On the General Graph Embedding Problem
With Applications to Circuit Layout

Jonathan S. Turner

WUCS-84-1
February 1984

ABSTRACT

We consider the problem of embedding one graph in another, where the cost of an embedding is the maximum distance in the target graph separating vertices that are adjacent in the source graph. An important special case, known as the bandwidth minimization problem, is when the target graph is a path. This author has shown that for random graphs having bandwidth at most k, a well-known heuristic produces solutions having cost not more than $3k$ with high probability. This paper considers generalizations of this heuristic and analyzes their performance in other classes of target graphs. In particular, we describe a heuristic that for random graphs having a cost k embedding in a rectangular grid, produces embeddings having cost not more than $3k$ with high probability. This problem has applications to laying out circuits in the plane so as to minimize the length of the longest wire. Similar results can be obtained for multi-dimensional grids, as well as triangular and hexagonal grids.

On the General Graph Embedding Problem
With Applications to Circuit Layout

Jonathan S. Turner
Washington University

1. Introduction

We define a graph $G = (V, E)$ as a set of vertices $V = \{1, \ldots, n\}$ and a set of edges E consisting of unordered pairs of vertices. Let $H = \{H_m = (W_m, F_m) : m \geq 1\}$ be an infinite family of graphs. An *embedding* of a graph $G = (V, E)$ in H is a function $\tau : V \to W_m$ for some $m \geq 1$. The *embedding cost* of G with respect to τ (and H) is given by

$$\varphi_\tau(G) = \max\{d(\tau(u), \tau(v), H_m) : \{u, v\} \in E\}$$

where $d(x, y, J)$ is the length of the shortest path joining vertices x and y in the graph J. (We will usually write just $d(x, y)$, when the graph in question is clear from the context.) For example, Figure 1 shows embeddings of a graph G in the line graph L_8 and the rectangular grid R_3 having costs 6 and 3 respectively. The embedding cost of G (with respect to H) is given by

$$\varphi(G) = \min\{\varphi_\tau(G) : \tau \text{ is an embedding of } G \text{ in } H\}$$

Several other authors have studied this embedding problem. Rosenberg [8,9], DeMillo, Eisenstat and Lipton [3] and Hong and Rosenberg [5] consider embeddings of graphs in trees. Hong, Melhorn and Rosenberg [6] consider trade-offs between the cost measure used here (which they call the *dilation cost*) and the ratio of the sizes of the target graph to the source graph (which they call the *expansion cost*). Aleliunas and Rosenberg [1] study embeddings of rectangular grids of arbitrary aspect ratio in square grids, which has applications to circuit layout.

In this paper, we consider the probable performance of heuristics for graph embedding. This is based on earlier work by this author on the bandwidth minimization problem [10,11]. We describe a class of heuristics that can be applied to many families of target graphs, and establish a framework for analyzing their performance for specific families. In particular, we show that there is an algorithm which for random graphs having a cost k embedding in a rectangular grid will produce embeddings having cost not more than $3k$. Similar results can be obtained for other regular structures, such as triangular and hexagonal grids and multi-dimensional grids of various sorts.

2. Review of Results for Bandwidth Minimization

Before proceeding, we briefly review earlier results for the bandwidth minimization problem, that provide the foundation for what is to follow. Let $L = \{L_m = (W_m, F_m) : m \geq 1\}$ be the family of paths or line graphs (see Figure 1). In this section, the embeddings are understood to be with respect to L. The resulting special case of the graph embedding problem is called the bandwidth minimization problem. It is known principally for its application to matrix bandwidth minimization and is known to be NP-complete [4,7]. In 1969, Cuthill and McKee

[2] proposed a heuristic that has met with great practical success in systems that routinely perform bandwidth minimization on large matrices. The first convincing analytical explanation for its success appeared only recently in [10,11]. The algorithm described by Cuthill and McKee is a member of the class of *level algorithms*. An algorithm is classified as a level algorithm if for all graphs $G = (V,E)$, the embedding τ produced by the algorithm satisfies

$$d(\tau^{-1}(1),u) < d(\tau^{-1}(1),v) \Rightarrow \tau(u) < \tau(v)$$

for all vertices u, v in V (the vertices of L_m are numbered consecutively, with one of the end vertices numbered 1). Figure 2 shows an embedding of a graph G in L_{14} that was produced by a level algorithm. The level algorithms arrange the vertices in the order of their distance from a starting vertex, which is mapped to one endpoint of the target graph. Notice that the cost of the embedding is at least as large as the largest set of vertices that are at the same distance from the starting vertex. The cost is smaller than the largest set of vertices whose distance from the starting vertex differs by at most one. The lower bound can be used to show that the performance of the level algorithms can be arbitrarily bad in the worst case.

Let $G = (V,E)$ be a random vertex graph $\{1, \ldots, n\}$, in which edges are generated independently with probability p. This distribution is denoted $\Gamma_n(p)$, and we say that $G \in \Gamma_n(p)$, meaning that G is a random variable generated by such an experiment. It is shown in [11] that for almost all such graphs G, $\varphi(G) > n - o(n)$; that is, most graphs have only very expensive embeddings in L_m. To understand the practical success of the level algorithms, we're forced to consider probability distributions that allow us to focus on random graphs with small bandwidth. Using such distributions, we can discover properties that occur frequently in small bandwidth graphs and can be exploited by heuristics such as the level algorithms.

Let $G = (V,E)$ be a graph in $\Gamma_n(p)$ and let ψ be a positive integer. Select at random an embedding τ of G in L_n and remove from E all edges $\{x,y\}$ such that $|\tau(x) - \tau(y)| > \psi$. The probability distribution described by this experiment is denoted $\Omega_n(\psi,p)$ and we write $G \in \Omega_n(\psi,p)$ to denote that G is a random graph generated in this way. Note that $\varphi(G) \leq \psi$. In [11], it was shown that there is a level algorithm which for random graphs $G \in \Omega_n(\psi,p)$, produces an embedding with cost at most 3ψ, with high probability. The key to the proof is a probabilistic upper bound on the size of the sets of vertices that are equidistant from the starting vertex. Similar results are given in the next section for a generalized level algorithm for embedding graphs in grids.

3. Embedding Graphs in Grids

In this section, we consider the problem of embedding graphs in rectangular grids. We are interested in efficient heuristics with good probable performance. One possibility is suggested by the level algorithms for embedding graphs in lines. The level algorithms select a starting vertex, which is placed at one endpoint, and then place the remaining vertices based on their distance from the starting vertex. We can use a similar strategy to embed graphs in grids, but two starting vertices are needed. These are placed at non-opposite corners of the target rectangular grid, then the remaining vertices are placed depending on the pair of distances from the two starting vertices. The idea is illustrated in Figure 3. The diamond shaped regions in the top figure contain vertices that are equidistant from the two starting vertices in the lower corners. The bottom

figure shows a mapping onto an 8×8 grid.

Let $R = \{R_m : m \geq 1\}$ where $R_m = (W_m, F_m)$ is an $m \times m$ rectangular grid. Two non-opposite corner vertices are designated the *basis* of R_m and are numbered 1 and 2. Every other vertex is assigned a unique integer between 3 and $n = m^2$.

Let A be an algorithm that produces an embedding of a graph in some rectangular grid. We say that A is a *level algorithm* if for all graphs $G = (V, E)$, the embedding $\tau : V \to W_m$ produced by A satisfies the following properties.

(1) There are vertices u, v in V such that $\tau(u) = 1$ and $\tau(v) = 2$.

(2) For all x, y in V

$$d(u,x) < d(u,y) \Rightarrow d(1,\tau(x)) \leq d(1,\tau(y))$$
$$d(v,x) < d(v,y) \Rightarrow d(2,\tau(x)) \leq d(2,\tau(y))$$

As in the case of mapping graphs onto lines, the size of the sets of equidistant vertices provides a mechanism for proving a lower bound on the performance of any level algorithm. This bound is contained in our next lemma. Define

$$V(d_1, d_2, u_1, u_2) = \{x \text{ in } V : d(u_1, x) = d_1 \text{ and } d(u_2, x) = d_2\}$$
$$Max(u_1, u_2) = \max \{ |V(d_1, d_2, u_1, u_2)| : d_1, d_2 \geq 0 \}$$
$$Max_2(G) = \min \{ Max(u_1, u_2) : u_1, u_2 \in V \}$$
$$level(G) = \sqrt{2 Max_2(G) - 1} - 1$$

Lemma 3.1. Let $G = (V, E)$ be a graph and let τ be an embedding of G produced by a level algorithm. $level(G) \leq \varphi_\tau(G)$.

We can use this lemma to show that the level algorithms can in the worst-case produce solutions that differ from optimal by an arbitrarily large factor. For example, Figure 4 shows a tree T with $\varphi(T) = 1$ and $level(T) = 4$.

Our next lemma shows that there are level algorithms that can produce solutions with cost close to the lower bound of Lemma 3.1. Define $LEVEL(G) = 2\sqrt{2 Max_2(G)} + 2$.

Lemma 3.2. There is a level algorithm which for any graph G produces an embedding τ that satisfies $\varphi_\tau(G) \leq LEVEL(G)$.

Our first theorem gives a probable lower bound on the embedding cost of a random graph.

Theorem 3.1. Let $0 < p < 1$ be fixed, m be the smallest integer larger than $\sqrt{n}$. For almost all G in $\Gamma_n(p)$, $\varphi(G) \geq 2m - o(m)$.

Theorem 3.1 implies that few graphs have good embeddings in rectangular grids. We now turn to the study of random graphs that do have good embeddings in the hope of discovering properties that can be used to construct good embeddings. The first task is to select a probability distribution. Let $G = (V, E)$ be a random graph in $\Gamma_n(p)$, where $n = m^2$ for some positive integer m, and let ψ be a positive integer. Select at random an embedding τ of G in H_m and remove from E all edges $\{u, v\}$ such that $d(\tau(u), \tau(v)) > \psi$. The distribution implied by this experiment is denoted $\Omega_n(\psi, p)$. Note that $\Omega_n(\psi, p)$ generates only graphs with embedding cost $\leq \psi$. The next theorem shows that most of these graphs have embedding cost close to ψ.

Theorem 3.2. Let $0 < p < 1$ be fixed, $n = m^2$, ψ be an unbounded function of n, $\psi \leq 2(m-1)$. For almost all G in $\Omega_n(\psi, p)$, $\varphi(G) \geq \psi - o(\psi)$.

The main theorem of this section gives a probable upper bound on $LEVEL(G)$.

Theorem 3.3. Let $0<p<1$ be fixed. $n=m^2$. $\ln n = o(\psi^2)$. $\psi \leq 2(m-1)$. For almost all $G \in \Omega_n(\psi,p)$. $LEVEL(G) < 3\varphi(G)$.

This is proved by a probable upper bound on $Max_2(G)$. The spirit of the proof is similar to that of Theorem 3.1 in [11], although the details are more involved. Taken together with Lemma 3.2, this implies that there is a level algorithm that produces embeddings within a factor of three of optimal with high probability.

4. The General Problem

We now turn to the general problem of embedding one graph in another. We describe a generalized level algorithm for graph embedding and give a framework that can be used to evaluate its performance for specific families of target graphs.

A *basis* for a graph $J=(W,F)$ is a subset $\{v_1, \ldots, v_r\}$ of W such that for every u in W, the vector $[d(u,v_1), \ldots, d(u,v_r)]$ is unique. An *r-basis* is a basis with r vertices. We say that a family of graphs has an r-basis if each of its members has an r-basis.

Let $H = \{H_m : m \geq 1\}$ be a family of graphs where $H_m = (W_m, F_m)$. Each H_m is assumed to have an r-basis $\{1, \ldots, r\}$. In this section, we consider only embeddings in H.

Let A be an algorithm that embeds graphs in H. We say that A is a *generalized level algorithm* if for any graph $G=(V,E)$, the embedding τ produced by A satisfies the following.

(1) For $1 \leq x \leq r$, there is a vertex u in V such that $\tau(u)=x$.

(2) For $1 \leq x \leq r$ and all vertex pairs u, v in V

$$d(\tau^{-1}(x),u) < d(\tau^{-1}(x),v) \Rightarrow d(x,\tau(u)) \leq d(x,\tau(v))$$

Notice that the line graph L_m has a 1-basis, the rectangular grid R_m has a 2-basis and the r-dimensional rectangular grid has an r-basis. Also notice that the triangular grid shown in Figure 4 has a 2-basis and each of its r-dimensional analouges has an r-basis. For all of these classes of target graphs, one can prove theorems like those of section 2. In particular, one can show that with high probability, the generalized level algorithms produce embeddings having costs within a small constant factor of optimal for randomly generated graphs. This observation provides the motivation for a detailed study of the performance of the generalized level algorithms in arbitrary families of target graphs.

Let $G = (V,E)$ be a graph. Define

$$V(d_1, \ldots, d_r, u_1, \ldots, u_r) = \{x \text{ in } V : d(u_i,x) = d_i \ 1 \leq i \leq r\}$$

$$Max(u_1, \ldots, u_r) = \max \{ |V(d_1, \ldots, d_r, u_1, \ldots, u_r)| : d_1, \ldots, d_r \geq 0\}$$

$$Max_r(G) = \max \{Max(u_1, \ldots, u_r) : u_1, \ldots, u_r \in V\}$$

If a level algorithm makes the best possible choice for the r vertices to be mapped to the basis vertices of a graph in H, then $Max_r(G)$ is the size of the largest set of vertices that are equidistant from the selected set of r. Now, let $J = (W,F)$ be any graph with an r-basis $\{1, \ldots, r\}$. Let W' be a subset of W and define the *breadth* of W' as follows.

$$B(W') = \max \{d(x,u) - d(x,v) : u,v \in W' \text{ and } 1 \le x \le r\}$$

Define the *h-breadth* of J as

$$B_h(J) = \min \{B(W') : W' \text{ is an } h \text{ vertex subset of } W\}$$

If H is a family of graphs,

$$B_h(H) = \min \{B_h(J) : J \in H\}$$

Finally, let $level(G) = B_h(H)$ where $h = Max_r(G)$.

Lemma 4.1. Let G be a graph and let τ be an embedding of G produced by a level algorithm. $\varphi_\tau(G) \ge level(G)$.

One can use the lemma to show that for many classes of target graphs, the performance of the level algorithms can differ from optimal by an arbitrarily large factor. The construction used to show this is indicated in Figure 5. For any family of target graphs H with an r-basis, $level(T_{s,r}) = s$. For many such classes, $\varphi(T_{s,r}) \le 2$. In particular, this is true of r-dimensional grids. The next lemma is an generalization of Lemma 3.2; it gives an upper bound on the performance possible with certain level algorithms. For $H_m = (W_m, F_m)$, define

$$W_{m,h}(d_1, \ldots, d_r) = \{y \in W_m : h(d_x - 1) < d(x,y) \le hd_x \text{ for } 1 \le x \le r\}$$

Let $G = (V,E)$ be a graph and let h be the smallest integer for which there is a subset $\{u_1, \ldots, u_r\}$ of V and an integer m that satisfy

$$|V(u_1, \ldots, u_r, d_1, \ldots, d_r)| \le |W_{m,h}(d_1, \ldots, d_r)|$$

for all $d_1, \ldots, d_r \ge 0$. Finally, let

$$LEVEL(G) = \max \{d(u,v) : u \in W_{m,h}(d_1, \ldots, d_r) \text{ and } v \in W_{m,h}(f_1, \ldots, f_r)$$
$$\text{where } |d_i - f_i| \le 1 \text{ for } 1 \le i \le r\}$$

Lemma 4.2. There is a level algorithm which for any graph G produces an embedding τ that satisfies $\varphi_\tau(G) \le LEVEL(G)$.

The definitions and lemmas of this section provide a uniform framework that can be used to analyze the probable performance of level algorithms for various classes of target graphs. As indicated above, versions of Theorem 3.3 can be derived for certain regular families of target graphs using this framework. A continuing goal of this research is to formulate and prove meta-theorems, so that a result like Theorem 3.3 becomes a special case of a more general statement.

5. Open Problems

This work began from an earlier study of the bandwidth minimization problem, which is in some ways the simplest case of graph embedding. The fascinating part of this research has been to see how the significance of the earlier results has changed as they were extended and generalized. The process of generalization forces one to focus more clearly on the essential features of the original problem that made the proofs work. The result has been a much deeper appreciation and a keener understanding of their significance.

This process of generalization is not yet complete. It remains to establish meta-theorems describing the probable performance of the level algorithms in arbitrary classes of graphs. In addition, there are several other results established for the bandwidth minimization problem that may be also be extended.

Modified level algorithms similar to those described in [11] can be applied to the general embedding problem as well, and at least for regular graphs (like the rectangular grids), their performance should be superior to the level algorithms. The level and modified level strategies define only the gross structure of an embedding. Practical algorithms based on these strategies also need heuristics for properly positioning vertices within the levels. Good heuristics are known for bandwidth minimization, but it's not clear how best to extend them to other families of target graphs.

The results described here are based on a probability distribution that is simpler than may be appropriate in some cases. For example, the distributions used in section 3 generate only graphs on m^2 vertices. It may be useful to extend these results to more general distributions.

Hong, Melhorn and Rosenberg [6] discuss two cost measures for graph embedding. We have only addressed what they call the dilation cost. However, the analysis that leads to our results can also provide good bounds on the expansion cost of the level algorithms, at least for rectangular grids and similar target graphs.

References

[1]   Aleliunas, Romas, Arnold L. Rosenberg. "On Embedding Rectangular Grids in Square Grids". In *IEEE Transactions on Computers*, 907-913, 9/82.

[2]   Cuthill, E., J. McKee. "Reducing the Bandwidth of Sparse Symmetric Matrices". In *ACM National Conference Proceedings 24*, 157-172, 1969.

[3]   DeMillo, Richard A., Stanley C. Eisenstat, Richard J. Lipton. "Preserving Average Proximity in Arrays". In *Communications of the ACM 21*, 228-231, 3/78.

[4]   Garey, Michael R., R. L. Graham, David S. Johnson, D. E. Knuth. "Complexity Results for Bandwidth Minimization". In *SIAM Journal of Applied Mathematics 34*, 477-495, 5/78.

[5]   Hong, Jia-Wei, Arnold L. Rosenberg. "Graphs That are Almost Binary Trees". In *SIAM Journal of Computing*, 227-242, 1982.

[6]   Hong, Jia-Wei, Kurt Melhorn, Arnold L. Rosenberg. "Cost Trade-offs in Graph Embeddings, with Applications". In *Journal of the ACM 30*, 709-728, 10/83.

[7]   Papadimitriou, Christos H. "The NP-Completeness of the Bandwidth Minimization Problem". In *Computing 16*, 263-270, 1976.

[8]   Rosenberg, Arnold L. "Encoding Data Structures in Trees". In *Journal of the ACM 26*, 668-689, 10/79.

[9]   Rosenberg, Arnold L. "Preserving Proximity in Arrays". In *SIAM Journal of Computing 4*, 443-460, 1975.

[10] Turner, Jonathan S. "Bandwidth and Probabilistic Complexity". Northwestern University, Ph.D. thesis, 6/82.

[11] Turner, Jonathan S. "Probabilistic Analysis of Bandwidth Minimization Algorithms". In *ACM Symposium on the Theory of Computing*, 467-476, 4/83.
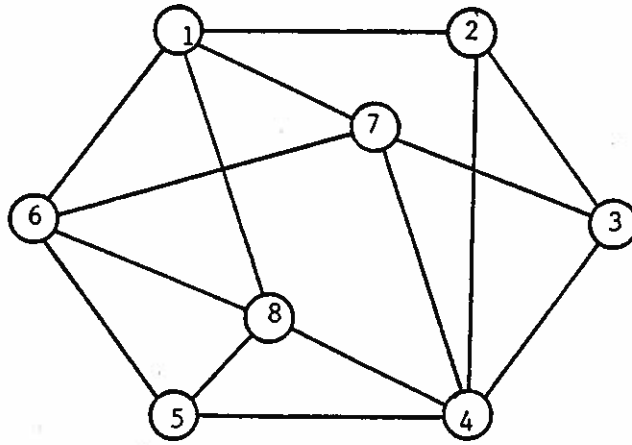
(a) Graph G
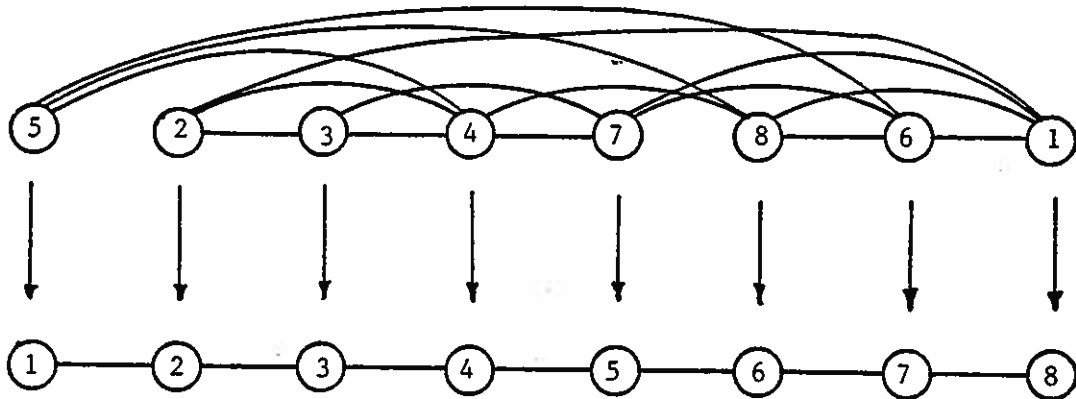
(b) Embedding of G in L_8
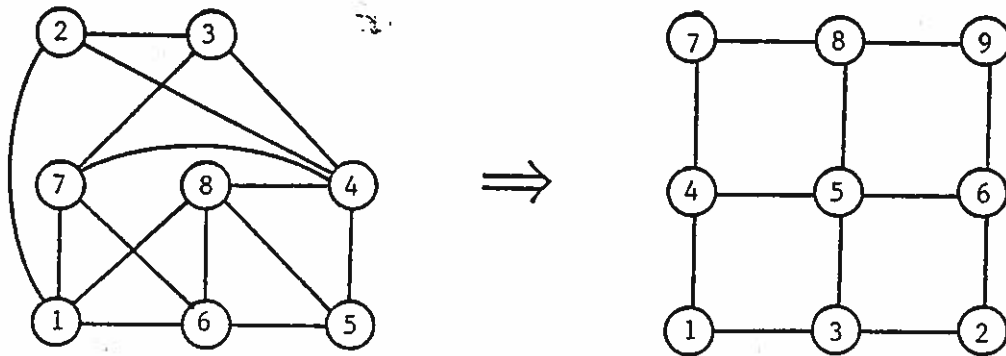
(c) Embedding of G in R_3
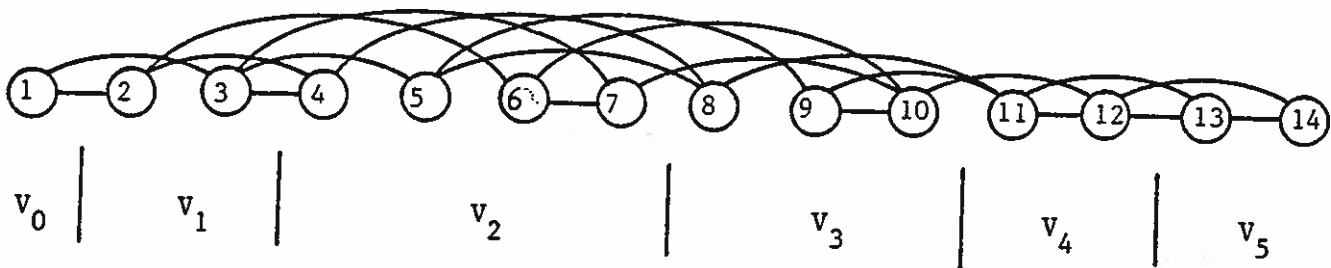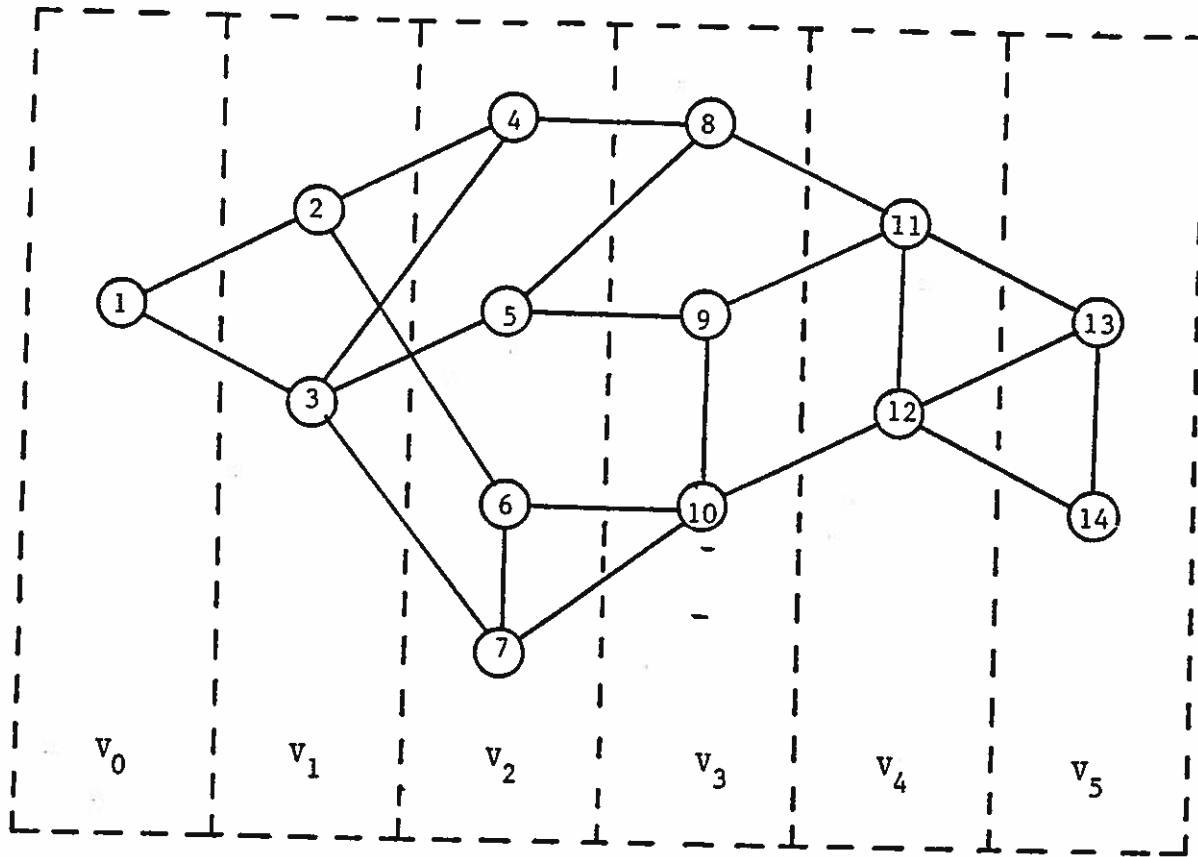
Figure 1. Embeddings of a graph in a line and a grid

Figure 2. Embedding in a line graph produced by a level algorithm
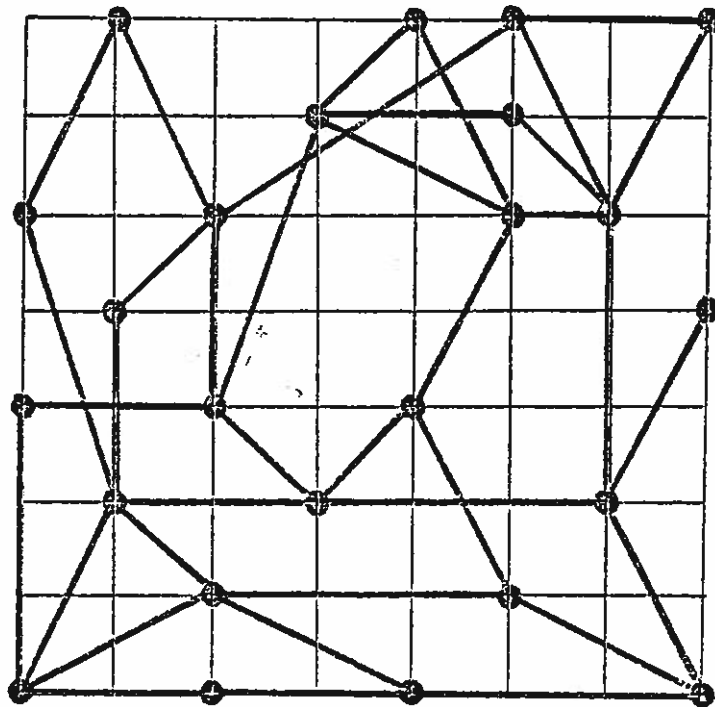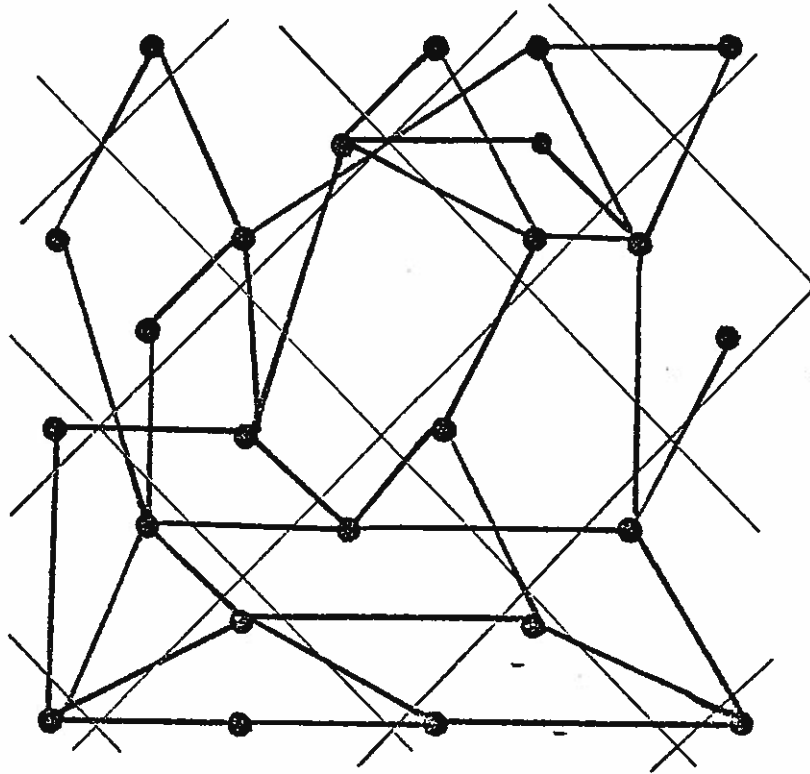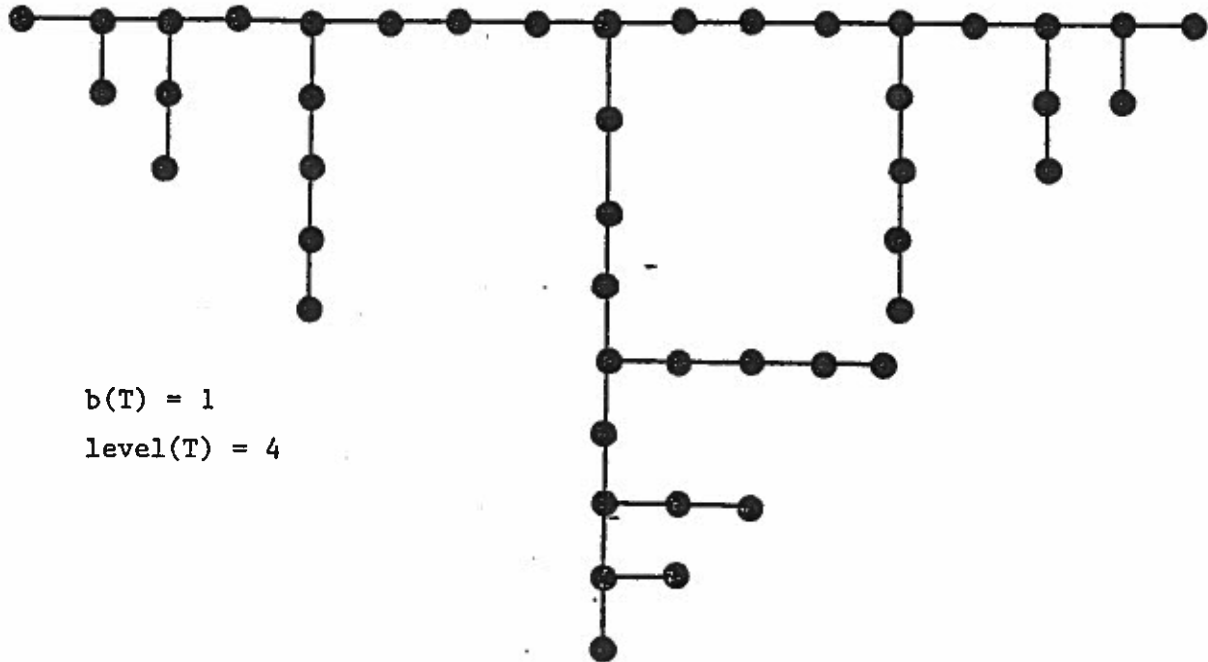
Figure 3. Embedding in a grid produced by a level algorithm

T:

b(T) = 1

level(T) = 4

Figure 4. Poor worst case performance of level algorithms for grids
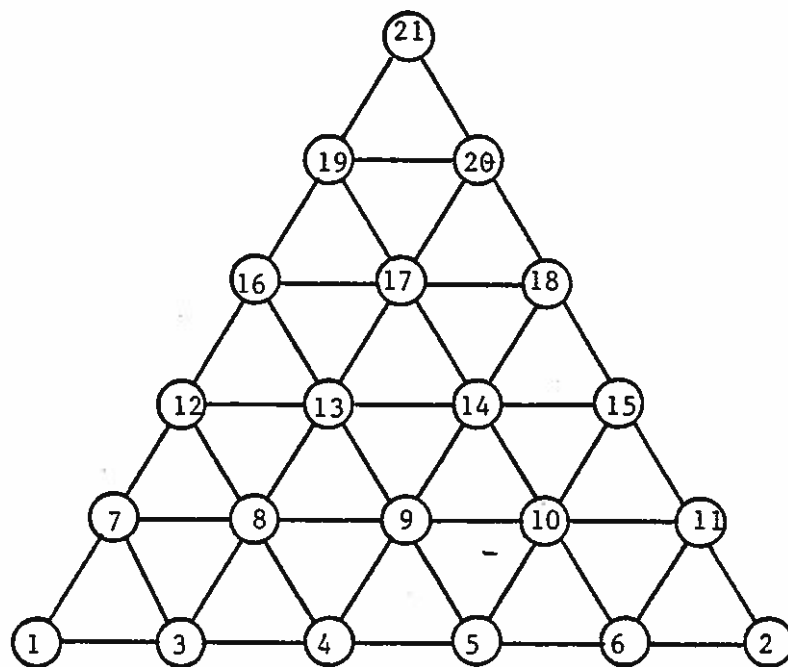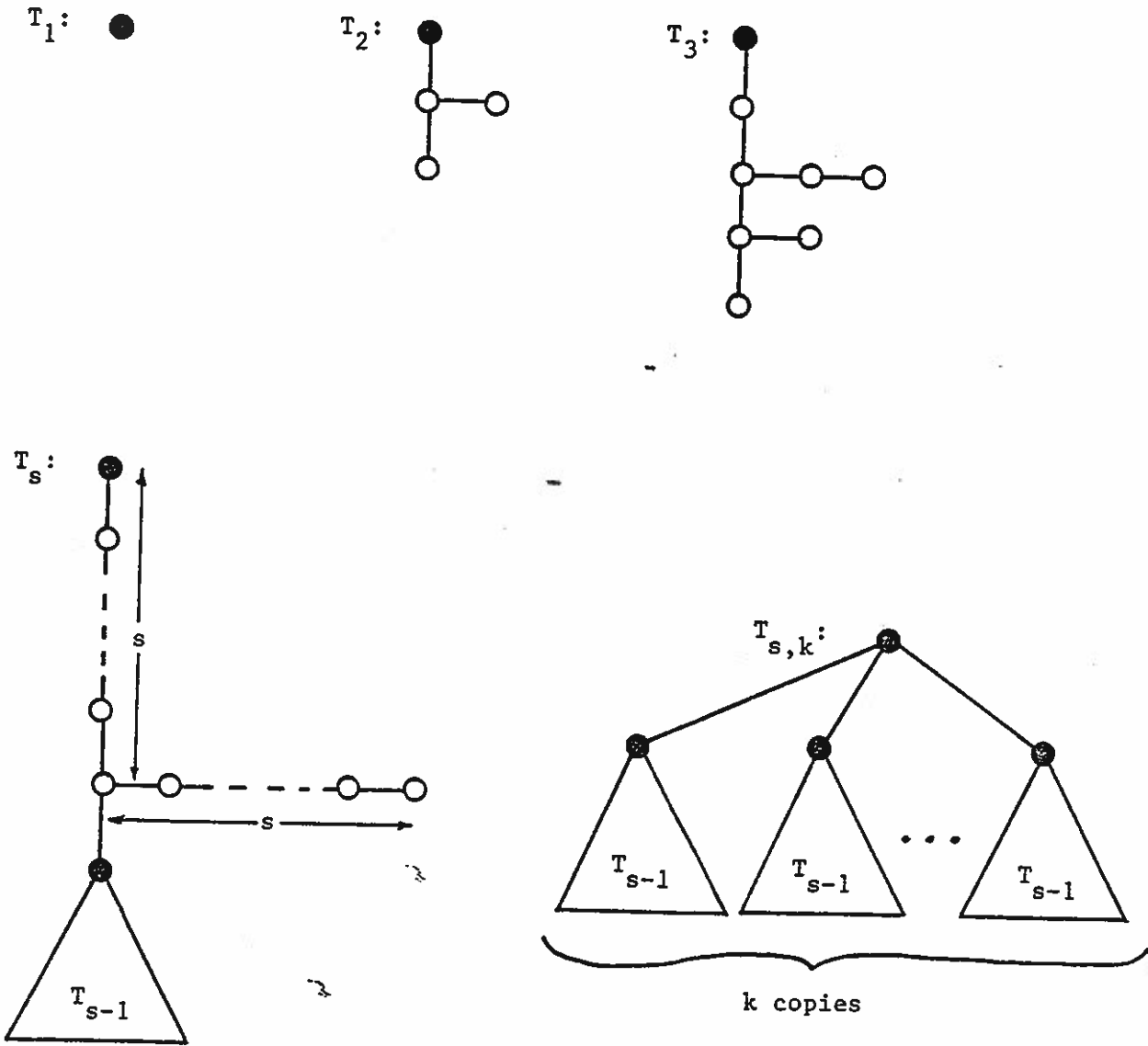
Figure 5. Triangular grid

Figure 6. Poor worst case performance of level algorithms for general graphs