the smaller objects. In a way, the SDP can be seen as the opposite problem to the Irregular Strip Packing Problem. The shapes and sizes of the objects to be packed in the Irregular Strip Packing Problem are predetermined. For the large object, the shape is known, but the length is to be minimized. On the contrary, in the SDP, the shape and size of the large object, the site, are predetermined, while the area of the smaller objects (the buildings) is maximized.

Gomes and Oliveira (2006) are two of many that study the Strip Packing Problem with irregular-shaped small objects. To detect overlap between the small objects, they rely on a concept known as the *nofit polygon*. The nofit polygon is obtained by fixing one polygon and then tracing the locus of a reference point on another polygon, as it traces around the edge of the fixed polygon (Bennell et al., 2010). This creates the nofit polygon. The overlap detection test can now be reduced to a simple point inclusion test - if the reference point is placed in the interior of the nofit polygon, the two shapes overlap. If the reference point is on the boundary of the nofit polygon, the two shapes touch, and if the reference point is in the exterior of the nofit polygon, then the pieces neither overlap nor touch. The nofit polygon is the most cited tool for dealing with the geometry of irregular shape cutting and packing problems (Bennell and Dowsland, 2001). This is also reflected in this literature review. Apart from Gomes and Oliveira (2006), other articles that consider the Irregular Strip Packing Problem and use the nofit polygon to detect overlap include Bennell and Dowsland (2001), Imamichi et al. (2009), Sato et al. (2012), Sato et al. (2016), Cherri et al. (2016) and Pinheiro et al. (2016).

Apart from the ”opposite” formulation of the objective functions, the SDP also differs from the Strip Packing Problems that utilize the nofit polygon in that they only allow discrete rotation or no rotation at all. The shape of the nofit polygon is different for each new orientation of the polygons, complicating the overlap detection when continuous rotation is introduced. Out of the problems with continuous rotation in Table 2.3, only Martins and Tsuzuki (2010) use the nofit polygon to detect overlap. However, in order to use the nofit polygon, Martins and Tsuzuki (2010) place the items one at a time in the container. As illustrated, the continuous rotation present in the SDP complicates the overlap detection as the nofit polygon cannot be used directly.

Fixed dimensions of small items

A fundamental characteristic distinguishes the SDP from the packing problems considered in the articles. This difference is illustrated in the fourth column in Table 2.3, which indicates whether or not the dimensions of the small items are fixed and predetermined. Table 2.3 shows that, except for one article, all the other problems have fixed dimensions of the small items. The exception is Ji et al. (2017). Ji et al. (2017) consider a packing problem known as the soft rectangle packing problem. This is a packing problem where the area of the small items is fixed and the aspect ratio (the ratio of rectangle height to rectangle width) can be adjusted in certain ranges. The solution method is an iterative merging algorithm that iteratively merges two rectangles with the least area and then places the merged rectangles onto the larger object, and determines the aspect ratio of the sub-rectangles recursively. This packing problem is the only one of the problems in Table 2.3 that also allows adjustments of the length and width of the small items, like the SDP. However, the area is still fixed. This is a fundamental difference, as the maximization of the area or the volume of the smaller items is the objective of the SDP.

Zoning regulations

Yet another feature that separates the SDP from the various packing problems described in the literature is the incorporation of zoning regulations related to the required space between the buildings. This creates a need for open zones around each of the smaller items. It could be possible to study the building with its open zones as one object. However this is a simplification in which the properties of the open zones are not considered. The open zones are allowed to overlap with each other and can exceed the boundary of the large object, i.e. the site.

Three-dimensional packing problems

Wäscher et al. (2007) report that one- and two-dimensional packing problems are considered significantly more than three dimensional packing problems, which constitute only 13% of the research papers published. This is reflected in the literature search, where only four of the articles left for full review examine three-dimensional packing. The problems defined in three-dimensions (Liu et al., 2015; Allen et al., 2011; Szykman and Cagan, 1995; Egeblad, 2009) have the same differences compared to the SDP as the two-dimensional problems. None consider open zones between buildings, all have fixed dimensions of the small items, and the shape of both the small and the large objects are different. When extending the SDP into three dimensions, the SDP and the packing problems deviate even more. Recall that the site in the three dimensional SDP is divided into different height zones which dictate the maximum allowable height of a building contained within a given zone. These constraints on the height of the smaller objects are not examined in the packing problems. The larger object is defined with a given height (Egeblad, 2009), or defined as a box-shaped container with fixed width and length, but unconstrained height (Allen et al., 2011; Liu et al., 2015). The latter problem type is the three-dimensional version of the Strip Packing Problem considered above, and the larger object is classified as *3D strip* in Table 2.3. Another characteristic that separates the two problem types is that the packing problem can stack multiple objects on top of each other. However, the 3D packing problem still shares some similarities with the SDP, and the solution methods and models developed are useful to study. In particular, Egeblad (2009) offers an interesting problem formulation. The author includes balance in the objective function, ensuring an evenly distributed placement of the smaller objects. Although the application considered by Egeblad (2009) is the loading of ships and trucks, where the importance of evenly distributed weight is crucial, the balance problem can also be related to the SDP in that evenly distributed buildings may allow higher density.

To summarize, the literature on cutting and packing problems can be helpful when studying the SDP as these problems also describe packing smaller objects onto a larger object to minimize waste of the larger object. Nonetheless, there are a number of aspects of the SDP that is not considered in the traditional packing problems due to the different application areas. For instance, the Strip Packing Problem arises in the furniture, shoe and garment industries, where the raw material is given in rolls. Thus, the objective becomes the minimization of the length of the larger object, that is, the minimization of the use of the raw material. Several of the articles mention the application of their work in these industries (e.g. Gomes and Oliveira, 2006; Imamichi et al., 2009; Chen et al., 2010). The SDP is a model for real estate development. Therefore, it involves other aspects that are irrelevant for traditional packing problems. These include the additional problem of optimizing the size of the smaller items and the consideration of the open zones around them.

2.3 The Facility Layout Problem

The Facility Layout Problem also exhibits similarities to the SDP. Typically, layout problems are related to the location of facilities (e.g. machines, departments) in a plant (Drira et al., 2007). To make sure the facilities do not overlap, the Facility Layout Problem needs no-overlap constraints similar to the SDP, where the buildings are not allowed to overlap. In addition, the space allocated to each facility in the Facility Layout Problem must take into account a buffer zone around the facility, for example to operate a machine. This can be compared to the zoning regulations related to the required space between the buildings in the SDP. However, based on the survey on the Facility Layout Problem by Drira et al. (2007), a decision is made not to consider this type of problems further in the literature review. The problem deviates from the SDP in the both the formulation of the objective and the constraints. In most of the articles on the Facility Layout Problem, the main objective is to minimize a function related to the transportation of parts between the facilities (Drira et al., 2007), an objective not applicable to the SDP. Although some constraints are useful, these constraints will not provide any further insight beyond that of packing problems. The no-overlap constraints are also considered in the packing problems, while the buffer zones in the Facility Layout Problem are included either in the area of the facility, eliminating the complications of allowing overlap between the open zones, or by considering the distances between the facilities as fixed (Heragu and Kusiak, 1988; Braglia, 1996). Thus, packing problems are studied instead of the Facility Layout Problem because these problems include the same constraints relevant for the SDP as the Facility Layout Problem includes, but these problems also have a similar objective to the SDP, namely to maximize the total area of the items to be packed.

Part I

The 2D problem: Maximizing the total saleable floor area of angled buildings on a non-convex or convex site

Chapter 3

Problem description I: The 2D problem

The problem addressed in this part of the thesis is the two dimensional Site Development Problem. Consider a real estate developer who wishes to develop a site for housing such that it maximizes the company's utility - e.g. saleable square meters. The task is then to find a layout of apartment buildings such that this measure of utility is maximized, subject to the zoning rules on the site. As the SDP is inherently multi-objective and involves complex zoning regulations in practice, problems of reduced complexity are studied in both parts of the thesis. In this part, the problem is first and foremost reduced to a single-objective problem with the only goal of maximizing the total saleable ground floor area. This is a core objective for commercial stakeholders in the planning process, and therefore an obvious objective to include in a reduced formulation of the problem. With this objective, only a subset of spatial zoning rules are formulated as constraints. Since the ground floor area is the objective to be maximized, the zoning rules included are the ones that apply to the footprint of the buildings, thus reducing the problem to a two-dimensional problem. The reduced problem may still yield valuable results, serving as an approximation of the maximal saleable area and thus the land coverage ratio, or *utilization*, of a site. The results from the formulation and solution can be used as bases for further conceptual work in the planning process. Finally, before modeling the larger, more complex SDP, an appropriate and well-functioning formulation of the purely spatial aspects of the problem is necessary.

3.1 Site

The property owned or evaluated by the developer either has been or will be *zoned* by the planning authorities resulting in some area, the site, being approved for housing development. In practice, the site is defined by a set of coordinates that constitute its boundary, which in this formulation of the SDP is assumed be a simple, convex or non-convex polygon. The buildings must be placed fully inside the site in order to comply with the regulations, as shown in Figure 3.1.

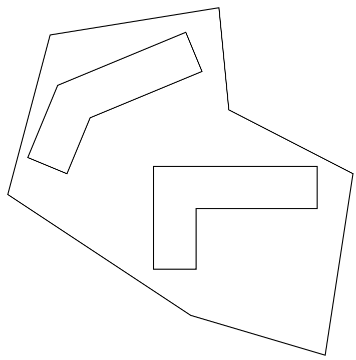


Figure 3.1: An example of a non-convex site with two buildings

3.2 Buildings

Since real estate developers or the planning authorities can require that a site should be developed with a certain number of buildings, this is considered fixed in the problem (Trondheim Kommune, 2015). The buildings are assumed to be angled along the length, with variable sizes for the angle, length, width and rotation of each building. Moreover, the study in this part of the thesis does not introduce any measure of utility or regulations that depend on the height of the buildings. Thus, the variables of the buildings will be those that vary in the plane, i.e. the angle, width, length and rotation of the building footprint. The lower and upper bounds on the width and length of the buildings are determined by best practice and depend on what the developer believes to suit the target market. This will also prevent any single building from covering most of the site area. We assume that there can be no internal overlap between the buildings so that they are treated as independent building areas.

3.3 Zoning regulations

Each site may have different zoning regulations, but some common constraints are assumed for the general formulation of the SDP. Certain zoning regulations require unoccupied space between buildings due to fire safety, daylight regulations and privacy. To incorporate these regulations, additional unoccupied space (”open zones”) outside each facade of the buildings must be enforced, as displayed in Figure 3.2. The open zones are allowed to overlap internally and lie outside the site boundary (see Figure 3.3), but building structures are not allowed to overlap with the open zones of other buildings.

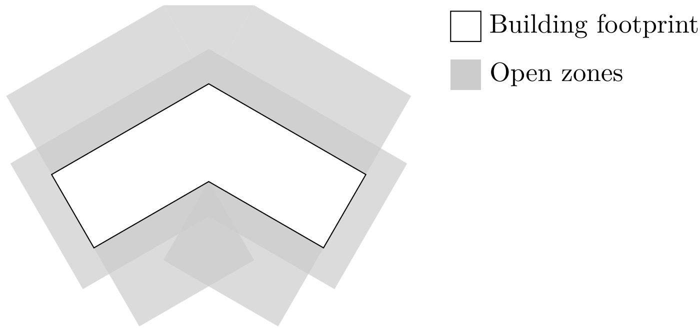


Figure 3.2: Building ground floor and open zones

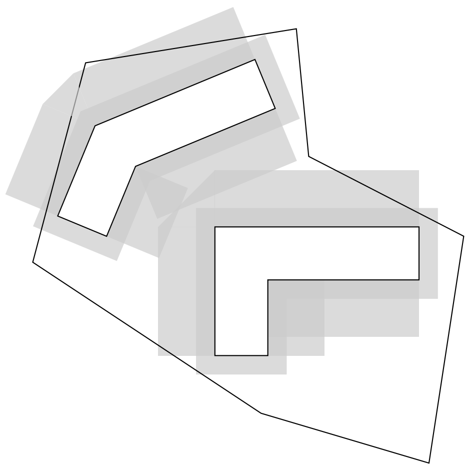


Figure 3.3: Overlap between the open zones, and the open zones and the site boundary

3.4 Objective

Although the objective will vary depending on different factors such as the site, geographic location and target market, to name a few, the real estate developer normally wishes to maximize the saleable floor area while complying with regulations. In practice, this is the combined area of all the apartments in the housing layout. In this formulation, where the height can be considered fixed, this can be modeled as the combined area of the building ground floors, or *building footprint*.

Chapter 4

Mathematical model I

In this chapter, a mathematical formulation of the SDP is presented. First, the notation used is presented. Next, a high-level formulation is given. The constraints are not described in detail in this chapter (they are further described in Chapter 5) as they have several possible formulations that each require notational and algorithmic explanation.

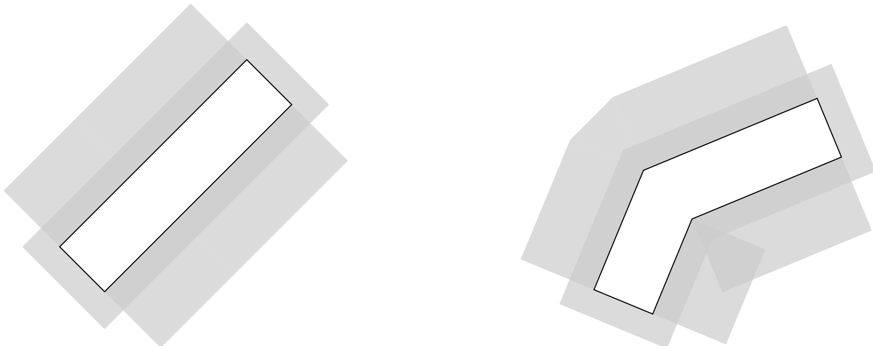
4.1 Assumptions and simplifications

4.1.1 Site

We assume that the boundary of the site is represented by a set of corner points. The site can be either convex or non-convex.

4.1.2 Buildings

The buildings are shaped by taking a rectangle as a starting point, and then creating an angle along the length of the rectangle, such that the buildings can be shaped as non-convex polygons with six corner points. This is illustrated in Figure 4.1. Each building has open zones on all sides, with fixed and equal width and length for all buildings. For a building with an angle, the open zones around the outer angle is created as a triangle with one corner in the angle vertex and two sides equal to the length of the open zones on the long sides. Figure 4.1 also illustrates the open zones of the building.



(a) A building rectangle without a perceptible angle (but actually an angle of 180°) and open zones in gray

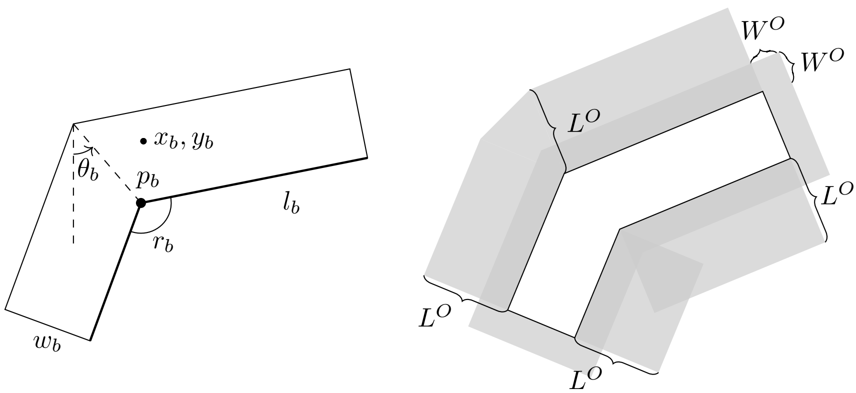
(b) A building with an angle and open zones in gray

Figure 4.1: Buildings

4.2 Notation

Sets	
$\mathcal{B}$	Set of buildings
$\mathcal{C}$	Set of site corner points
Parameters	
$\underline{W}, \overline{W}$	Minimum and maximum width of each building
$\underline{L}, \overline{L}$	Minimum and maximum length of each building
W^O	Width of open zone
L^O	Length of open zone
X_i, Y_i	Coordinates of ith corner point in $\mathcal{C}$
K	Minimum distance from angle point to neighboring site corner point
Variables	
$w_b \in \mathbb{R}$	Width of building b
$l_b \in \mathbb{R}$	Length of building b
$x_b, y_b \in \mathbb{R}$	Centroid of building b
$\theta_b \in \mathbb{R}$	Rotation of building b
$p_b \in \mathbb{R}$	Angle point of building b
$r_b \in \mathbb{R}$	Angle of building b

Let $\mathbf{v}_b = (x_b,\ y_b,\ w_b,\ l_b,\ \theta_b,\ p_b,\ r_b)$ denote the *configuration* vector of building $b \in \mathcal{B}$, and $F(\mathbf{v}_b)$ a building footprint with center at (x_b, y_b), width w_b, length l_b, building rotation θ_b, and angle point p_b with angle r_b. Figure 4.2a illustrates the variables of the configuration vector $\mathbf{v}_b$, and Figure 4.2b illustrates the parameters of the open zones. The building center, (x_b, y_b), is defined as the center of the building without an angle, that is, a rectangular building with the same length and width as the angled building. The length l_b of the building is defined as the "inner" length of the building when it is angled. This is marked with a thicker line in Figure 4.2a. This implies that the length of the building does not change with the angle of the building. The building rotation, θ_b, is measured as the angle between a vertical line starting in the upper angle point, and the line between the upper and the lower angle point p_b. These lines are illustrated with dashed lines in Figure 4.2a. The angle point, p_b, is placed along the length l_b of the building. For easier visualization, p_b is referred to as a point, although it is a single value in the model that determines the point's location relative to the building corners. The angle r_b of the building is 180° when the building is shaped as a rectangle, and decreases as the building bends more.



(a) The building configuration vector

(b) A building and its corresponding open zones

Figure 4.2: The building variables and parameters

4.3 Model

4.3.1 Objective function

$$\text{maximize} \sum_{b \in \mathcal{B}} w_b \cdot l_b + \frac{w_b^2}{\tan(\frac{r_b}{2})} \tag{4.3.1}$$

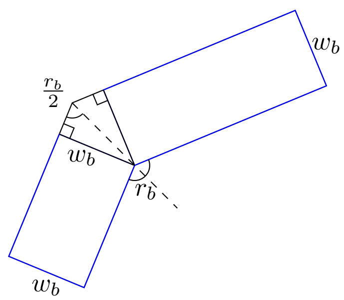
The objective of the program is to maximize the total area of all the buildings. The first term of Equation (4.3.1) is the area of the building if it is shaped as a rectangle, which equals the areas marked in blue in Figure 4.3a. The last term is the area added due to the angled building shape, which is the area of two equal-area right-angled triangles created by the dashed line in Figure 4.3a. Figure 4.3b illustrates one of the right-angled triangles. The area of the triangle in Figure 4.3b can be expressed as

$$A = \frac{1}{2} \cdot x \cdot w_b \tag{4.3.2}$$

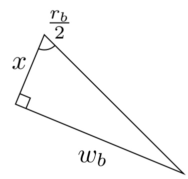
The tangent of the triangle is calculated and used to find x

$$\tan\left(\frac{r_b}{2}\right) = \frac{w_b}{x} \implies x = \frac{w_b}{\tan(\frac{r_b}{2})} \tag{4.3.3}$$

Substituting the expression for x into Equation (4.3.2) gives the area of one of the triangles. Multiplying by two gives the total area added by the angled shape of the building, and equals the last term in Equation (4.3.1).



(a) The building



(b) One of the right-angled triangles

Figure 4.3: Calculation of the objective function

4.4 Constraints

4.4.1 Building sizes and rotation

For each building $b \in \mathcal{B}$ we must have

$$\underline{W} \leq w_b \leq \overline{W} \tag{4.4.1}$$
$$\underline{L} \leq l_b \leq \overline{L} \tag{4.4.2}$$
$$0 \leq \theta_b \leq 2\pi \tag{4.4.3}$$
$$K \leq p_b \leq l_b - K \tag{4.4.4}$$
$$\frac{\pi}{2} \leq r_b \leq \pi \tag{4.4.5}$$

4.4.2 No overlap between buildings

Each building is divided into two convex parts to facilitate the detection of overlap. The buildings are divided along the two angle points as illustrated in Figure 4.4, creating two *half-buildings*, denoted $P^1(\mathbf{v}_b)$ and $P^2(\mathbf{v}_b)$, i.e. $F(\mathbf{v}_b) = P^1(\mathbf{v}_b) \cup P^2(\mathbf{v}_b)$. For each pair of buildings, the ground area of one cannot overlap with the

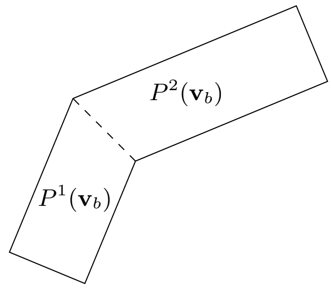


Figure 4.4: Decomposition of the non-convex angled building into two convex parts, $P^1(\mathbf{v}_b)$ and $P^2(\mathbf{v}_b)$

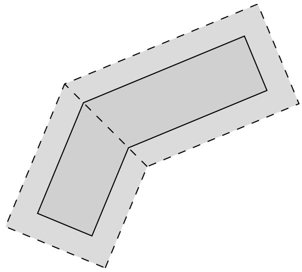
open zones, and ground area, of the other. Observe that the open zones around each building can be decomposed into three different components, the open zones created by extending the short side of the building, the open zones created by extending the long sides of the building, and the open zone created around the outer angle point of the building. Figure 4.5 illustrates this decomposition. From this decomposition, convex sub-shapes are created. The sub-shapes are represented by dashed lines in Figures 4.5b, 4.5c and 4.5d. This allows for a simpler formulation of the building overlap constraints. Let $T^1(\mathbf{v}_b)$ and $T^2(\mathbf{v}_b)$ be the trapezoids created from the open zones on the short sides of building b, illustrated in Figure 4.5b. Let $R^1(\mathbf{v}_b)$, $R^2(\mathbf{v}_b)$, $R^3(\mathbf{v}_b)$ and $R^4(\mathbf{v}_b)$ represent the four rectangles created from the open zones on the long sides of building b, as seen in Figure 4.5c. Finally, let $A(\mathbf{v}_b)$ be the triangle that covers the space between the open zones on the long side where the outer angle point is. For each pair of building $b, a \in \mathcal{B},\ b \neq a$ it must be enforced that

$$F(\mathbf{v}_b) \cap \left(T^1(\mathbf{v}_a) \cup T^2(\mathbf{v}_a)\right) = \emptyset \tag{4.4.6}$$

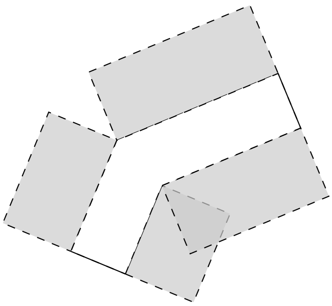
$$F(\mathbf{v}_b) \cap \left(R^1(\mathbf{v}_a) \cup R^2(\mathbf{v}_a) \cup R^3(\mathbf{v}_a) \cup R^4(\mathbf{v}_a)\right) = \emptyset \tag{4.4.7}$$

$$F(\mathbf{v}_b) \cap A(\mathbf{v}_a) = \emptyset \tag{4.4.8}$$

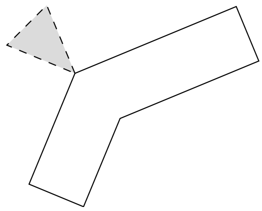
(a) A building and all of its corresponding open zones



(b) Open zones on the short sides, trapezoidal sub-shapes $T^1(\mathbf{v}_b)$ and $T^2(\mathbf{v}_b)$



(c) Open zones on the long sides, rectangular sub-shapes $R^1(\mathbf{v}_b)$, $R^2(\mathbf{v}_b)$, $R^3(\mathbf{v}_b)$ and $R^4(\mathbf{v}_b)$



(d) Open zone around the outer angle point, triangular sub-shape $A(\mathbf{v}_b)$

Figure 4.5: Decomposition of the open zones

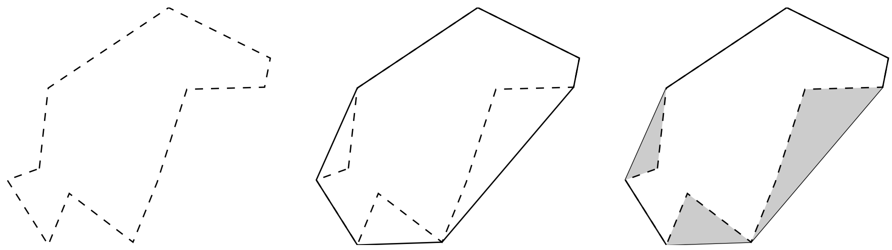
4.4.3 Containment within site

The open zones are allowed to overlap with the site boundary, while the building footprint is not. We require for each $b \in \mathcal{B}$ that

$$F(\mathbf{v}_b) \cap \operatorname{conv}(\mathcal{C}) = F(\mathbf{v}_b) \tag{4.4.9}$$

where $\operatorname{conv}(\mathcal{C})$ denotes the convex hull of the site corner points. To then enforce containment within a non-convex site, a constraint to prevent overlap with the difference $\operatorname{conv}(\mathcal{C}) \setminus \mathcal{C}$ is introduced, as shown in Figure 4.6c. Let $\mathcal{S}' = \operatorname{conv}(\mathcal{C}) \setminus \mathcal{C}$ and let $\mathcal{S}$ be the set of convex sub-shapes found by decomposing $\mathcal{S}'$. We require for each $b \in \mathcal{B}$ and $S \in \mathcal{S}$ that

$$F(\mathbf{v}_b) \cap S = \emptyset \tag{4.4.10}$$



(a) Non-convex site **(b)** Convex hull (solid) of non-convex site (dashed) **(c)** No-overlap shapes S (gray)

Figure 4.6: Containment within a non-convex site

4.5 Model properties

For convenience, the mathematical model for the SDP is repeated in full with the objective function and constraints presented in the previous sections of this chapter.

$$\max \sum_{b \in \mathcal{B}} w_b \cdot l_b + \frac{w_b^2}{tan(\frac{r_b}{2})} \tag{4.3.1}$$

$$\text{s.t. } F(\mathbf{v}_b) \cap \left(T^1(\mathbf{v}_a) \cup T^2(\mathbf{v}_a)\right) = \emptyset \qquad b, a \in \mathcal{B},\ b \neq a \tag{4.4.6}$$

$$F(\mathbf{v}_b) \cap \left(R^1(\mathbf{v}_a) \cup R^2(\mathbf{v}_a) \cup R^3(\mathbf{v}_a) \cup R^4(\mathbf{v}_a)\right) = \emptyset \qquad b, a \in \mathcal{B},\ b \neq a \tag{4.4.7}$$

$$F(\mathbf{v}_b) \cap A(\mathbf{v}_a) = \emptyset \qquad b, a \in \mathcal{B},\ b \neq a \tag{4.4.8}$$

$$F(\mathbf{v}_b) \cap \text{conv}(\mathcal{C}) = F(\mathbf{v}_b) \qquad b \in \mathcal{B} \tag{4.4.9}$$

$$F(\mathbf{v}_b) \cap S = \emptyset \qquad b \in \mathcal{B}, S \in \mathcal{S} \tag{4.4.10}$$

$$\underline{W} \leq w_b \leq \overline{W} \qquad b \in \mathcal{B} \tag{4.4.1}$$

$$\underline{L} \leq l_b \leq \overline{L} \qquad b \in \mathcal{B} \tag{4.4.2}$$

$$0 \leq \theta_b \leq 2\pi \qquad b \in \mathcal{B} \tag{4.4.3}$$

$$K \leq p_b \leq l_b - K \qquad b \in \mathcal{B} \tag{4.4.4}$$

$$\frac{\pi}{2} \leq r_b \leq \pi \qquad b \in \mathcal{B} \tag{4.4.5}$$

The model is a nonlinear program with continuous variables. There are nonlinearities both in the objective function and the constraints. The objective function is nonlinear as it maximizes the area of the footprint of the buildings, and the constraints (4.4.6) - (4.4.10) are nonlinear because they depend on the shape and rotation of each building.

It is also important to distinguish between convex and non-convex problems. A maximization problem is convex if the objective function is concave and the feasible region defined by the constraints is a convex set (Lundgren et al., 2010). Recall that a set $C \subset \mathbb{R}^n$ is convex if for all points $x, y \in C$ and $0 \leq \lambda \leq 1$ we have

$$\lambda x + (1 - \lambda)y \in C \tag{4.5.1}$$

That is, for any two points x and y in the set, the line segment joining the two points is fully contained in the set (Nocedal and Wright, 2006). To show that the given model is non-convex, a counterexample to the proposition that the feasible set is convex is provided. Consider a building that lies along the site boundary such that one of its sides is perfectly aligned with the boundary. Assume then that the rotation is increased such that one corner lies outside the site, yielding an infeasible placement. If the rotation is increased enough, the same corner can enter the site again, and the building placement is feasible. Thus, a linear combination of these two feasible rotational values (for which both corner points are inside the site boundary) might give a rotation where the corner point is outside the site, and infeasible. Consequently, the model is non-convex. Another counterexample of convexity can be provided based on the overlap constraints. Consider a solution in which two buildings do not overlap. Another solution, where the building indices are switched, is visually identical and feasible. Taking the mean of these two identical solutions results in overlap of the buildings and thus infeasibility.

Chapter 5

Solution method I

In the following chapter, the solution method used to solve the mathematical model introduced in Chapter 4 is presented. The proposed formulation is implemented in MATLAB and solved using Sequential Quadratic Programming (SQP). First, different formulations of the constraints are presented in Section 5.1. The constraint functions in the formulations are defined such that they provide a measure of feasibility and infeasibility to the solution method applied. The chosen solution method used in MATLAB, SQP, is described further in Section 5.2. As seen in Section 4.5, the formulation of the problem is nonlinear in the objective function, which maximizes area, and in the constraints, which are dependent on the shape and rotation of each building. SQP is therefore chosen because it is one of the most effective methods for nonlinear, constrained optimization problems (Nocedal and Wright, 2006). Next, alternative definitions of the building variables are presented in Section 5.3. Lastly, different strategies for constructing initial solutions are proposed in Section 5.4.

5.1 Constraints

In the following sections, formulations and implementations of the high-level constraints (4.4.6)-(4.4.10) in the mathematical model are presented. While the constraints in the model are equality constraints, they are implemented as inequality constraints due to easier implementation. First, a concept used to detect overlap is introduced, before the constraint formulation that impose no-overlap between buildings is presented. This is followed by the formulation of the site containment constraints.

5.1.1 The Hyperplane Separation Algorithm

To detect overlap between the buildings, and between the buildings and the open zones, the *hyperplane separation theorem* is applied. Let $\mathcal{C}$ and $\mathcal{D}$ be two convex sets in $\mathbb{R}^n$ that do not intersect (i.e. $\mathcal{C} \cap \mathcal{D} = \emptyset$). Then, there exists an $a \in \mathbb{R}^n, a \neq 0,$ and $b \in \mathbb{R}$, such that $a^T x \leq b$ for all $x \in \mathcal{C}$ and $a^T x \geq b$ for all

$x \in \mathcal{D}$ (Gottschalk et al., 1996). In other words, the theorem states that if the two sets do not intersect, then there exists a hyperplane that separates them. In $\mathbb{R}^2$ the hyperplane is a line that separates the two sets. The theorem applied in $\mathbb{R}^2$ is illustrated in Figure 5.1.

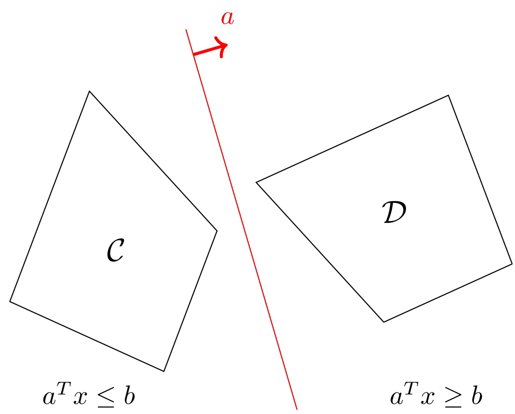


Figure 5.1: Illustration of the hyperplane separation theorem in $\mathbb{R}^2$

An algorithm to detect overlap between two buildings based on this theorem is developed. This algorithm is referred to as the *hyperplane separation algorithm* throughout the thesis. To determine whether or not there exists a line separating the two convex polygons, as illustrated in Figure 5.1, the shapes are projected onto a line. This line is called an *axis*, and the projections are checked for overlap on this axis. The hyperplane separation algorithm may test many axes for overlap. However, if the algorithm finds an axis where the projections do not overlap, it can immediately exit and conclude that there is no overlap between the two shapes. This is illustrated in Figure 5.2, where the trapezoids in Figure 5.1 are projected onto the y- and the x-axis. The projections overlap on the y-axis, but not on the x-axis. Thus, the conclusion can be made that the shapes do not overlap, without finding an actual hyperplane. An analogy to this method is to think of a person holding a torch, shining on the two shapes while moving around them, projecting their shadows onto a wall. If there is a gap in the shadow at any time, the two shapes do not overlap. Luckily, the hyperplane separation algorithm does not need to project the shapes onto an axis for all angles. It is enough to project the shapes onto the normals of the sides of the shapes. Also, since parallel surfaces will provide normals that lie on the same axis, it is enough to test one of the parallel sides (Gottschalk et al., 1996).

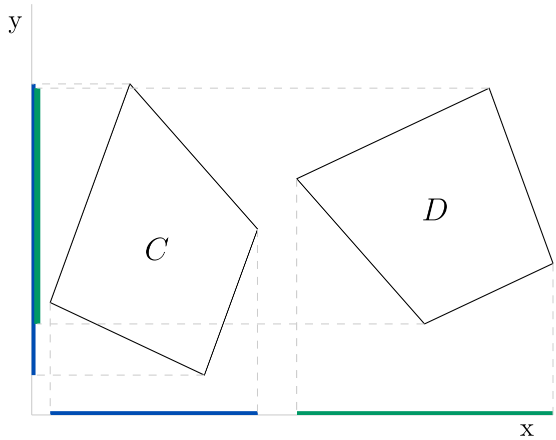


Figure 5.2: Projections of the two trapezoids onto the x- and y-axis

5.1.2 Building overlap

The hyperplane separation algorithm is used for each pair of convex sub-shape and half-building as stated in the constraints given in (4.4.6)-(4.4.8). The hyperplane separation algorithm needs to test all axes that are not parallel of each pair of sub-shapes and half-buildings. Both the half-buildings and the sub-shapes created from extending the open zones on the short sides are trapezoids. Two sides are parallel in a trapezoid and therefore at most six axes need to be tested for each pair of half-building and sub-shape in constraints (4.4.6). The sub-shapes created from the open zones on the long sides are rectangles. Thus, at most five axes needs to be tested for each pair of half-building and rectangle in constraints (4.4.7). The sub-shape created from the open zone around the outer angle point is a triangle, and at most six axes need to be tested for each pair of half-building and sub-shape in constraints (4.4.8).

In addition to detect overlap, it is necessary to find the minimum distance the buildings have to be moved in order to no longer overlap, or the distance separating them if they do not overlap. This is a way to model the overlap constraints as continuous functions. The hyperplane separation algorithm can return a minimum translation vector (Gottschalk et al., 1996) which will provide this information. While checking the projections of the shapes for overlap, the algorithm can keep track of the minimum distance and corresponding axis. However, now the algorithm cannot exit early if it concludes that there is no overlap because there might be an axis which has not yet been tested with a smaller minimum distance than the current one.

An algebraic formulation for the algorithm is provided in the following, where the two sub-shapes are assumed to be trapezoids. Let $\hat{\mathbf{n}}_{bl}$ be the unit vector that is normal (to the right) to the edge ending in corner point c_{bl}, as shown in Figure

5.3. Assume the building edge vectors are created in the same order, i.e. clockwise or counter-clockwise. Then the constraint functions can be formulated as

$$D_{a,b} = \max_{i=1\ldots4}\Big(\min_{j=1\ldots4}\{\langle c_{aj}, \hat{\mathbf{n}}_{bi}\rangle - \langle c_{bi}, \hat{\mathbf{n}}_{bi}\rangle\}\Big) \tag{5.1.1}$$

$$D_{b,a} = \max_{i=1\ldots4}\Big(\min_{j=1\ldots4}\{\langle c_{bj}, \hat{\mathbf{n}}_{ai}\rangle - \langle c_{ai}, \hat{\mathbf{n}}_{ai}\rangle\}\Big) \tag{5.1.2}$$

and it must be enforced that

$$\max(D_{a,b}, D_{b,a}) \geq 0 \tag{5.1.3}$$

for all unique pairs of half-buildings $P(\mathbf{v}_a), P(\mathbf{v}_b),\ (a,b) \in \mathcal{B},\ b \neq a$ and the corners of the sub-shapes. The notation $\langle\cdot\rangle$ is used to denote the inner product. The inner min-problem determines which corner point of building a that lies closest to building b on the axis defined by vector $\hat{\mathbf{n}}_{bi}$. If the shortest distance in all directions is negative, then the buildings overlap. Therefore, the constraint requires that *at least* one axis has a positive distance (i.e. a separating hyperplane exists), enacted by the outer max-problem over axes. Although the implementation only needs to check three axes, since only one of the parallel sides of the trapezoid needs to be tested (i.e. $\max_{i=1,3}(\cdot)$), all are included here as it allows a simpler formulation.

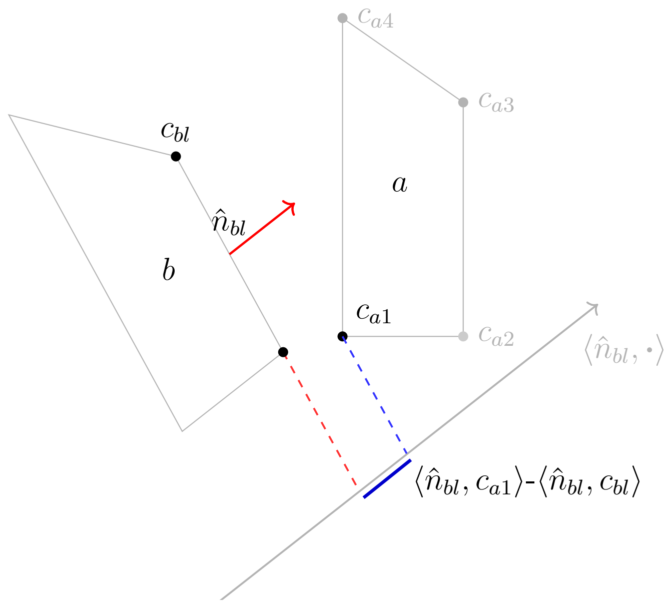


Figure 5.3: Illustration of constraint (5.1.1). The expression finds the normal vector, or axis, $\hat{\mathbf{n}}_{bl}$ such that the shortest distance (the blue line) is the maximum over all the axes

5.1.3 Containment within site

Constraints (4.4.9) require that the building footprint is contained within the convex hull of the site. These constraints are fulfilled by ensuring that all corner points of a building are inside the convex hull of the site. The convex hull can be viewed as an intersection of linear inequalities, so that every building corner point must be contained in this intersection. There are different ways to define these linear inequalities depending on the expression used to define the lines that pass through the site edges. Although the selected formulation is more tedious to develop, it allows for a natural interpretation. Let $(X_i, Y_i), (X_j, Y_j)$ be two site corner points which also are included in the convex hull, and (x_{kl}, y_{kl}) the lth corner point of building b, as shown in Figure 5.4. An expression for the distance d, the blue line in Figure 5.4, is developed. It is defined to be the shortest (perpendicular) distance from a corner point to a site edge, and it is negative when the corner point is inside the site, zero when it touches the border, and positive outside. Figure 5.4 shows that the distance can be expressed as the projection of the vector $\mathbf{v} = (X_i - x_{bl},\ Y_i - y_{bl})^T$ onto the unit normal vector $\hat{\mathbf{n}}_{\mathbf{ij}}$. A normal vector for the edge can be defined as $\mathbf{n}_{\mathbf{ij}} = (-(Y_j - Y_i),\ X_j - X_i)^T = (Y_i - Y_j,\ X_j - X_i)^T$ and its length is thus $\|\mathbf{n}_{\mathbf{ij}}\| = \sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2}$. The distance d is calculated by the projection (Adams and Essex, 2013)

$$
\begin{aligned}
d = \frac{\mathbf{n}_{\mathbf{ij}} \cdot \mathbf{v}}{\|\mathbf{n}_{\mathbf{ij}}\|} &= \frac{1}{\|\mathbf{n}_{\mathbf{ij}}\|}\Big((Y_i - Y_j)(X_i - x_{bl}) + (X_j - X_i)(Y_i - y_{bl})\Big) \\
&= \frac{1}{\|\mathbf{n}_{\mathbf{ij}}\|}\Big((Y_i - Y_j)x_{bl} + (X_j - X_i)y_{bl} + (X_iY_j - X_jY_i)\Big)
\end{aligned}
\tag{5.1.4}
$$

Now, as the corner points $(X_i, Y_i), (X_j, Y_j)$ for a site are constant and known, the expression is linear in the building corner points (x_{bl}, y_{bl}), and can be used to ascertain that all buildings are inside the convex hull of the site by introducing the constraint

$$
\frac{1}{\|\mathbf{n}_{\mathbf{ij}}\|}\Big((Y_i - Y_j)x_{bl} + (X_j - X_i)y_{bl} + (X_iY_j - X_jY_i)\Big) \leq 0 \tag{5.1.5}
$$

for $(i, j) \in (\mathcal{C}') \cup \mathcal{C}'_1,\ j = i + 1,\ b \in \mathcal{B},\ l = 1, ..., 6$ where $\mathcal{C}' = \mathrm{conv}(\mathcal{C})$.

Next, constraints (4.4.10) enforce containment within a non-convex site by the introduction of no-overlap shapes, as previously illustrated in Figure 4.6. The hyperplane separation algorithm is utilized again to ensure no overlap between the half-buildings and the no-overlap shapes. The use of the hyperplane separation algorithm assumes that the no-overlap shapes are convex. If some of the shapes are non-convex, they are decomposed into convex shapes.

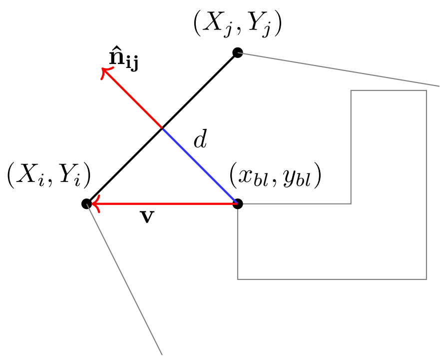


Figure 5.4: Calculation of distance to edge

5.2 Sequential Quadratic Programming

The basic idea of SQP is to model the nonlinear optimization problem as a quadratic subproblem at a given approximate solution, $\mathbf{x}_k$. The quadratic subproblem is solved, and the solution is used to construct a better approximation, $\mathbf{x}_{k+1}$, to the solution. The method is iterative and creates a sequence of approximations that, if the method is successful, will converge to a (local) minimum $\mathbf{x}^*$ (Boggs and Tolle, 1996).

The SQP method implemented in MATLAB assumes a general problem of the form (MathWorks, 2017)

$$\min_x \; f(x) \tag{5.2.1}$$

$$\text{subject to} \quad g_i(x) \geq 0 \qquad i = 1, ..., n \tag{5.2.2}$$

$$g_i(x) = 0 \qquad i = n+1, ..., m \tag{5.2.3}$$

where there can be nonlinearities in both the objective function and the constraints. It follows that the Lagrangian of the problem is given by

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) \tag{5.2.4}$$

The quadratic programming subproblem is obtained by linearizing the nonlinear constraints, and formulating the objective function as a quadratic approximation of the Lagrangian. The quadratic programming subproblem is given by

$$\min_d \quad \frac{1}{2} d^T H_k d + \nabla f(x_k)^T d \tag{5.2.5}$$

$$\text{subject to} \quad \nabla g_i(x_k)^T d + g_i(x_k) \geq 0 \qquad i = 1, ..., n \tag{5.2.6}$$

$$\nabla g_i(x_k)^T d + g_i(x_k) = 0 \qquad i = n+1, ..., m \tag{5.2.7}$$

where H_k is a positive definite approximation of the Hessian of the Lagrangian at x_k, $\nabla^2_{xx} L(x_k, \lambda_k)$ (MathWorks, 2017; Nocedal and Wright, 2006), and the Lagrange multipliers λ are defined as free variables for equality constraints and greater than or equal to zero for inequality constraints. The step d is used to construct a better approximation to the solution of the original problem. As seen from the expression, which includes both the Hessian and gradients, the SQP assumes that the objective and constraints are twice differentiable. The Hessian of the Lagrangian needs to be approximated with a positive definite Hessian to ensure that the subproblem is well defined.

The quadratic programming subproblem is solved using an active set strategy (MathWorks, 2017). An active set strategy is based on trying to identify the set of constraints that are active at the solution, and treat these constraints as equality constraints when solving the quadratic programming problem. The solution of this problem defines the new iterates as $(x_k + \alpha_k d_k, \lambda_{k+1})$ where d_k and λ_{k+1} are the solution and corresponding Lagrange multiplier of the quadratic programming subproblem. The step length α_k is determined based on creating a sufficient decrease in a merit function. The merit function is created to balance the often competing goals of reducing the objective function and satisfying the inequality constraints. The merit function combines the decrease in the objective with a measure of constraint violation, such that the step length will improve the objective function without too much violation of the constraints (Nocedal and Wright, 2006).

The implementation of SQP in MATLAB treats the objective and constraint functions $f(x)$, $g_i(x)$ as *black-box* functions, in that it only needs the values returned by a user-defined constraint function and no further information about their structure (other than the assumptions of SQP). To evaluate the gradients $\nabla f(x_k), \nabla g_i(x_k)$ at an iterate x_k, the method of *finite difference approximation* is used if analytic gradients are not provided. One might observe that the overlap constraint function, as exemplified in Equation (5.1.1), is not everywhere twice differentiable (as is assumed by SQP), due to use of max- and min-operations. More specifically, it is not differentiable in the case when the argmax or argmin of Equation (5.1.1) can be more than one axis or corner, as shown in Figure 5.5. However, as these situations should be rare, the constraint is *almost everywhere differentiable*. In addition, SQP only evaluates the gradient of the constraints when they are active.

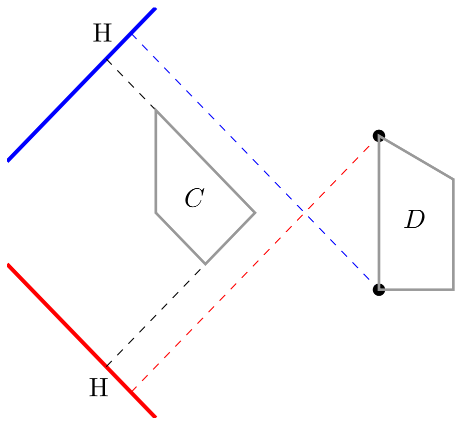


Figure 5.5: Both the red and the blue line may be selected as the axis that have the maximum shortest distance between half-buildings C and D, with distance H

5.3 Alternative building variables

Although the selected variable set for defining the buildings is clear and intuitive, it is not immediately obvious whether it is the most effective in terms of computing time, or whether it yields the best results. Therefore, four alternatives to the building variable set are proposed in this section. All of the alternatives are based on the common goal to avoid using an angle to define the orientation of the building, as the original variable set does with θ_b. The rotation variable θ is bounded within the interval $[0,\ 2\pi]$. One may observe that if the building is oriented with a rotation $\theta' = 2\pi$ and the optimal solution is obtainable by an additional rotation $\Delta\theta$ such that $\theta^* = \theta' + \Delta\theta$, then with the defined bounds, the building must be rotated in the other direction by $2\pi - \Delta\theta$ to find the same orientation. Indeed, this may be solved by increasing the size of the interval, e.g. to $[-4\pi, 4\pi]$, as has been implemented in practice. However, this does not eliminate the problem completely. To avoid this problem, two of the proposed variable alternatives use certain points on the building and distances to define a new variable set, while the other two alternatives use vectors.

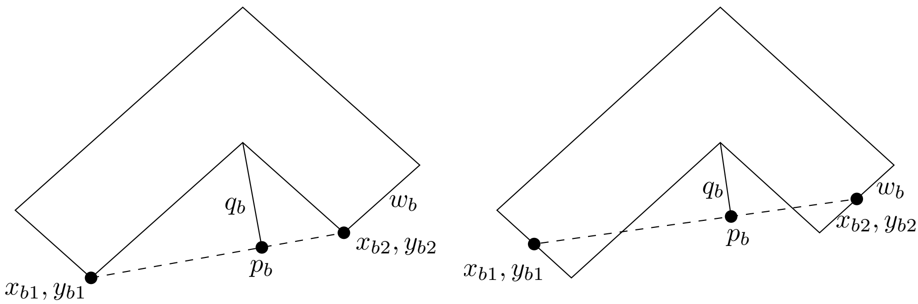
5.3.1 Points

Two-corner-points

Let the length of the building be defined by the line segment with end points (x_{b1}, y_{b1}) and (x_{b2}, y_{b2}), a point p_b on the line segment and a height q_b, where the height denotes the perpendicular distance from the point p_b to the angled corner in the building. That is, if the height equals zero, the building will be shaped as a rectangle. The width of the building is defined as before. The configuration vector of this formulation is thus defined as $\mathbf{v}_b = (x_{b1},\ y_{b1},\ x_{b2},\ y_{b2},\ w_b,\ p_b,\ q_b)$. The variables are illustrated in Figure 5.6a.

Mid-width-points

This formulation is similar to the previous, but the end points of the line segment are now defined as the points in the middle of the width of the building. The idea behind this alternative to the previous is that the angle point of the buildings can change from lower angle point to upper angle point without having to rotate the whole building. That is, the building can bend the other way more easily. Except for the alternation to the end points of the line segment, the configuration vector is identical to the other one, that is, $\mathbf{v}_b = (x_{b1},\ y_{b1},\ x_{b2},\ y_{b2},\ w_b,\ p_b,\ q_b)$. The variables that constitute the configuration vector are exemplified in Figure 5.6b.



(a) Two corner points **(b)** Mid-width points

Figure 5.6: The alternative variable sets with defined points

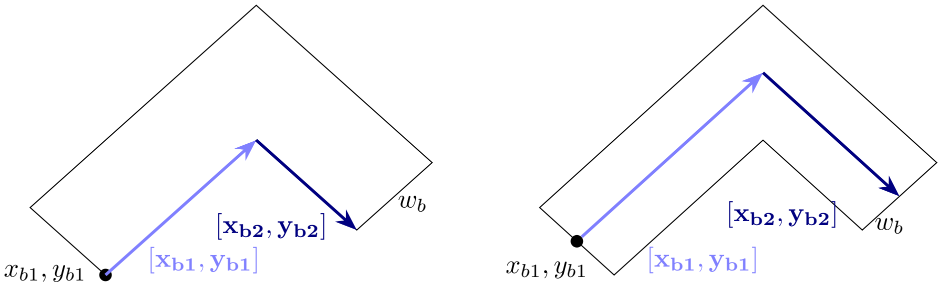
5.3.2 Vector variables

Length-vectors

This variable alternative is called length-vectors because the total length of the defined vectors equal the length in the original variable set. The variables of this alternative are a corner point (x_{b1}, y_{b1}), a vector with starting point in the given corner point, $[\mathbf{x}_{b1}, \mathbf{y}_{b1}]$, and a second vector with starting point in the end point of the first vector, given by $[\mathbf{x}_{b2}, \mathbf{y}_{b2}]$. The width, w_b, is defined as in the previous alternatives. Thus, the configuration vector becomes $\mathbf{v}_b = (x_{b1}, y_{b1}, \mathbf{x}_{b1}, \mathbf{y}_{b1}, \mathbf{x}_{b2}, \mathbf{y}_{b2}, w_b)$. The variables are seen in Figure 5.7a.

Mid-width-vectors

The configuration vector of this alternative is very similar to the length vector alternative. However, the starting point and the vectors are defined differently. In this alternative, the starting point for the first vectors is the mid-width point, and the first vector ends between the upper and lower angled point of the building. This is where the second vector begins, and it ends in the mid-width point on the other side. Again, the idea of this alternation to the length vector alternative is that the building can bend the other way more easily. The other alternatives, except mid-width, must be rotated 180° to bend the opposite direction. The width, w_b, is defined as before. This is illustrated in Figure 5.7b. The configuration vector is $\mathbf{v}_b = (x_{b1}, y_{b1}, \mathbf{x}_{b1}, \mathbf{y}_{b1}, \mathbf{x}_{b2}, \mathbf{y}_{b2}, w_b)$.



(a) Length vectors **(b)** Mid-width vectors

Figure 5.7: The alternative variable sets with vectors

5.4 Constructing initial solutions

The SDP might have multiple local optima, and as described in Section 5.2, SQP most likely returns a local optimum. There is no guarantee that the local optimum returned by SQP is a good solution to the problem. In order to obtain satisfactory results, the initial solution provided to the solver should be close to a good, locally optimal solution in order for the solver to more easily converge to such a solution. Different strategies for producing initial solutions are developed and described in the following. The SQP is not a feasible-point method (Nocedal and Wright, 2006), meaning that an initial solution can be infeasible and the method could still converge to a good solution.

5.4.1 Random

In this approach, the building center points are chosen randomly within the *envelope*, or *bounding box*, of the site, and the dimensions and rotations of the buildings are drawn uniformly at random from within the variable bounds. The buildings drawn have a rectangular shape. Pseudocode for the strategy is provided in Algorithm 1. The solution constructed may be infeasible as buildings might overlap or be placed partly or fully outside the site, depending on their shape and size. This naive approach is developed to allow for easier comparison with other heuristics, in order to determine whether or not the other heuristics actually provide better results. An example of output is shown in Figure 5.8.

Algorithm 1: RandomInitialSolution

Data: Set of site corner points $\mathcal{C}$ and number of buildings N
Result: A set buildings $\mathcal{B}$ and a set $\mathcal{V}$ of initial solution configuration vectors, v_b, for each $b \in \mathcal{B}$

begin
  $\mathcal{B} \longleftarrow \emptyset$
  $\mathcal{V} \longleftarrow \emptyset$
  $\underline{X},\ \overline{X} \longleftarrow \min_{X_i} \mathcal{C},\ \max_{X_i} \mathcal{C}$
  $\underline{Y},\ \overline{Y} \longleftarrow \min_{Y_i} \mathcal{C},\ \max_{Y_i} \mathcal{C}$
  for $b = 1 \ldots N$ **do**
    $x_b \longleftarrow \text{Uniform}[\underline{X}, \overline{X}]$
    $y_b \longleftarrow \text{Uniform}[\underline{Y}, \overline{Y}]$
    $w_b \longleftarrow \text{Uniform}[\underline{W}, \overline{W}]$
    $l_b \longleftarrow \text{Uniform}[\underline{L}, \overline{L}]$
    $\theta_b \longleftarrow \text{Uniform}[0, \pi]$
    $v_b = (x_b,\ y_b,\ w_b,\ l_b,\ \theta_b)$
    $\mathcal{V} \longleftarrow \mathcal{V} \cup \{v_b\}$
    $\mathcal{B} \longleftarrow \mathcal{B} \cup \{b\}$
  end
  return $\mathcal{B}, \mathcal{V}$
end

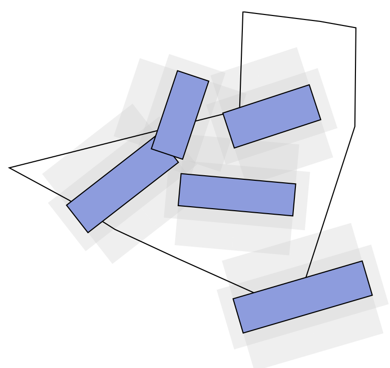


Figure 5.8: Initial solution with buildings randomly placed within the bounding box of the site

5.4.2 Open zones outside of the site boundary

The rationale behind the following initial solution strategy is that if as much of the open zones as possible lie outside the site boundary, the more area of the site is available to cover with saleable area, that is, the area of the buildings. The algorithm draws random site edges and places rectangular buildings with the long side along the selected edge, such that the entire open zone of this side is outside the site boundary. The length of the building is selected at random within the bounds, but shorter than the random site edge. The width is maximal to cover as much as possible of the open space towards the center of the site. The part of the site edge that is now covered by the recently placed building is removed from the set of site edges, and if the remaining parts of the edge are longer than the minimum building length, then they are included in the set to draw new random edges from. The strategy may produce infeasible initial solutions when buildings overlap with the open zones of other buildings, or if two buildings on different edges overlap. The pseudocode in Algorithm 2 presents the strategy more precisely and an illustration of an initial solution generated by this method is shown in Figure 5.9.

Algorithm 2: OpenZonesOutsideInitialSolution

```
Data: Set of site corner points C and number of buildings N
Result: A set buildings B and a set V of initial solution configuration
        vectors, v_b, for each b ∈ B
begin
    B ⟵ ∅
    V ⟵ ∅
    /* Set of feasible site edges, i.e.  edges with length
       longer than the minimum length L as specified in the
       model                                                  */
    E ⟵ {E ∈ Edges(C) : ‖E‖ ≥ L}
    while |B| < N do
        if E = ∅ then
            /* No remaining edges to be selected, terminate and
               retry                                              */
            return
        end
        b ⟵ |B| + 1
        E_b ⟵ DrawRandom(E)
        v_b, E_b1, E_b2 ⟵ PlaceBuildingOnEdge(E_b, W̄)
        V ⟵ V ∪ {v_b}
        B ⟵ B ∪ {b}
        E ⟵ E \ {E_b}
        E ⟵ E ∪ {E_b1, E_b2}
    end
    return B, V
end
```

Algorithm 3: PlaceBuildingOnEdge

Data: An edge E_b and on which to place a building of random length and input width w_b

Result: A configuration vector of v_b of the placed building and the remaining parts E_{b1}, E_{b2} of the edge that are longer than the minimum building length $\underline{L}$

```
begin
    /* Select random length and compute remaining variables so
       that building lies along E_b                            */
    x_b, y_b, l_b, θ_b ⟵ ConfigureRandomlyAlongEdge(E_b)
    v_b ⟵ (x_b, y_b, w_b, l_b, θ_b)
    /* Split E_b into the remaining feasible part of edges.  If
       no remaining parts, return ∅, ∅                         */
    E_b1, E_b2 ⟵ RemainingFeasibleEdgeParts(E_b, v_b)
    return v_b, E_b1, E_b2
end
```

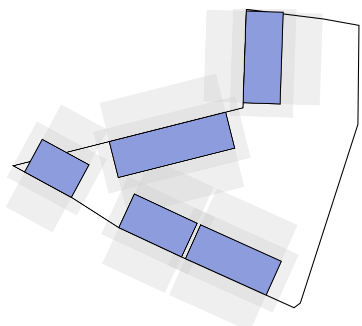


Figure 5.9: Initial solution with open zones outside the site boundary

5.4.3 Angled building placed in site corner

This initial solution strategy places the buildings in the site corners, with an angle equal to the angle of the site corner. The idea is to place as much area possible of the open zones outside the site boundary, while utilizing the increased area gained by angled buildings. The algorithm draws random site corners, and depending on whether the site corner is a convex corner or a reflex corner, one of the angled points of the building is placed in the same point as the selected site corner. A convex corner is a corner with an angle smaller than 180°, while a reflex corner is a corner with an angle greater than 180°. The building length and width is chosen randomly within their bounds, while the rotation and the angle of the building is determined such that the building fits perfectly within the corner point of the site. The drawn corner point is then removed from the list of available corner points such that no two buildings are placed in the same corner. If there are more buildings than corner points, the algorithm chooses one of the buildings already placed, and places a new building with the same orientation and angle next to the building placed towards the center of the site, and such that the open zones of the buildings perfectly overlap. This initial solution strategy may also produce infeasible initial solutions. Pseudocode for the strategy is presented in Algorithm 4, and an example of an initial solution generated by this strategy is shown in Figure 5.10.

Algorithm 4: AngleInCornerInitialSolution

Data: Set of site corner points $\mathcal{C}$ and number of buildings N

Result: A set of buildings $\mathcal{B}$ and a set $\mathcal{V}$ of initial solution configuration vectors, v_b, for each $b \in \mathcal{B}$

```
begin
    B ⟵ ∅
    V ⟵ ∅
    /* Set of feasible site edges                                  */
    E ⟵ {E ∈ Edges(C) :  ‖E‖ ≥ L̲}
    C_E ⟵ UniqueCorners(E)
    while |B| < N do
        if C_E = ∅ then
            /* No remaining corners to be selected                 */
            b ⟵ |B| + 1
            v_j ⟵ DrawRandom(V)
            /* Copy a random building and place the new building
               next to it                                          */
            v_b ⟵ CopyAndMoveInwards(v_j)
            V ⟵ V ∪ {v_b}
            B ⟵ B ∪ {b}
        else
            b ⟵ |B| + 1
            C_b ⟵ DrawRandom(C_E)
            v_b ⟵ PlaceBuildingInCorner(C_b)
            V ⟵ V ∪ {v_b}
            B ⟵ B ∪ {b}
            C_E ⟵ C_E \ {C_b}
        end
    end
    return B, V
end
```

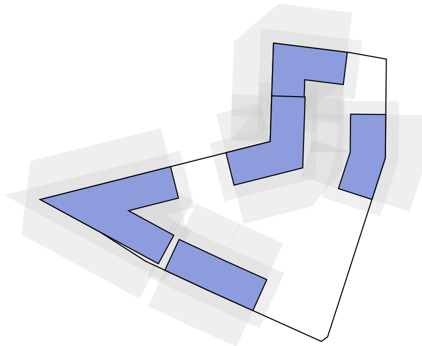


Figure 5.10: Initial solution with the angle of buildings equal to inner angle of the site corner points

5.5 Smooth approximation of overlap constraint functions

As described, the model for the SDP is currently non-smooth in the overlap constraints. More specificially, the max-min-operation applied in Equation (5.1.1) is non-differentiable. As SQP and many other solution techniques require that the problem is continuously differentiable, a smooth approximation of the overlap constraint is developed to compare the quality of the solutions found and the impact on the solution method. Recall that the overlap constraints for two trapezoids are defined as

$$\max_{i=1\ldots6}\Big(\min_{j=1\ldots6}\{\langle c_{aj},\hat{\mathbf{n}}_{bi}\rangle-\langle c_{bi},\hat{\mathbf{n}}_{bi}\rangle\}\Big)\geq 0 \tag{5.1.1}$$

Tsoukalas et al. (2009) provide a smooth approximation of functions of type

$$\Phi(x)=\max_{i\in I}\min_{j\in J}f_{i,j}(x) \tag{5.5.1}$$

Here, the inner function $f_{i,j}$ can for Equation (5.1.1) be defined as

$$f_{i,j}(\mathbf{c}_a,\mathbf{c}_b)=\langle c_{aj},\hat{\mathbf{n}}_{bi}\rangle-\langle c_{bi},\hat{\mathbf{n}}_{bi}\rangle \tag{5.5.2}$$

Now, let $M_I=|I|, M_J=|J|$ and $\epsilon_I>0, \epsilon_J<0$. Tsoukalas et al. (2009) propose that

$$\widetilde{\Phi}(x)=\frac{1}{\epsilon_I}\ln\left[\sum_{i\in I}\left[\sum_{j\in J}\exp\left(\epsilon_J f_{i,j}(x)\right)\right]^{\frac{\epsilon_I}{\epsilon_J}}\right] \tag{5.5.3}$$

approximates $\Phi(x)$ with lower and upper bound

$$\Phi(x)+\frac{\ln M_J}{\epsilon_J}\leq\widetilde{\Phi}(x)\leq\Phi(x)+\frac{\ln M_I}{\epsilon_I} \tag{5.5.4}$$

For the SDP model, M_I is the number of axes and M_J is the number of points tested for each axis. So, a differentiable overlap constraint can be defined as

$$\frac{1}{\epsilon_I} \ln \left[\sum_{i \in I} \left[\sum_{j \in J} \exp \left(\epsilon_J \cdot [\langle c_{aj}, \hat{\mathbf{n}}_{bi} \rangle - \langle c_{bi}, \hat{\mathbf{n}}_{bi} \rangle] \right) \right]^{\frac{\epsilon_I}{\epsilon_J}} \right] \geq 0 \tag{5.5.5}$$

for only one shape in each pair, to simplify notation. Here, increasing $|\epsilon_I|$ and $|\epsilon_J|$ will tighten the bounds and yield a more precise approximation.

Chapter 6

Computational Study I

The computational study tests and compares the alternative building variables, the different initial solution strategies, and the smooth approximation of the overlap functions proposed in Chapter 5. In addition, to study the impact by allowing a more complex shape of the buildings, angled buildings are compared to rectangular buildings. The tests performed within each solution method category are summarized in Table 6.1. The utilization of the sites, convergence to feasible solutions and computation time is considered in the evaluation of the different alternatives.

Table 6.1: The alternatives tested within each category

Initial solutions	Variables	Building shape	Approximations
Random generation	Original	Rectangle	Original
Open-zones-outside	Two-corner-points	Angled	Smoothing
Angled-building-in-site-corner	Mid-width-points		
	Length-vectors		
	Mid-width-vectors		

The model is solved using MATLAB as described in Section 5.2. The specifications of the hardware and software used to solve the model is presented in Table 6.2. The computational study has been conducted on the computing cluster *Solstorm* at the Department of Industrial Economics and Technology Management (IØT, 2018).

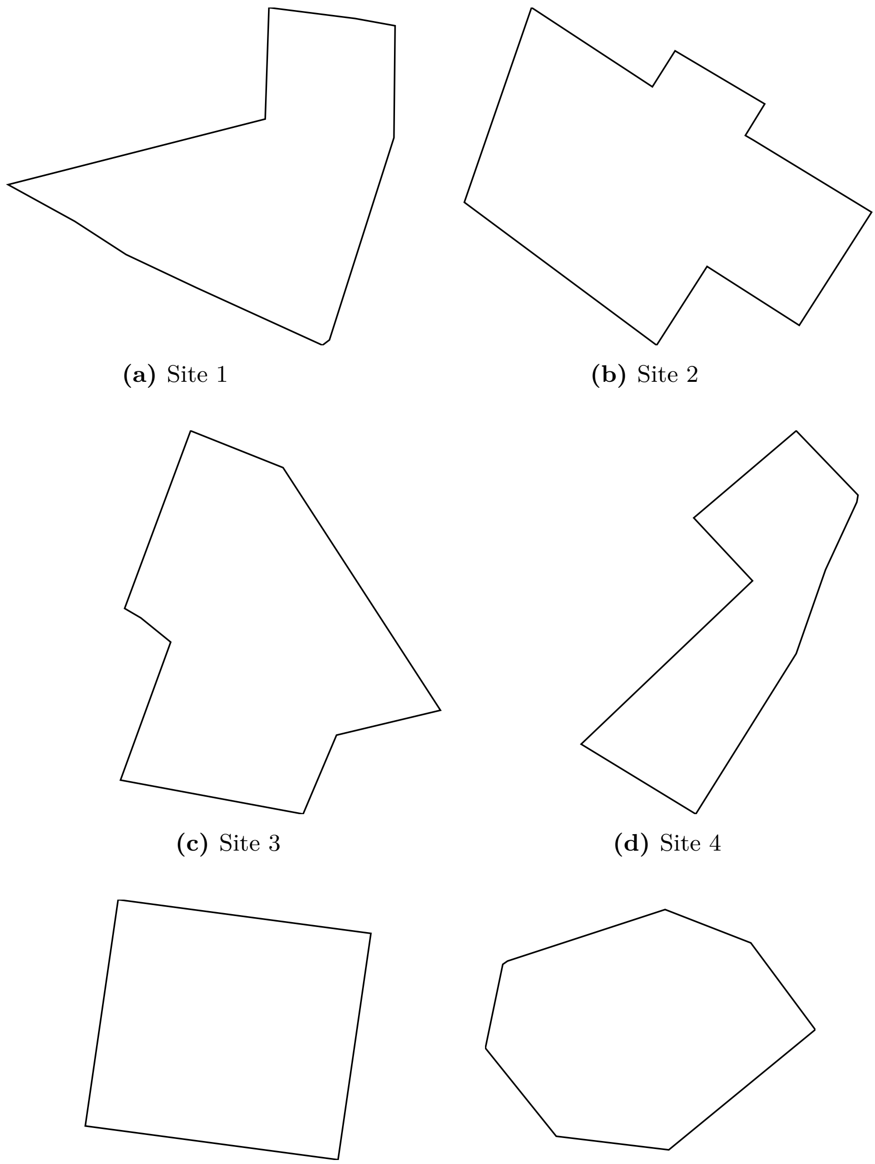
Table 6.2: Details of the computer hardware and software used

Server	Lenovo NextScale nx360 M5
CPU	2x 3,4GHz Intel E5-2643v3 – 6 core
RAM	512 GB
MATLAB version	R2017a 64-bit

6.1 Test instances

Rather than testing the model and solution method on fabricated sites, real sites are gathered for the study. Through *Felles KartdataBase (FKB)* (Geonorge, 2017), Norway's public cartographical series in digital form, all registered sites in Trondheim are collected. Out of the sites, only the sites of area between 5 000 − 10 000 m^2 are kept for the study. From the remaining sites, four non-convex sites and two convex sites are selected manually to be used in the computational study. These sites are chosen to reflect the variation in geometry of real sites. The selected sites can be seen in Figure 6.1. Note that Figure 6.1 does not display the true relative sizes of the sites.

The upper and lower bounds on the length and the width of the buildings, and the fixed width and length of the open zones, are given in Table 6.3, together with the minimum angle point distance. This distance dictates the minimum distance to the edge of the building along the length where the angle of the building can be placed. The bounds on the length of the buildings are determined by best practice, while the bounds on the width are dictated by the common choice to have only one housing unit through the width of the building, using a rule of thumb to ensure enough daylight penetration inside each housing unit (Uytenhaak, 2008). The value of the open zone on the short side (W^O) of the buildings is based on fire safety regulations (PBE, 2012). As the choice should not affect the success of the formulation and solution method, the shortest allowable open zone distance is chosen to study the maximum potential of the sites. The selected length of the open zones on the long sides (L^O) is determined based on a zoning rule that is meant to ensure enough daylight for the apartments of each building (PBE, 2012). The amount of daylight on the facade of a building depends on the height of the surrounding buildings and the orientation of each building. Since the height is not considered in this part of the study, and the orientation of each building is a variable, an approximate and fixed value for the open zones on all the long sides is used. Note that all of these values are approximate and based on best practice, and the exact values (within a reasonable interval) should not affect the general conclusion about the mathematical formulation and solution method.



(a) Site 1 **(b)** Site 2

(c) Site 3 **(d)** Site 4

(e) Site 5 **(f)** Site 6

Figure 6.1: The sites used in the computational study

Recall that the number of buildings on each site is assumed fixed, since it is often a part of the decision maker's specifications. However, to make sure that the optimal number of buildings on a site is tested, an approximate interval for the number of buildings on each site is calculated based on the area of the respective site. According to the values for the maximum and minimum length and width

Table 6.3: Values of parameters except site corner coordinates, which are given by the selected site

Parameter	Value
Maximum length $\overline{L}$	60 m
Minimum length $\underline{L}$	20 m
Maximum width $\overline{W}$	14 m
Minimum width $\underline{W}$	12 m
Long side open zone L^O	15 m
Short side open zone W^O	5 m
Minimum angle point distance K	10 m

given in Table 6.3, the lower bound in the interval is computed as the number of maximally-dimensioned buildings, without an angle, that can be placed within the area of the site such that there is no remaining space. The calculation of the upper bound is identical, except that the buildings are the smallest possible. The calculation assumes that the open zones are a part of the building area, that is, no open zones can overlap or lie outside the site boundary (contrary to the problem description). This leaves less area for saleable area, such that the bounds in the interval underestimates the true bounds in the case where the buildings can be placed perfectly with overlapping open zones internally and outside the site. However, as the shape of the sites often complicate the placement of the buildings, it is highly unlikely that buildings can be placed perfectly, making up for the approximation error in the bounds. The intervals with the number of buildings tested for the sites in Figure 6.1 are shown in Table 6.4, next to the area of each site.

Table 6.4: Area of the sites and the intervals for the number of buildings tested

Site	Area of the site	Minimum number of buildings	Maximum number of buildings
1	6911 m^2	2	7
2	5393 m^2	2	5
3	8292 m^2	2	8
4	8036 m^2	2	7
5	5247 m^2	2	5
6	5180 m^2	2	5

Each test is constructed to test one of the solution approaches or formulations in Table 6.1. A test is performed on the four non-convex sites or on both the non-convex and convex sites, with the number of buildings specified in the interval for that site, given in Table 6.4. 30 generated initial solutions are used for one combination of site and number of buildings.

6.2 Methodology

The utilization of the sites, convergence to feasible solutions and computation time are considered when evaluating the different solution methods. The utilization is computed by dividing the combined area of the buildings by the area of the site. Only the feasible solutions are considered in the utilization computation and included in the mean and maximum utilization. For practical and theoretical purposes, both the mean and the maximum utilization are interesting to study. For the decision maker, the maximum utilization is the most important when the goal is to maximize the total saleable area. If a selection of feasible site layout designs are presented, one may argue that if at least one layout provides a degree of utilization that is acceptable for the decision maker, then the overall problem and solution approach can be considered successful. However, the value of a single maximum might not give a true indication of how well a solution approach generally performs compared to another. Although an average solution in itself is not of much practical value to the decision maker, comparing the mean utilization produced by two different alternatives can help determine which strategy to apply to a new site. The convergence to feasible solutions is computed by dividing the number of times the algorithm produces a feasible solution as the final result, by the number of generated initial solutions (which is 30 for each combination of site and number of buildings). The computation time is given in seconds and is the average of each combination of site and number of buildings, only considering feasible solutions.

To evaluate whether the solution method that provides the highest mean utilization is significantly better than the method that provides the second highest mean utilization, a statistical hypothesis test is conducted. The test evaluates the mean utilization for each combination of site and a number of buildings. To choose a suitable test, the samples are tested for normality as this is a prevalent assumption for some of the common hypothesis tests (e.g. a t-test). *Normal quantile plots* are created from the difference in utilization for two different variable sets. A normal quantile plot is a graph of points where the x-value is from the original set of sample data, and the y-value is the corresponding z-score, that is, the quantile value expected from the standard normal distribution (Triola, 2012). If the sample comes from a normal distribution, the plot should follow a straight line. The plots are presented in Figure 6.2 and show the difference in utilization provided by the two variable sets for two, three, five and seven buildings on site 1. A similar pattern is seen for tests on the other sites and number of buildings. As the plots illustrate, the plotted points (in blue) show a systematic pattern that does not follow a straight line (in red). The pattern indicates that the distribution has thinner tails than the normal distribution. Thus, the normality assumption cannot be applied to the data.

Since the normality assumption does not hold, the *Wilcoxon signed rank test* is used. The Wilcoxon signed rank test is a nonparametric test. That is, the test does not require that the samples come from populations with normal distributions or any other particular distribution (Triola, 2012). The Wilcoxon signed rank test assumes that the two samples are dependent or *paired*. Two samples are consid-

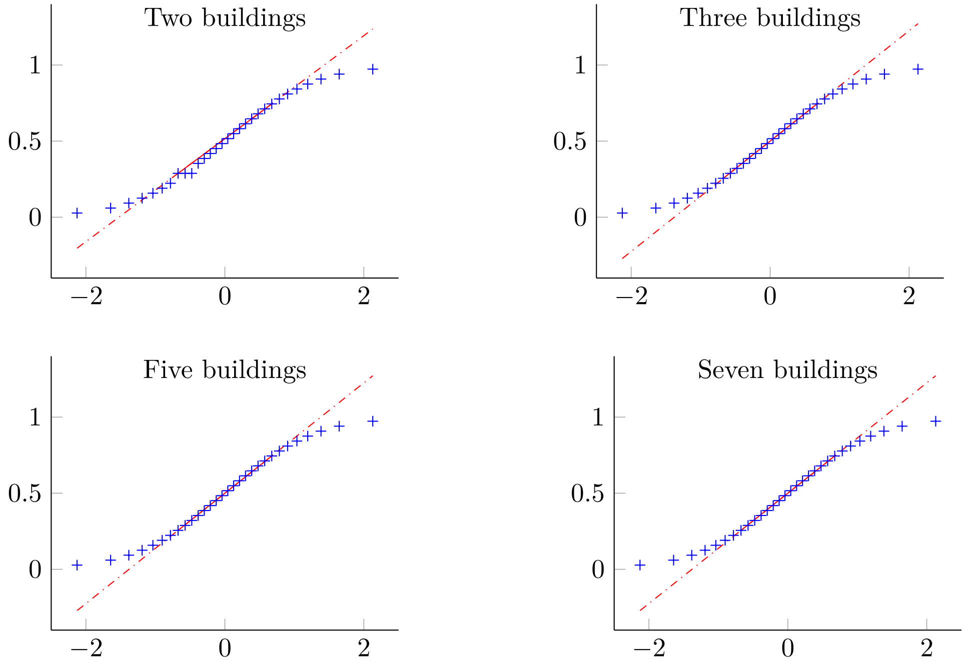


Figure 6.2: Normal quantile plots

ered paired if the samples are based on some inherent relationship (Triola, 2012). This is reasonable to assume for the utilization of the sites when testing two different solution methods because the only difference between two tests is the solution method to be tested, since the same 30 initial solutions are used for both methods. The Wilcoxon signed rank test also assumes that the population of differences has a distribution that is approximately symmetric. This is acceptable to assume in this case. The null hypothesis of the Wilcoxon signed rank test is that the populations of the matched pairs have differences with a median equal to zero (Triola, 2012). The level of significance in the Wilcoxon signed rank test is set to 0.05 by convention. The test is performed on the solution method that provides the highest average utilization and the solution method that provides the second highest average utilization, as the Wilcoxon signed rank test compares the two to assess whether one solution method has systematically larger values than another.

The Wilcoxon signed rank test cannot be applied to extreme values of a sample. To determine if the highest maximum utilization for a given site and number of buildings is better than the second highest maximum utilization, a value defining the minimum difference between two solutions is chosen. The value is set to 0.01. If the difference is larger than 0.01, the highest maximum value is regarded as considerably higher than the second highest maximum value. If not, the difference in utilization is too small to be considered important. Recall that sites between

5 000 – 10 000 m^2 are considered in this study, so a value of 0.01 equals an area between 50 – 100 m^2 depending on the area of the site. The size of a normal sized apartment lies within this interval, motivating the choice of 0.01 as it represents a difference of one apartment (per story of the building) between two highest values.

6.3 Alternative variable sets

In the following, the alternative building variable sets are tested and compared. To isolate the effect of a change from one variable set to another, the sets are compared to the original one using the same randomly generated initial solutions. The test is performed on all of the non-convex sites, with the number of buildings varying within the interval for each site as specified in Table 6.4. The convex sites are not used in this test as the variable sets apply to the buildings, and should be independent of the shape of the sites and whether they are convex or non-convex. However, all four of the non-convex sites are used in the test to provide more robust results. The variable sets tested were introduced in Section 5.3 and are for convenience repeated in Table 6.5.

Table 6.5: The different variable sets tested

Number	Variable set
1	Original set
2	Mid-width points
3	Two corner points
4	Length vector
5	Mid-width vector

6.3.1 Utilization

The results for the mean and maximum utilization of the sites are shown in Table 6.6 and Table 6.7, respectively. The numbers for the variable alternatives correspond with those in Table 6.5. If an utilization value is considerable higher than the utilization yielded by the second highest alternative according to the statistical significance test for the mean, or the minimum difference for the maximum utilization, the result is marked in blue. The tests are performed for each combination of site and number of buildings. If none of the numbers in a row are marked, the best mean or maximum utilization is not considerably better than the second best.

As seen in Table 6.6, variable alternative 5, mid-width vector, provides the significant highest mean utilization in most of the cases where the Wilcoxon signed rank test yields a significant result. The mean utilization is significantly best for variable alternative 5 in almost one third of the cases. The considerable highest maximum

Table 6.6: The mean utilization with different variable sets, the best result (if significant) in each row is marked in blue

Site	No. of buildings	Variable alt.1[%]	Variable alt.2[%]	Variable alt.3[%]	Variable alt.4[%]	Variable alt.5[%]
1	2	27.5	28.6	27.2	29.1	29.0
	3	38.5	38.6	35.2	39.6	40.2
	4	41.7	43.1	41.1	41.9	43.1
	5	42.8	43.6	40.9	43.5	41.2
	6	43.7	43.0	37.6	39.8	0.0
	7	43.2	41.4	38.2	0.0	0.0
2	2	35.3	31.9	32.1	36.4	35.6
	3	42.8	40.1	35.5	39.5	42.9
	4	44.1	42.7	40.4	45.4	47.7
	5	39.9	43.5	40.4	44.6	48.6
3	2	24.7	24.5	24.6	25.0	25.0
	3	35.7	34.0	34.0	35.6	36.1
	4	43.0	40.2	37.1	42.5	41.9
	5	42.2	41.1	40.9	43.0	44.7
	6	43.4	44.0	39.9	42.7	46.6
	7	41.5	44.1	37.5	46.8	44.9
	8	40.3	42.0	38.4	0.0	43.3
4	2	24.8	24.1	23.7	25.1	25.2
	3	34.2	32.9	33.3	33.1	36.3
	4	41.2	40.6	37.7	44.4	44.1
	5	43.3	43.2	39.9	42.0	44.6
	6	44.7	45.8	40.3	45.5	46.2
	7	43.2	43.7	41.5	0.0	0.0

values marked in blue in Table 6.7 show a different pattern. Here, variable alternative 1, the original variable formulation, provides most of the highest maximum values. However, there are only a few cases in which the maximum value in a row is considerably higher than the second highest, making it difficult to separate the different variable alternatives based these results. For instance, variable alternative 1 provides only two more highest values compared to variable alternative 5. Further, Tables 6.6 and 6.7 do not suggest that any of the variable alternatives provides better results for a particular site or a number of buildings.

Table 6.7: The maximum utilization with different variable sets, the best result in each row (according to the test in Section 6.2) is marked in blue

Site	No. of buildings	Variable alt.1[%]	Variable alt.2[%]	Variable alt.3[%]	Variable alt.4[%]	Variable alt.5[%]
1	2	30.0	30.0	30.0	30.0	30.0
	3	44.6	44.4	44.0	45.0	45.0
	4	47.5	46.3	48.0	45.5	46.7
	5	49.3	49.8	43.7	48.8	46.0
	6	50.0	48.9	48.0	45.2	0.0
	7	47.6	47.9	42.7	0.0	0.0
2	2	38.4	38.4	38.4	38.4	38.4
	3	52.7	46.8	44.0	46.0	49.6
	4	49.2	52.8	50.4	46.9	53.6
	5	45.6	53.2	48.2	49.2	56.4
3	2	25.0	25.0	25.0	25.0	25.0
	3	37.5	37.5	37.5	37.5	37.5
	4	47.3	45.5	44.9	47.6	47.2
	5	46.8	45.5	49.0	45.5	48.1
	6	48.1	47.9	47.2	45.6	48.8
	7	46.2	48.0	46.3	46.8	46.0
	8	46.6	44.9	44.8	0.0	44.5
4	2	26.1	26.1	26.1	26.1	26.1
	3	39.1	38.8	39.1	39.1	39.1
	4	49.7	46.3	46.4	48.1	48.5
	5	48.3	46.5	47.6	46.6	45.7
	6	51.2	50.3	46.2	47.6	52.8
	7	46.5	46.6	47.0	0.0	0.0

To study the relationship between the utilization and the number of buildings on the sites, the maximum utilization is plotted for each site and number of buildings in Figure 6.3. The steep and straight decrease seen for site 1, 3 and 4 for variable alternative 4, and in some cases alternative 5, is due to the fact that these alternatives do not find any feasible solutions for these combinations of site and number of buildings. Apart from this, the variable sets behave quite similar. The increase in utilization for each site and variable set is steepest when the number of buildings is in the lower end of the building interval, before it flattens out and slowly starts to decrease for most of the sites and variable sets. An exception to this observation is site 2, where most of the variable sets reach their peak at the higher end of the building interval. A reason for this may be the shape of site 2. As seen in Figure 6.1b, most of the site's corners are approximately 90° (either inner or outer corner angles), which is not as acute as the many of the corners in the other non-convex sites. Therefore, a significant amount of the open zones of a building can be placed outside the site boundary by placing the angle of the building in a corner. This allows for a high utilization even when the number of buildings is at the end of the building interval. In addition, since the site corners are approximately 90°, the angle of the building can be at its minimum (i.e. 90°), which is the angle that provides the largest building area. In fact, site 2 (combined with variable set 5) obtains a solution with utilization of 56.4 %, the highest utilization of all of the sites. This solution is shown in Figure 6.4.

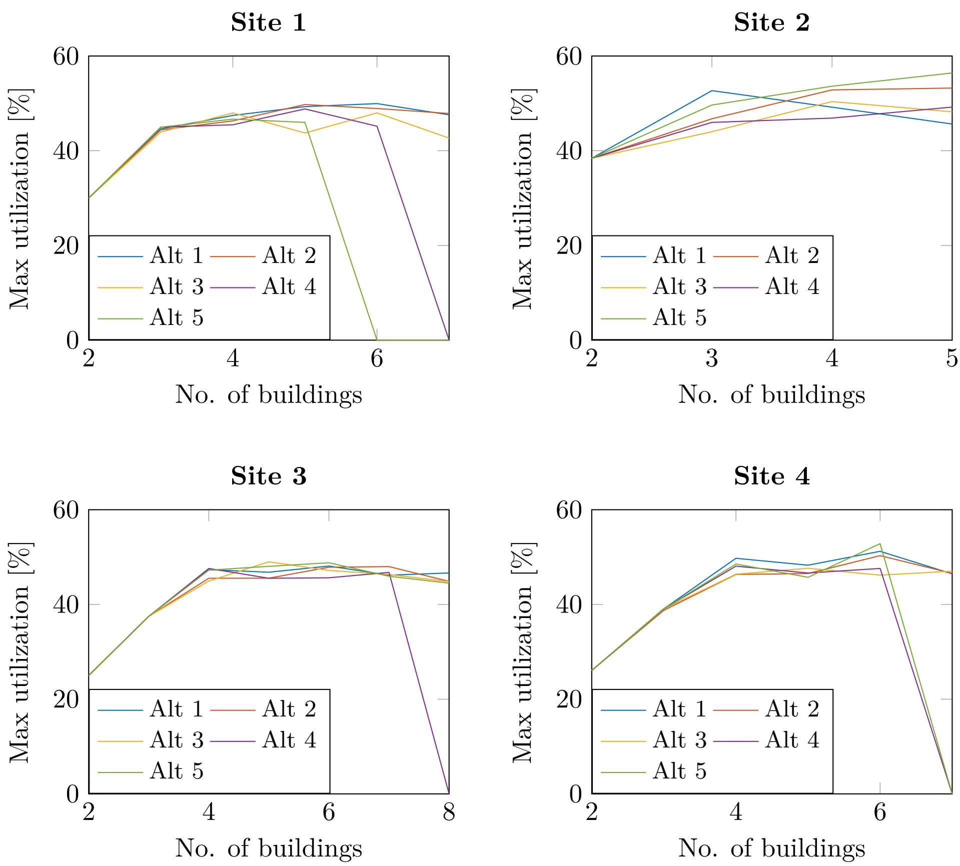


Figure 6.3: Maximum utilization for the different variable alternatives

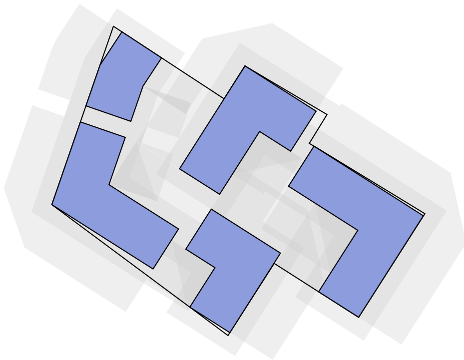


Figure 6.4: The highest utilization obtained for all the variable alternatives and sites - 56.4 % utilization on site 2 with variable alternative 5

Figure 6.3 and Table 6.7 show that all of the variable alternatives find the same maximum utilization on all the sites when the number of buildings is fixed to two buildings. This also happens for three buildings on site 3. This indicates that all the variable alternatives find solutions where the the two (or three) buildings have the maximal area possible. The utilization for these solutions are naturally the highest for the smallest site (site 2) and lowest for the largest site (site 3).

6.3.2 Convergence

Although variable alternative 5 yields the best average utilization for all of the sites, it converges poorly. The convergence to feasible solutions is important to be able to find good solutions in a tolerable amount of time. The convergence to feasible solutions for each variable set and number of buildings are shown for each site in Figure 6.5. The y-axis shows the percentage convergence to feasible solutions. The numbers associated with the different lines refer to the number of the variable set alternative. The graphs reveal a pattern of decreasing convergence as the number of buildings increases. This is compatible with the expected behavior as more buildings lead to less available space on the site, which makes it harder to find feasible solutions. The green and purple line are variable sets 4 and 5, the alternatives using vectors. As seen in all four graphs in Figure 6.5, these two alternatives have the lowest convergence to feasible solutions. This may be due to the more complex formulation of the constraints when vectors are used. The three other variable alternatives are more or less equal, although variable set 3 (the yellow line) generally provides the highest convergence, while variable set 1 (the blue line) provides the lowest of the three.

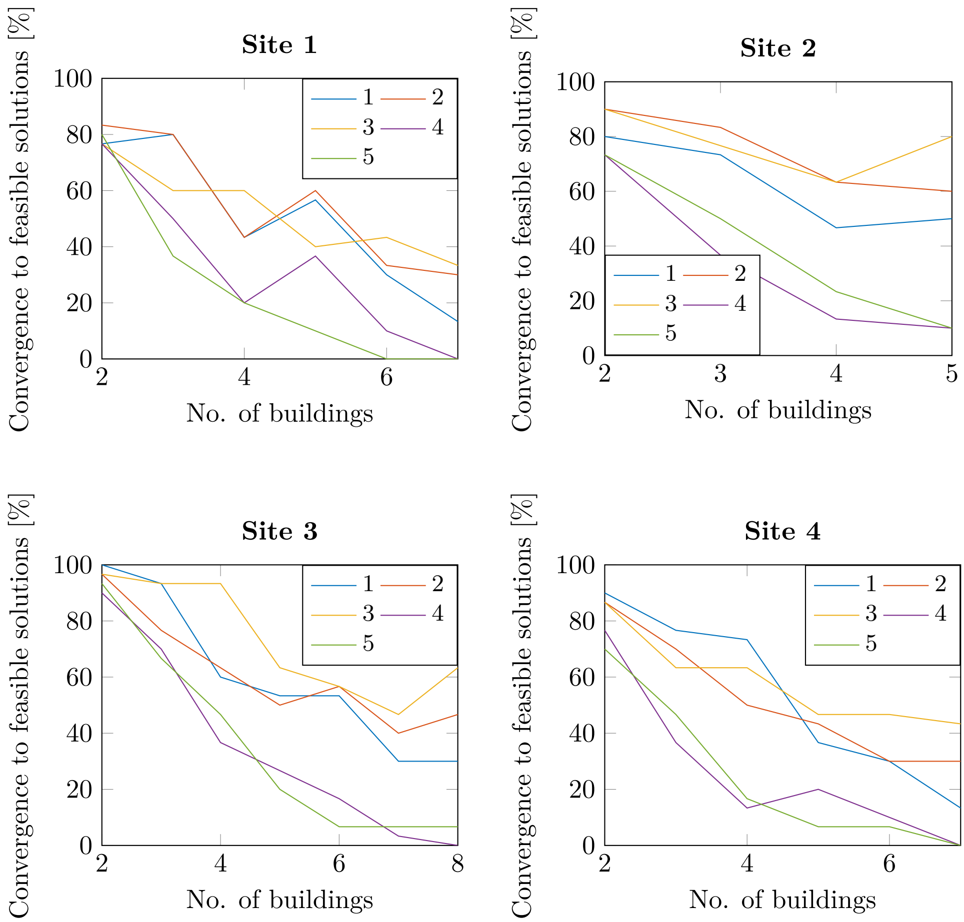


Figure 6.5: Convergence to feasible solution for the different variable sets

6.3.3 Computation time

The average time to reach a feasible solution is the final property tested for comparison of the different variable sets. The time is given in seconds. The time is not included in the average if the solution is infeasible. The results of the average computation time are presented in Figure 6.6. There is one graph for each site. The different variable alternatives are shown on the x-axis, where each column is divided into the different number of buildings tested for the site. The average time is shown in the y-axis. Since the average time only includes the feasible solutions, the graph can be misleading if one tries to compare the height of each column. For instance, variable sets 4 and 5 (the vector formulations) do not find any feasible solutions for seven buildings on site 4. Therefore, the yellow boxes are not included in these columns. It does not necessarily mean that variable sets 4 and 5 use less time overall.

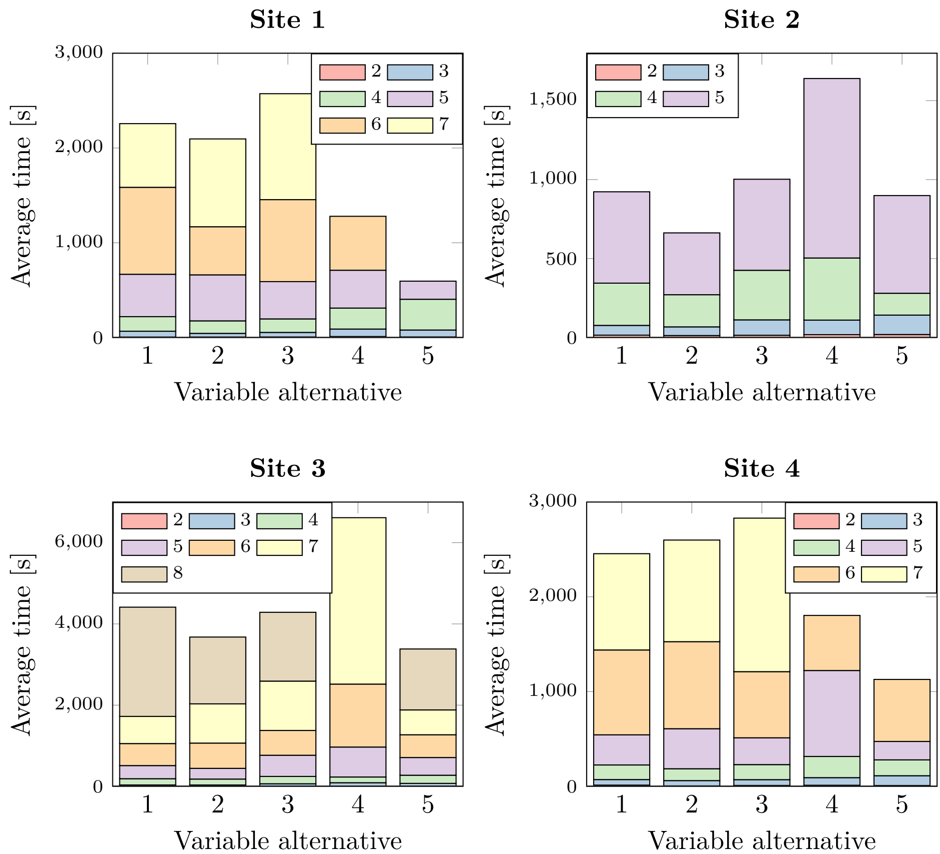


Figure 6.6: Average time for each variable set to reach a feasible solution

A few observations can be made by studying Figure 6.6. First, the average time to find a feasible solution increases with the number of buildings. This is true for almost all of the combinations tested. Again, this is consistent with expectations as it is harder to find feasible solutions when the number of buildings increases. This also influences the time used for each site, where the variable alternatives clearly use less time on site 2 which at most has five buildings, compared to the largest site, site 3, which has eight buildings at the most. Second, when comparing the size of the columns to find which variable set that uses the least amount of time, no clear pattern is found. Variable set 5 generally has a low column, but this is mostly due to the fact that the highest number of buildings are not included as it does not find any feasible solutions. Thus, no conclusion can be made about which variable set is the best with regards to the computation time.

To conclude, it is not clear which variable set is preferable. The mid-width-vector alternative (alternative 5) yields the highest mean utilization, but the convergence to feasible solutions is too low to produce results within a satisfactory computation time. The convergence is also quite low for the other vector alternative, the length-vector formulation (alternative 4). Studying the maximum utilization indicates that variable alternative 1, the original variable set, is slightly better than the others. The computation time does not provide enough information to separate the alternatives. Since a clear conclusion is hard to provide, a choice is made to continue using the original variable set (alternative 1) in the remaining part of the computational study.

6.4 Initial solutions

In this section, the initial solution strategies presented in Section 5.4 are tested. All of the six sites, both the non-convex and the convex sites are used in the tests of the initial solution strategies. The original variable set is used in all the tests.

In Table 6.8, the mean and maximum utilization are compared for the three different initial solution strategies. The solution strategy that places angled buildings in the site corners yields the highest utilization, both in terms of mean and maximum utilization. There are not many statistically significant results for the mean utilization on the non-convex sites (site 1-4), but for the convex sites, the angle-in-corner-strategy provides the significantly highest mean for 75 % of the combinations tested. In terms of the maximum utilization, there are more considerably best results on the non-convex sites, but all sites support the conclusion that the angle-in-corner-strategy is the best. In fact, this strategy provides a considerably highest maximum utilization for more than half of the combinations tested. On average, an increase of 2.5 percentage points can be obtained using the angle in corner strategy compared to both the random and open zones strategy, which provide approximately the same maximum utilization on average. For sites between 5 000 - 10 000 m^2, this amounts to an area of 125 - 250 m^2 more saleable floor area per story of a building. Examining the individual sites, even larger differences are found. For instance, the maximum utilization attained by the angle in corner strategy and the strategy providing the second highest utilization (the random strategy) on site 3, have a difference of 4 percentage points. This equals a floor area of 332 m^2 per story of a building.

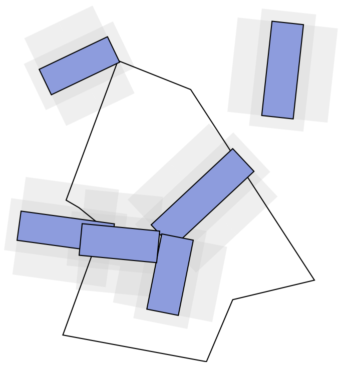
An illustration of some of the results obtained from the different initial solution strategies are shown in Figure 6.7. The figure shows the initial solutions that produce the maximum utilization on site 3 for the three strategies. The random generation of initial solutions and the open-zones-outside-strategy obtain a maximum utilization on site 3 with six buildings, while the strategy that places the angled buildings in the site corner points provides the maximum utilization with five buildings.

The convergence and computation time of the different initial solution strategies are also compared. The convergence to feasible solutions is similar for the three strategies and does not add any further insights. The comparison of the computation time is more interesting to study. The computation time is plotted in Figure 6.8 for each site. The heights of the columns show the mean computation time for the given strategies (the strategies are labeled on the x-axis), and the different colors of the boxes are the different number of buildings. As observed when testing the alternative variable sets, the average computation time increases as the number of buildings increases. In addition, a pattern is revealed for the three smallest sites, site 2, 5 and 6. Here, the random generation of initial solutions clearly has the highest computation time, followed by the open-zones-outside-strategy, while the angled-building-in-corner-strategy has the lowest computation time on average. This can be explained by the fact that the random initial solution strategy places the buildings randomly, possibly far from any (local) minimum, such that the solver requires some time to find a good solution. The strategy that places angled buildings in the site corners may possibly start with an initial solution close to a (local) minimum, which could explain the low computation time and the high mean and maximum utilization seen in Table 6.8. However, the same pattern is not observed for the larger sites, site 1, 3 and 4.

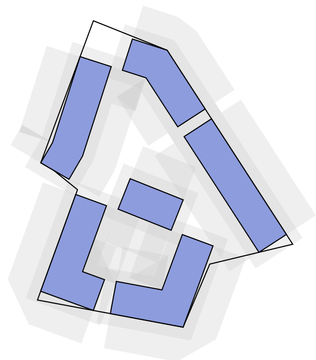
The considerable increase in utilization attained by the angled-building-in-corner-strategy can lead to a notable increase in the monetary value of a site. In addition, the computation time on small sites is shorter when using this strategy. Moreover, the convergence is similar to the different strategies. Thus, a conclusion can be made that the angle building in corner strategy provides the best results.

Table 6.8: The utilization of the initial solution strategies, the best results (according to the tests in Section 6.2) are marked in blue

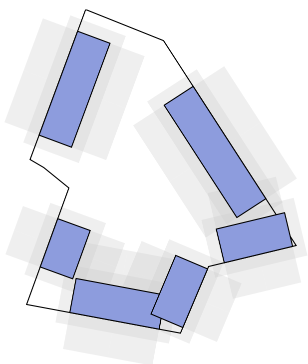
		Mean utilization [%]			**Maximum utilization [%]**		
Site	No. of buildings	Random	Open zones	Angle in corner	Random	Open zones	Angle in corner
1	2	27.5	29.7	29.7	30.0	30.0	30.0
	3	38.5	40.5	40.3	44.6	45.0	45.0
	4	41.7	43.8	44.5	47.5	48.7	52.4
	5	42.8	43.1	44.2	49.3	48.6	50.8
	6	43.7	42.1	44.9	50.0	48.3	50.7
	7	43.2	42.3	43.1	47.6	48.6	46.3
2	2	35.3	36.7	37.0	38.4	38.4	38.4
	3	42.8	41.8	47.6	52.7	49.3	53.9
	4	44.1	42.6	49.3	49.2	49.5	55.3
	5	39.9	41.9	49.4	45.6	48.3	54.5
3	2	24.7	24.8	25.0	25.0	25.0	25.0
	3	35.7	35.6	37.1	37.5	37.5	37.5
	4	43.0	43.0	44.1	47.3	46.7	48.4
	5	42.2	41.2	46.6	46.8	45.6	52.1
	6	43.4	41.9	45.1	48.1	47.1	50.0
	7	41.5	42.2	46.8	46.2	45.2	50.3
	8	40.3	40.6	44.3	46.6	42.9	49.2
4	2	24.8	25.7	25.7	26.1	26.1	26.1
	3	34.2	36.5	37.5	39.1	39.1	39.1
	4	41.2	42.3	45.9	49.7	50.0	52.1
	5	43.3	44.1	49.4	48.3	48.7	56.4
	6	44.7	45.1	48.6	51.2	52.3	55.4
	7	43.2	43.2	49.6	46.5	49.0	54.5
5	2	37.6	37.0	39.5	39.5	39.5	39.5
	3	46.1	43.3	54.4	53.1	52.0	54.7
	4	46.4	46.1	57.1	55.2	58.9	57.1
	5	44.1	42.6	59.1	53.2	51.4	61.8
6	2	38.6	39.3	39.7	40	40	40
	3	43.1	44.8	48.6	49.5	49.2	51.9
	4	41.2	44.4	47.6	49.6	48.3	50.4
	5	41.8	43.1	46.6	47.2	49.3	50.3



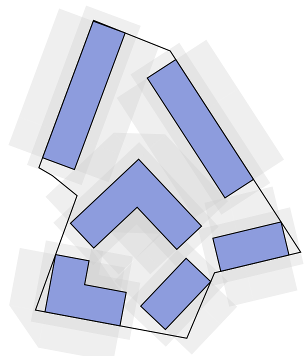
(a) Initial solution - random generation



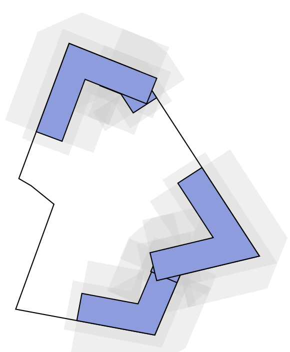
(b) 48.1 % utilization after optimization starting from 6.7a



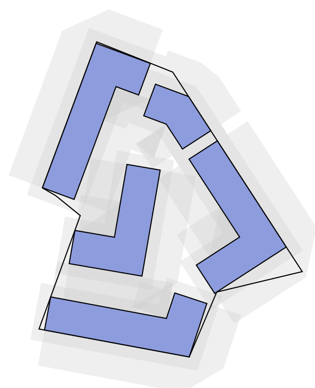
(c) Initial solution - open zones outside



(d) 47.1 % utilization after optimization starting from 6.7c



(e) Initial solution - angled building in site corners (two buildings under the others)



(f) 52.1 % utilization after optimization starting from 6.7e

Figure 6.7: The highest utilization obtained for each initial solution strategy on site 3

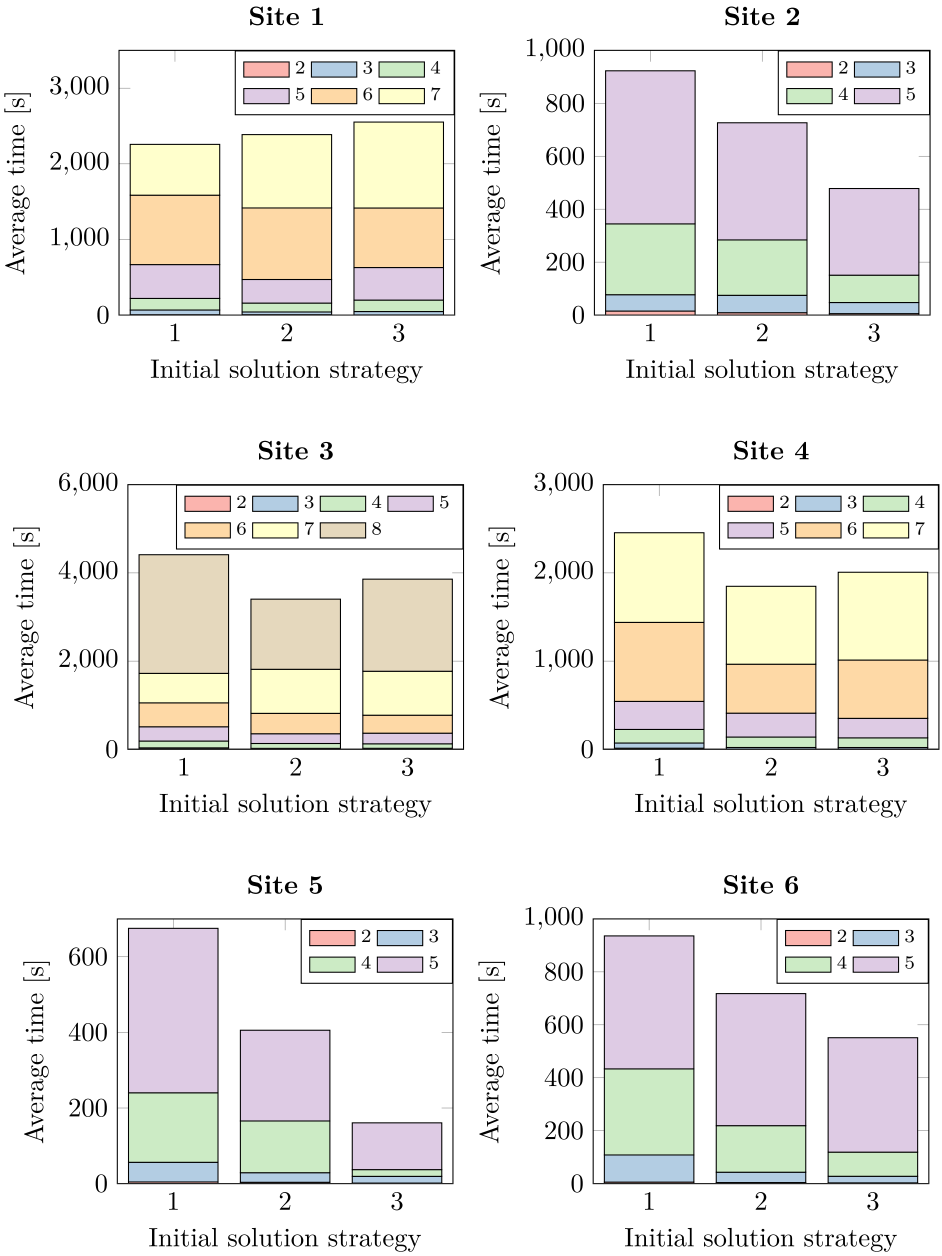


Figure 6.8: Average time to reach feasible solutions for the three different initial solution strategies

6.5 Rectangular versus angled buildings

Allowing an angle in the building footprint increases the flexibility in the layout optimization process. However, it also increases the complexity of the model. An interesting comparison is therefore the gain in utilization by allowing angled buildings compared to rectangular buildings as studied in Bratsberg and Mellbye (2017). The model developed in Bratsberg and Mellbye (2017) is utilized for the rectangular buildings on the convex sites, and extended to use on the non-convex sites. The original variable set is used for both rectangles and angled buildings, though less the angle point and size of angle for the rectangles. Moreover, the random initial solution generation is used for both. In order to make the comparison fair, the maximum area possible to achieve for a building is equal for both the angled and rectangular buildings.

As shown in Table 6.9, a significant gain in utilization is achieved by using angled buildings and the increased complexity does not inhibit the solution method from producing dense layouts. Almost one half on the mean utilizations with angled buildings are significantly larger than the mean utilizations with rectangular buildings. The maximum utilization provides even more convincing results, showing that a substantial gain in utilization can be obtained by allowing angled buildings. The largest difference is observed on site 2, where a gain in utilization of 7.3 percentage points is attained by allowing angled buildings (the difference between the maximum values for the rectangular and angled buildings on site 2). This amounts to a floor area of 394 m^2 per story of a building.

The increased complexity introduced by the angled buildings does however increase the computational time substantially. This is illustrated in Figure 6.9. Again, the computation time is considerably shorter for the smaller sites, and it increases with the number of buildings for all sites.

The difference in convergence to feasible solutions for rectangular and angled buildings is not as considerable as the difference in computation time. The convergence to feasible solutions is plotted in Figure 6.10 for all of the sites. The difference in convergence for the two building shapes is highest for site 2, 3, 6 and to some extent site 5, while it is quite similar for site 1 and 4. Generally, the convergence is higher for the convex sites. However, this is true also for site 2. Thus, the convex shape of the sites might not be the reason for a high convergence, but rather the lower number of buildings and smaller area of the sites.

Table 6.9: Mean and max utilization using rectangular buildings versus angled buildings. The best results (according to the tests in Section 6.2) are marked in blue

		Mean utilization [%]		**Max utilization** [%]	
Site	Number of buildings	Rectangular	Angled	Rectangular	Angled
1	2	23.2	23.9	24.3	24.3
	3	32.7	34.9	36.5	36.5
	4	38.3	41.4	44.2	48.3
	5	40.3	41.4	43.7	46.8
	6	40.6	43.2	43.8	48.4
	7	41.0	43.2	45.6	44.6
2	2	26.7	30.7	31.1	31.1
	3	35.4	39.3	41.2	46.7
	4	38.1	41.8	44.7	52.3
	5	38.2	40.0	45.0	50.6
3	2	20.1	20.3	20.3	20.3
	3	28.6	29.7	30.4	30.4
	4	35.2	38.4	39.4	40.5
	5	38.5	41.3	44.1	49.6
	6	38.3	42.3	43.0	46.7
	7	38.7	43.1	43.4	46.1
	8	39.4	41.4	44.2	48.7
4	2	20.1	20.8	20.9	21.1
	3	30.1	30.6	31.4	31.7
	4	37.8	37.9	41.8	42.3
	5	41.9	41.7	47.6	50.0
	6	42.2	43.6	45.9	49.9
	7	44.0	42.6	46.3	45.8
5	2	29.6	30.8	32.0	32.0
	3	37.5	43.8	48.0	48.0
	4	42.8	45.0	51.5	55.7
	5	41.1	44.1	52.3	52.8
6	2	31.2	32.3	32.4	32.4
	3	38.0	43.2	42.5	48.6
	4	37.5	41.9	42.9	48.8
	5	37.4	38.9	43.3	45.7

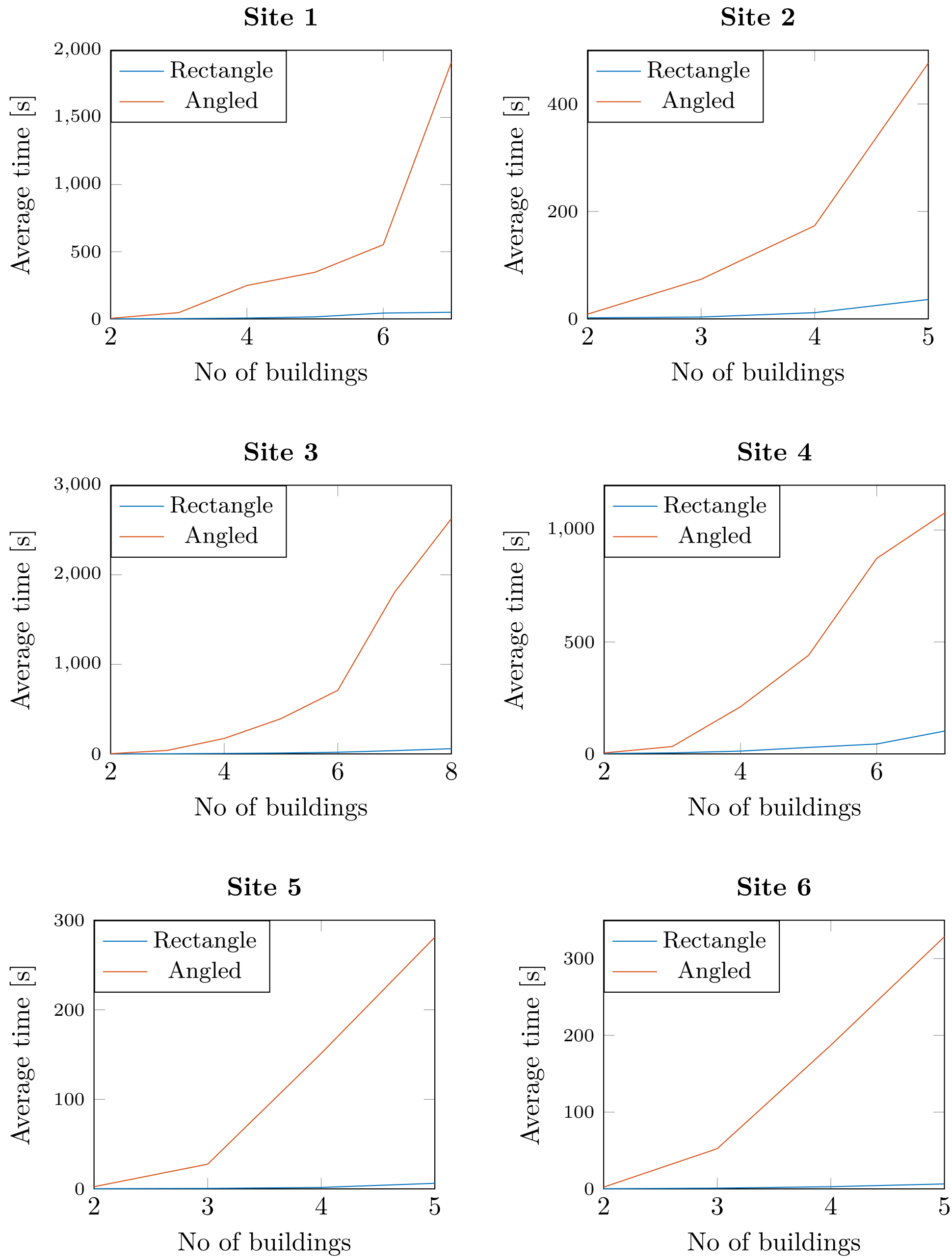


Figure 6.9: Average time to reach feasible solutions, rectangular versus angled buildings

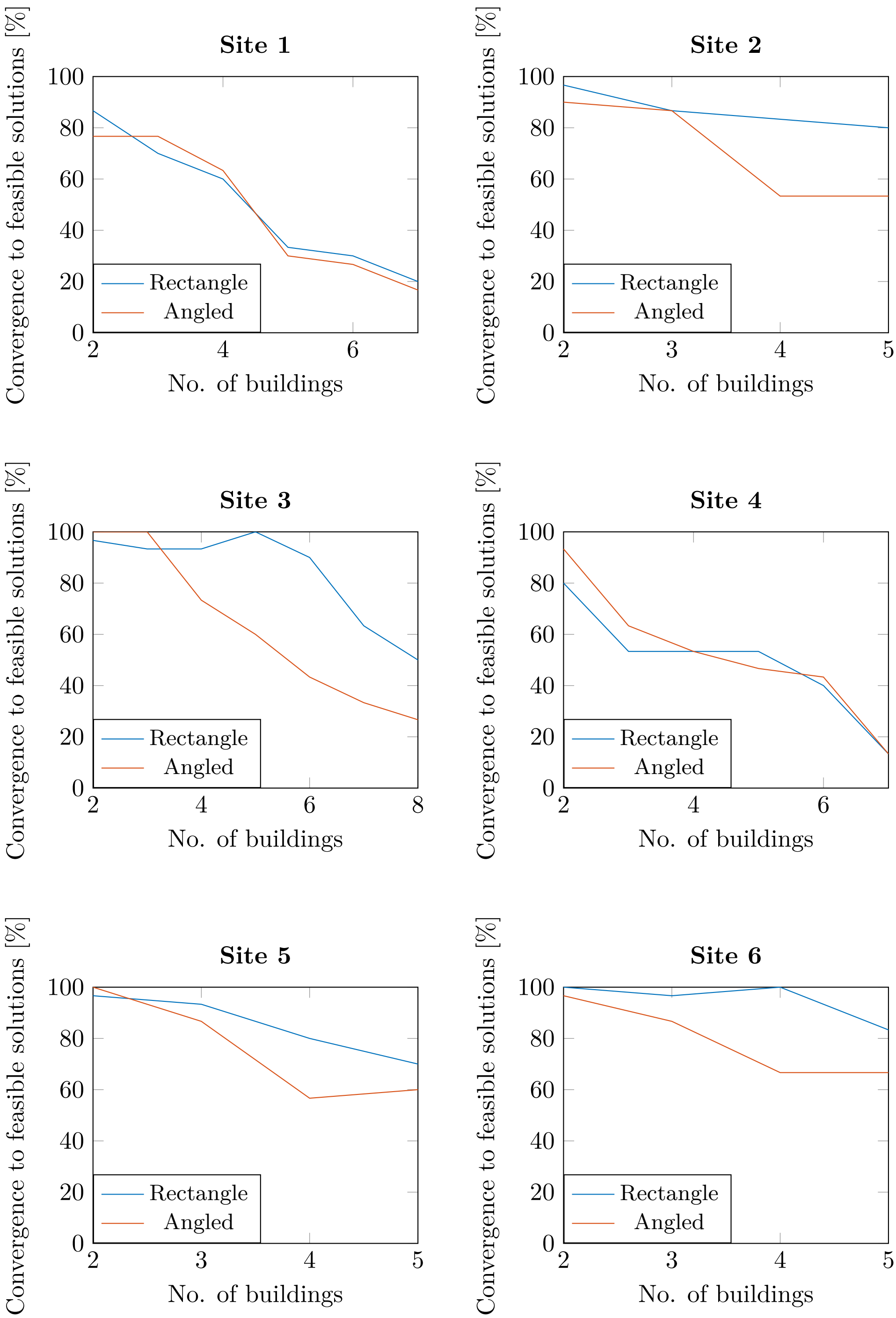


Figure 6.10: Convergence to feasible solutions, rectangular versus angled buildings

6.6 Smooth approximation of overlap constraint functions

SQP requires that the problem to be solved is continuously differentiable, which due to the overlap constraints, the SDP is not. However, as shown previously in this chapter, SQP still provides acceptable results. Nonetheless, it is interesting to compare the quality of the solutions produced when a smooth approximation of the overlap constraints is used, in relation to the original non-continuously differentiable overlap constraints of the SDP. The four non-convex sites are used in the tests, together with the original variable set and the random generation of initial solutions. The results for the mean and maximum utilization are shown in Table 6.10.

As shown in Table 6.10, smoothing of constraints yields higher mean and maximum utilization in almost all cases. In nearly half of the cases, a significant increase in mean utilization can be obtained. For max utilization, smoothing provides considerable increase in more than half of the test cases. Most notably, it increases the overall maximum utilization of each site, by an average of approximately 2.1 percentage points. In practice, this may lead to an increase of 105-210 m^2 of additional saleable floor area per story, for sites between 5 000-10 000 m^2. The maximum utilization obtained using smoothing is seen on site 4 with a utilization of 54.0%. This solution is presented in Figure 6.11.

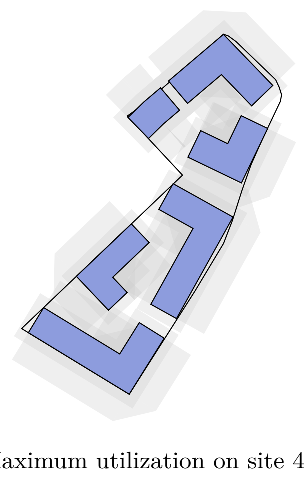


Figure 6.11: Maximum utilization on site 4 using smoothing

If studying Figure 6.11 closely, it can be seen that the buildings in the upper part of the site slightly overlap with the site boundary. When comparing the constraint violation of the exact and smooth approximation formulation of the overlap constraints, it shows that the smooth approximation does indeed allow a larger constraint violation. However, the gain in utilization is larger than the area the constraint violation may contribute with, so that a smooth approximation may produce better results. To counter the constraint violation, one can increase the parameters ϵ until a satisfactory tolerance level for overlap is reached.

Table 6.10: The mean and max utilization of the smooth approximation versus the original formulation. The best results (according to the tests in Section 6.2) are marked in blue

		Mean utilization [%]		**Max utilization** [%]	
Site	Number of buildings	Original	Smooth	Original	Smooth
1	2	27.5	28.4	30.0	30.0
	3	38.5	39.6	44.6	45.0
	4	41.7	43.0	47.5	51.5
	5	42.8	44.2	49.3	51.2
	6	43.7	45.9	50.0	52.9
	7	43.2	44.2	47.6	49.7
2	2	35.3	35.1	38.4	38.4
	3	42.8	41.4	52.7	53.2
	4	44.1	45.8	49.2	53.9
	5	39.9	41.7	45.6	48.5
3	2	24.7	24.6	25.0	25.0
	3	35.7	36.3	37.5	37.5
	4	43.0	41.0	47.3	46.9
	5	42.2	42.9	46.8	49.6
	6	43.4	43.0	48.1	48.6
	7	41.5	43.5	46.2	48.8
	8	40.3	40.9	46.6	48.5
4	2	24.8	24.7	26.1	26.1
	3	34.2	36.6	39.1	39.1
	4	41.2	43.1	49.7	52.1
	5	43.3	44.9	48.3	51.1
	6	44.7	47.1	51.2	54.0
	7	43.2	44.7	46.5	47.3

Figure 6.12 shows that the use of smooth constraints leads to more feasible solutions than exact constraints in nearly all cases. With smooth constraints, the average share of feasible solutions is approximately 19 percentage points higher than with exact constraints. SQP may struggle to converge around non-differential points in the solution space possibly leading to too many iterations or termination. However, as demonstrated in Section 5.2, these points should not be frequent and the difference in convergence is larger than expected and than what can be explained.

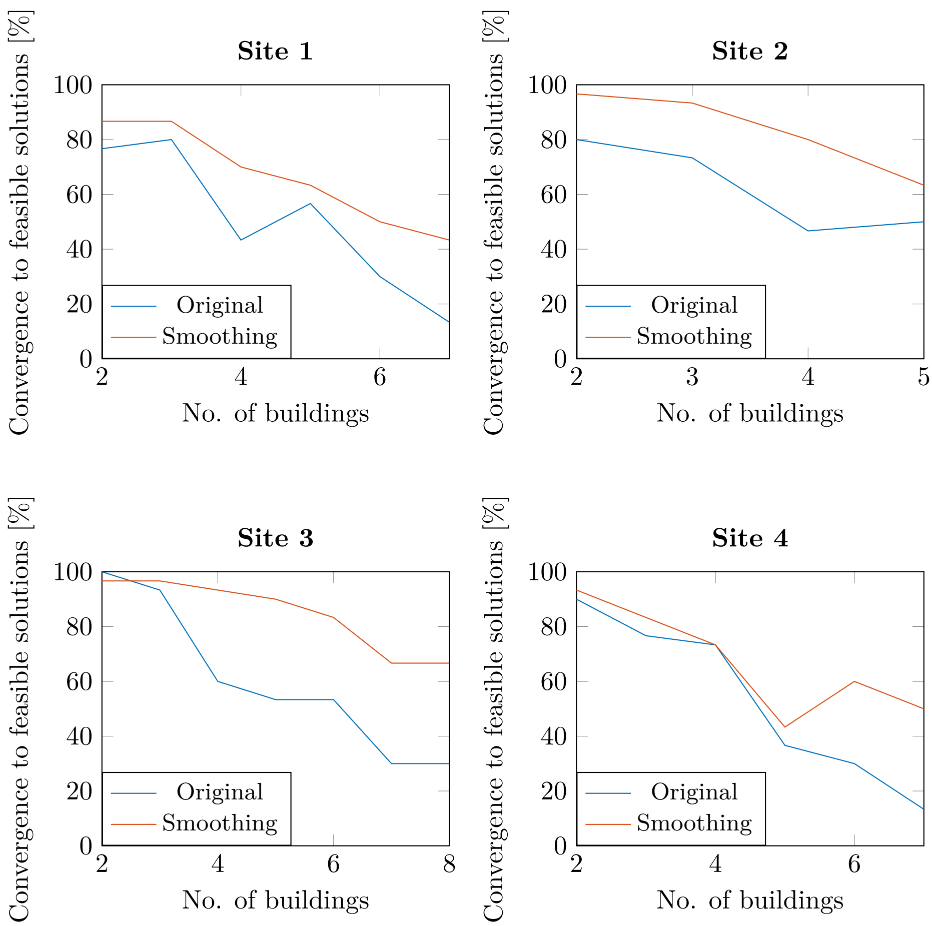


Figure 6.12: Convergence to feasible solutions, smoothing versus the exact formulation

Figure 6.13 shows that smoothing leads to longer convergence time than exact constraints. Although it is difficult to determine the cause, it may be due to a source of error in the computing cluster used to solve the model. The tests with the smooth formulation of overlap constraints were conducted later in time than the tests on the exact model. Increased computing traffic in this period may have caused an increase in the computation time.

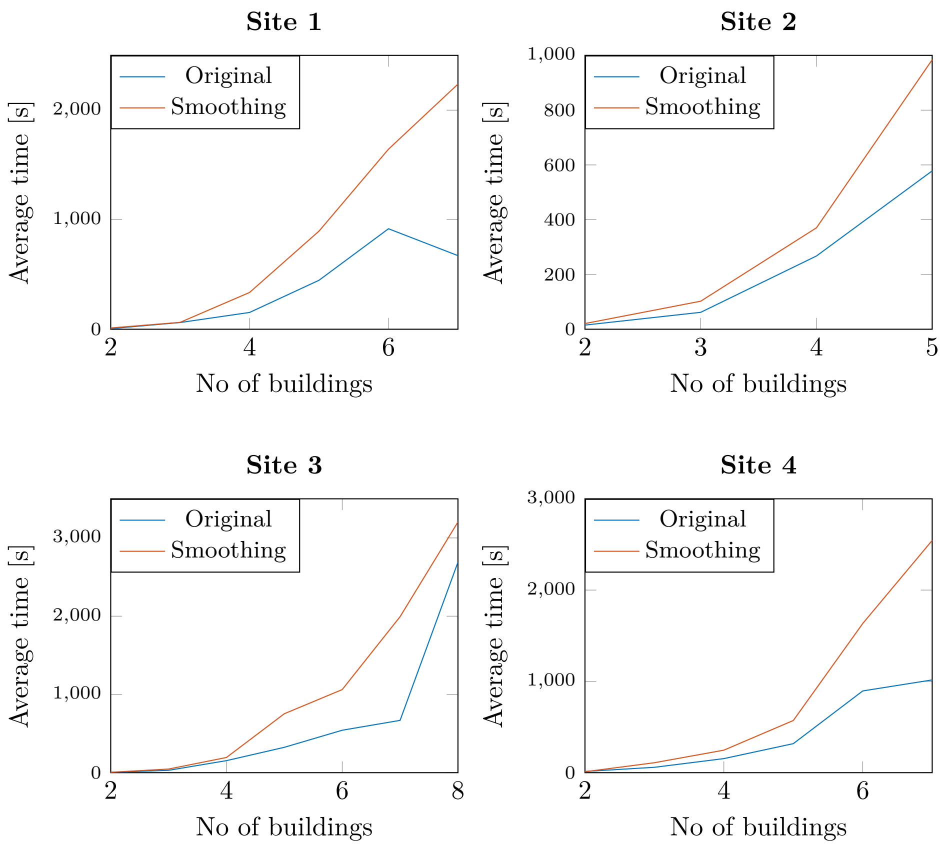


Figure 6.13: Average time to reach feasible solutions, smoothing versus the exact formulation

6.7 The highest utilization obtained on each site

The results shown could be different if different combinations had been tested. All the tests for the variable sets use the random generation of initial solutions, while all the initial solution strategies use the original variable set. For instance, an interesting combination to test could be the mid-width-vectors (variable alternative 5) together with the angled-building-in-corner-strategy. The mid-width-vectors set provides most of the highest mean utilization values but convergences poorly. Combined with a better initial solution strategy than the random approach, the convergence may improve. Nevertheless, a choice is made to test only some of the combinations to avoid all potential combinations of variable sets, initial solution strategies, shapes of buildings and formulation of the overlap constraints. The solutions with the highest utilization produced for each site with the combinations tested in this computational study are presented in Figure 6.14. The solutions using the smooth approximation of the overlap constraints are not included since these allow some constraint violation.

It is interesting to note that almost all of the highest utilization values are obtained with the angle-in-corner-strategy as the initial solution strategy. This is consistent with the results seen in Section 6.4. Since this strategy is only combined with the original variable set in the tests, the results shown in Figure 6.14 for site 1 and site 3-6 are produced with the original variable set. Only the result for site 2 is produced with the random generation of initial solutions and variable alternative 5.

(a) Site 1 - A utilization of 52.4 %

(b) Site 2 - A utilization of 56.4 %

(c) Site 3 - A utilization of 52.1 %

(d) Site 4 - A utilization of 56.4 %

(e) Site 5 - A utilization of 61.8 %

(f) Site 6 - A utilization of 51.9 %

Figure 6.14: The solutions with the highest utilization obtained for each site out of all the combinations tested with the exact formulation

Part II

The 3D problem: Maximizing the total volume of angled buildings on a non-convex or convex site

Chapter 7

Problem description II: The 3D problem

The problem addressed in this part of the thesis is still the Site Development Problem (SDP). In Part I, the aggregated footprint area of the buildings was considered a proxy for saleable square meters. This is a reduction of the problem as it does not consider building heights. In this part, the SDP is extended into three dimensions by introducing building heights. The problem is still a single-objective problem, now with the goal to maximize the total volume of the buildings.

7.1 Site

The site is defined as in Part I, by a set of coordinates that constitute its boundary, which can be either a convex or non-convex polygon. With the introduction of building heights, the site is divided into different *height zones*. These height zones are sub-areas of the site and dictate the maximum height of the building that is placed within, or intersects with, the zones. If a building overlaps with two or more height zones, the lowest of the zones determines the maximal height. A site divided into height zones is illustrated in Figure 7.1. In addition to height zones, a maximum allowable mean height on the site is imposed on the buildings.

7.2 Buildings

The footprint shape of the buildings is defined as in Part I, and the number of buildings on each site is considered fixed. The height of a building assumed to be continuous and the same over the whole area of the building. A minimum bound on the height is set according to a normal residential building story height (Uytenhaak, 2008). An example building is illustrated in Figure 7.2.

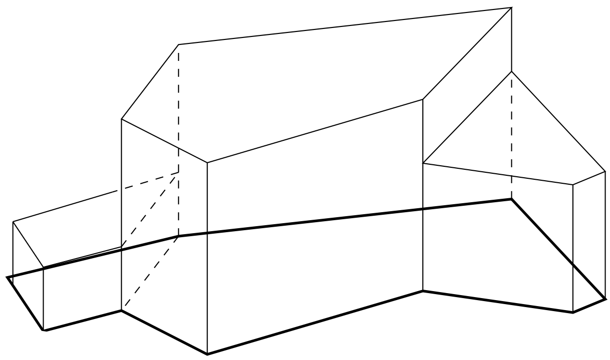


Figure 7.1: A site with three different height zones

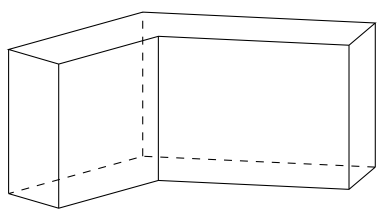
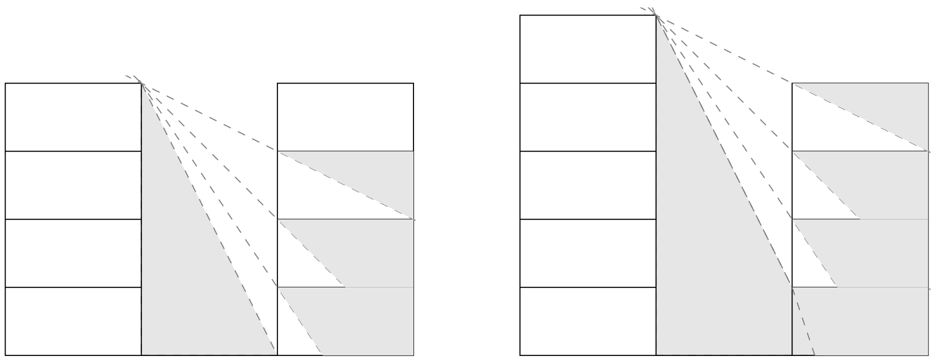


Figure 7.2: A building

7.3 Zoning regulations

As in Part I, additional unoccupied space is enforced around the facade of each building. Ensuring adequate natural light inside the buildings and enough sunlit outdoor area are important reasons for requiring additional open space between the buildings. Naturally, the amount of daylight accessible to each building and sunlit outdoor area are highly dependent on the height of the buildings that surround it. Therefore, the size of the open zones is adjusted depending on the height of the buildings. This is illustrated in Figure 7.3, in which a building casts a shadow in the direction of another building, and the dashed lines show the light penetration into the second building. Figure 7.3a shows a feasible solution where the light penetration into the ground floor of the second building is acceptable. Figure 7.3b exemplifies what happens if one more story is added to the first building without increasing the distance between the buildings. The ground floor on the second building does not receive sufficient daylight, and the buildings have to be moved further apart in order to correct this. The shadow of the building is dependent on where the sun is relative to the building, which will change throughout the course of the day and throughout the year, and with the placement and orientation of the building. Regulations are developed specifically for different places in the world, measuring whether the outdoor area is sunlit more than a certain amount of time on a specified date (PBE, 2012).



(a) Natural light penetration of a building next to a four story building

(b) Natural light penetration of a building next to a five story building

Figure 7.3: Obstruction of natural light penetration by neighboring buildings

7.4 Objective

In practice, the objective is to maximize the saleable floor area, which is the combined area of all the apartments in the housing layout. The combined volume of the buildings is used as a proxy for this area.

Chapter 8

Mathematical model II

In this chapter, a formulation of the extended SDP with heights of the buildings is presented. First, the notation used is presented. Second, a high-level formulation of the problem is given.

8.1 Assumptions and simplification

8.1.1 Site

Like before, the site is assumed to be represented by a set of corner points and can be either convex or non-convex. As a simplification, the site is assumed to be flat. The site is divided into height zones, sub-areas of the site with different allowable maximum heights of the buildings. If a building is placed within two or more height zones, the height zone with the lowest allowable maximum height applies. All the buildings has to be placed fully inside the site to produce feasible solutions.

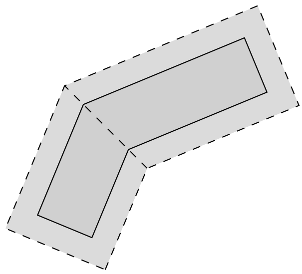
8.1.2 Buildings

The building footprints can be shaped either as rectangles or as angled buildings. A building is assumed to have only one height. A maximum allowable average height for all the buildings is given. The contribution to the average height of one building is independent of its footprint area. Moreover, a minimum allowable height, lower than the mean height, is also given, which applies to all of the height zones.

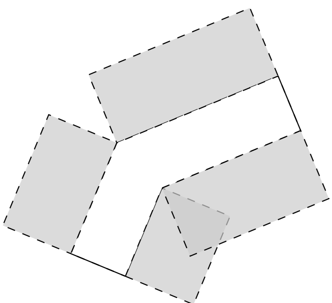
8.1.3 Zoning regulations

For convenience, the illustration of the decomposition of the open zones from Part I is included here. To ensure adequate daylight inside each building and sufficient sunlit outdoor area, the size of the open zones on the long side of each building (Figure 8.1c) and around the angle of the building (Figure 8.1d) increases when the height of the building is increased. The shadow of the building is dependent on where the sun is relative to the building, which changes with the the placement and orientation of each building. Since the latter are variables in the SDP, a simplification is made. A simple rule of thumb stating that the *angle of obstruction* should be no greater than 45 degrees is used in order to find the relationship between the height of the building and the size of the open zones (Uytenhaak, 2008). The angle of obstruction is the angle between the ground and the shadow from the building. Figure 8.2 illustrates the angle of obstruction for a seven story building. The orientation of the building relative to the sun is not considered, so an angle of obstruction of 45 degrees is assumed on all of the long sides of the buildings, including the open zone around the outer angle point of the building. By assuming this angle on all sides, the total open zone area is overestimated. If the orientation of the buildings in relation to the sun is known, some open zones do not need an angle of obstruction as large as 45 degrees. However, this approach guarantees compliance with the zoning regulations. The open zones on the short sides of the building (Figure 8.1b) are required due to fire safety regulations, and do not depend on the height of the building.

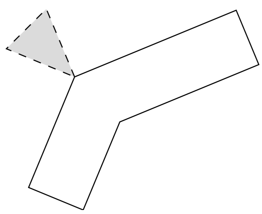
(a) A building and all of its corresponding open zones



(b) Open zones on the short sides, trapezoidal sub-shapes $T^1(\mathbf{v}_b)$ and $T^2(\mathbf{v}_b)$



(c) Open zones on the long sides, rectangular sub-shapes $R^1(\mathbf{v}_b)$, $R^2(\mathbf{v}_b)$, $R^3(\mathbf{v}_b)$ and $R^4(\mathbf{v}_b)$



(d) Open zone around the outer angle point, triangluar sub-shape $A(\mathbf{v}_b)$

Figure 8.1: Decomposition of the open zones

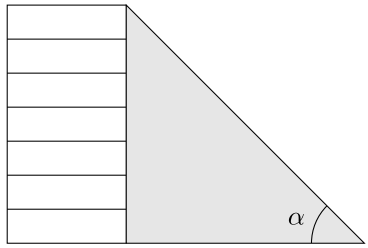


Figure 8.2: The angle of obstruction, α

8.2 Notation

Sets	
$\mathcal{B}$	Set of buildings
$\mathcal{I}$	Set of height zones
Parameters	
$\underline{H}$	Minimum allowable height for all buildings
H_i	Maximum allowable height in height zone i
$\widetilde{H}$	Maximum allowable mean height
N	Number of buildings
Variables	
$w_b \in \mathbb{R}$	Width of building b
$l_b \in \mathbb{R}$	Length of building b
$x_b, y_b \in \mathbb{R}$	Centroid of building b
$\theta_b \in \mathbb{R}$	Rotation of building b
$p_b \in \mathbb{R}$	Angle point of building b
$r_b \in \mathbb{R}$	Angle of building b
$h_b \in \mathbb{R}$	Height of building b
$\delta_{bi} \in$	$\begin{cases} 1, & \text{if building } b \text{ overlaps or is contained within height zone } i, \\ 0, & \text{otherwise} \end{cases}$

Let $\mathbf{v}_b = (x_b,\ y_b,\ w_b,\ l_b,\ \theta_b,\ p_b,\ r_b,\ h_b)$ denote the configuration vector of building $b \in \mathcal{B}$, and $F(\mathbf{v}_b)$ a building footprint with centroid at (x_b, y_b), width w_b, length l_b, building rotation θ_b, angle point p_b with angle r_b, and height h_b. The footprint of the building is defined as in Part I, but the configuration vector has one more variable since the model is extended to include the building height. The variables of the configuration vector are illustrated in Figure 8.3. Notice that all the variables except h_b are defined in the plane.

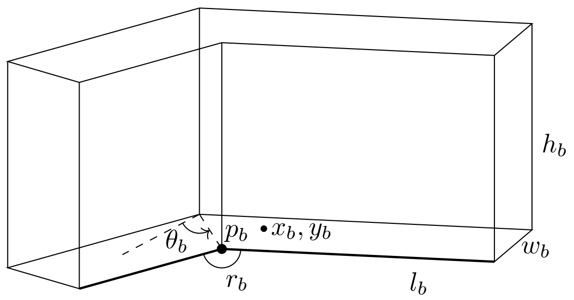


Figure 8.3: The extended building configuration vector

8.3 Model

8.3.1 The objective function

$$\text{maximize} \sum_{b \in \mathcal{B}} h_b \cdot \left(w_b \cdot l_b + \frac{w_b^2}{\tan(\frac{r_b}{2})} \right) \tag{8.3.1}$$

The objective function is the volume of the buildings. The footprint area of each building is calculated as in Section 4.3.1 and multiplied with the height of the corresponding building.

8.3.2 Constraints

All of the constraints from Section 5.1 still hold, that is, constraints (4.4.1)-(4.4.10). The additional constraints added when the SDP is extended with building heights are presented here.

A lower bound on building heights h_b for all $b \in \mathcal{B}$ is included

$$h_b \geq \underline{H} \tag{8.3.2}$$

For each building $b \in \mathcal{B}$ and height zone $i \in \mathcal{I}$ it must be enforced that

$$\text{overlap}(F(v_b), i) \leq M_1 \delta_{bi} \tag{8.3.3}$$

$$h_b \leq H_i \delta_{bi} + M_2(1 - \delta_{bi}) \tag{8.3.4}$$

where M_1 and M_2 are "sufficiently large" constants, and overlap$(F(v_b), i)$ is a function that is positive in case of overlap between a building footprint and a height zone i, and negative otherwise. A mathematical expression for the function is given in Part I, Equation (5.1.1). For each building b, constraints (8.3.3) determine if a building b lies inside or overlaps with a height zone i. M_1 can be set equal to the largest possible building area to ensure that it always is sufficiently large. Constraints (8.3.4) ensure that the height of building b is lower than each of its overlapping height zones. M_2 can be set equal to the height of the maximum height zone for those constraints that involve zones which do not overlap with the building b.

To ensure that the mean height of the buildings is lower than or equal to the maximum allowable mean height, constraint (8.3.5) is added

$$\frac{1}{|\mathcal{B}|}\sum_{b\in\mathcal{B}} h_b \leq \widetilde{H} \tag{8.3.5}$$

8.4 Model properties

The model is given by the following objective and constraints. For the full model, see Appendix A.1.

$$\begin{aligned} \max\ & (8.3.1) \\ \text{s.t.}\ & (4.4.1) - (4.4.10) \\ & (8.3.2) - (8.3.5) \end{aligned}$$

The model has the same properties as the model introduced in Part I since all of the constraints from the model in Part I also apply here. Thus, the model is nonlinear in both the objective and the constraints, and still non-convex due to the non-convex set formed by the constraints. In addition, due to the binary variables introduced in constraints (8.3.3) and (8.3.4), the problem becomes a Mixed-Integer Nonlinear Program (MINLP).

Chapter 9

Solution method II

While the extended model is more complex, there is significant overlap with the two-dimensional model of Part I. In the plane, the only extension is the variable length of the open zones on the long sides of the buildings. As this only changes the size of the open zones, and not the shapes, the solution methods of Part I are maintained for the planar parts of the extended problem. However, the height of a building dictates its possible placements, and binary variables must be introduced to constrain the building from overlapping with lower height zones. The combination of integer and continuous variables presents significant challenges. This is outlined in Section 9.1, where a short introduction to solution methods for general MINLPs are given. Next, a heuristic and a naive solution method for the extended model are presented in Section 9.2 and Section 9.3, respectively.

9.1 Theoretical considerations

The problem introduced in this part of the thesis is a non-convex mixed-integer non-linear program (MINLP). Although integer variables make an optimization problem non-convex (Nocedal and Wright, 2006), a distinction can be made between convex and non-convex MINLPs. For convex MINLPs, the continuous relaxation is convex. Thus, even though both convex and non-convex MINLPs are NP-hard in general, convex MINLPs are far easier to solve than non-convex ones, in both theory and practice (Burer and Letchford, 2012).

Several efficient and exact solution methods have been devised for the convex MINLP based on the fact that the continuous relaxation is convex. Two of these methods include branch-and-bound and outer approximation (Burer and Letchford, 2012). Branch-and-bound is a classical algorithm for solving MINLPs. The basic idea of such methods is to relax the integer requirement of the variables, and approach the optimal solution by introducing constraints, or *branches*, on the variables. The relaxed problem constitutes an NLP problem, and a variety of NLP algorithms are available to solve such problems, for instance SQP utilized in this thesis. Since the continuous relaxation of a convex MINLP is convex, each local feasible solution found is a lower bound (assuming a maximization problem) on the optimal objective value. Therefore, the solution at a node may be used to *prune* branches that have a lower objective function value to avoid enumeration of all solutions (Bonami et al., 2011). In the outer-approximation algorithm, the problem is first solved as an NLP problem where the integer variables are fixed. Due to convexity, the optimal solution found from solving the NLP problem constitutes a lower bound on the optimal value of the original MINLP problem (asssuming it is a maximization problem). Next, a linearization around the optimal solution of the NLP problem is carried out, and the linearized functions are added to the problem. The resulting model is a MILP problem. The MILP problem is solved, and again due to convexity, the optimal solution constitutes an upper bound on the optimal solution of the MINLP. The steps in the solution method is repeated until the difference between the lower and upper bounds is within a specified tolerance (Duran and Grossmann, 1986).

On the contrary, in the non-convex case, the continuous relaxation is in itself a global optimization problem (Burer and Letchford, 2012). As seen in the previous paragraph, both the branch-and-bound algorithm and the outer-approximation method relies on convexity to efficiently exclude certain solutions in order to find the global optimum. The same is not as easy to do for non-convex MINLP problems as there are several locally optimal solutions. Nonetheless, there exist concepts that can be applied to non-convex MINLP problems such that exact algorithms can be applied to find the optimal solution. Some of these concepts require the MINLP problem to have certain properties not applicable to the SDP, such as twice differentiable constraint functions, while others are quite complex (Burer and Letchford, 2012). Therefore, a choice is made to develop a heuristic to solve the non-convex MINLP presented in Chapter 8. The use of SQP on the two-dimensional non-convex SDP solved in Part I produced good solutions despite the non-convexities present, and a lot of information about the problem is known and can be utilized in a heuristic.

9.2 The heuristic solution method

The heuristic developed for solving the extended problem is presented in this section. It is based on solving a series of subproblems where buildings are bound to be contained in certain height zones, i.e. a fixed value for the binary variables δ present in the constraints (8.3.3)-(8.3.4). This is based on the fact that the height of a building is restricted to the lowest of the height zones it overlaps with. Hence, if an optimal solution contains a building with height equal to that of the highest height zone, then this building must be fully contained within that zone. Similarly, a building with height equal to that of the second tallest zone must be in one or both of the two tallest zones. Each subproblem can therefore be defined as the number of buildings that is restricted to each height zone, and can be solved with the methods presented in Part I. Let

$$M(z_i) = \bigcup_{\substack{j \in \mathcal{I} \\ H_j \geq H_i}} z_j \tag{9.2.1}$$

denote the union of height zone z_i, $i \in \mathcal{I}$, with the adjacent zones that are higher than it. If building b is placed in zone i, then constraints (4.4.9) that enforce containment within the site is changed to

$$F(\mathbf{v}_b) \cap \operatorname{conv}(M(z_i)) = F(\mathbf{v}_b) \tag{9.2.2}$$

As it cannot be guaranteed that these zones are convex, the constraints that restrain the buildings from overlapping with the non-convex part of the zones (4.4.10) is also altered to

$$F(\mathbf{v}_b) \cap \big(\operatorname{conv}(M(z_i)) \setminus M(z_i)\big) = \emptyset \tag{9.2.3}$$

These constraints replace the previous site containment constraints (4.4.9)-(4.4.10) and binary zone containment constraints (8.3.3)-(8.3.4). With the new restrictions, the resulting subproblem is an NLP that can solved using SQP and the solution methods presented in Chapter 5.

The task is then to optimally allocate buildings to height zones, solve several different instances of the corresponding subproblem and return the best solution found. However, due to the non-convexity of the subproblems, it cannot be guaranteed that the objective value is globally optimal. Consequently, allocations cannot be conclusively compared and ordered, and the optimal allocation must be approximated iteratively. The idea is to assume that the buildings can be perfectly placed inside the zones, such that there is no unoccupied space. As a first measure, the buildings are distributed with this assumption. Since it is not possible to perfectly place building footprints into height zones due to variable open zones and irregular shapes, such an allocation will most likely be too dense in certain zones. If there is an estimated gain in volume by moving a building from a more dense zone to another, the buildings are re-allocated accordingly and a new set of subproblem instances is solved. This procedure continues until a re-allocation proves to be infeasible, worse than the previous, denser allocation, or until an earlier allocation is produced again. Pseudocode for the heuristic solution method is presented in Algorithm 6.

Algorithm 5: AllocateBuildings

Data: Set of height zones $\mathcal{Z} = \{z_i \ : \ i \in \mathcal{I}\}$, maximum mean height $\widetilde{H}$, number of buildings N, and upper bounds $\overline{N}_i$ on the number of buildings in each zone $i \in \mathcal{I}$, maximal building area $\overline{A}$

Result: Number of buildings N_i in each zone $i \in \mathcal{I}$

begin

    $N_i := 0$ for all $i \in \mathcal{I}$
    $A_i := \mathrm{Area}(z_i)$ for all $i \in \mathcal{I}$
    $h_j := 0$ for all $j \in \{1, 2, \ldots, N\}$
    for $n = 1, \ldots, N$ **do**
        $H := N \cdot (\widetilde{H} - \underline{H}) - \sum_{j=1}^{N} h_j$
        $\widetilde{I} := \{i \ : \ i \in \mathcal{I}, \ N_i < \overline{N}_i\}$
        $k := \operatorname*{argmax}_{i \in \widetilde{I}} \left\{ \min(A_i, \overline{A}) \cdot \min(H, \overline{H}_i) \right\}$
        if $A_k = 0$ *or* $\widetilde{I} = \emptyset$ **then**
            No more space in zones given upper bounds.
            Terminate and return $\emptyset$.
        end
        $N_k = N_k + 1$
        $h_n = \min(H, \overline{H}_k)$
        $A_k = A_k - \min(A_k, \overline{A})$
    end

end

Algorithm 6: HeuristicSolutionMethod

Data: Site corner points $\mathcal{C}$, Set of height zones $\mathcal{Z}$, maximum mean height $\widetilde{H}$ and number of buildings N. Let $M(z)$ denote the union of zone z and the adjacent zones that are taller than $z \in \mathcal{Z}$.
Result: Optimal solution of building footprints and heights V^*
begin
    $V^* = \emptyset$
    $A = \text{AllocateBuildings}(\mathcal{Z}, \widetilde{H}, N, \overline{N})$
    while *$A \neq \emptyset$ and A not previously tested* **do**
        $V_A = \emptyset$
        for $k = 0, 1, \ldots, K$ **do**
            $V_k :=$ Solution to problem in Chapter 8 with (9.2.1)-(9.2.3) and random initial values
            if *feasible and Volume(V_k) > Volume(V_A)* **then**
                $V_A = V_k$
            end
        end
        if $V_A = \emptyset$ **then**
            $j = \underset{j}{\text{argmax}}\{A_j/\text{Area}(z_j)\}$
            $N_j = N_j - 1$
            $A = \text{AllocateBuildings}(\mathcal{Z}, \widetilde{H}, N, \overline{N})$
        end
        else if $V_A, V^* \neq \emptyset$ *and* $V_A < V^*$ **then**
            /* Previous allocation was better */
            Break
        end
        else
            $\mathcal{Z}_s := \mathcal{Z}$ sorted by decreasing maximum height
            for $i = 1, \ldots, N-1$ **do**
                $v_{\min} :=$ Volume of smallest building in zone z_i
                if $v_{min} < \textit{UnusedVolume}(z_{i+1}, V_A)$ **then**
                    $A_i = A_i - 1$
                    $A_{i+1} = A_{i+1} + 1$
                end
            end
        end
    end
    Return V^*
end

9.3 The naive solution approach

To determine if the heuristic method yields good results, a naive solution method is developed for comparison. The naive approach is performed in two steps. First, the model from Part I is used to optimize the layout of the buildings. The original variable set formulation is used together with the initial strategy from Section 5.4 that provided the best results, that is, the building-angle-in-site-corner strategy. The layouts of the feasible solutions are fixed and next the heights are optimized using SQP in MATLAB.

A few alterations are made to the model in Part I to be able to fairly compare the naive approach to the heuristic. Recall that the open zones on the long sides and outside the angle of the buildings depend on the heights of the buildings. In Part I, these are assumed fixed since the heights of the buildings are not considered in the model. To prevent the solution of the model from Part I to dictate the heights based on the size of the open zones, the model is solved with different fixed sizes of the open zones along the length and angle of the buildings.

Chapter 10

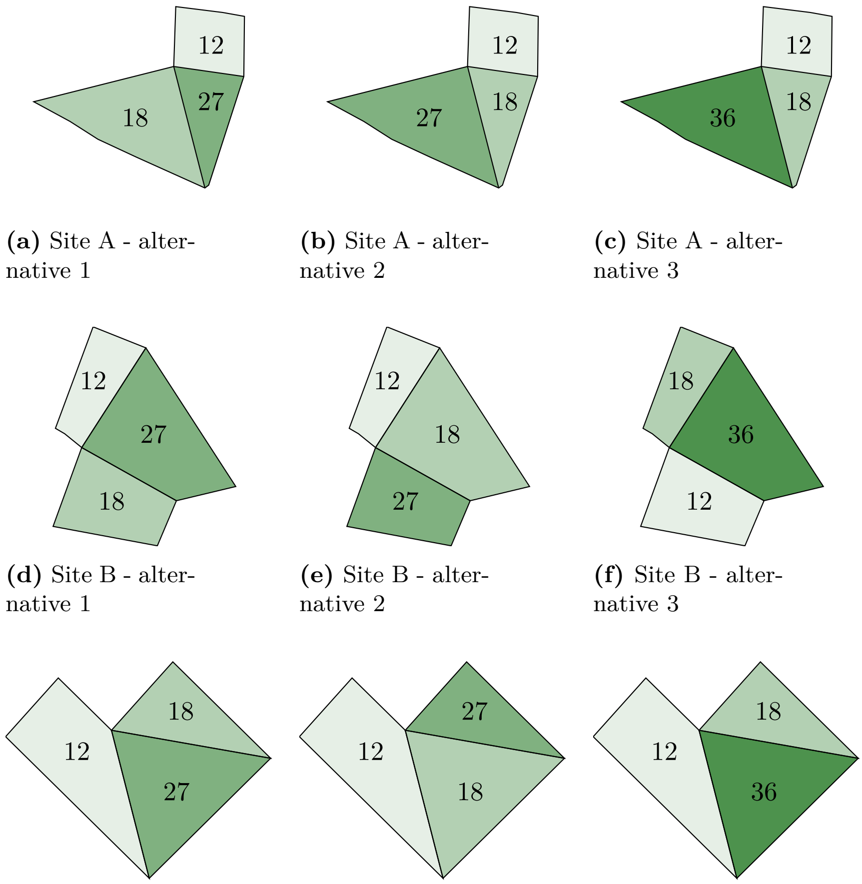
Computational study II

The computational study tests the heuristic method presented in Section 9.2, and compares it to the naive approach introduced in Section 9.3. The aggregated volume of the buildings is studied to evaluate the two solution approaches. The model is solved using MATLAB, and the specifications of the hardware and software used to solve the model are the same as in Part I, presented in Table 6.2.

10.1 Test instances

Like in Part I, real sites are gathered for the study through *Felles KartdataBase (FKB)* (Geonorge, 2017), Norway's public cartographical series in digital form. Only non-convex sites are used in the computational study in this part, but the model developed can also be utilized on convex sites. Site 1 and 3 from the computational study in Part I are used here as well. In addition, a larger site is chosen from the real sites with area between $11\ 000 - 18\ 000\ \text{m}^2$. The sites are divided into different height zones. Three alternative compositions of the maximum allowable heights in the height zones are made for each site. Two of the alternatives have the same maximum allowable heights, but the distribution of the zones within the site is different. One alternative ensures increasing height of the buildings into the middle of the site, as the maximum height zone is placed there. The other alternative allow increasing heights from one side of the site to the other, as the minimum and maximum height zone are placed on separate sides of the site. The last alternative has a larger difference between the maximum allowable heights in the height zones. Figure 10.1 illustrates the sites and the three different alternatives. The sites are illustrated from above, with different colors for the different height zones, and a number indicating the maximum allowable height for the corresponding height zone (in meters).

The number of buildings tested on each site, together with the area of the sites, are presented in Table 10.1. The mean height on each site is also stated in Table 10.1, which is set to 15 meters throughout. For sites A and B (sites 1 and 3 in Part I), the number of buildings tested on each site is reduced compared to Part I based



(a) Site A - alternative 1 (b) Site A - alternative 2 (c) Site A - alternative 3 (d) Site B - alternative 1 (e) Site B - alternative 2 (f) Site B - alternative 3 (g) Site C - alternative 1 (h) Site C - alternative 2 (i) Site C - alternative 3

Figure 10.1: The sites with the three alternative distribution of height zones

on the results seen in Section 6.3. That is, the utilization is the highest when the number of buildings is in the middle to higher end of the building interval. Since this information is available, five and six building are tested on sites A and B to provide the most interesting results and limit the computation time. On site C, the number of buildings tested is fewer than that which the area would maximally allow. Fewer buildings are used to, hopefully, produce more freedom and flexibility in the heuristic.

To fairly compare the heuristic to the naive approach, the naive approach is solved with different fixed sizes of the open zones along the length and angle of the buildings. These are chosen based on the maximum allowable heights in the height zones. Recall that the relationship between the height and the length of the open

Table 10.1: The area, mean height and number of buildings tested on each site

	Area of the site	Mean height	No. of buildings		
Site A	6 911 m^2	15 m	5	6	
Site B	8 292 m^2	15 m	5	6	
Site C	13 401 m^2	15 m	6	7	8

zones is one to one, such that the maximum allowable height in a height zone is also a test case for the fixed length of the open zones. However, 12 meters is smaller than the mean height, which may prevent the buildings from reaching the mean height for the site. For this reason, 15 meters is used as the shortest fixed length of the open zones. In addition, 36 meters, the largest maximum allowable height, is only used as fixed length on site C due to the size of sites A and B being too small. The fixed lengths tested for each site are seen in Table 10.2. To find the best naive solution, all of the feasible solutions found by solving the problem in two dimensions are used when optimizing the heights. 30 generations of initial solutions with the angle-in-corner-strategy are used for each combination of site, height zone, number of buildings and fixed open zone lengths. The solution with the highest objective value (largest aggregated volume of buildings) is chosen as the best solution.

Table 10.2: The fixed length of the open zones on the long sides and around the angle points of the buildings

	Fixed length of open zones			
Site A	15 m	18 m	27 m	
Site B	15 m	18 m	27 m	
Site C	15 m	18 m	27 m	36 m

The best solution obtained using the heuristic is compared to the best solution obtained using the naive approach for each combination of site, number of buildings and height zone alternative.

10.2 The heuristic versus the naive approach

In this section, the heuristic is compared to the naive approach. The maximum aggregated volume of each solution is compared when evaluating the solutions. The maximum volume obtained for the three height zone distribution alternatives (see Figure 10.1) are presented in Table 10.3. The difference in aggregated volume between the heuristic and the naive approach is marked in bold.

As described in Section 10.1, the naive solution method is tested with different fixed open zone lengths. Only the best naive solutions are presented in Table 10.3.

On site A, the best naive solutions are obtained when the length of the open zones are fixed to 15 meters for all combinations of number of buildings and height zone distributions, with one exception. With six buildings and height zone alternative 2, the result when the open zones are fixed to 18 meters is slightly better than the others. On site B, the best naive solutions for five buildings are obtained when the length of the open zones are fixed to 15 meters for all height zone alternatives, while the best solutions for six buildings are obtained when the lengths are fixed to 18 meters for all alternatives. These observations can be explained by the fact that site A is smaller than site B, such that in order for a dense enough packing to occur in the plane, smaller open zones are necessary. The difference in the best open zone lengths for five and six buildings on site B may be explained by the fact that there is more available space on a site when five buildings are placed. Thus, the open zones do not need to be as large in order to retain enough space to increase its height after the initial placement. When the number of buildings (i.e. density) is increased, the open zones need to be larger to ensure that the buildings can increase the heights. Site A and B do not share the same pattern, which may be caused by the different site shapes. One shape may allow for more open zone outside the site, giving more space for building footprint. The fixed open zone length of 27 meters is too large for both sites. The potential in increased heights does not make up for the lower utilization in the plane due to longer open zones. For site C, the best naive solutions are obtained with an open zone length fixed to 18 meters for height zone alternative 1 and 3. For height zone alternative 2, the best solutions are obtained with a fixed open zone length of 15 meters.

As seen in Table 10.3, the heuristic finds a higher maximum volume for all height zones alternatives and number of buildings on sites A and B. The same does not hold for site C. The largest difference between the heuristic and the naive approach is for most of the cases observed on site A. Site A is the smallest site, and thus the most challenging site as the buildings have to be placed closely together to fit on the site, while leaving enough space to increase the heights sufficiently. The naive approach does not consider this balance simultaneously, which may explain the large difference in the approaches for site A. The overall highest difference is found on site A with height zone alternative 3 and five buildings. These solutions are illustrated in Figure 10.2. The figure shows the site from above, with the height of the buildings given by the number on each building. The large number outside the site area is the maximum allowable height in the nearest height zone. The height zones are separated by lines on the site. The difference is 3 851 m^3 between the two solutions, which constitutes a significant amount of saleable area. A large gain in volume is obtained by the heuristic from the building that is 36 meters high. The naive approach has placed the buildings with the largest area in the lower height zones, and does not manage to utilize the maximum height zone in the same manner as the heuristic. Moreover, the buildings placed in the maximum height zone do not have enough surrounding space to take on maximum height.

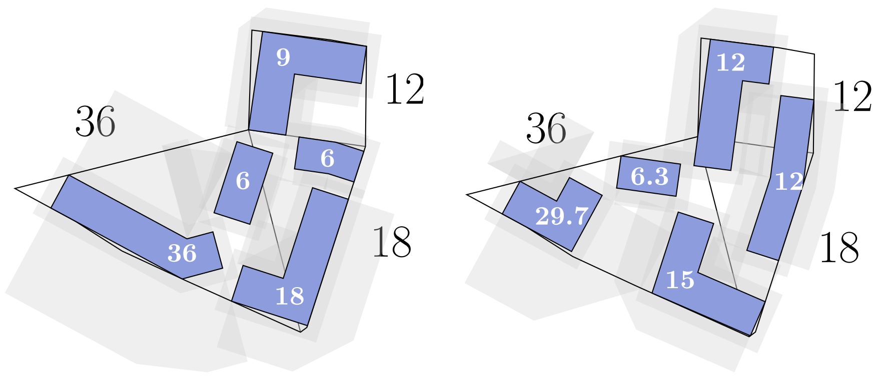
Site B is larger than site A, but the results still show that the heuristic is better for all cases also on this site, although generally with a lower difference between the

Table 10.3: The maximum aggregated volume of the buildings obtained by the heuristic and the naive solution approach. The difference between the approaches marked in bold

Site	Number of buildings	Solution approach	Alternative 1 [m^3]	Alternative 2 [m^3]	Alternative 3 [m^3]
A	5	Heuristic	17 301	19 515	20 449
		Naive	15 779	16 347	16 598
		Difference	**1 522**	**3 167**	**3 851**
	6	Heuristic	18 415	18 345	18 662
		Naive	16 189	16 802	17 096
		Difference	**2 226**	**1 543**	**1 566**
B	5	Heuristic	22 838	22 453	21 482
		Naive	20 574	21 273	20 678
		Difference	**2 264**	**1 180**	**804**
	6	Heuristic	21 093	21 865	22 372
		Naive	20 686	20 686	20 686
		Difference	**407**	**1 179**	**1 686**
C	6	Heuristic	30 626	31 005	30 896
		Naive	30 762	31 080	30 762
		Difference	**-136**	**-75**	**134**
	7	Heuristic	32 792	33 759	32 940
		Naive	32 345	33 626	32 345
		Difference	**447**	**133**	**595**
	8	Heuristic	32 237	32 610	33 610
		Naive	32 311	33 124	32 311
		Difference	**-74**	**-513**	**1 230**

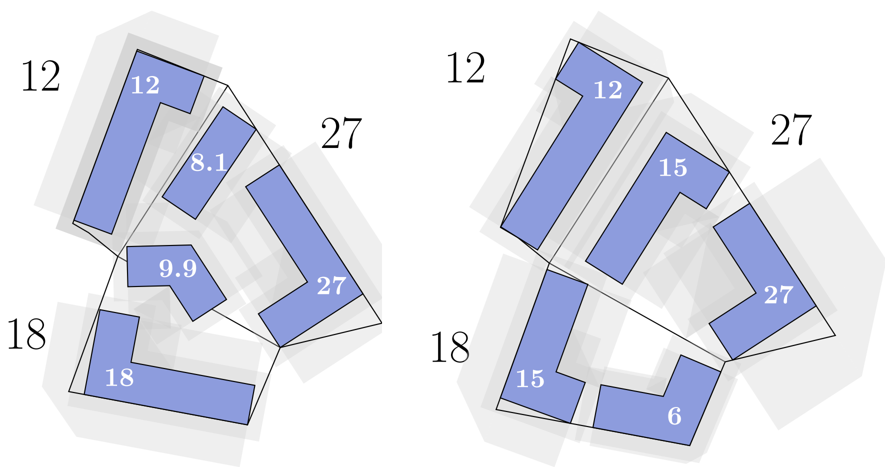
two approaches. The combination with the largest difference on site B is seen for five buildings and height zone alternative 1. The difference is 2 264 m^3, thus again, the heuristic provides a significant gain in saleable area. The heuristic and the corresponding naive solution are illustrated in Figure 10.3. The heuristic solution is also the solution with the highest aggregated volume found on site B. As seen in Figure 10.3, the reason the heuristic finds a better solution is because it places a large building in the highest height zone, and manages to maximize the height of this building. In addition, a large building is placed in the second highest height zone, and the height of this building is also maximized. In the naive approach, one building obtains the maximum height, but the building has a smaller footprint area than the one in the heuristic solution.

In the results for site C, the largest site, Table 10.3 shows that the naive solution sometimes finds better solutions than the heuristic, indicated by a negative number for the difference. The number of buildings tested on site C is smaller than the



(a) Heuristic solution - 20 449 m^3 **(b)** Naive approach - 16 598 m^3

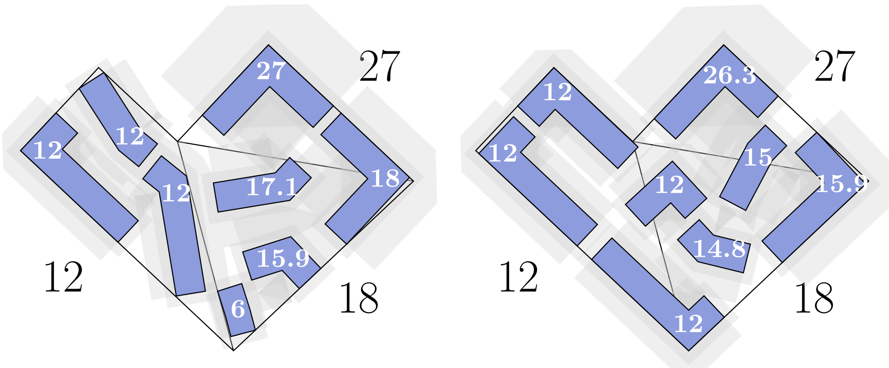
Figure 10.2: The combination with the overall highest difference. The heuristic solution is also the solution with the highest aggregated volume on site A



(a) Heuristic solution - 22 838 m^3 **(b)** Naive approach - 20 574 m^3

Figure 10.3: The combination with the highest difference on site B. The heuristic solution is also the solution with the highest aggregated volume on site B

area of the site dictates, such that there is more space available on the site. The naive approach finds good solutions since there is enough available space on the site for the buildings to have a large area, in addition to adequate spacing between them. This enables a sufficient increase in the heights after the fixed placement of the footprint areas to attain a large aggregated building volume. The solutions in which the difference between the two approaches is largest in favour of the naive approach are illustrated in Figure 10.4. The difference in volume is 513 m^3. The naive approach utilizes the corners of the site better than the heuristic, which may be due to the initial solution strategy used by the naive approach. The heuristic uses a random approach for the construction of initial solutions. In addition, the naive approach, like the heuristic, manages to place a large building in the highest height zone and obtain a large height for this building. The naive approach has fewer constraints than the heuristic, and may therefore be able to obtain better solutions than the heuristic for simpler cases, like the one illustrated in Figure 10.4.

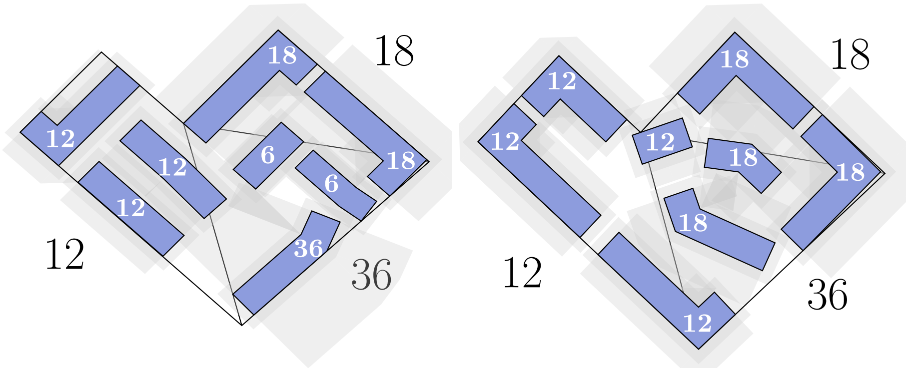


(a) Heuristic solution - 32 610 m^3 **(b)** Naive approach - 33 124 m^3

Figure 10.4: The combination on site C with the largest difference between the two approaches in favour of the naive approach

The results for site C reveal that there is one height zone alternative in which the heuristic is better than the naive approach for all the number of buildings tested. This is height zone alternative 3. One may argue that this is the most difficult height zone alternative to handle. The contribution to the aggregated volume is potentially huge if a building attains the maximum height, and conversely, a lot of potential volume can be lost to a poor placement of the building footprint areas. Since the naive approach places the buildings without knowledge of the maximum allowable height in each zone, the approach does not manage to take advantage of the large height allowed in the maximum height zone. The heuristic, on the other hand, takes the heights into account while the buildings are placed, and manages to utilize the maximum allowable height. The combination with the largest difference in volume between the naive approach and the heuristic on site C is 1 230 m^3. This is with eight buildings, and the difference is in favor of the heuristic. Figure 10.5 illustrates the heuristic and naive solutions. As seen in Figure 10.5b, the highest buildings produced by the naive approach are 18 meters tall, while the highest building in Figure 10.5a (using the heuristic) is 36 meters tall, producing a significant gain in volume on the site.

The same observation made for height zone alternative 3 on site C is not seen for sites A and B. There is no particular pattern that shows which height zone alternative has the largest difference between the heuristic and the naive approach. However, the largest difference between the heuristic and the naive approach seen



(a) Heuristic solution - 33 610 m^3 (b) Naive approach - 32 311 m^3

Figure 10.5: The combination with the largest difference between the two approaches on site C

for site A is indeed with height zone alternative 3, the solution illustrated in Figure 10.2a. From this Figure, it is clear that the heuristic is the better approach because it manages to place a large building in the height zone with the largest allowable height, and maximizing its height. Nonetheless, the same is not observed for six buildings on site A or at all on site B. No particular pattern is observed for height zone alternative 1 and 2. With a larger site and more height zones, the distribution of the zones may be more important than revealed in these results.

The layouts with highest aggregated volume achieved on the sites are illustrated in 3D in Figure 10.6 - 10.8. The lighting is added for illustrative purposes. The 2D plot of the same solutions for sites A and B are presented in Figures 10.2a and 10.3a. The 2D plot of the solution with the highest aggregated volume on site C is shown in Figure 10.9.

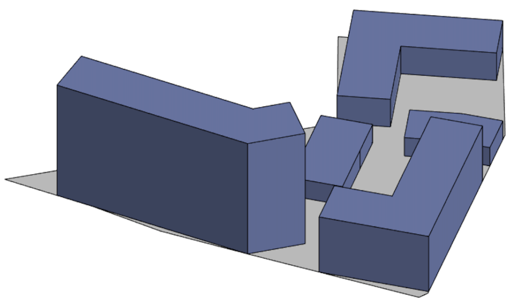


Figure 10.6: 3D plot of the highest aggregated building volume (20 449 m^3) on site A. The corresponding 2D plot is presented in Figure 10.2a

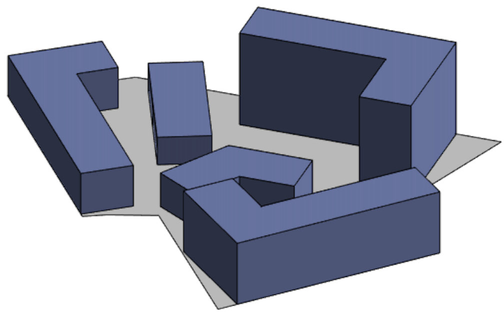


Figure 10.7: 3D plot of the highest aggregated building volume (22 838 m^3) on site B. The corresponding 2D plot is presented in Figure 10.3a

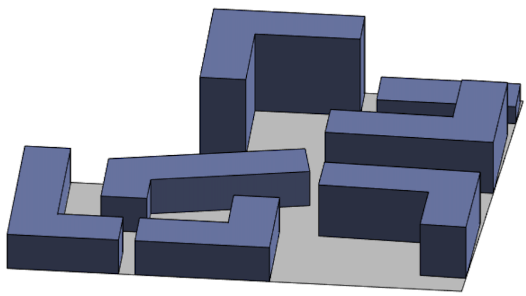


Figure 10.8: 3D plot of the highest aggregated building volume (33 759 m^3) on site C. The corresponding 2D plot is presented in Figure 10.9

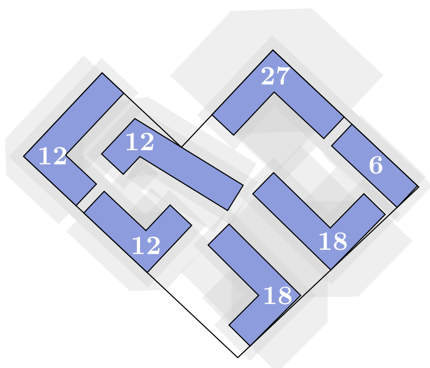


Figure 10.9: 2D plot of the highest aggregated building volume on site C - 33 759 m^3

To conclude, the results show that the heuristic is better than the naive approach when the number of buildings placed on a site is relatively large in relation to the area of the site. As seen in Section 6.3, this number of buildings is the number of buildings that maximizes the utilization of the site, and most likely the number of buildings the developer wants on the site. Moreover, a study of site C, the largest site, shows that the heuristic finds the highest aggregated volume for the most challenging case on this site. This is height zone alternative 3, where the difference between the maximum allowable heights of the height zones is larger. Hence, the heuristic shows promising results on cases that are more complicated and complex, cases that are more likely to appear in real life.

Chapter 11

Concluding remarks

This thesis has discussed the problem of planning the layout of an undeveloped site regulated for housing, and presented a mathematical formulation and solution method for generating layout designs. Today, the planning process is burdened by complex regulations and contradictory goals, making it hard to estimate the value of a site and create proposals of how to develop it. Even though most aspects of the problem are quantifiable, little research has been done on the particular subject. The framework proposed in this thesis can serve as decision support to stakeholders in the planning process by providing many different, good layouts. Overall, the layouts appear to have high degrees of utilization without violating the zoning regulations imposed on the site.

An extensive literature search is conducted in order to explore existing research on similar problems. Due to the sparsity of comparable problems and results, a new, nonlinear mathematical formulation is proposed and developed in two parts.

In Part I, the problem is reduced to two dimensions, and the mathematical formulation maximizes the area of a given number of buildings, which are subject to spatial constraints that have been formulated using a distance-based measure of feasibility and infeasibility. Using SQP, problem instances with up to eight buildings are solved to local optimua on different test sites within reasonable time. The average and maximum utilization of the layouts produced appear to be high, and the results indicate that the optimal number of buildings depend on the site and is not easily computed analytically. Alternative mathematical formulations of the buildings are developed for comparison with the more intuitive formulation. Although they do not yield significantly better solutions, they can be useful in exploring other layouts in addition to those found by the original formulation. Heuristics for constructing initial solutions that can be provided to SQP are developed and tested, one of which shows improvement over a random approach.

In Part II, the model is extended to three dimensions by incorporating building heights. The mathematical formulation proposed is a mixed-integer nonlinear program, and a heuristic is developed to solve the model. SQP and the solution methods for Part I is used to solve the subproblems developed in the heuristic. The difference in aggregated volume between the heuristic and the naive approach is greatest for the smallest site tested, which is the site with the highest number of buildings compared to site area. The difference decreases for larger sites. This indicates that the heuristic is most efficient for the more complicated cases where a large amount of building area has to be placed on a smaller site.

Based on the results, the mathematical formulation and solution method appear to be promising for the use of optimization within site layout design and merit further research.

Chapter 12

Future research

There are many possible future areas of research for the SDP, most of which were too comprehensive or too complex to include in this thesis. This chapter highlights the research questions considered to be the most relevant to increase performance and broaden the area of application.

12.1 Modeling

Although the SDP studied in this report can provide valuable insights to stakeholders in the planning process, several extensions to both the objective function and the constraints are needed to broaden the type of sites and regulations that can be handled. An evident extension is to extend the model to a multi-objective optimization problem. For this purpose, other important qualities or goals such as daylight or view can be optimized. This may help in producing site layout designs that not only preserve living qualities through constraints, but maximize them where possible, according to the preferences of the decision maker. In this context, daylight, or qualities based on illumination by the sun in general, are important to include as it is highly valued by residents, and may be part of zoning regulations. This extension will further complicate the objective function and constraints, possibly to a degree where SQP and other gradient-based methods may fail to converge if the functions are not formulated sufficiently smooth and efficient. Other qualities to introduce may be noise pollution, wind, and mobility. However, many such qualities cannot be evaluated analytically and must be simulated. This changes the nature of the problem completely, with the main challenge of solving a multi-objective problem where one or more objectives require the use of simulation.

Another natural extension of the current model is to allow for more general shapes, both of the buildings and sites by allowing a different topography. Other shapes of the buildings may yield better results as they might be easier to place and size on certain sites. Further, extending the model to handle uneven terrain of the sites may create a need for different building shapes. Other building shapes may include buildings with more than one angle point, terraced apartment buildings, or ”shifted buildings” (two or more buildings connected without an open zone in between).

The constraints included in the model in this thesis ascertain that the site layout design complies with some common zoning rules that dictate the minimum allowable distance between the buildings. Some zoning regulations require that a road or walkway should pass through the site, which complicates the placement and sizing of buildings. Possible extensions to admit this constraint could be to require that a set of scenarios of possible walkway placements are provided, or to include it as a variable, effectively making the boundary and containment constraints dependent on this variable. All in all, the model can be extended by many quantitative (or quantifiable) zoning regulations, depending on the location and type of problem at hand.

12.2 Formulation

In this study, the approach to defining the constraint functions has been to provide a distance measure of feasibility and infeasibility. An alternative approach is to evaluate the constraints between buildings, and between the buildings and the site, by computing the area of overlap instead of the distance. A comprehensive study into the algorithms and models used in computational geometry, vision and graphics might reveal approaches that are transferable to the SDP.

12.3 Solution methods

While SQP yields promising results in this thesis, alternative solutions methods could be tested for comparison. For instance, nonlinear interior-point methods are together with SQP considered the most powerful algorithms for nonlinear programming (Nocedal and Wright, 2006) and may be worth testing for comparison with SQP. As SQP only guarantees to find local optima, the initial solutions provided to the algorithm will significantly influence the quality of the results. Even if some of the algorithms for generating initial solutions proposed in this thesis produce good results, other global search strategies might adapt better to the variety of sites, parameters and number of buildings in practice. These can produce several different approximations of the global optimum, which can be used as input to SQP for further local optimization. Another research question might be to discretize parts of the problem and solve it using techniques from integer programming and bin packing. Local search can then be applied to optimize the discretized problem further.

Some of the results obtained in the computational study in Part I indicate that a solution approach using a heuristic may provide good results. One of the heuristics developed to construct initial solutions provides considerably better utilization than a random approach. In addition, the comparison between rectangular and angled buildings show that the computation time increases significantly for the angled buildings. Thus, developing a more advanced heuristic for the model could yield interesting results.

The heuristic developed in Part II can be improved by developing more accurate estimates of good allocations of buildings to height zones. Another area that is interesting to explore further is whether an established method can be applied to the non-convex, mixed-integer nonlinear problem in Part II.

Bibliography

Adams, R. A., Essex, C., 2013. Calculus, 8th Edition. Pearson.

Allen, S., Burke, E., Kendall, G., 2011. A hybrid placement strategy for the three-dimensional strip packing problem. European Journal of Operational Research 209 (3), 219–227.

Bennell, J., Scheithauer, G., Stoyan, Y., Romanova, T., 2010. Tools of mathematical modeling of arbitrary object packing problems. Annals of Operations Research 179 (1), 343–369.

Bennell, J. A., Dowsland, K. A., 1999. A tabu thresholding implementation for the irregular stock cutting problem. International Journal of Production Research 37 (18), 4259–4275.

Bennell, J. A., Dowsland, K. A., 2001. Hybridising tabu search with optimisation techniques for irregular stock cutting. Management Science 47 (8), 1160–1172.

Birgin, E., Martínez, J., Nishihara, F., Ronconi, D., 2006. Orthogonal packing of rectangular items within arbitrary convex regions by nonlinear optimization. Computers & Operations Research 33 (12), 3535–3548.

Boggs, P. T., Tolle, J. W., 1996. Sequential quadratic programming. Acta Numerica 4, 1–51.

Bonami, P., Lee, J., Leyffer, S., Wächter, A., 2011. More branch-and-bound experiments in convex nonlinear integer programming. Unknown publisher. Retrieved on May 25, 2018 from `http://www.optimization-online.org/DB_FILE/2011/09/3191.pdf`.

Braglia, M., 1996. Optimisation of a simulated-annealing-based heuristic for single row machine layout problem by genetic algorithm. Operations Research 3 (1), 37–49.

Bratsberg, N. M., Mellbye, A.-M., 2017. Site development optimization. NTNU Project Report.

Burer, S., Letchford, A. N., 2012. Non-convex mixed-integer nonlinear programming: A survey. Surveys in Operations Research and Management Science 17 (2), 97–106.

Cassioli, A., Locatelli, M., 2011. A heuristic approach for packing identical rectangles in convex regions. Computers & Operations Research 38 (9), 1342–1350.

Chen, D., Liu, J., Fu, Y., Shang, M., 2010. An efficient heuristic algorithm for arbitrary shaped rectilinear block packing problem. Computers & Operations Research 37 (6), 1068–1074.

Cherri, L. H., Mundim, L. R., Andretta, M., Toledo, F. M., Oliveira, J. F., Carravilla, M. A., 2016. Robust mixed-integer linear programming models for the irregular strip packing problem. European Journal of Operational Research 253 (3), 570–583.

Dalalah, D., Khrais, S., Bataineh, K., 2014. Waste minimization in irregular stock cutting. Journal of Manufacturing Systems 33 (1), 27–40.

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), 1632–1643.

Drira, A., Pierreval, H., Hajri-Gabouj, S., 2007. Facility layout problems: A survey. Annual Reviews in Control 31 (2), 255–267.

Duran, M. A., Grossmann, I. E., 1986. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Mathematical programming 36 (3), 307–339.

Dyckhoff, H., 1990. A typology of cutting and packing problems. European Journal of Operational Research 44 (2), 145–159.

Egeblad, J., 2009. Placement of two- and three-dimensional irregular shapes for inertia moment and balance. International Transactions in Operational Research 16 (6), 789–807.

Geonorge, 2017. Fkb produktspesifikasjoner. Retrieved on November 17, 2017 from `https://register.geonorge.no/register/produktspesifikasjoner?register=Produktspesifikasjoner&text=fkb`.

Gomes, A. M., Oliveira, J. F., 2006. Solving irregular strip packing problems by hybridising simulated annealing and linear programming. European Journal of Operational Research 171, 811–829.

Gottschalk, S., Lin, M. C., Manocha, D., 1996. Obbtree: a hierarchical structure for rapid interference detection. Proceeding SIGGRAPH '96 Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, 171–180.

He, K., Jin, Y., Huang, W., 2013. Heuristics for two-dimensional strip packing problem with 90 ° rotations. Expert Systems with Applications 40 (14), 5542–5550.

He, S., Perret, J., Brasebin, M., Brédif, M., 2015. A stochastic method for the generation of optimized building layouts respecting urban regulations. Advances in Spatial Data Handling and Analysis 31 (5), 265–288.

Heragu, S. S., Kusiak, A., 1988. Machine layout problem in flexible manufacturing systems. Operations Research 36 (2), 258–268.

Imamichi, T., Yagiura, M., Nagamochi, H., 2009. An iterated local search algorithm based on nonlinear programming for the irregular strip packing problem. Discrete Optimization 6 (4), 345–361.

IØT, 2018. Solstorm. `https://solstorm.iot.ntnu.no/wordpress/`.

Jakobs, S., 1996. On genetic algorithms for the packing of polygons. European Journal of Operational Research 88 (1), 165–181.

Ji, P., He, K., Jin, Y., Lan, H., Li, C., 2017. An iterative merging algorithm for soft rectangle packing and its extension for application of fixed-outline floorplanning of soft modules. Computers & Operations Research 86, 110–123.

Jin, J.-T., Jeong, J.-W., 2014. Optimization of a free-form building shape to minimize external thermal load using genetic algorithm. Energy and Buildings 85, 473–482.

Leung, S. C., Zhang, D., 2011. A fast layer-based heuristic for non-guillotine strip packing. Expert Systems with Applications 38 (11), 13032–13042.

Liu, X., min Liu, J., xi Cao, A., le Yao, Z., 2015. Hape3d—a new constructive algorithm for the 3d irregular packing problem. Frontiers of Information Technology & Electronic Engineering 16 (5), 380–390.

Lundevall, T., 2015. Arkitektarbeid - Profesjonskunnskap for arkitekter, 1st Edition. Fagbokforlaget.

Lundgren, J., Rönnqvist, M., Värbrand, P., 2010. Optimization. Studentlitteratur.

Martins, T., Tsuzuki, M., 2010. Simulated annealing applied to the irregular rotational placement of shapes over containers with fixed dimensions. Expert Systems with Applications 37 (3), 1955–1972.

MathWorks, 2017. Constrained nonlinear optimization algorithms. Retrieved on November 19, 2017 from https://se.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html.

Michalek, J., Choudhary, R., Papalambros, P. Y., 2002. Architectural layout design optimization. Engineering Optimization 34 (5), 461–484.

Nocedal, J., Wright, S. J., 2006. Numerical Optimization, 2nd Edition. Springer.

PBE, 2012. Utearealnormer. normer for felles leke- og uteoppholdsarealer for boligbygging i indre oslo. Retrieved on April 20, 2018 from `http://www.landskapsarkitektur.no/data/artikler/normer-og-standarder?iid=228718&pid=NLA-Artikkel-Filer.Native-InnerFile-File&attach=1`.

Pinheiro, P. R., Júnior, B. A., Saraiva, R. D., 2016. A random-key genetic algorithm for solving the nesting problem. International Journal of Computer Integrated Manufacturing 29 (11), 1159–1165.

Pressman, A., 2012. Designing architecture - the elements of process. Routledge.

Ratcliffe, J., Stubbs, M., Keeping, M., 2009. Urban Planning and Real Estate Development, 3rd Edition. Routledge.

Reinhard, K., 2015. Urban design synthesis for building layouts based on evolutionary many-criteria optimization. International Journal of Architectural Computing 13 (3-4), 257–269.

Santoro, M. C., Lemos, F. K., 2015. Irregular packing: Milp model based on a polygonal enclosure. Annals of Operations Research 235 (1), 693–707.

Sato, A. K., de Sales Guerra Tsuzuki, M., de Castro Martins, T., 2016. A pairwise exact placement algorithm for the irregular nesting problem. Annals of Operations Research 235 (1), 693–707.

Sato, A. K., Martins, T. C., Tsuzuki, M. S. G., 2012. An algorithm for the strip packing problem using collision free region and exact fitting placement. Computer-Aided Design 44 (8), 766–777.

Song, X., Bennell, J., 2014. Column generation and sequential heuristic procedure for solving an irregular shape cutting stock problem. Journal of the Operational Research Society 65 (7), 1037–1052.

Szykman, S., Cagan, J., 1995. A simulated annealing-based approach to three-dimensional component packing. Journal of Mechanical Design 117 (2a), 308–314.

Tay, F. E., Chong, T., Lee, F., 2002. Pattern nesting on irregular-shaped stock using genetic algorithms. Engineering Applications of Artificial Intelligence 15 (6), 551–558.

Theodoracatos, V. E., Grimsley, J. L., 1995. The optimal packing of arbitrarily-shaped polygons using simulated annealing and polynomial-time cooling schedules. Computer Methods in Applied Mechanics and Engineering 125 (1-4), 53–70.

Tian, Z. C., Chen, W. Q., Tang, P., Wang, J. G., Shi, X., 2015. Building energy optimization tools and their applicability in architectural conceptual design stage. Energy Procedia 78, 2572–2577.

Tong, Z., 2016. A genetic algorithm approach to optimizing the distribution of buildings in urban green space. Automation in Construction 72, 46–51.

Triola, M. F., 2012. Elementary Statistics - technology update, 11th Edition. Pearson Education International.

Trondheim Kommune, 2015. Eksempelsamling for reguleringsbestemmelser. Retrieved on December 15, 2017 from https://www.trondheim.kommune.no/attachment.ap?id=48817.

Tsoukalas, A., Parpas, P., Rustem, B., 2009. A smoothing algorithm for finite min–max–min problems. Optimization Letters 3 (1), 49–62.

United Nations, 2015. World urbanization prospects: The 2014 revision. Tech. Rep. ST/ESA/SER.A/352, United Nations, Department of Economic and Social Affairs, Population Division, retrieved on April 20, 2018 from `https://esa.un.org/unpd/wup/Publications/Files/WUP2014-Report.pdf`.

Uytenhaak, R., 2008. Cities full of space - qualities of density. 010 Publishers Rotterdam.

Watson, P., Tobias, A., 1999. An effcient algorithm for the regular w1 packing of polygons in the infnite plane. Journal of the Operational Research Society 50 (10), 1054–1062.

Wei, L., Hu, Q., Leung, S. C. H., Zhang, N., 2017. An improved skyline based heuristic for the 2d strip packing problem and its efficient implementation. Computers & Operations Research 80, 113–127.

Wäscher, G., Haußner, H., Schumann, H., 2007. An improved typology of cutting and packing problems. European Journal of Operational Research 183 (3), 1109–1130.

Yi, Y. K., Kim, H., 2015. Agent-based geometry optimization with genetic algorithm (ga) for tall apartment's solar right. Solar Energy 113, 236–250.

Yu, M. T., Lin, T. Y., Hung, C., 2009. Active-set sequential quadratic programming method with compact neighbourhood algorithm for the multi-polygon mass production cutting-stock problem with rotatable polygons. International Journal of Production Economics 121 (1), 148–161.

Yu, M. T., Lin, T. Y., Hung, C., 2012. Sequential quadratic programming method with a global search strategy on the cutting-stock problem with rotatable polygons. Journal of Intelligent Manufacturing 23 (3), 787–795.

Appendix A

A.1 Full mathematical model - Part II

$$
\begin{aligned}
\max\ & \sum_{b\in\mathcal{B}} w_b \cdot l_b \cdot h_b + \frac{w_b^2 \cdot h_b}{\tan(\frac{r_b}{2})} && && (8.3.1)\\
\text{s.t.}\ & F(\mathbf{v}_b) \cap \big(T^1(\mathbf{v}_a) \cup T^2(\mathbf{v}_a)\big) = \emptyset && b, a \in \mathcal{B},\ b \neq a && (4.4.6)\\
& F(\mathbf{v}_b) \cap \big(R^1(\mathbf{v}_a) \cup R^2(\mathbf{v}_a) \cup R^3(\mathbf{v}_a) \cup R^4(\mathbf{v}_a)\big) = \emptyset && b, a \in \mathcal{B},\ b \neq a && (4.4.7)\\
& F(\mathbf{v}_b) \cap A(\mathbf{v}_a) = \emptyset && b, a \in \mathcal{B},\ b \neq a && (4.4.8)\\
& F(\mathbf{v}_b) \cap \mathrm{conv}(\mathcal{C}) = F(\mathbf{v}_b) && b \in \mathcal{B} && (4.4.9)\\
& F(\mathbf{v}_b) \cap S = \emptyset && b \in \mathcal{B}, S \in \mathcal{S} && (4.4.10)\\
& h_b \geq \underline{H} && b \in \mathcal{B} && (8.3.2)\\
& \mathrm{overlap}(F(v_b), h_i) \leq M_1 \delta_{bi} && b \in \mathcal{B}, i \in \mathcal{I} && (8.3.3)\\
& h_b \leq H_i \delta_{bi} + M_2(1 - \delta_{bi}) && b \in \mathcal{B}, i \in \mathcal{I} && (8.3.4)\\
& \frac{1}{|\mathcal{B}|} \sum_{b\in\mathcal{B}} h_b \leq \widetilde{H} && && (8.3.5)\\
& \underline{W} \leq w_b \leq \overline{W} && b \in \mathcal{B} && (4.4.1)\\
& \underline{L} \leq l_b \leq \overline{L} && b \in \mathcal{B} && (4.4.2)\\
& 0 \leq \theta_b \leq 2\pi && b \in \mathcal{B} && (4.4.3)\\
& K \leq k_b \leq l_b - K && b \in \mathcal{B} && (4.4.4)\\
& \frac{\pi}{2} \leq r_b \leq \pi && b \in \mathcal{B} && (4.4.5)
\end{aligned}
$$