Minimum Cost Polygon Overlay with Rectangular Shape Stock Panels

Wilson Siringoringo
Software Engineering Research Lab, Auckland University of Technology
Auckland, New Zealand
wsiringo@aut.ac.nz

ABSTRACT
Minimum Cost Polygon Overlay (MCPO) is a unique two dimensional optimization problem that involves the task of covering a polygon shaped area with a series of rectangular shaped panels. This work examines the MCPO problem in order to construct a model that captures essential parameters of the problem to be solved by generic optimization algorithms.

Three algorithms have been implemented to perform the actual optimization task: the greedy search, the Monte Carlo method, and the Genetic Algorithm.

Categories and Subject Descriptors
J.6.a [Computer Applications]: Computer-Aided Engineering – Computer-Aided Design

Keywords
Two-dimensional modeling, layout optimization, greedy search, Monte Carlo, Genetic Algorithm

1. INTRODUCTION
In manufacturing industries, the optimization of material plays important part in minimizing the production cost, which is necessary to attain the competitive edge for the organization. The importance of material optimization is especially evident in manufacturing goods consisting of large amount of two dimensional material components such as sheet metal or fabric material.

A unique variant of the optimum two-dimensional layout problem is found in the residential house construction industry. A polygon shaped area such as wall or ceiling is to be tiled with covering sheet material such as cardboard or plywood. With such tiling, it is essential that the entire surface is covered with no gaps or overlaps. This class of problem has been identified as the Minimum Cost Polygon Overlay (MCPO).

When the panel is homogenous, such as with sheet metal, it is desirable to reuse the off cuts to cover irregular regions at other places, as this has the potential to reduce the total number of sheets required. A particular example is the reuse of off cuts from corrugated iron roofs [1]. The justification for such effort is provided by the high cost of delivering the roofing material.

1.1 Motivation
Apart from reducing the waste and reducing the associated cost, automating the panel placement design also greatly assists the builder in calculating the required material. When the calculation is done by hand, the common practice is to have a human expert work on the layout and to estimate the number of panels needed to cover a particular part of the building. A few extra panels must then be allocated to anticipate the possible errors in the calculation. As the solution only applies to a particular part of the building, the work must be repeated for all other parts as well. The process becomes more tedious when different sizes of the panels exist and therefore must be considered. Exploring more than a few different configurations by hand is such situation becomes an impractical proposition.

1.2 Related Works
Considerable research has been done in various fields of two dimensional layout optimization problems, primarily due to the practical needs of industry. Dyckhoff makes an attempt to provide a systematic classification of such optimization problems [2]. He uses the term cutting and packing (C&P) as a generic name for the problem and all its variants. He further postulates that there are four properties of each problem which determine to which class it belongs to. The properties are the problem dimensionality, the kind of assignment, the assortment of the containers, and the assortment of the pieces.

Dyckhoff asserts that there exist 96 classes of C&P problems that result from the combination of the four characteristics. For the purpose of this study, only the most important variants are considered. The significance of such variants is evident by the amount of research done and the publications that follow. The majority of such problems can be modelled in one of the four main variants: the sheet layout, the bin packing and strip packing, rectangular floor planning and cutting stock problem.

This paper was published in the proceedings of the New Zealand Computer Science Research Student Conference 2008. Copyright is held by the author/owner(s).

Figure 1: Wall Overlay with Fixed Size Panels.

The most generic and unrestricted form of the two-dimensional layout optimization is the sheet layout problem [3], which is also commonly known as sheet nesting and polygon containment. In essence, the sheet layout problem calls for
This decomposition into two sub-problems can potentially mask of numerical and graphical information: At the end of the calculation process, the desired output consists optimum solutions. that the problem is complex with potentially many locally potentially be varied. With this in mind, it becomes apparent applications, the panel size will remain fixed for the two sub-actual size of the panels is in itself a design parameter. In some important to recognize that in the construction industry, the one another [4][5]. The optimization task is accomplished by

packing as many polygon-shaped pieces within a polygon-shaped container without any restrictions apart from the basic requirement that the pieces should never overlap. The pieces are allowed to rotate, translate, and to flip about any axis.

2. PROBLEM MODELLING

Upon closer examination, the MCPO is found to be composed of two sub-problems which must be resolved sequentially, with each sub-problem belonging to the same two-dimensional layout optimization. Thus for a given enclosed area and a given dimensions of rectangular panels, the requirement is twofold:

1. Find the optimum arrangement of whole panels in which the covered area within the enclosure is maximized. The by-product of this process is a set of irregular shapes which represent the remaining exposed areas.
2. Resolve how such irregular shapes can be nested within the minimum number of panels.

This decomposition into two sub-problems can potentially mask the complexity of the task of finding the optimum solution. It is important to recognize that in the construction industry, the actual size of the panels is in itself a design parameter. In some applications, the panel size will remain fixed for the two sub-problems whilst for other applications the panel size could potentially be varied. With this in mind, it becomes apparent that the problem is complex with potentially many locally optimum solutions.

At the end of the calculation process, the desired output consists of numerical and graphical information:

1. The total number of panels, consisting of panels to be fitted whole and the remainder to be cut to produce the irregular shapes
2. The nesting plan with which irregular shapes are cut from whole panels
3. The area overlay plan with which whole panels and irregular cuts are fitted to the enclosed area

It is also important to note that although the two sub-problems are similar, they are resolved with mutually unrelated and potentially conflicting objectives. As an example, the lowest cost for first sub-problem may be to cover as much area as possible with the least number of panels. However, the optimum solution second sub-problem may be the least amount of cutting. Hence a cheap solution in the first phase may lead to expensive penalties in the second.

3. OPTIMISATION ALGORITHMS

3.1 Placement Strategies

Sequential placement algorithms are characterized by populating the container with one piece after another. When a piece is placed on the container, an irregularly shaped smaller container is created in effect. The algorithms greedily conserve the size of the newly created restricted area when it picks subsequent pieces. The process is repeated until either the pieces are exhausted or the container is unable to accommodate more pieces. An equally important aspect of sequential placement algorithms is the optimization of coordinates and orientation of the pieces.

With simultaneous placement, pieces are paired and placed in the container without using any sequence allocation list. Instead, other data structures such as trees and graphs are used to represent the nesting and the position of each piece relative to one another [4][5]. The optimization task is accomplished by finding the configuration of such structures which provides the best value for the objective function.

3.2 Greedy Algorithm

The assumed posture of the Greedy Algorithm is to choose the solution with the highest immediate value at every turn. The major weakness of this naïve strategy is it inability to escape local optima traps. In the classic hill climbing problem, the algorithm makes its ascent by successively selecting the highest neighbouring node until the peak is reached and no more climbing is possible. Obviously, this approach is prone to premature convergence in multimodal search space.

Greedy algorithms seldom find the global optimum solutions; yet in many cases they are capable of finding reasonable solutions quickly [6]. Because of its simplicity and speed of execution, the greedy method is quite powerful and well suited for a range of problems. Greedy methods are used in a number of important algorithms such as minimum-spanning-tree algorithms, Dijkstra’s single-source-shortest-path, and for data compression using Huffman codes [6].

3.3 Monte Carlo Technique

Monte Carlo method is a blanket term used to describe any method characterized by the use of a random number generator and the complete disregard of dynamics involved in reaching the results. Weissens defines Monte Carlo technique in general as [7]:

[Monte Carlo technique is] any method which solves a problem by generating suitable random numbers and observing that fraction of the numbers obeying some property or properties. The method is useful for obtaining numerical solutions to problems which are too complicated to solve analytically.

In its most basic form, a memory-less random walk is all that is involved in implementing Monte Carlo optimization method. With such unrestricted search, completely lacking in decision making rules and record keeping makes Monte Carlo optimization much simpler to implement than the heuristic algorithms.

3.4 Genetic Algorithm

The algorithm uses the concept of a population of individuals which is subject to a series of probabilistic operators such as mutation, selection and recombination. Each individual, in the form of chromosomes, represents a potential solution to a given optimization problem. During the computation process, the population will undergo a draconian process in which stronger individuals will thrive while the weaker perish.

From the optimization point of view, the chromosome serves as the representation of the coded parameters of the optimization problem. To determine how ‘good’ an individual is as a solution, its chromosome is decoded to retrieve the actual values, which is then fed to the objective function of the original optimization problem. The routine that decodes the gene string and calculates its objective function is called the fitness function, and the result of the examination is called the fitness value. Gene strings with better fitness values represent the stronger individuals within the population. Such individuals are favoured by the system and more likely to survive and reproduce.

4. DATA REPRESENTATION

The MCPO presents a major challenge when simultaneous placement strategy is employed, such as that found in MC and GA. Unlike other C&P problems documented in the literature
where only a single container is used, the number of containers required to accommodate the nested shapes in MCPO is itself a variable. Consequently a direct mapping of the rest of the optimization parameters to the chromosome string cannot be made. To resolve this problem, an indirect coding utilizing the concept of clusters is used.

4.1 Cluster Coding
In this technique, static blocks are mapped to the nested panel. Each panel is associated with a fixed-width block of bits in the chromosome. This block contains only a single variable of integer type, namely the cluster ID. Figure 2 shows the association between the panels and the blocks in the chromosome.

![Figure 2: Gene to Panel Mapping](image)

The value of each variable points to an imaginary cluster to which the panel belongs. Figure 3 shows an example of a populated chromosome with the imaginary clusters that result. Using Figure 2 as reference, it is easy to decode the chromosome to find that the Panel 0 is a member of Cluster 2, whereas Panel 1 is a member of Cluster 8, and so on. Similarly, Cluster 0 appears to have only a single member, i.e. Panel 4, whereas Cluster 2 has two members: Panel 0 and Panel 5.

A cluster is regarded legal if all its members can be nested in a single stock panel. Part of the fitness function’s task is to discover whether such nesting is possible. In the case of invalid cluster being encountered, there are a number of possible ways to respond.

![Figure 3: Interpreting a Candidate Chromosome](image)

4.2 Valid/Invalid Chromosomes
A chromosome in the context of layout optimization is accepted as valid only when all pieces can be successfully nested in their associated stock panel. Because the search performed by both GA and MC algorithms is set-oriented, there is no guarantee that all the clusters extracted from a chromosome are valid. Invalid clusters are found very frequently in the actual tests because many of the individual pieces are quite large compared to the size of stock panels, invariably claiming most of the available area after only one or two nested pieces.

![Figure 4: Complex Roof Layout](image)

Invalid chromosomes have much less impact on Monte Carlo technique than they do on Genetic Algorithm because the MC technique generates a new bit pattern on completely random basis at each cycle. The bit pattern of the chromosome at any particular point has no influence on the shaping of the bit pattern in the next iteration. As a result, the MC algorithm needs only to retain the best known valid chromosome somewhere in memory and ignore the invalid varieties.

5. EXPERIMENTAL RESULTS
The number of parameters used in solving a layout optimization problem is such that possible combinations exist is very large. This problem rules out exhaustive investigation because the actual optimization run is computationally expensive for each combination of parameters.

![Figure 5: A Fragment of Nesting Solution for Complex Roof Problem](image)

The experiments therefore were configured and executed in such a way that allows the behaviour of the software to be
monitored and measured through only a small number of optimization runs. With such limitations in mind, the experiments have been conducted with the aim of observing the optimization processes.

A number of experiments have been conducted that compare the performances of the three algorithms. The actual optimization problem presented in this paper is taken from one of the sample problems used by Sibley-Punnett and Bossomaier for their roof layout optimization [1]. In this case, a complex roof consisting of multiple sections is used. Figure 4 shows the top view of the roof, whilst Figure 5 shows a certain part of the nesting solution presented as the cutting plan for the roofing material.

Table 1 compares the performances of greedy method, Monte Carlo search, and Genetic Algorithm in solving the same layout problem. In this experiment, the same best fit strategy was employed and the material was allowed to rotate 180° only.

6. DISCUSSION

Solution quality is defined by the efficiency of material usage (or minimization of wasted material) and the length of shared edges (indicating the effort of cutting the pieces off the stock panels). Greedy search proves to be very effective by consistently outperforming other algorithms on both accounts. Greedy search has decisive advantage in computation time because the task of constructing the solution using this method is equivalent to just one fitness function evaluation on MC and GA. It also converges to a local optimum reliably, of which the efficiency is always higher than that of MC and GA.

Table 1: Solution Quality Criteria

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Greedy</th>
<th>MC</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Panels</td>
<td>25</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>No. Offcuts to Nest</td>
<td>129</td>
<td>129</td>
<td>129</td>
</tr>
<tr>
<td>Total No. Pieces</td>
<td>154</td>
<td>154</td>
<td>154</td>
</tr>
<tr>
<td>Stock Panels Used</td>
<td>64</td>
<td>93</td>
<td>85</td>
</tr>
<tr>
<td>Shared Edge Length</td>
<td>9930</td>
<td>9662</td>
<td>9814</td>
</tr>
<tr>
<td>Area to be Covered</td>
<td>124153</td>
<td>124153</td>
<td>124153</td>
</tr>
<tr>
<td>Stock Panels Area</td>
<td>128000</td>
<td>186000</td>
<td>170000</td>
</tr>
<tr>
<td>Wasted Material</td>
<td>3847</td>
<td>61847</td>
<td>45847</td>
</tr>
<tr>
<td>Solution Efficiency</td>
<td>97%</td>
<td>67%</td>
<td>73%</td>
</tr>
<tr>
<td>Search Duration</td>
<td>0:00:03</td>
<td>0:48:05</td>
<td>0:45:35</td>
</tr>
</tbody>
</table>

Table 1: Solution Quality Criteria

Such finding naturally raises a question of why such a crude algorithm can perform so much better than its much more sophisticated counterpart. Especially when compared to a GA, which is widely accepted as a powerful tool for solving multivariable optimization class of problems to which the second-stage problem of MCPO belong.

The fundamental problem with the use of GA in solving MCPO problem is identified as the parameter modelling, which also applies to MC method. The concept of clustering is used in the prevailing model to address the problem of mapping the irregular panels to an undetermined number of stock panels. While the model solves this particular problem quite well, it completely disregards a host of crucial parameters to be solved in the individual nesting tasks.

From the user perspective, the experiment results reveal that the use of novel optimization algorithms such as MC and GA has not been justified at the current stage of the software maturity. Employing the greedy search is the most logical choice for solving MCPO problems due to its low resource requirements and high quality solutions.

7. CONCLUSION

Decomposing MCPO into a two-stage optimization model provides a solid ground for constructing a well-functioning solution. The study has also proven that with the support of appropriate analysis, software application to solve complex problem such as MCPO can be successfully implemented using standard modelling and programming tools. Successful implementation of the software in turn proves the feasibility of constructing MCPO solution automatically for commercial use with current computing technologies.

Successful use of Monte Carlo technique and the Genetic Algorithm has proven that generic optimization algorithms can be used to solve the second-stage problem. The implication is that other optimization techniques could also be used in their place. Various optimization techniques such as Swarm Intelligence, Simulated Annealing, and Tabu Search can potentially increase the effectiveness of the search for the second-stage optimization.

8. ACKNOWLEDGEMENT

This research has been supported by Technology New Zealand through the Technology for Industry Fellowships scheme under grant number BISCO502 and this support is gratefully acknowledged.

9. REFERENCES


Minimum Cost Polygon Overlay with Rectangular Shape Stock Panels