Optimization of Newspaper Pagination using the Simulated Annealing Algorithm and the Genetic Algorithm

K.J.L Sirisena, K.A.P.S Perera, C.D.D Karunarathna, H.A.T.D Hettimulla, Shahani Weerawarana and Ravindra Koggalage

Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka.

Abstract— **Newspaper pagination has become an NP-hard problem with the need to optimize the space of a newspaper. A well paginated newspaper is a newspaper which includes a high number of advertisements and articles along with specific pagination rules. The research problem is to find an efficient and suitable algorithm to generate a well paginated newspaper. Most of the literature related to newspaper pagination indicates the use of the Simulated Annealing algorithm to solve the problem. In this research study, we introduce an improved method of using the Genetic Algorithm to solve the newspaper pagination problem along with a method of deriving an improved solution using Simulated Annealing. We use some heuristic methods within the Genetic Algorithm and the Simulated Annealing algorithm to achieve the basic pagination rules. This research study includes a comparison of statistical data from the two algorithms.**

***Index Terms*-newspaper pagination, space optimization, Simulated Annealing, Genetic Algorithm, optimized pagination**

I. INTRODUCTION

Newspaper pagination may be described as the process of producing good page layouts for given material. Automatic pagination will be useful as pagination is one of the important phases of the newspaper production process. [1] Automatic pagination systems might be an answer to the newspaper publications, struggling with high production costs and need for increased speed of production. The standards of pagination are mostly based on goals of the publishing organization. In general the automatic pagination is an important NP-hard problem and the size of the problem naturally contributes to the complexity of the task. [2] The only way to find an optimal solution for a NP-hard problem is to do a computationally-intensive, exhaustive analysis, in which possible outcomes are tested. [3] So, the main objective of our research is to implement a better solution for this problem by analyzing some possible ways of solving.

II. LITERATURE REVIEW

Our literature survey was primarily focused on finding existing solutions for newspaper pagination problems as well as for other optimization problems. So we firstly researched similar kind of problems and found that this problem goes to the category of NP-Hard problems [2]. That is a mathematical problem for which, even in theory, no shortcut or smart algorithm that would lead to a simple or rapid solution is possible. An algorithm for solving a NP-Hard problem can be translated into an algorithm for solving any NP-problem of the same kind. The only way to find an optimal solution for this kind of problems is to do a computationally-intensive, exhaustive analysis, in which all possible outcomes are, tested [3], [4].

Then we studied some of the algorithms used in literature for solving optimization problems like this. Below sections contains the details on the problems and the approaches used in studied algorithms.

A. Problems Studied

1) Bin Packing Problem

Bin Packing Problem (BPP) can be defined using three parameters,

O - Finite set of objects
B - Bin size
N - Possible bins that all the objects can be pack.

Bin Packing Problem is a transportation problem, where there are various size packages to transport from one place to another using containers, such that the requirement is to transport all the packages using minimum number of containers.

Some pagination problems fit this formulation approximately. Each page can be represented by a bin, and the given advertisements can be represented by the items to be packed. Furthermore, many pagination problems have different kinds of constraints that would need to be integrated into the pure Bin Packing Algorithms [6], [7], [8]. To do this we want to place maximum number of packages in a container.

2) Two-Dimensional Strip Packing Problem (2SPP)

The two-dimensional strip packing problem (2SPP) exists in many real-world applications, such as in glass, paper, textile, and other industries, with some variations on its basic formulation. Typically, the 2SPP consists of a set of M rectangular pieces, each defined by a width w_i and a height h_i ($i = 1, \ldots, M$), which have to be packed in to a larger rectangle, the *strip*, with a fixed width W and unlimited length. The objective is to find a layout of all the pieces in the strip that minimizes the required strip length, taking into account that the pieces have to be packed with their sides parallel to the sides of the strip, without overlapping. This problem belongs to NP-hard category [2].

3) Discussion of Problems

As we have described in above two problems, the three dimensional Bin Packing Problem is modeled to two dimensions easily and then it will be transformed in to a space optimization problem, which is in two dimensions. The 2- Dimensional Strip Packing problem is a similar transformation and our problem also can transform in to a similar model. Furthermore, since both the Bin Packing Problem and 2 – Dimensional Strip Packing Problem belong to NP- hard category we can conclude that our space optimization problem also belongs to NP- hard category.

B. Algorithms Studied

1) The Simulated Annealing Algorithm

The Simulated Annealing algorithm is defined using six major steps while maintaining two solutions at a time with the purpose of reaching the best solution. The current solution is considered to be the best solution we have found up to the point of execution, and the working solution is the solution we are working on to make in the next iteration. During the assessing state, if the working solution is better than the current solution, the working solution is assigned to the current solution [9].

SA (Simulated Annealing) is widely used to solve combinatorial problems like optimizing web newspaper layout. The browser window is divided in to several columns with fixed width, fitting in the available surface. There are various types of articles, but web browser has an infinite length so the problem to be solved is how to fill all the columns of the webpage equally, minimizing gaps between articles [11], [12].

2) Heuristic Best-First Forward Search Utilizing Branch-And-Bound Technique

The initial goal of Fax Newspaper pagination had been defined as pagination of articles with all kinds of sizes and rectangular shapes. However, the requirement of any article size had been relaxed and problem space considerably discredited to reduce it. Instead of allowing any article size starting from any location on the page, a sparse grid (typically 5×5 grid for an A4 page) had been defined and article sizes were accordingly modularized. Division into columns was natural, since in a typical newspaper the text usually resides in columns of equal width [13], [14].

The pagination method used is heuristic best-first forward search utilizing branch-and-bound technique. The contents of each article were first paginated into several different modularized shapes--- *article forms*---by rough article layout design..

3) Graph Theoretic Approach

One way to solve a problem is to describe it in terms of a graph with nodes as states in the problem-solving process and edges as transitions between states. There are many methods to search a graph but a heuristic search technique is used here [14].

4) Heuristic Graph Searches

Heuristics are rules of thumb that guide the search of good solutions to a problem. In graph searches heuristics are used to select the most promising-looking branch of the search tree or graph. These branches may then be examined first, before others, or even to cut out parts of the search tree altogether. Good heuristics can reduce the time required to find a good-enough solution, compared to brute-force methods. The issue is how to design heuristic rules that lead to good solutions instead of bad ones [14].

5) Genetic Algorithm

The Genetic Algorithm is an optimization technique that simulates the phenomenon of natural evolution (as first observed by Charles Darwin). In natural evolution, species search for increasingly beneficial adaptations for survival within their complex environments. The search takes place in the species' chromosomes where changes, and their effects, are graded by the survival and reproduction of the species. This is the basis for survival of the fittest—survival and the passing on of these characteristics to future generations. Survival in nature is the ultimate utility function [9].

6) Discussion of Algorithms

After considering all the algorithms in our literature survey we realized that they have different usages in different kinds of optimization problems. To use any of these algorithms to solve our problem, it was necessary to model them to suit our problem. When modeling an algorithm to a particular problem, there are some constraints we need to consider such that the outcome of the algorithm is productive. If we consider the Heuristic Best-First Forward Search Utilizing Branch-And-Bound Technique or Heuristic Graph Search method, they both mainly depend on heuristics, so that reaching to the best solution is somewhat impossible. And also in the graph theoretic approach the search space is a larger, so it will be a waste of memory.

But when we consider the Genetic Algorithm regarding its usage in Bin Packing Problem, if we think Bin Packing in two dimensional geometry, instead of three dimensions it is similar to our problem. Because of that, we thought to model Genetic Algorithm for our problem.

Also the Simulated Annealing algorithm has a more generic flow which can model to our problem. So that, we thought of implementing Simulated Annealing algorithm while using some heuristic functions to get the optimum solution easily.

III. METHODOLOGY

A. Modeling the Solution

For the implementation of algorithms we have mathematically modeled our pagination problem. Publishing organizations has divided the newspaper space into rows and columns. Most of the time a page has 11 columns and 52 rows in English newspaper.

Therefore we modeled the page layout as a two dimensional Array (matrix). Then we can specify a start position for every advertisement within the layout by giving a value for given positions in the matrix. To model advertisements we also used the matrices with all the values filled by its unique advertisement id. Then we have defined some constant values to define the spaces of the dummy matrix as free space = -1, article space=0, etc.

So the input variables to the algorithms are follows

- Page matrices (Dummy)- Fixed size several matrixes ,initial value=-1
  Ex: -two 16x5 matrices

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1

Fig. 1: Page Matrices

- Advertisement matrices with various sizes (size<size of a page matrix)
  Values in each matrix is same for each element of matrix and it is a unique identity for that matrix

  Ex:- 2x2, 4x2,1x2,5x1,3x4,2x4,4x5,7x2,8x4,5x1 matrixes

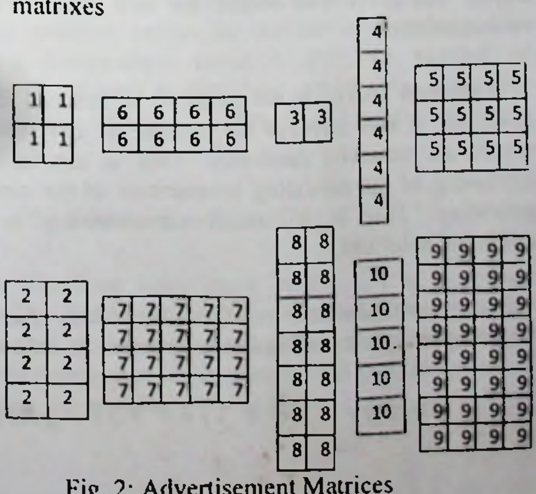

Fig. 2: Advertisement Matrices

Outputs of the Algorithms:

The output of the algorithms contains the page matrixes which filled by the advertisement matrixes and the remaining advertisement matrixes which couldn't be in to the page matrixes. The advertisement matrixes will remain in the output if the free spaces of the page matrixes are not enough to put all the advertisement matrixes. To get the best solutions those outputs should have the following properties.

- The remaining spaces of the page matrixes should be near to a square shape
  Ex:-

1st Solution

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
				-1
				-1
				-1
				-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	
-1	-1	-1	-1	
-1	-1	-1	-1	
-1	-1	-1	-1	
6	6	6	6	
6	6	6	6	

2nd Solution

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
6	6	6	6	
6	6	6	6	

Fig. 3: Solutions

Here the second solution is better than the first one.

- Remaining distinct spaces should be minimum

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
5	5	5	5	10
5	5	5	5	10
5	5	5	5	10
5	5	5	5	10
-1	-1	-1	-1	10
-1	-1	-1	-1	4
-1	-1	-1	-1	4
-1	-1	-1	-1	4
-1	-1	-1	-1	4
6	6	6	6	4
6	6	6	6	4

-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	10
-1	-1	-1	-1	10
-1	-1	-1	-1	10
-1	-1	-1	-1	10
-1	-1	-1	-1	10
5	5	5	5	4
5	5	5	5	4
5	5	5	5	4
5	5	5	5	4
6	6	6	6	4
6	6	6	6	4

Fig. 4: Solutions

Here the second solution is better than the first one.

- We should try to put all the advertisement matrices in to the remaining spaces (if there are -1 values)
- We have to put advertisements according to the rules and regulations of the publication company.

    1. If the advertisement has a specific position that defined by the agent, we first place them in the template correctly.
    2. If the advertisements do not have a defined specific position.
        a. Advertisements are placed according to the descending order of the width of the advertisements from bottom to top.
        b. Also advertisements are placed according to the descending order of the height of the advertisements. (Higher priority is given for width)

Firstly we implemented the two algorithms according to our problem separately. The following sections describe the procedures of implementing the said algorithms.

B. The Simulated Annealing (SA) Algorithm

In the Simulated Annealing algorithm, normally the initial configuration (layout) of advertisements is a random configuration. But we created the initial configuration of the layout according to the advertisement placing rules of the newspaper publication corporation. We also optimized the working solution according to the rules of the newspaper publishing corporation. Therefore we can converge in to a global minimum in a short period of time.

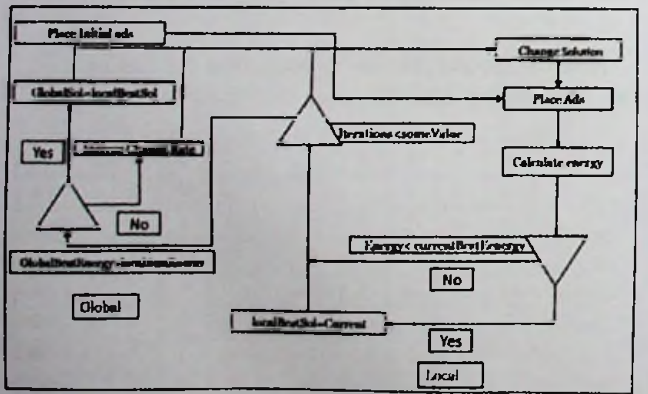

Fig. 5: Simulated Annealing Model

Given below is the simplified version of the Simulated Annealing approach. At first, we create an initial solution by using following steps.

- Sort the advertisement matrices according to their sizes
- Place the biggest advertisements in the bottom right corner of the every even page (If it is odd page place it in the bottom left corner) matrices and name them as initial large advertisements.
- After placing one large advertisement in all the page matrices fill remaining spaces by using other advertisements according to the advertisement placing algorithm.

There are three steps when we place an advertisement to the page matrix in the algorithm. This is the algorithm mentioned as advertisement placing algorithm above. Following these steps helps to converge quickly in to a better solution. This algorithm checks the free space to put advertisements in following order.

- Width and height fit advertisements – First put advertisements which are an exact fits to given free spaces .(Advertisement and the free space have the same height and width)
- Width or height fit advertisement- Try to find advertisements which have either the width or the height as same as the given free space.
- Random advertisements which reduce the energy of the current solution.

After creating the initial solution, the algorithm will optimize the solution by removing some advertisements and placing them again. When removing these advertisements the algorithm will not remove above mentioned initial large advertisements that had been put according to the rules and regulations of the publisher. Without optimizing the solution randomly the algorithm is using a better way to find the areas which we want to improve directly. When removing advertisements in the page matrix it uses following two steps.

- Remove advertisements that are near to the free spaces
- Remove some random advertisements

These steps help to converge to a better solution rather than tweaking the solution randomly.

After optimizing the solution algorithm calculates the energy of the new solution and it will be compared with the current solution. If the new energy is less than the current energy it will accept the new solution as the current solution.

If the new energy is not going to be reduced in given iterations, it will increase the number of advertisements which are removed randomly. That is similar to the increasing of the annealing temperature of the simulated annealing. This is to avoid concentrating in local minimum solutions.

Energy calculation in the Simulated Annealing algorithm is done by using four parameters as below.

$$Energy = (D + 1) * P * SP * (RSP)^{0.4}$$

D: - distance to the square shape

$$D = (P) - 4 * (SP)^{1/2}$$

This variable is used to measure the shape of the free space. If the free space is near to a square shape this will be minimum

P: perimeter of the empty space this is also related to the shape of the free space. If the shape is a regular square shape or the remaining free spaces are less, this will be minimum.

SP: area of the remaining space this will be reduced if free spaces are less. So to minimize this algorithm will try to fill all the free spaces of the dummy.

RSP: number of distinct empty spaces to reduce this parameter, algorithm needs to find solutions which have all the free spaces gathered to one free space.

In the energy function +1 added to the _D' to avoid energy being zero. If the remaining free space is exactly a square shape _D' will be zero.

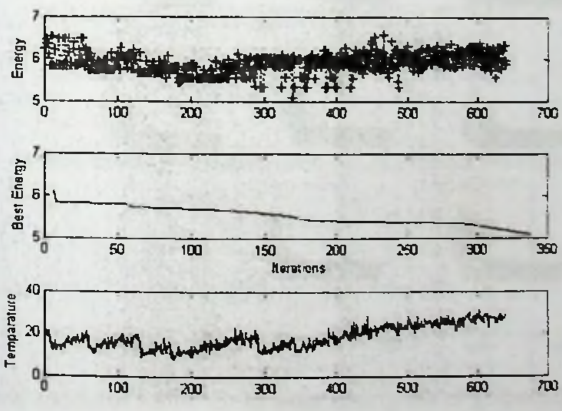

Fig. 6: Energy and temperature variations

In figure 16 the first two graphs show the variation of the energies of the working solution and the global optimum solution versus the number of iterations. The annealing temperature variation with the number of iterations is shown in the third graph. The annealing temperature is varied according to the global optimum solution

C. *The Genetic Algorithm (GA)*

There are three basic steps in the GA as described below. We used some heuristic methods along with the algorithm to model it to our problem. According to GA the whole newspaper can be considered as a chromosome and a page can be considered as a gene.

a) Create the initial population

We used Descending Next First (DNF) heuristic method to enter the highest aliel (advertisement) (highest width and height) to genes (papers). We enter it to the bottom right corner of the paper according to the DNF to satisfy with the conventions we used.

After the first iteration there is one advertisement in all genes at the bottom right corner of the paper. Then we take the 1st gene (paper) and get the combination of the advertisements that fit to the width of the most bottom limit.

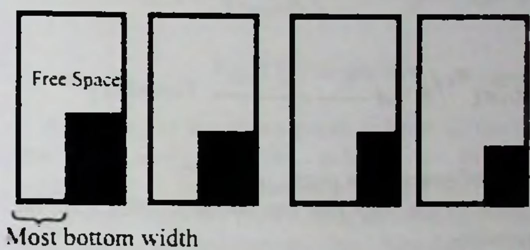

Fig. 7: Initial Solution

If there are advertisements have the same width but different heights, then we randomly select an advertisement. According to that method we fill the lowest level of the paper using available advertisements. Then we do this process page by page (across all the genes of the chromosome) according to the Next Fit heuristic considering a combination of few advertisements as a one.

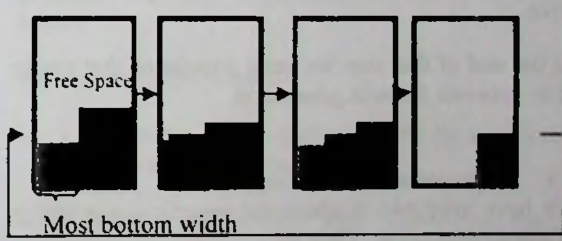

Fig. 8: After Adding Advertisements

Likewise we create one chromosome. According to that process we change the initial page and do the same to generate chromosomes (paper or section) that equals to the number of genes (pages) of the chromosome. This is the initial population.

That is, if there are n pages for a section, then we create n number of chromosomes. Then there are n numbers of chromosome in the initial population.

b) Assess the population

We use the below cost functions to assess the individual genes (Equation 1) of the chromosome and to assess each chromosome (Equation 2).

We firstly assess the individual gens of the chromosome and then assess each chromosome. So at first energy for the free space within a page is calculated.

If the shape type of the free space is a rectangle or a square, then energy of the free space (e_i).

$e_i = E_{normal}$

else, $e_i = E_{normal} + (E_{normal}/S_f)*(t_p)$
t_p: Number of turn Points of the free space
S_f: full space that need to fill from the advertisements.

Then we calculate the energy for each of the pages.

$$E_{gi} = (e_m)^n + \left(A_{remain}/A_{total}\right) * e_i$$

$$E_m = \sum_{i=1}^{n} e_i/n \quad \longleftarrow \quad \text{Equation 1}$$

N: Number of spaces in a page.
A_{remain}: Area of the page that has not been filled from the advertisements.
A_{total}: Area of the page that is reserved to insert advertisements

Finally calculate the energy of the solution

$$E_{chromosome} = \sum_{i=0}^{n} E_{gi}/N \quad \longleftarrow \quad \text{Equation 2}$$

According to the energy of each chromosome we decided which chromosomes should be used to generate the next generation and which chromosomes couldn't survive.

At the end of that step we get a population that can be used to generate the next generation.

- Generate next generation

We have used two fundamental genetic operations in the Genetic Algorithm. They are,

- Cross over
- Mutation

Before doing these operations we first select the chromosome with the lowest energy and the chromosome with the highest energy.

Then select the lowest energy gene of the lowest energy chromosome. Do the cross over operation using these genes. Replace these genes to the corresponding positions of the highest energy gene of the highest energy chromosome. From this process to this process we get two new children.

Then select each of the chromosomes and remove the genes that repeated by the advertisements and get the advertisements from the removed genes that are not in the chromosome and try to inject these to the existing genes according to the pagination rules. If we could not enter to any of the existing gene then we introduce a new gene to the chromosome and replace it if the existing genes in the chromosome are less than the number of required genes. Otherwise, we destroy the children which are very bad solution that cannot survive in the environment (solution space).

Likewise we do this process to all the chromosomes in the population and we get a huge population

Population = new population + parent population.

Assess the population according to the 2nd step and select a best population that use to generate the next generation. Then by doing the 3rd step we can generate new population. Likewise we can do this iterative process until we get the best chromosome (Lowest energy) within the population. It will be the global optimum solution of the Genetic Algorithm.

IV. RESULTS AND DISCUSSION

A. *Optimization for GA*

1) Results for the Energy Variation of Generations within Iterations

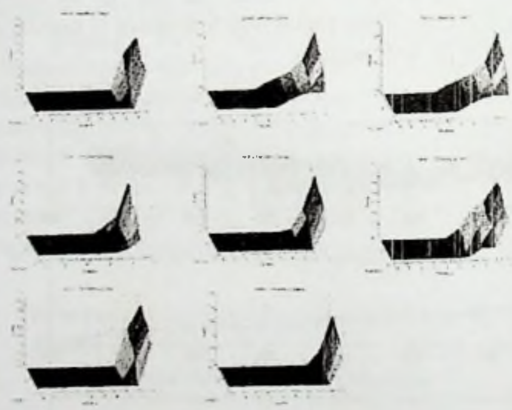

Fig. 9: Results for GA

According to the graphs most of the times solutions are converge to a same solution within five iterations. Then we limit number of iterations to five for the Genetic Algorithm.

2) Impact of Selection Size to the Global Energy of the Final Solution

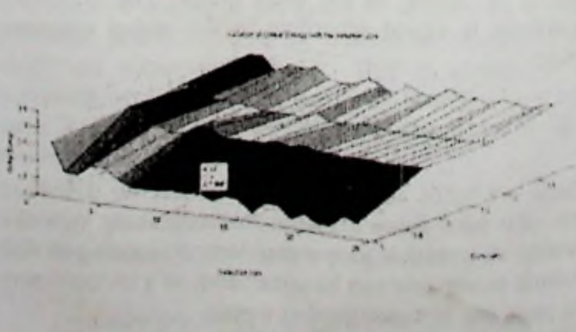

Fig. 10: Impact of Selection Size

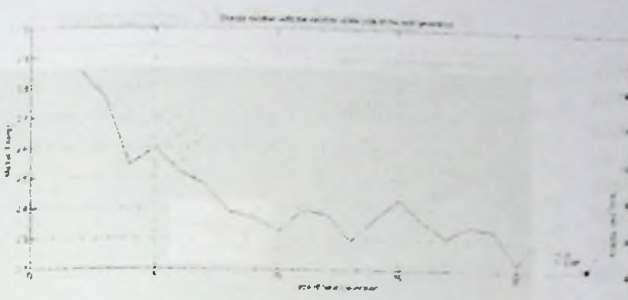

Fig. 11: Impact of Selection size

According to the graph, if the selection size is 23 it gives a better solution for all the test data set. Then In this algorithm we use 23 as the internal selection size.

B. Optimization for SA

Energy Variations with the Iterations:

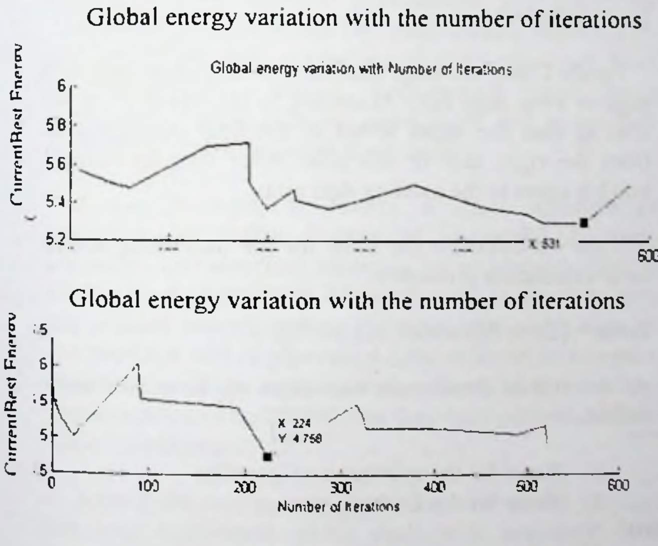

Fig. 12: Results for SA

Number of iterations that related to minimum energy for ten different data sets,

28, 81, 19, 531, 224, 121, 532, 12, 482, 147

Figure 22: show the results for the two of the data sets. After considering this data set, we selected 300 as the average iterations to get effective solution.

C. Comparison Between GA and SA

Fig. 13: Comparison

According to the above graph in most of the iterations, the final energy of the solution of the Simulated Annealing is lower than the energy of the Genetic Algorithm. Then for the existing conditions the procedure that used to implement the Simulated Annealing algorithm is better than the procedure that used to implement Genetic Algorithm.

D. Error Evaluation

There are two energy calculation processes for SA and GA algorithms. These two energy calculation process are specific to internal implementation of these two algorithms. Then we use a common energy assign function to check the fitness of the final solution. The procedure that used to calculate the energy is described below.

We use two major steps.

- Calculate the energy that related to the advertisement configuration of the page.
- Calculate the energy that related to the final appearance of the configuration

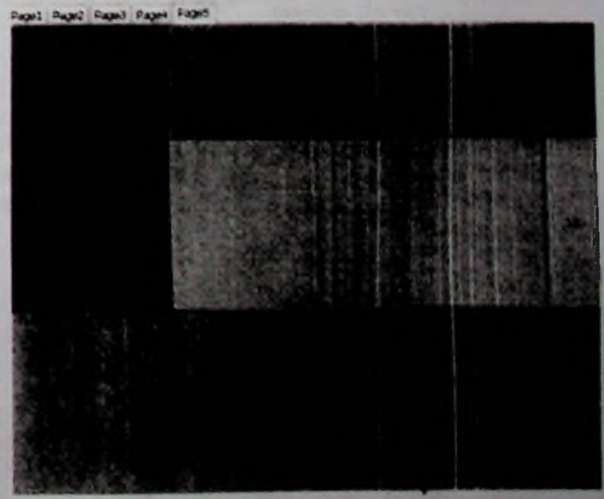

Fig. 14: Dummy Viewer

E. Energy of Advertisement Configurations

As the first step check the page number and check whether it is an odd numbered page or an even numbered page. If it is an odd numbered page, we start to read the advertisement from the bottom right corner and if it is an even numbered page, then start to read the advertisement from the bottom left corner. Likewise, after we checked the paper we read advertisements one by one and insert width of the advertisement to an array. It is the existing configuration.

Then we sort the existing configuration to descending order of width and it is considered as the expected order.

Using existing order and expected order we calculate the error for the configuration as described below.

Example:
Existing order = 7, 3, 7, 4, 6
Expected order = 7, 7, 6, 4, 3

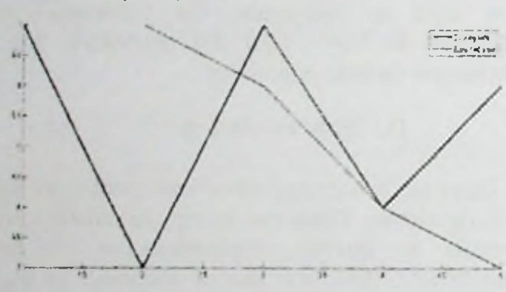

Fig. 15: Error Calculation

Then we calculate the difference between expected data and existing data. If the difference is greater than zero then we calculate the error for all the points, that difference is greater than zero.

$$Error = \sqrt{\sum_{for\ all\ spacess}^{diference>0} \frac{difference^2}{differenceToActPosition}}$$

differenceToActposition is the difference between the actual position and the existing position of the data. Likewise we estimate the error for the advertisment configuration.

F. Energy of the Final Configuration

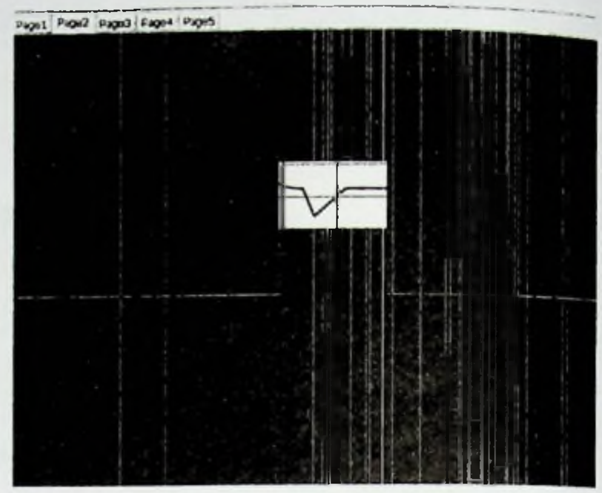

Fig. 16: Final Configuration

Process used to this procedure is also same as the above method. But only difference is instead of the width of the advertisement we read the level height of the each step [see the figure]

Firstly Check the page number and identify it as a odd page or even page first. According to the odd or even we start to read the upper levels of the final configuration from the right side or left side. After that these level heights insert to the existing data array.

Then we calculate the error for the data using above error calculation procedure

Error = [Error *(Number of Levels)]

At the end of these main two steps we have two error values

1. Error for the existing configuration
2. Error for the final appearance

Then we calculate the final total error

$$\text{Total Error of a page} = [(Energy\ for\ final\ appearence)^2 + (\text{Energy For Configuration})]$$

Then we calculate the Global energy of a section as follows

$$Global\ Energy = \left(\frac{\sum_{i=0}^{number\ of\ pages}(\text{Total Error of a page})}{Number\ of\ pages}\right) * (Number\ of\ Remain$$

When we consider the outcome of each of the algorithms below results could be gain.

V. DISCUSSION

At the time the project was only an idea; the aim of this project was simply to build an advertisement booking and tracking system. After analyzing the requirements with our client we could identify that this system should be the core mechanism of extensive process. The real newspaper publication process is highly complex with respect to its multiple numbers of time critical sub processes. Instead of building this project for the generic process we selected to build the system in a manner such that it suits to abstract newspaper publication process, so that, we will be able to customize it to precede with different rules and regulations of different newspaper publication organizations.

Basic idea of this research was to find a solution for optimized newspaper pagination. But since this problem is a NP-hard problem there is no particular best algorithm developed to solve the problem. So with our extensive research our target was to develop an algorithm which solves this problem at its best.

After a comprehensive development life cycle, we could satisfy all the initial key requirements expected by the system, giving us the feeling of an excellent job. With the progress of project development some basic requirements were changed accordingly and some were added to obtain the desired functionality.

As an example, in finding a space optimization algorithm we firstly thought to implement a single algorithm and searched literature for algorithms used in similar kind of problems. But during the literature survey we selected two algorithms among several which suits to the problem and implemented both of them to compare the statistics. Then we compared and contrasted these two implementations and found that they both perform well in different situations.

Among the major components we have in our system, the core component which deals with important data processing is the dummy creation component. We have the two algorithms Simulated Annealing and Genetic Algorithm implemented in to the system where user can select an algorithm to proceed. Though these two algorithms deal with highly mathematical procedures we could model it to our system with a large amount of test data. Since this newspaper pagination problem is a NP-hard problem and our algorithms try to optimize the newspapers advertisement space while preserving advertisement placing rules of publication organization these two algorithm implementations add a great value to our system.

Since this system automates a completely manual process it will be a huge improvement in efficiency of the newspaper publication process. And as this system is provided in highly user friendly manner with attractive user interfaces the people in newspaper publication industry will be able to familiarize with the system in short period of time.

By considering the results obtained through testing of the system we realized that the system has met with its primary requirements of the two algorithms.

During the project's development life cycle the greatest achievement was the implementation of two algorithms to solve a NP-hard problem. According to the feedbacks we had got from our supervisors as well as from our client our solution is mostly applicable to the problem and since these two algorithm implementations are done in a more generic manner it can be modeled to other space optimization problems as well such that it realizes the real potential of making out a great product into the real world. However it does not conclude the effort on building the ultimate model. As these two algorithms can be further improve by enhancing the mathematical functions used for energy calculations, with more tested data. Also adding some artificial intelligence related components into the system it can be use to dynamically suggest the best algorithm for each optimization problem. And without executing two algorithms separately it will be more efficient to develop a improved algorithm by combining Simulated Annealing and Genetic Algorithm.

Since we are hoping to release this system as an open source space optimization project it will be helpful for interested parties to continue with improving the solution.

ACKNOWLEDGEMENT

While doing this research we had to go on a big journey, meeting many challenges and overcoming those with the contribution of the people who surrounded us with helping hands and courageous words. So as a team, we would like to express our sincere thanks to all the expert personnel for their great contribution.

First we would like to thank our project coordinator Mr. Shantha Fernando for his continued presence and assistance throughout the project. The guidance we have been getting step by step has been actually helpful throughout our project.

Special thanks go out my supportive supervisor, Dr. Shahani Weerawarna. The supervision and support that she is giving truly helps the progression and smoothness of the project. The co-operation and courage we are getting is indeed much appreciated.

We would like to offer our sincere gratitude to Dr. Ravindra Koggalage for his excellent coordination for outstanding guidance throughout the project

REFERENCES

[1] Anne Bruggemann-Klein, Rolf Klein, Stefan Wohlfeil, "*Pagination Reconsidered*", Technische University, Munchen, Fern University, Hagen, 1996.

[2] Enrique Alba, Christian Blum, Pedro Isasi, Coromoto Leo'n, Juan Antonio Gomez, "*Optimization Techniques for Solving Complex Problems*", Hoboken, New Jersey John Wiley & Sons, Inc., 2009.

[3] The Internet Encyclopedia of Science, "*NP-hard Problem*". Available: http://www.daviddarling.info/encyclopedia/N/NP-hard_problem.html. [last accessed: 2009, Aug].

[4] Eric Weisstein, "*NP-Hard Problem – from Wolfarm MathWorld*". Available: http://mathworld.wolfram.com/NP-HardProblem.html [last accessed: 2009, Aug].

[5] M. Hofri, "*Probabilistic Analysis of Algorithms*", Springer-Verlag, New York, 1987.

[6] Emanuel Falkenauer, "*Genetic Algorithms and Grouping Problems*".

[7] E. Falkenauer, A. Delchambre, "*A Genetic Algorithm for Bin Packing and Line Balancing*", CRIF - Research Centre for Belgian Metalworking Industry, Brussels, Belgium, 2002.

[8] Xianghe Jing, Xianzhong Zhou, Yangyong Xu, "*A Hybrid Genetic Algorithm for Bin Packing Problem Based on Item Sequencing*", Dept. of Automation, Nanjing University of Science & Technology, Nanjing, China, Dept. of Information & Control, Air Defence Command College, Zhengzhou, China, School of Management & Engineering, Nanjing University, Nanjing, China, 2005.

[9] M. Tim Jones, "*AI Application Programming*", Hingham, Massachusetts: Charles River Media, 2003.

[10] Andrew Williamson, (1996, Nov). "*The Metropolis Algorithm*", Available: http://www.tcm.phy.cam.ac.uk/~ajw29/thesis/node18.html, [Last accessed: 2009, Sep]

[11] J. Gonzalez, I.Rojas, H. Pomares, M. Salmeron, A. Prieto, J.J. Merelo, "*Optimization of web newspaper layout in real time*". Department of Computer Architecture and Computer Technology, University of Granada, Granada, Spain, 2000.

[12] J. Gonzales, J.J. Merelo, P.A. Castillo, V. Rivas, G. Romero, A. Prieto, "*Optimizing Web Newspaper Layout using Simulated Annealing*". Department of Computer Architecture and Computer Technology, University of Granada, Granada, Spain, 2001.

[13] Krista Lagus. (1994). "*Fax Newspaper Prototype*". Available: http://www.cis.hut.fi/krista/personal/dippa/node18.html, [Last accessed: 2009, Aug].

[14] Krista Lagus, "*Automated pagination of the generalized newspaper using Simulated Annealing*, Master's thesis, Otaniemi, Finland, 1995.

[15] J.H. Holland, "*Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to biology, control and artificial intelligence*." MIT Press, ISBN, 1975.

[16] Krista Lagus, Ilkka Karanta, Juha Yla-Jaaski, "*Paginating the generalized newspaper (a comparison of Simulated Annealing and a heuristic method)*", VTT Information Technology, FIN-02044 VTT, Finland, 1996.