boosting approach, applied to them, minimizes the output size. These pages have provided convincing evidence that the Burrows–Wheeler Transform is an elegant and efficient permutation $\mathcal{P}$. Surprisingly enough, other classic Data Compression problems fall into this framework: Shortest Common Superstring (which is MAX-SNP hard), Run Length Encoding for a Set of Strings (which is polynomially solvable), LZ77 and minimum number of phrases (which is MAX-SNP-Hard). Therefore, the boosting approach is general enough to deserve further theoretical and practical attention [5].

Experimental Results

An investigation of several compression algorithms based on boosting, and a comparison with other state-of-the-art compressors is presented in [4]. The experiments show that the boosting technique is more robust than other bwt-based approaches, and works well even with less effective 0th order compressors. However, these positive features are achieved using more (time and space) resources.

Data Sets

The data sets used in [4] are available from http://www.mfn.unipmn.it/~manzini/boosting. Other data sets for compression and indexing are available at the Pizza&Chili site http://pizzachili.di.unipi.it/.

URL to Code

The Compression Boosting page (http://www.mfn.unipmn.it/~manzini/boosting) contains the source code of all the algorithms tested in [4]. The code is organized in a highly modular library that can be used to boost any compressor even without knowing the bwt or the boosting procedure.

Cross References

- ► Arithmetic Coding for Data Compression
- ► Burrows–Wheeler Transform
- ► Compressed Text Indexing
- ► Table Compression
- ► Tree Compression and Indexing

Recommended Reading

1. Bell, T.C., Cleary, J.G., Witten, I.H.: Text compression. Prentice Hall, NJ (1990)
2. Burrows, M. Wheeler, D.: A block sorting lossless data compression algorithm. Tech. Report 124, Digital Equipment Corporation (1994)
3. Ferragina, P., Giancarlo, R., Manzini, G., Sciortino, M.: Boosting textual compression in optimal linear time. J. ACM **52**, 688–713 (2005)
4. Ferragina, P., Giancarlo, R., Manzini, G.: The engineering of a compression boosting library: Theory vs practice in bwt compression. In: Proc. 14th European Symposium on Algorithms (ESA). LNCS, vol. 4168, pp. 756–767. Springer, Berlin (2006)
5. Giancarlo, R., Restivo, A., Sciortino, M.: From first principles to the Burrows and Wheeler transform and beyond, via combinatorial optimization. Theor. Comput. Sci. **387**(3):236-248 (2007)
6. Karp, R., Pippenger, N., Sipser, M.: A Time-Randomness tradeoff. In: Proc. Conference on Probabilistic Computational Complexity, AMS, 1985, pp. 150–159
7. Manzini, G.: An analysis of the Burrows-Wheeler transform. J. ACM **48**, 407–430 (2001)
8. Navarro, G., Mäkinen, V.: Compressed full text indexes. ACM Comput. Surv. **39**(1) (2007)
9. Salomon, D.: Data Compression: the Complete Reference, 4th edn. Springer, London (2004)
10. Schapire, R.E.: The strength of weak learnability. Mach. Learn. **2**, 197–227 (1990)

Branchwidth of Graphs

2003; Fomin, Thilikos

FEDOR FOMIN[1], DIMITRIOS THILIKOS[2]

[1] Department of Informatics, University of Bergen, Bergen, Norway

[2] Department of Mathematics, National and Kapodistrian University of Athens, Athens, Greece

Keywords and Synonyms

Tangle Number

Problem Definition

Branchwidth, along with its better-known counterpart, treewidth, are measures of the "global connectivity" of a graph.

Definition

Let G be a graph on n vertices. A *branch decomposition* of G is a pair (T, τ), where T is a tree with vertices of degree 1 or 3 and τ is a bijection from the set of leaves of T to the edges of G. The *order*, we denote it as $\alpha(e)$, of an edge e in T is the number of vertices v of G such that there are leaves t_1, t_2 in T in different components of $T(V(T), E(T) - e)$ with $\tau(t_1)$ and $\tau(t_2)$ both containing v as an endpoint.

The *width* of (T, τ) is equal to $\max_{e \in E(T)}\{\alpha(e)\}$, i. e. is the maximum order over all edges of T. The *branchwidth* of G is the minimum width over all the branch decompositions of G (in the case where $|E(G)| \leq 1$, then we define the branchwidth to be 0; if $|E(G)| = 0$, then G has no branch decomposition; if $|E(G)| = 1$, then G has a branch decomposition consisting of a tree with one vertex – the width of this branch decomposition is considered to be 0).

The above definition can be directly extended to hypergraphs where τ is a bijection from the leaves of T to the hyperedges of G. The same definition can easily be extended to matroids.

Branchwidth was first defined by Robertson and Seymour in [25] and served as a main tool for their proof of Wagner's Conjecture in their Graph Minors series of papers. There, branchwidth was used as an alternative to the parameter of treewidth as it appeared easier to handle for the purposes of the proof. The relation between branchwidth and treewidth is given by the following result.

Theorem 1 ([25]) *If G is a graph, then* $\text{branchwidth}(G) \leq \text{treewidth}(G) + 1 \leq \lfloor 3/2\,\text{branchwidth}(G) \rfloor$.

The algorithmic problems related to branchwidth are of two kinds: first find fast algorithms computing its value and, second, use it in order to design fast dynamic programming algorithms for other problems.

Key Results

Algorithms for Branchwidth

Computing branchwidth is an NP-hard problem ([29]). Moreover, the problem remains NP-hard even if we restrict its input graphs to the class of split graphs or bipartite graphs [20].

On the positive side, branchwidth is computable in polynomial time on interval graphs [20,24], and circular arc graphs [21]. Perhaps the most celebrated positive result on branchwidth is an $O(n^2)$ algorithm for the branchwidth of planar graphs, given by Seymour and Thomas in [29]. In the same paper they also give an $O(n^4)$ algorithm to compute an optimal branch decomposition. (The running time of this algorithm has been improved to $O(n^3)$ in [18].) The algorithm in [29] is basically an algorithm for a parameter called carving width, related to telephone routing and the result for branchwidth follows from the fact that the branch width of a planar graph is half of the carving-width of its medial graph.

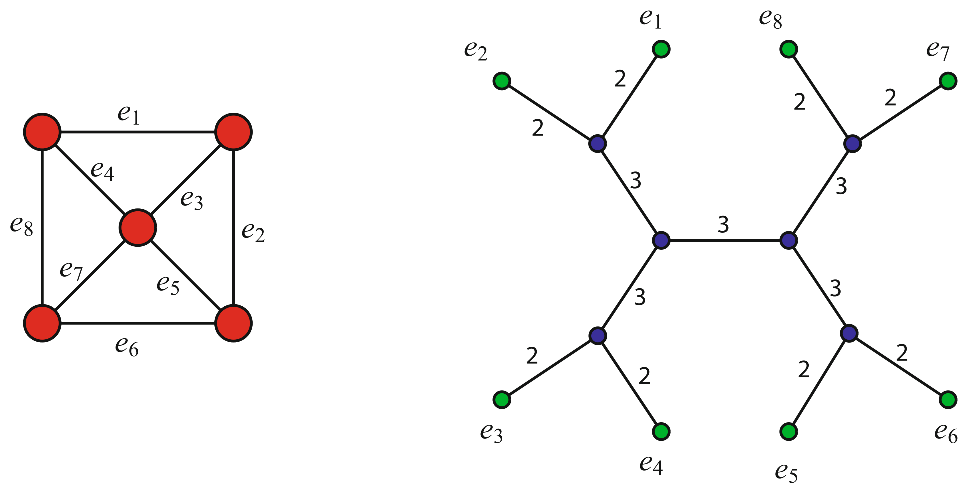
The algorithm for planar graphs [29] can be used to construct an approximation algorithm for branchwidth of some non-planar graphs. On graph classes excluding a single crossing graph as a minor branchwidth can be approximated within a factor of 2.25 [7] (a graph H is a *minor* of a graph G if H can be obtained by a subgraph of G after applying edge contractions). Finally, it follows from [13] that for every minor closed graph class, branchwidth can be approximated by a constant factor.

Branchwidth cannot increase when applying edge contractions or removals. According to the Graph Minors theory, this implies that, for any fixed k, there is a finite number of minor minimal graphs of branchwidth more than k and we denote this set of graphs by $\mathcal{B}_k$. Checking whether a graph G contains a fixed graph as a minor can be done in polynomial time [27]. Therefore, the knowledge of $\mathcal{B}_k$ implies the construction of a polynomial time algorithm for deciding whether $\text{branchwidth}(G) \leq k$, for any fixed k. Unfortunately $\mathcal{B}_k$ is known only for small values of k. In particular, $\mathcal{B}_0 = \{P_2\}, \mathcal{B}_1 = \{P_4, K_3\}, \mathcal{B}_2 = \{K_4\}$ and $\mathcal{B}_3 = \{K_5, V_8, M_6, Q_3\}$ (here K_r is a clique on r vertices, P_r is a path on r edges, V_8 is the graph obtained by a cycle on 8 vertices if we connect all pairs of vertices with cyclic distance 4, M_6 is the octahedron, and Q_3 is the 3-dimensional cube). However, for any fixed k, one can construct a linear, on $n = |V(G)|$, algorithm that decides whether an input graph G has branchwidth $\leq k$ and, if so, outputs the corresponding branch decomposition (see [3]). In technical terms, this implies that the problem of asking, for a given graph G, whether $\text{branchwidth}(G) \leq k$, parameterized by k is fixed parameter tractable (i. e. belongs in the parameterized complexity class FPT). (See [12] for further references on parameterized algorithms and complexity.) The algorithm in [3] is complicated and uses the technique of characteristic sequences, which was also used in order to prove the analogous result for treewidth. For the particular cases where $k \leq 3$, simpler algorithms exist that use the "reduction rule" technique (see [4]). We stress that $\mathcal{B}_4$ remains unknown while several elements of it have been detected so far (including the dodecahedron and the icosahedron graphs). There is a number of algorithms that for a given k in time $2^{O(k)} \cdot n^{O(1)}$ either decide that the branchwidth of a given graph is at least k, or construct a branch decomposition of width $O(k)$ (see [26]). These results can be generalized to compute the branchwidth of matroids and even more general parameters.

An exact algorithm for branchwidth appeared in [14]. Its complexity is $O((2 \cdot \sqrt{3})^n \cdot n^{O(1)})$. The algorithm exploits special properties of branchwidth (see also [24]).

In contrast to treewidth, edge maximal graphs of given branchwidth are not so easy to characterize (for treewidth there are just k-trees, i. e. chordal graphs with all maximal cliques of size $k+1$). An algorithm for generating such graphs has been given in [23] and reveals several structural issues on this parameter.

It is known that a large number of graph theoretical problems can be solved in linear time when their inputs are restricted to graphs of small (i. e. fixed) treewidth or branchwidth (see [2]).



Branchwidth of Graphs, Figure 1
Example of a graph and its branch decomposition of width 3

Branchwidth appeared to be a useful tool in the design of exact subexponential algorithms on planar graphs and their generalizations. The basic idea behind this approach is very simple: Let $\mathcal{P}$ be a problem on graphs and $\mathcal{G}$ be a class of graphs such that

- For every graph $G \in \mathcal{G}$ of branchwidth at most ℓ, the problem $\mathcal{P}$ can be solved in time $2^{c \cdot \ell} \cdot n^{\mathcal{O}(1)}$, where c is a constant, and;
- For every graph $G \in \mathcal{G}$ on n vertices a branch decomposition (not necessarily optimal) of G of width at most $h(n)$ can be constructed in polynomial time, where $h(n)$ is a function.

Then for every graph $G \in \mathcal{G}$, the problem $\mathcal{P}$ can be solved in time $2^{c \cdot h(n)} \cdot n^{\mathcal{O}(1)}$. Thus, everything boils down to computations of constants c and functions $h(n)$. These computations can be quite involved. For example, as was shown in [17], for every planar graph G on n vertices, the branchwidth of G is at most $\sqrt{4.5n} < 2.1214\sqrt{n}$. For extensions of this bound to graphs embeddable on a surface of genus g, see [15].

Dorn [9] used fast matrix multiplication in dynamic programming to estimate the constants c for a number of problems. For example, for the MAXIMUM INDEPENDENT SET problem, $c \le \omega/2$, where $\omega < 2.376$ is the matrix product exponent over a ring, which implies that the INDEPENDENT SET problem on planar graphs is solvable in time $O(2^{2.52\sqrt{n}})$. For the MINIMUM DOMINATING SET problem, $c \le 4$, thus implying that the branch decomposition method runs in time $O(2^{3.99\sqrt{n}})$. It appears that algorithms of running time $2^{O(\sqrt{n})}$ can be designed even for some of the "non-local" problems, such as the HAMILTONIAN CYCLE, CONNECTED DOMINATING SET, and STEINER TREE, for which no time $2^{O(\ell)} \cdot n^{\mathcal{O}(1)}$ algorithm on general graphs of branchwidth ℓ is known [11]. Here one needs special properties of some optimal planar branch decompositions, roughly speaking that every edge of T corresponds to a disk on a plane such that all edges of G corresponding to one component of $T - e$ are inside the disk and all other edges are outside. Some of the subexponential algorithms on planar graphs can be generalized for graphs embedded on surfaces [10] and, more generally, to graph classes that are closed under taking of minors [8].

A similar approach can be used for parameterized problems on planar graphs. For example, a parameterized algorithm that finds a dominating set of size $\le k$ (or reports that no such set exists) in time $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$ can be obtained based on the following observations: there is a constant c such that every planar graph of branchwidth at least $c\sqrt{k}$ does not contain a dominating set of size at most k. Then for a given k the algorithm computes an optimal branch decomposition of a palanar graph G and if its width is more than $c\sqrt{k}$ concludes that G has no dominating set of size k. Otherwise, find an optimal dominating set by performing dynamic programming in time $2^{\mathcal{O}(\sqrt{k})} n^{\mathcal{O}(1)}$. There are several ways of bounding a parameter of a planar graph in terms of its branchwidth or treewidth including techniques similar to Baker's approach from approximation algorithms [1], the use of separators, or by some combinatorial arguments, as shown in [16]. Another general approach of bounding the branchwidth of a planar graph by parameters, is based on the results of Robertson et al. [28] regarding quickly excluding a planar graph. This brings us to the notion of *bidimensionality* [6]. Parameterized algorithms based on branch decompositions can be generalized from planar graphs to graphs embedded on surfaces and to graphs excluding a fixed graph as a minor.

Applications

See [5] for using branchwidth for solving TSP.

Open Problems

1. It is known that any planar graph G has branchwidth at most $\sqrt{4.5} \cdot \sqrt{|V(G)|}$ (or at most $\frac{3}{2} \cdot \sqrt{|E(G)|+2}$) [17]. Is it possible to improver this upper bound? Any possible improvement would accelerate many of the known exact or parameterized algorithms on planar graphs that use dynamic programming on branch decompositions.
2. In contrast to treewidth, very few graph classes are known where branchwidth is computable in polynomial time. Find graphs classes where branchwidth can be computed or approximated in polynomial time.
3. Find $\mathcal{B}_k$ for values of k bigger than 3. The only structural result on $\mathcal{B}_k$ is that its planar elements will be either self-dual or pairwise-dual. This follows from the fact that dual planar graphs have the same branchwidth [29,16].
4. Find an exact algorithm for branchwidth of complexity $O^*(2^n)$ (the notation $O^*()$ assumes that we drop the non-exponential terms in the classic $O()$ notation).
5. The dependence on k of the linear time algorithm for branchwidth in [3] is huge. Find an $2^{O(k)} \cdot n^{O(1)}$ step algorithm, deciding whether the branchwidth of an n-vertex input graph is at most k.

Cross References

► Bidimensionality
► Treewidth of Graphs

Recommended Reading

1. Alber, J., Bodlaender, H.L., Fernau, H., Kloks, T., Niedermeier, R.: Fixed parameter algorithms for dominating set and related problems on planar graphs. Algorithmica **33**, 461–493 (2002)
2. Arnborg, S.: Efficient algorithms for combinatorial problems on graphs with bounded decomposability – A survey. BIT **25**, 2–23 (1985)
3. Bodlaender, H.L., Thilikos, D.M.: Constructive linear time algorithms for branchwidth. In: Automata, languages and programming (Bologna, 1997). Lecture Notes in Computer Science, vol. 1256, pp. 627–637. Springer, Berlin (1997)
4. Bodlaender, H.L., Thilikos, D.M.: Graphs with branchwidth at most three. J. Algorithms **32**, 167–194 (1999)
5. Cook, W., Seymour, P.D.: Tour merging via branch-decomposition. Inf. J. Comput. **15**, 233–248 (2003)
6. Demaine, E.D., Fomin, F.V., Hajiaghayi, M., Thilikos, D.M.: Bidimensional parameters and local treewidth. SIAM J. Discret. Math. **18**, 501–511 (2004)
7. Demaine, E.D., Hajiaghayi, M.T., Nishimura, N., Ragde, P., Thilikos, D. M.: Approximation algorithms for classes of graphs excluding single-crossing graphs as minors. J. Comput. Syst. Sci. **69**, 166–195 (2004)
8. Dorn, F., Fomin, F.V., Thilikos, D.M.: Subexponential algorithms for non-local problems on H-minor-free graphs. In: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms (SODA 2008). pp. 631–640. Society for Industrial and Applied Mathematics, Philadelphia (2006)
9. Dorn, F.: Dynamic programming and fast matrix multiplication. In: Proceedings of the 14th Annual European Symposium on Algorithms (ESA 2006). Lecture Notes in Computer Science, vol. 4168 , pp. 280–291. Springer, Berlin (2006)
10. Dorn, F., Fomin, F.V., Thilikos, D.M.: Fast subexponential algorithm for non-local problems on graphs of bounded genus. In: Proceedings of the 10th Scandinavian Workshop on Algorithm Theory (SWAT 2006). Lecture Notes in Computer Science. Springer, Berlin (2005)
11. Dorn, F., Penninkx, E., Bodlaender, H., Fomin, F.V.: Efficient exact algorithms on planar graphs: Exploiting sphere cut branch decompositions. In: Proceedings of the 13th Annual European Symposium on Algorithms (ESA 2005). Lecture Notes in Computer Science, vol. 3669, pp. 95–106. Springer, Berlin (2005)
12. Downey, R.G., Fellows, M.R.: Parameterized complexity. In: Monographs in Computer Science. Springer, New York (1999)
13. Feige, U., Hajiaghayi, M., Lee, J.R.: Improved approximation algorithms for minimum-weight vertex separators. In: Proceedings of the 37th annual ACM Symposium on Theory of computing (STOC 2005), pp. 563–572. ACM Press, New York (2005)
14. Fomin, F.V., Mazoit, F., Todinca, I.: Computing branchwidth via efficient triangulations and blocks. In: Proceedings of the 31st Workshop on Graph Theoretic Concepts in Computer Science (WG 2005). Lecture Notes Computer Science, vol. 3787, pp. 374–384. Springer, Berlin (2005)
15. Fomin, F.V., Thilikos, D.M.: Fast parameterized algorithms for graphs on surfaces: Linear kernel and exponential speed-up. In: Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP 2004). Lecture Notes Computer Science, vol. 3142, pp. 581–592. Springer, Berlin (2004)
16. Fomin, F.V., Thilikos, D.M.: Dominating sets in planar graphs: Branch-width and exponential speed-up. SIAM J. Comput. **36**, 281–309 (2006)
17. Fomin, F.V., Thilikos, D.M.: New upper bounds on the decomposability of planar graphs. J. Graph Theor. **51**, 53–81 (2006)
18. Gu, Q.P., Tamaki, H.: Optimal branch-decomposition of planar graphs in $O(n^3)$ time. In: Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP 2005). Lecture Notes Computer Science, vol. 3580, pp. 373–384. Springer, Berlin (2005)
19. Gu, Q.P., Tamaki, H.: Branch-width, parse trees, and monadic second-order logic for matroids. J. Combin. Theor. Ser. B **96**, 325–351 (2006)
20. Kloks, T., Kratochvíl, J., Müller, H.: Computing the branchwidth of interval graphs. Discret. Appl. Math. **145**, 266–275 (2005)
21. Mazoit, F.: The branch-width of circular-arc graphs. In: 7th Latin American Symposium on Theoretical Informatics (LATIN 2006), 2006, pp. 727–736

22. Oum, S.I., Seymour, P.: Approximating clique-width and branch-width. J. Combin. Theor. Ser. B **96**, 514–528 (2006)
23. Paul, C., Proskurowski, A., Telle, J.A.: Generating graphs of bounded branchwidth. In: Proceedings of the 32nd Workshop on Graph Theoretic Concepts in Computer Science (WG 2006). Lecture Notes Computer Science, vol. 4271, pp. 205–216. Springer, Berlin (2006)
24. Paul, C., Telle, J.A.: New tools and simpler algorithms for branchwidth. In: Proceedings of the 13th Annual European Symposium on Algorithms (ESA 2005), 2005 pp. 379–390
25. Robertson, N. Seymour, P.D.: Graph minors. X. Obstructions to tree-decomposition J. Combin. Theor. Ser. B **52**, 153–190 (1991)
26. Robertson, N. Seymour, P.D.: Graph minors. XII. Distance on a surface. J. Combin. Theor. Ser. B **64**, 240–272 (1995)
27. Robertson, N. Seymour, P.D.: Graph minors. XIII. The disjoint paths problem. J. Combin. Theor. Ser. B **63**, 65–110 (1995)
28. Robertson, N., Seymour, P.D., Thomas, R.: Quickly excluding a planar graph. J. Combin. Theor. Ser. B **62**, 323–348 (1994)
29. Seymour, P.D., Thomas, R.: Call routing and the ratcatcher. Combinatorica **14**, 217–241 (1994)

Broadcasting in Geometric Radio Networks

2001; Dessmark, Pelc

ANDRZEJ PELC
Department of Computer Science,
University of Québec-Ottawa, Gatineau, QC, Canada

Keywords and Synonyms

Wireless information dissemination in geometric networks

Problem Definition

The Model Overview

Consider a set of stations (nodes) modeled as points in the plane, labeled by natural numbers, and equipped with transmitting and receiving capabilities. Every node u has a *range* r_u depending on the power of its transmitter, and it can reach all nodes at distance at most r_u from it. The collection of nodes equipped with ranges determines a directed graph on the set of nodes, called a *geometric radio network* (GRN), in which a directed edge (uv) exists if node v can be reached from u. In this case u is called a *neighbor* of v. If the power of all transmitters is the same then all ranges are equal and the corresponding GRN is symmetric.

Nodes send messages in synchronous *rounds*. In every round every node acts either as a *transmitter* or as a *receiver*. A node gets a message in a given round, if and only if, it acts as a receiver and exactly one of its neighbors transmits in this round. The message received in this case is the one that was transmitted. If at least two neighbors of a receiving node u transmit simultaneously in a given round, none of the messages is received by u in this round. In this case it is said that a *collision* occurred at u.

The Problem

Broadcasting is one of the fundamental network communication primitives. One node of the network, called the *source*, has to transmit a message to all other nodes. Remote nodes are informed via intermediate nodes, along directed paths in the network. One of the basic performance measures of a broadcasting scheme is the total time, i. e., the number of rounds it uses to inform all the nodes of the network.

For a fixed real $s \geq 0$, called the *knowledge radius*, it is assumed that each node knows the part of the network within the circle of radius s centered at it, i. e., it knows the positions, labels and ranges of all nodes at distance at most s. The following problem is considered:

How the size of the knowledge radius influences deterministic broadcasting time in GRN?

Terminology and Notation

Fix a finite set $R = \{r_1, \ldots, r_\rho\}$ of positive reals such that $r_1 < \cdots < r_\rho$. Reals r_i are called *ranges*. A *node* v is a triple $[l, (x, y), r_i]$, where l is a binary sequence called the *label* of v, (x, y) are coordinates of a point in the plane, called the *position* of v, and $r_i \in R$ is called the *range* of v. It is assumed that labels are consecutive integers 1 to n, where n is the number of nodes, but all the results hold if labels are integers in the set $\{1, \ldots, M\}$, where $M \in O(n)$. Moreover, it is assumed that all nodes know an upper bound Γ on n, where Γ is polynomial in n. One of the nodes is distinguished and called the *source*. Any set of nodes C with a distinguished source, such that positions and labels of distinct nodes are different is called a *configuration*.

With any configuration C the following directed graph $\mathcal{G}(C)$ is associated. Nodes of the graph are nodes of the configuration and a directed edge (uv) exists in the graph, if and only if the distance between u and v does not exceed the range of u. (The word "distance" always means the geometric distance in the plane and not the distance in a graph.) In this case u is called a neighbor of v. Graphs of the form $\mathcal{G}(C)$ for some configuration C are called *geometric radio networks* (GRN). In what follows, only configurations C such that in $\mathcal{G}(C)$ there exists a directed path from the source to any other node, are considered. If the size of the set R of ranges is ρ, a resulting configuration and the corresponding GRN are called a ρ-configuration and