Virtual Path Network Topology Optimization Using Random Graphs

András Faragó Imrich Chlamtac Stefano Basagni

Erik Jonsson School of Engineering and Computer Science
The University of Texas at Dallas
E-mail: {farago,chlamtac,basagni}@utdallas.edu

Abstract— **An algorithm is presented for designing the logical topology of the virtual path (VP) network, an important task in ATM network design. We prove that the algorithm provides a VP network topology that is asymptotically optimal with respect to both connectivity and the diameter of the network. These optimality properties are combined with algorithmic simplicity and polynomial running time, thus overcoming the notorious "optimality vs. scalability" dilemma. This result is made possible by applying the theory of *random graphs* to this type of networks. This theory has the methodological advantage of increased accuracy with growing network size, thus turning the "curse of dimensionality" into a blessing. Therefore, the paper exemplifies that the theory of random graphs, beyond supporting analysis purposes, may serve as a useful tool in the design of algorithms that overcome the "scalability bottleneck," a problem that prevents current approaches from finding near-optimal solutions as today's networks grow in size and complexity.**

I. Introduction

The *virtual path (VP)* concept plays a central role in the resource management of ATM networks. The optimal design of the VP system in a large network is a highly complex task whose solution bears heavily on network performance. Therefore, it is not surprising that a large number of different solutions to this problem have been proposed in the technical literature (for an overview see e.g. [1]).

The general design of a VP system for an ATM network deals with one or more of the following tasks:

- **VP network topology design**: which (pairs of) nodes should be connected by VPs, i.e., what is the *logical* topology of the VP network. Note that, at this level, each VP appears as a single logical link between its two terminator nodes.
- **VP layout design**: how should the VPs be mapped in the physical network topology, i.e., what is the physical route of each VP.
- **VP dimensioning**: how much transmission capacity should be assigned to each VP, while obeying physical network constraints.

This work was supported in part by the Army Research Office under contract No. DAAG55-96-1-0382.

This paper deals with the first task above, the *VP network topology design.*

II. The Optimality vs. Scalability Dilemma

Each of the tasks mentioned in the Introduction is difficult to solve optimally, as from the algorithmic point of view they all involve NP-hard subproblems. This is why any approach to these problems faces the following dilemma: either give up *optimality* and be satisfied with some heuristic solution that can be arbitrarily far from the optimum, or, if optimum is searched for, either exactly or at least with a bounded error, then *scalability* is lost in the sense that for realistic, i.e., large, networks the algorithm becomes computationally infeasible.

In this paper we show that, by proposing a new solution to the problem of the *VP network topology design*, the notorious optimality vs. scalability dilemma can be resolved. The methodology we use to accomplish this is derived from the *theory of random graphs* (for a thorough introduction to the mathematical foundations see [2]).

The key feature we use from this theory is its power of providing rigorous analysis tools for various properties on large statistical manifolds of graphs. In this way we can avoid the difficulties caused by worst cases, since they usually occur with very small probability. Using the theory of random graphs, in many cases it is possible to rigorously prove that the probability of the worst case to occur tends to zero as the graph size grows.

Let us explain this situation with a short example based on a classical NP-complete problem. Let G be a graph with n vertices and m edges. Assume a number k is given and we want to decide whether G contains a complete (i.e., fully connected) subgraph of k vertices. This is well known to be an NP-complete problem [3]. Moreover, the corresponding optimization version (the *maximum clique* problem) where we wish to compute the maximum size of a complete subgraph, cannot even be approximated by a polynomial time algorithm (assuming $P \neq NP$) within an error factor of $n^{1-\epsilon}$ for any fixed $\epsilon > 0$, as proven in [4]. Thus, this problem clearly defies scalability, as with growing n no fast algorithm can guarantee even an ac-

0-7803-5417-6/99/$10.00 ©1999 IEEE.

ceptable approximation for the worst case. On the other hand, in a random graph setting with the same parameters the maximum size of a complete subgraph can be easily computed, as it is shown to be asymptotically

$$\frac{2\log n}{\log \frac{n(n-1)}{2m}}$$

for a wide range of m values, depending on n [2, Chapter XI]. For a finite graph this holds *almost surely*. This means that the fraction of all graphs that deviate from the formula tends to 0 as n grows. Thus, if we have a graph that is chosen at random from all graphs of n nodes with m edges, we can determine the desired quantity with vanishing error probability, despite its *worst-case* algorithmic intractability.

Of course, when applying such results, we have to make sure that the graph can "truly" be considered random. In the case of VP network topology design, an essential element of the solution will be choosing the graph such that certain properties are enforced, apparently destroying uniform randomness, while maintaining sufficient randomness to allow us to rigorously prove the main properties using the theory of random graphs. Since these theorems are asymptotic, the growth of the network size *increases* their accuracy, rather than diminishing it. In summary, in this way we can overcome the *scalability bottleneck*, or, putting it differently, from the viewpoint of this methodology, having a very large network becomes a blessing, rather than a curse.

III. Problem Definition

We consider the following problem. As input, we are given the number n of nodes in the network and a bound Δ for the *VP processing capacity* of the nodes, which means that no node can be the endpoint of more than Δ links in the logical VP network topology. The goal is to construct a VP network topology with the desirable properties discussed below.

We represent the VP network topology by a graph on n vertices, where each vertex corresponds to a network node. The edges of the graph represent the virtual paths, that is, an edge between vertices i and j in the graph signifies that a VP exists between the corresponding nodes in the network.[1] The parameter Δ is the maximum allowed degree in the graph.

Our goal is to choose the logical VP topology such that:

- Its *connectivity*[2] is maximum under the given VP processing capacity bound;
- The *diameter* D of the resulting VP network is minimum, that is, any two nodes can reach each other in at most D VP-hops.

These goals are naturally justified from the networking point of view, since high connectivity yields both safer network operation and better load balancing opportunities, while small diameter puts a limit on the number of needed VP-hops, resulting in reduced end-to-end delay, as well as faster call setup time. Additionally, the reduced number of needed VP-hops also means that any given connection places load on fewer links, which helps in decreasing the average link load, thus, lowering the probability of congestion. This also means that a prescribed blocking probability can be achieved with less total transmission capacity in the network, that is, at lower cost.

In the next section we present a randomized algorithm, that constructs the desired VP network topology in polynomial time. Then, using the theory of random graphs, we prove that with a probability approaching 1, as n grows, the constructed VP network topology will be *optimal* with respect to *both* above objectives.

IV. The Algorithm

Our strategy is to construct a random VP network topology that meets the node processing constraints. Then we prove that it optimizes both connectivity and diameter, with probability approaching 1 as the network size grows.

Note that we cannot simply generate a random graph in the most common way (by adding edges randomly), as this would violate the degree constraint with high probability. On the other hand, if the degree constraint is artificially enforced, e.g., by excluding nodes that became saturated, then uniform randomness does not remain intact, preventing the direct applicability of random graph theorems. Another possibility is to generate a random Δ-*regular* graph.[3] This is, however, quite involved, and an algorithm is only known for $\Delta = O(n^{1/3})$ (see [5]), while we would like to eliminate such range restrictions.

To avoid the above difficulties, we choose the following solution. We generate the graph randomly by enforcing regularity, but giving up the condition that the generated graph is uniformly distributed over *all* Δ-regular graphs. The graph will be chosen randomly from an appropriate

[1] Note that we focus here on designing the *logical* VP network topology, that is why VPs are represented by edges in the graph, rather than routes. The underlying *physical* network topology plays no role at this level.

[2] By connectivity we mean the minimum number of nodes whose deletion can disconnect the network. Note that this is stronger than considering the minimum number of disconnecting links, since if the network cannot be disconnected by any k failed nodes, then the same holds for any k links, too, but not necessarily vice versa.

[3] A Δ-regular graph is defined as a graph in which each vertex has the same degree Δ. Note that, in networking, the term "regular topology" often means a topology with some regular structure, such as a ring, hypercube, etc. Here we stick to the graph theoretic definition, where regularity only means the equality of degrees, allowing an otherwise arbitrary topology.

subclass of these graphs (those that arise as the union of random perfect matchings) and, as we show in the proof, this choice is already sufficient to guarantee the desired properties of asymptotically optimal connectivity and diameter.

To exclude trivial cases, we assume that $\Delta \geq 3$. (If $\Delta = 1$ the graph cannot be connected with $n > 2$ and if $\Delta = 2$ then the only optimal solution is a single ring.) Since both the algorithm and the proof are easier to describe if n is an even number, we assume that n is even (for large networks the parity of the number of nodes does not make an essential difference, anyway). Note that we consider simple graphs, that is, parallel edges are not allowed. This also implies that $n \geq \Delta + 1$, since otherwise no Δ-regular graph would exist on n nodes.

The algorithm proceeds in phases. We start with the set of all possible edges collected in a set H. That is, initially $|H| = \frac{n(n-1)}{2}$ and it defines a complete graph. During each phase a *random perfect matching*[4] is selected from H, such that it is disjoint from all the previously selected ones, i.e., contains no common edge with any of them. This is ensured by selecting the matching in every phase using only the edges that are still in H and when the matching is found, its edges are removed from H, so the next matching will necessarily be disjoint.

In the selection of the matchings we have to ensure that the obtained matching is a *random* perfect matching in the graph defined by the edges that are still in H. For this purpose, at the beginning of each phase we randomly assign weights to the edges in H and then we use a deterministic algorithm to find a *maximum weight perfect matching.* Finding the maximum weight perfect matching is a standard, well solved task in combinatorial optimization that can be performed in $O(n^3)$ steps for a graph of n vertices [6], or even more efficiently (see, e.g., [7]), so we can use it as a subroutine. We will show that, as a result of choosing the weights as powers of 2 randomly, the obtained perfect matching is randomly and *uniformly* selected among all the perfect matchings in H. The uniformity is essential to guarantee the desired properties. We also show that the procedure does not get stuck in the sense that, with probability approaching 1, there is always a perfect matching in the remaining graph, as long as it is not empty. We finally show that, after Δ phases, the graph that results from the union of the produced random disjoint perfect matchings has the desired properties of being Δ-regular and optimal with respect to both connectivity and diameter.

In the following, V denotes the set of nodes, given as input to the algorithm together with Δ, and E is the set of the chosen edges that represent the VPs.

[4] A perfect matching is a set of edges that together cover all vertices exactly once, that is, a perfect matching consists of $n/2$ disjoint edges.

Algorithm *Random VP Network (RVPN)*
$E := \emptyset$;
$H := (V \times V) \setminus \{\{v, v\} : v \in V\}$;
for $i := 1$ **to** Δ **do** {
  Generate a random permutation of the edges in H;
  For each i, assign to the i^{th} edge in the permutation the weight 2^i;
  Find a *Maximum Weight Perfect Matching* M in H;
  If no perfect matching exists in H
    then **return** $G = (V, E)$ and Stop;
  $E := E \cup M$;
  $H := H \setminus M$;
  };
return $G = (V, E)$.

V. Analysis

In this section we prove the fundamental properties of the algorithm RVPN. Specifically, we show that the algorithm produces a VP network topology in polynomial time, and that the resulting graph is Δ-regular, with probability approaching 1 as $n \to \infty$, so that the VP processing capacity of each node is fully utilized (as Δ VPs end at each node). Moreover, both the connectivity and the diameter of the constructed graph are asymptotically optimal, in the sense stated precisely in the theorem below.

For concise notation let $D_{min}(n, \Delta)$ denote the smallest possible diameter that a graph on n nodes can have if the degree of each node is at most Δ (if a graph is disconnected, then the diameter is defined to be infinite). Note that computing the exact value of $D_{min}(n, \Delta)$ in general is a well known unsolved problem in graph theory. Although certain bounds exist (see, e.g., [2]), the exact value is known only for a few exceptional cases.

The properties of our algorithm are summarized in the following theorem.

Theorem. *Algorithm* RVPN *has the following properties:*

1. *It constructs the graph in polynomial time, bounded by* $O(\Delta n^3)$.
2. *The constructed graph is regular of degree at most* Δ *and with probability approaching 1 as* $n \to \infty$, *the degree is exactly Delta.*
3. *With probability approaching 1 as* $n \to \infty$, *the constructed graph has the highest possible connectivity among all graphs that satisfy the same degree constraints.*
4. *If* $\Delta \to \infty$ *as* $n \to \infty$, *then the diameter* D *of the constructed graph is asymptotically optimal in the sense that for any fixed* $\epsilon > 0$

$$\lim_{n \to \infty} \mathrm{P}\left(1 \leq \frac{D}{D_{min}(n, \Delta)} < 1 + \epsilon\right) = 1$$

holds.

For the proof we need two lemmas that show some important properties.

Lemma 1. *In each phase when the algorithm finds a maximum weight perfect matching in the graph H, this perfect matching is selected uniformly at random from the family of all perfect matchings of H.*

Proof. We show that, under the special random weighting, any two perfect matchings have exactly the same probability to become a maximum weight perfect matching, which will imply that any perfect matching has the same probability of being selected, that is, the found perfect matching is chosen uniformly at random. Let M_1, M_2 be any two different perfect matchings (they are viewed as sets of edges). Let us define the following two events, whose occurences depend on the random weighting:

A : M_1 is of maximum weight.
B : M_2 is of maximum weight.

We wish to show that the two events have equal probability, that is, $\mathrm{P}(A) = \mathrm{P}(B)$. Using the simple fact that for any two events $A(A+B) = A$ holds[5] and applying the definition of conditional probability, we can write:

$$\mathrm{P}(A) = \mathrm{P}\left(A(A+B)\right) = \mathrm{P}(A|A+B)\mathrm{P}(A+B). \quad (1)$$

Similarly, using $B(A+B) = B$, we have

$$\mathrm{P}(B) = \mathrm{P}\left(B(A+B)\right) = \mathrm{P}(B|A+B)\mathrm{P}(A+B). \quad (2)$$

Hence, if we can show that $\mathrm{P}(A|A+B) = \mathrm{P}(B|A+B)$ holds, then (1), (2) will imply $\mathrm{P}(A) = \mathrm{P}(B)$. Thus, the only thing we have to prove is that M_1, M_2 have equal chance to be of maximum weight, given that one of them is of maximum weight. To show this, let us write the weights of M_1, M_2 as

$$w(M_1) = w(M_1 - M_2) + w(M_1 \cap M_2),$$
$$w(M_2) = w(M_2 - M_1) + w(M_1 \cap M_2).$$

Since the term $w(M_1 \cap M_2)$ is the same in both the above expressions, the weight relationship of M_1, M_2 depends only on which of the first terms is larger. Let 2^j be the largest weight that occurs in $(M_1 - M_2) \cup (M_2 - M_1)$, i.e., in the symmetric difference of the two matchings (which cannot be empty, as $M_1 \neq M_2$ is assumed). Since

$$2^j > \sum_{i=1}^{j-1} 2^i$$

holds, therefore, it does not matter how the other (smaller) weights are distributed in $(M_1 - M_2) \cup (M_2 - M_1)$, the only decisive factor is that which of $M_1 - M_2$ and $M_2 - M_1$ has this largest weight 2^j. Now, as M_1, M_2 have equal number of edges (both being perfect matchings), implying $|M_1 - M_2| = |M_2 - M_1|$, it follows from the random construction of weights that 2^j has the same chance to fall in either of them. Thus, given that one of M_1, M_2 is a maximum weight perfect matching, both of them have exactly the same probability of being this maximum weight perfect matching, i.e., $\mathrm{P}(A|A+B) = \mathrm{P}(B|A+B)$ holds. By (1) and (2) this implies $\mathrm{P}(A) = \mathrm{P}(B)$, which proves the lemma.

□

Next we show that we cannot get stuck in any phase of the algorithm, that is, there is always a perfect mathing in the remaining graph H, with probability approaching 1 as the number of nodes grow.

Lemma2. *For each phase of the algorithm*

$$\lim_{n\to\infty} \mathrm{P}(H \text{ has a perfect matching}) = 1.$$

Proof. Consider the i^{th} phase of the algorithm, i.e., $i-1$ random perfect matchings have been already selected and we wish to prove that the remaining graph H has a perfect matching with probability approaching 1. At the beginning, when $i = 1$, H is still a complete graph (on an even number of vertices), so it trivially has a perfect matching. For $i = 2$ one can still directly see that H must have a perfect matching: index the vertices such that the previously chosen matching is $(1,2),(3,4),\ldots,(n-1,n)$, then $(2,3),(4,5),\ldots,(n-2,n-1),(n,1)$ is a perfect matching in H, showing that H must have one. A similar argument still works for $i = 3$, making use that the union of the first two perfect matchings consists of disjoint even circuits. For $i > 3$, however, the situation is becoming more difficult. First we observe that the disjointness of the perfect matchings that have been removed from H in the earlier phases implies that they together constitute the removal of an $i-1$-regular graph. Consequently, the remaining H is a regular graph of degree $k = n - i$, which alone, however, does not imply the existence of a perfect matching in H. Having a randomized construction, at this point we would like to apply the following random graph theorem [2]:

(A) For any fixed $k \geq 1$, almost every[6] random k-regular graph has a perfect matching.

The direct application of (A) is prevented, however, by the fact that H is not a random k-regular graph as required by the theorem, as it is not chosen uniformly at random among all k-regular graphs, being the complement of the union of $i-1$ disjoint perfect matchings,

[5] Multiplication and addition of events denote the AND and OR operations, respectively.

[6] The term "almost every" is often used in the theory of random graphs, with the precisely defined meaning that the fraction of graphs that have the considered property approaches 1 as $n \to \infty$. We also often say in this case that the property holds *almost surely*.

which is not necessarily true for all regular graphs of degree $k = n - i$. To help this, we use a (deep) theorem published in [8]. This theorem is based on the concept of *contiguity* of probability spaces. Two probability spaces that depend on a parameter n are called contiguous if for any sequence of events (one event for each n) the following holds: if the sequence is *almost sure* in one of the probability spaces, i.e., the sequence has probability approaching 1 as $n \to \infty$, then it is also almost sure in the other probability space in the same sense. It is proven in [8] that:

(B) *For any $\Delta \geq 3$ the probability space generated by random Δ-regular graphs is contiguous to the probability space generated by taking the union of Δ disjoint random perfect matchings.*

Now, when starting the i^{th} phase, what we have already generated is the union of $i-1$ disjoint perfect matchings. Let us call this graph F_{i-1}. By Lemma 1, the matchings are all chosen uniformly at random from the ones available at the time of the choice. Thus, F_{i-1} is the union of $i-1$ disjoint random perfect matchings. By (B) we have the following property:

(C) For $i > 3$ any property holds almost surely for F_{i-1} if and only if it holds almost surely for a random regular graph of the same degree.

Now, taking into account that the complement of a random $i-1$ regular graph is a random k-regular graph with $k = n - i$, and by (A) a random k-regular graph almost surely has a perfect matching, we have that the complement of a random $i-1$-regular graph almost surely has a perfect matching. By (C) this implies that the same property holds for F_{i-1}, as well, so H, being the complement of F_{i-1}, almost surely has a perfect matching, too, which proves the lemma.

□

Now, being armed with the two lemmas, we can prove the main theorem.

Proof of the Theorem.

1. The first claim directly follows from that the complexity of the algorithm is dominated by computing the maximum weight perfect matching that can be done in $O(n^3)$ time (see [6]). This is executed at most Δ times and, since the side computations can clearly be included in the $O(n^3)$ time, therefore, the overall complexity is bounded by $O(\Delta n^3)$.

2. According to the algorithm, in each of the phases we add a disjoint random perfect matching to E. Since there are altogether at most Δ phases and each node gets exactly one additional degree in each phase from the chosen perfect matching, therefore, the returned graph $G = (V, E)$ must be regular of degree at most Δ. Since by Lemma 2 we almost surely do not get stuck in any phase, it follows that with probability approaching 1 as $n \to \infty$, the degree is *exactly* Δ.

3. The connectivity of a Δ-regular graph can be *at most* Δ, since removing all the Δ neighbors of a given node would separate this node from the rest of the graph. Thus, to show the desired optimality it is enough to prove that the constructed graph is Δ-connected with probability approaching 1, as $n \to \infty$. From [2], it is known that:

   (D) *For any $\Delta \geq 3$ almost every random Δ-regular graph is Δ-connected.*

   To apply (D), however, we again have to resolve the problem, just as in the proof of Lemma 2, that our graph is not chosen uniformly at random from the set of all Δ-regular graphs on n nodes. By Lemma 1, our graph is the union of Δ disjoint random perfect matchings (almost surely). This class, does not contain all Δ-regular graphs on n nodes. Here, however, we can make use again of the theorem stated in (B) (in the proof of Lemma 2), which implies, together with Lemma 1 and (D), that if the constructed graph is Δ-regular, then it is almost surely Δ-connected. By 2., the graph is almost surely Δ-regular, so it has the optimal connectivity with probability approaching 1.

4. Let G be any graph on n vertices with maximum degree $\leq \Delta$ and diameter D. Then, according to the well known *Moore bound* (see, e.g., [2]),

$$n \leq \frac{\Delta(\Delta-1)^D - 2}{\Delta - 2} \leq \Delta^D + 1$$

   holds.[7]

   Now, recall from the proof of 3. that, by (B) and Lemma 1, any theorem that holds for a random Δ-regular graph ($\Delta \geq 3$) also holds for our construction. We use the following random graph theorem from [2]:

   (E) *Let $\Delta \geq 3$ and fix any $\rho > 0$. If d is the smallest integer for which*

$$(\Delta - 1)^{d-1} \geq (2 + \rho)\Delta n \log n \qquad (3)$$

   holds, then almost every Δ-regular graph on n vertices has diameter less than or equal to d.

   By (B) and Lemma 1 this property holds for our randomly constructed graph, too. Thus, if the constructed graph has diameter D, then from (3) we obtain (using the $O(.)$ order notation to avoid carrying

[7] The extreme graphs that satisfy the Moore bound with equality are called *Moore graphs*. They have the maximum number of nodes under the degree and diameter constraints. They are very rare, however, for example, it is proven in [9] that no Moore graph can exist for $\Delta \geq 3$ and $D \geq 3$.

constants) that with probability approaching 1

$$D \leq \frac{\log n}{\log(\Delta - 1)} + O\left(\frac{\log\log n}{\log \Delta}\right)$$

must hold. On the other hand, from the Moore bound we obtain

$$D_{min}(n, \Delta) \geq \frac{\log(n-1)}{\log \Delta}.$$

Since $D \geq D_{min}(n, \Delta)$ must always hold, $D_{min}(n, \Delta)$ being the smallest possible diameter, the combination of the above expressions yields that, with probability approaching 1

$$1 \leq \frac{D}{D_{min}(n, \Delta)} \leq \frac{\log n \log \Delta}{\log(n-1)\log(\Delta - 1)} + O\left(\frac{\log\log n}{\log n}\right) \quad (4)$$

holds. If $\Delta \to \infty$ as $n \to \infty$ then the upper bound in (4) becomes smaller than $1 + \epsilon$ for any fixed $\epsilon > 0$. Thus, we obtain:

$$\lim_{n\to\infty} \mathrm{P}\left(1 \leq \frac{D}{D_{min}(n, \Delta)} < 1 + \epsilon\right) = 1,$$

which completes the proof.

□

Remark. For finite n the algorithm may halt before a Δ-regular graph is constructed, since the probability of getting stuck in some phase tends to zero only asymptotically. On the other hand, if we run the algorithm several times with independent random numbers, then the probability of getting stuck in *each* run decreases exponentially, so the probability of producing the desired graph in at least one of the runs can be made arbitrarily close to 1.

VI. Conclusions

This paper presented an algorithm for designing the logical topology of the virtual path network, which is a practically important task in ATM network management. We have proven that the algorithm creates a VP network topology that is asymptotically optimal with respect to both connectivity and diameter, under given VP processing capacity constraints. While the proof of these properties is quite involved, the algorithm itself exhibits attractive simplicity and ease of implementation. The paper thus shows, for the first time, that the theory of random graphs may serve as a useful tool in generating *algorithms* in networking to overcome the "scalability bottleneck" that prevents current approaches from finding near-optimal solutions as today's networks grow in size and complexity.

References

[1] V. J. Friesen, J. J. Harms, and J. W. Wong, "Resource management with virtual paths in ATM networks," *IEEE Network*, vol. 10, no. 5, pp. 10–20, September–October 1996.

[2] B. Bollobás, *Random Graphs*, Academic Press Inc., London, UK, 1985.

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, CA, 1979.

[4] J. Håstad, "Clique is hard to approximate within $n^{1-\epsilon}$", in *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, Burlington, VT, 14–16 October 1996, pp. 627–636.

[5] B. D. McKay and N. C. Wormand, "Uniform generation of random regular graphs of moderate degree," *Journal of Algorithms*, vol. 11, no. 1, pp. 52–67, March 1990.

[6] M. O. Ball and U. Derigs, "An analysis of alternative strategies for implementing matchings algorithms," *Networks*, vol. 13, pp. 517–549, January 1983.

[7] H. N. Gabow and R. E. Tarjan, "Faster scaling algorithms for generating graph-matching problems," *Journal of the ACM*, vol. 38, no. 4, pp. 815–853, October 1991.

[8] M. S. O. Molloy, H. Robalewska, R. W. Robinson, and N. C. Wormand, "1-factorizations of random regular graphs," *Random Structures & Algorithms*, vol. 10, no. 3, pp. 305–321, May 1997.

[9] E Bannai and T. Ito, "On finite Moore graphs," *J. Fac. Sci. Univ. Tokyo, Sect. IA, Math.*, vol. 20, pp. 191–208, 1973.