Chapter 1

Introduction

1.1 Introduction

Graph layout problems (GLPs) belong to the set of combinatorial optimization problems whose aim is to find a labeling of the vertices of an input graph with distinct natural numbers in such a way that a certain objective function is optimized. In its simpler form, a layout is an embedding of a graph into line or circle. A large number of relevant problems in different domains can be formulated as graph layout problems. These include optimization of networks for parallel computer architecture, VLSI circuit design, information retrieval, numerical analysis, computational biology, scheduling and archaeology [JDJPMS, 2002]. VLSI circuit design has enabled the construction of very complex interconnection networks in recent years. The management of communication becomes very complex as the number of units increases. Hence search of a suitable topological structure for the interconnection networks has become important [RS, 2007]. On the other hand, in the area of numerical analysis, reordering the rows and columns of large sparse symmetric matrices is desired such that their non-zero entries lie as close as possible to the diagonal.

Though many of the graph layout problems are NP-hard [JDJPMS, 2002] and traditional optimization software (Cplex solver [ADRMMGCRRMAS, 2011]) have been inefficient for large graphs, yet a feasible solution with an almost optimal cost is adequate for most of their applications. Thus, in practice, approximation algorithms and effective heuristics are also acceptable. In this context, metaheuristic search techniques, referred to as just metaheuristic techniques here, offer much promise.
This thesis focuses on the design and development of metaheuristic techniques for some of the graph layout problems. In this work, the usefulness of these techniques has been assessed by extensive experiments conducted on standard graphs, benchmark graphs taken from matrix market collection [Harwell], randomly generated connected graphs (RCGs) and Rome graphs [GDToolkit]. The efficacy of a metaheuristic is measured by comparing the obtained results with known optimal results available in the literature. The metaheuristics have thus also been simulated on those classes of graphs for which the optimal values of the objective function is unknown in the literature surveyed by us. Further, trends observed in the results obtained by simulation of metaheuristics on these graphs have been examined for conjectures.

Further in this chapter the graph layout problems are discussed in Section 1.2. Section 1.3 discusses the motivation behind graph layout problems and the organization of thesis is presented in Section 1.4.

1.2 Graph Layout Problems

A graph $G = (V, E)$ consists of a set of vertices $V = \{ v_1, v_2, \ldots \}$ and the set of edges $E = \{ e_1, e_2, \ldots \}$, such that each undirected edge $e_k$ is identified with an unordered pair of vertices $(v_i, v_j)$ and is denoted by $v_i v_j$.

A graph is finite if $V$ and $E$ are finite sets. In this work, graphs are assumed to be finite, undirected and without loops.

The edge density of a graph $G$ with $n$ vertices and $m$ edges is taken as $2m/n(n-1)$.

A layout (labeling / ordering) of an undirected graph $G = (V, E)$ with $n = |V|$ vertices, is a bijective function $\varphi : V \rightarrow \{1, 2\ldots n\}$. The set of all layouts of a graph $G$ is denoted by $\psi(G)$ and clearly $|\psi(G)| = n!$. In the context of graph drawing, a labeling corresponds to placing its vertices along a line and the edges being drawn as semi circles.
The length of an edge \( uv \) for the layout \( \phi \) is defined as
\[
\lambda (uv, \phi) = |\phi(u) - \phi(v)|, \quad uv \in E.
\]

The edge cut at position \( i \) (\( i \in [1,n] \)) of \( \phi \) is defined as
\[
\theta(i, \phi) = |\{uv \in E: u \in L(i, \phi) \land v \in R(i, \phi) \}|
\]
and the modified edge cut at position \( i \) of \( \phi \) as
\[
\zeta(i, \phi) = |\{uv \in E: u \in L(i, \phi, G) \land v \in R(i, \phi, G) \land \phi(u) \neq i \}|
\]
where \( L(i, \phi) = \{u \in V: \phi(u) \leq i\} \) and \( R(i, \phi) = \{u \in V: \phi(u) > i\} \).

For a layout \( \phi \) of a graph \( G = (V, E) \), its reversed layout is denoted \( \phi^R \) and is defined by
\[
\phi^R = |V| - \phi(u) + 1 \quad \text{for all } u \in V.
\]

Layout cost is a map \( \psi : \varphi(G) \to Z \), that associates to each layout \( \phi \) of a graph \( G \), an integer \( T(\phi, G) \). GLP associated with \( T \) consists in determining some layout \( \varphi^* \) of an input graph \( G \) such that \( T(\varphi^*, G) \) is minimum/maximum among all \( n! \) layouts. Thus, each of the GLP is defined by some objective function whose value depends on the layout of the graph.

The layout problems considered in this thesis are related to embedding vertices of the graph along the line as well as along the circle. For the case when the vertices are placed along a cycle, the length of an edge \( uv \) for the layout \( \phi \) is defined as
\[
\lambda_c (uv, \phi) = \min (|\phi(u) - \phi(v)|, n - (|\phi(u) - \phi(v)|))
\]
which will be referred to as cyclic length of \( uv \). If the line is treated as the spine of a book then the edges may be drawn as half circles on the pages of the book. Such a drawing is known as the book embedding of a graph. Two of the problems considered in this work, are based on the book embedding of graphs.

We now present various GLPs with the help of an example (Figure 1.1). In Figure 1.1 \( V = \{a, b, c, d, e, f, g, h, i\} \) and the edges are as shown in the figure. A layout of the graph results in placing its vertices along the line (spine) (Figure 1.2) and for another example along a circle (cycle) (Figure 1.3). Further, for a given labeling of vertices the edges can be distributed on the pages of a book as shown in Figure 1.4. Edges drawn on a page may cross as shown in Figure 1.5. In Figure 1.4, the black edges drawn above the spine are on the first page and those below
the spine correspond to the second page. Edges of third page are shown in red colour above the spine.

Figure 1.1: An example graph with nodes and edges

Figure 1.2: Drawing of the example graph corresponding to a labeling and one of the cut points
Bandwidth Minimization Problem (BMP):

Bandwidth of a graph $G$ for layout $\phi$ is defined as

$$BW(\phi, G) = \max_{uv \in E} \lambda(uv, \phi)$$

and BMP is to find a layout which minimizes $BW(\phi, G)$ among all the layouts $\phi \in \psi(G)$. In Figure 1.2, the edge $ai$ has the maximum length which is 8.
Minimum Linear Arrangement (MinLA) or Bandwidth Sum Minimization Problem (BSP):

Bandwidth sum of a graph $G$ for layout $\varphi$ is defined as

$$BS(\varphi, G) = \sum_{uv} \lambda(uv, \varphi)$$

and BSP is to find a layout which minimizes $BS(\varphi, G)$ among all the layouts $\varphi \in \psi(G)$. For the layout shown in Figure 1.2, $BS$ is 45.

Cutwidth Minimization Problem (CMP):

Cutwidth of a graph $G$ for layout $\varphi$ is defined as

$$CW(\varphi, G) = \max_i \zeta(i, \varphi)$$

and CMP is to find a layout which minimizes $CW(\varphi, G)$ among all the layouts $\varphi \in \psi(G)$. In Figure 1.2, the edge $cg$ has maximum value of $CW$, i.e. 7.

Antibandwidth Maximization Problem (AMP):

Antibandwidth of a graph $G$ for layout $\varphi$ is defined as

$$ABW(\varphi, G) = \min_{uv \in E} \lambda(uv, \varphi)$$

and AMP is to find a layout which maximizes $ABW(\varphi, G)$ among all the layouts $\varphi \in \psi(G)$. In Figure 1.2, the edges $de$ and $hc$ have minimum length which is 1.

Cyclic Antibandwidth Maximization Problem (CAMP):

Cyclic antibandwidth of a graph $G$ for layout $\varphi$ is defined as

$$CAB(\varphi, G) = \min_{uv \in E} \lambda_c(uv, \varphi).$$

CAMP is to find a layout which maximizes $CAB(\varphi, G)$ among all the layouts $\varphi \in \psi(G)$. For the layout shown in Figure 1.3, cyclic antibandwidth is 1 (between vertices 6 and 7).

Profile Minimization Problem (PMP):

Profile of a graph $G$ for layout $\varphi$ is defined as

$$PR(\varphi, G) = \sum_{u \in V} \left( \varphi(u) - \min_{v \in \Gamma^*(u)} \varphi(u) \right)$$

where,

$$\Gamma^*(u) = \{u\} \cup \{v \in V : uv \in E\}$$

and PMP is to find a layout which minimizes $PR(\varphi, G)$ among all the layouts $\varphi \in \psi(G)$. In Figure 1.2 profile is 20.

K-page Crossing Number Minimization Problem (KPMP):

KPMP is a book embedding problem. It is concerned with finding a layout of vertices and distribution of edges on $K$ pages of a book such that the total number
of crossings (see crossing condition in Figure 1.5) between the edges are minimized. In Figure 1.2 the graph is drawn on a 2 page book where edges drawn above the spine correspond to first page of the book and those below the spine correspond to second page. Here number of crossing among all the edges on the two pages is 4.

**Pagenumber Minimization Problem (PNP):**

PNP is another book embedding problem wherein the vertices are placed along the spine and the edges are distributed on the minimum number pages of the book in such a way that the edges do not cross on any page. In Figure 1.4, minimum number pages used to embed the graph are three.

**Cyclic Bandwidth Minimization Problem (CBMP):**

Cyclic bandwidth of a graph $G$ for layout $\phi$ is defined as

$$CBW(\phi, G) = \max_{uv \in E} \lambda_c (uv, \phi)$$

and CBMP is to find a layout which minimizes $CBW (\phi, G)$ among all the layouts $\phi \in \psi (G)$. In Figure 1.3, vertices and edges of a graph are placed along a circle. $CBW$ in this case is 4 (between vertices 7 and 11, 10 and 2, 7 and 3).

**Cyclic Bandwidth Sum Minimization Problem (CBSP):**

Cyclic bandwidth sum of a graph $G$ for layout $\phi$ is defined as

$$CBS (\phi, G) = \sum_{uv \in E} \lambda_c (uv, \phi)$$

and CBSP is to find a layout which minimizes $CBS (\phi, G)$ among all the layouts $\phi \in \psi (G)$. In Figure 1.3, for the given layout, cyclic bandwidth sum is 30.

GLPs along with their complexity are summarized in Table 1.1.
Table 1.1: Graph layout problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>Alternate Names</th>
<th>Cost</th>
<th>Optimality Type</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth minimization problem (BMP)</td>
<td>-</td>
<td>$\text{BW}(\phi, G) = \max_{uv \in E} \lambda(uv, \phi)$</td>
<td>Minimization</td>
<td>NP-complete [CHP, 1976]</td>
</tr>
<tr>
<td>Minimum Linear Arrangement problem (MinLA)</td>
<td>Optimal linear ordering, Edge sum problem, Bandwidth sum, Minimum-1-sum</td>
<td>$\text{LA}(\phi, G) = \left{ \sum_{uv \in E} \lambda(uv, \phi), \sum_{i=1}^n \theta(i, \phi) \right}$</td>
<td>Minimization</td>
<td>NP-hard [JDJPMS, 2002]</td>
</tr>
<tr>
<td>Cutwidth minimization problem (CMP)</td>
<td>Min cut linear arrangement, Network migration scheduling</td>
<td>$\text{CW}(\phi, G) = \max_{i=1}^n \zeta(i, \phi)$</td>
<td>Minimization</td>
<td>NP-complete [JDJPMS, 2002]</td>
</tr>
<tr>
<td>Profile minimization problem (PMP)</td>
<td>Envelope size</td>
<td>$\text{PR}(\phi, G) = \sum_{u \in V} \left( \varphi(u) - \min_{v \in \Gamma^*(u)} \varphi(v) \right)$</td>
<td>Minimization</td>
<td>NP-complete [YLJY, 1994]</td>
</tr>
<tr>
<td>Antibandwidth Separation number</td>
<td></td>
<td>$\text{ABW}(\phi, G) = \min_{uv \in E} \lambda(uv, \phi)$</td>
<td>Maximization</td>
<td>NP-hard [ARHSOS]</td>
</tr>
<tr>
<td>Problem Description</td>
<td>Equation</td>
<td>Complexity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximization problem (AMP)</td>
<td>Dual bandwidth</td>
<td>Maximization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cyclic Antibandwidth maximization problem (CAMP)</td>
<td>$\text{CAB}(\varphi, G) = \min_{uv \in E} \lambda_c(\varphi) \lambda_c(\varphi)$</td>
<td>NP-hard</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cyclic Sum minimization problem (CSMP)</td>
<td>Cyclic bandwidth sum, directed circular arrangement, minimum circular arrangement</td>
<td>NP-Hard</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cyclic Bandwidth minimization problem (CBMP)</td>
<td>$\text{CBW}(\varphi, G) = \max_{uv \in E} \lambda_c(\varphi)$</td>
<td>Minimization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$K$-page crossing number minimization problem (KPMP)</td>
<td>Total number of edge crossings over all $K$-page book drawings of $G$</td>
<td>Minimization</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>where crossing condition is, $1 \leq \varphi(u) &lt; \varphi(p) &lt; \varphi(v) &lt; \varphi(q) \leq n$ , $uv$ and $pq$ are a pair of edges and both lying</td>
<td>NP-hard</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pagenumber minimization problem (PNP)</td>
<td>Book embedding</td>
<td>Total number of pages of the book into which $G$ can be embedded without crossing of edges</td>
<td>Minimization</td>
<td>NP-complete [NKMRIS, 2002]</td>
</tr>
</tbody>
</table>

### 1.3 Motivation

Layout problems have received attention due to the early application of these problems to the optimal layout of circuits. GLPs have applications in different fields, e.g., BMP corresponds to speeding up several computations on sparse matrices, MinLA is useful in *designing error-correcting codes* with minimal average absolute errors on certain classes of graphs ([LHH, 1964], [LHH, 1966]). MinLA was also considered as an over-simplified model of some nervous activity in the cortex by Mitchison and Durbin [GMRD, 1986]. This also has applications in *single machine job scheduling* [DLA, 1977] and [RRAAPK, 1991] and in *graph drawing* [FSOSLASIV, 2001]. For the aesthetic drawing of bipartite graphs on two layers it is required that the number of crossings between the edges is minimized. Shahrokhi et al. [FSOSLASIV, 2001] have proved that the problem of reducing the crossings is equivalent to reducing the total edge length, i.e. the linear arrangement problem. PMP was proposed as a way to reduce the amount of storage of sparse matrices. PMP turned out to be equivalent to the Interval Graph Completion problem [RRAAPK, 1991], which has applications in *archaeology* and *clone fingerprinting* [JDJPMS, 2002]. In numerical analysis reordering of rows and columns of very large sparse symmetric matrices is needed such that their non-zero entries lies as close as possible to the diagonal, which has importance in some engineering applications. Figures 1.6 and 1.7 show respectively the non-zero entries of a graph in the adjacency matrix of a graph before and after the bandwidth has been reduced. Here each dot represents a non-zero entry in the corresponding adjacency matrix of the graph. These Spy graphs are generated by MATLAB. PNP has applications in *Direct Interconnection*
Networks, Fault-Tolerant Processor Arrays, Sorting with Parallel Stacks, Single-
Row Routing and Ordered Sets [NKMRIS, 2002]. In Direct Interconnection
Networks an optimal labeling for processors and wires in VLSI implementation of
interconnection networks is needed. Processors are arranged on a line and bundles
(stacks) of wires with embedded switches run parallel to the line of processors
where minimum number of stacks are desired which is similar to PNP. PNP is also
similar to simplifying the problem of routing multilayer printed circuit boards
(PCBs) [NKMRIS, 2002].

Many layout problems are originally motivated as mathematical model of VLSI
layout. For a set of modules the VLSI layout problem consists in placing the
modules on a board in a non-overlapping manner. The wiring, together with
placement of the terminals on different modules is done according to a given
wiring specification and in such a way that wires do not interfere among them.
The placement problem consists in placing the modules on a board; the routing
problem consists in wiring together the terminals on different modules that should
be connected [RS, 2007]. If a VLSI circuit is modeled by a graph then the non
overlapping manner of wires is similar to PNP, where the number of crossings
between edges is not allowed. On the other hand if wiring can be done with
minimum overlapping of wires on different levels (pages of a book along the
spine) then it is similar to KPMP where the number of crossings between edges of
a graph needs to be reduced. Also minimizing the total wire length in the
placement phase is equivalent to MinLA problem. The cutwidth of a graph times
the order of graph gives a measure of the area needed to represent the graph in a
VLSI layout when vertices are laid out in a line.
1.4 Organization of Thesis

Chapter 1 is an introduction to the thesis. It briefly explains graph layout problems (GLPs), preliminaries and definitions used throughout the thesis. It explains motivation behind studying these problems and organization of the thesis.

Chapter 2 is dedicated to literature survey of GLPs. It gives a brief introduction of the metaheuristics used to solve GLPs and their working details. This is followed
by a survey of graph layout problems solved by these techniques along with their results.

Chapter 3 presents the general methodology used for the experiments. The methodology used at intermediate stages as well as in final experimentation is explained in this chapter. The chapter also presents the test graphs used for experimentation.

Chapter 4 addresses the pagenumber minimization problem (PNP). In this chapter a statistical evaluation of vertex ordering heuristics for minimum number of edge crossings and edge distribution heuristics for minimum number of pages is carried out. Using the results of vertex ordering heuristics and edge distribution heuristics a guided evolutionary simulated annealing (GESA) algorithm is implemented. The results of experiments with GESA are carried out on selected standard and random graphs. Its performance is also compared with an existing genetic algorithm for PNP.

Chapter 5 presents an improvement to the techniques presented in Chapter 4. A reduced variable neighbourhood search (RVNS) is designed for PNP and compared with GESA. RVNS starts with a single ordering. We have further designed a memetic algorithm (MA) which embeds features of VNS and genetic algorithm. A comparison of all the techniques is done on a test suite of known as well as unknown graphs. An important contribution of this work is a set of conjectures for the pagenumber of some classes of graphs with unknown optimal results obtained by the experimental results. These conjectures are also proved in this chapter.

In chapter 6, $K$-page crossing number minimization problem (KPMP) is considered. In this chapter a statistical evaluation of edge distribution heuristics for distributing edges on $K$-pages with minimum number of crossings for a fixed ordering of vertices on the spine is carried out. An implementation of guided evolutionary simulated annealing for solving KPMP and experimental results are presented in this chapter.
Chapter 7 covers another GLP, namely, the cyclic bandwidth sum minimization problem (CBSP). The GLPs considered in the earlier chapters deal with the ordering of the vertices along a path (line), but chapters 7 and 8 are dedicated to cyclic problems. To solve CBSP some metaheuristics namely, modified scatter search (MSS), general variable neighbourhood search (GVNS), a variant of GVNS, reduced variable neighbourhood search (RVNS), and genetic algorithm are used. Extensive experiments are conducted and results are presented on different classes of graphs with known cyclic bandwidth sum besides some other classes of graphs.

In chapter 8, cyclic bandwidth minimization problem (CBMP) is considered. A memetic algorithm (MA), ant colony optimization (ACO), greedy randomized adaptive search procedure (GRASP), and genetic algorithm (GA) are applied to solve the problem. Extensive experimentation is carried out and some conjectures based upon the experimentation are also presented.

Chapter 9 presents conclusion of the research.