Modeling Virus Self-Assembly Pathways: Avoiding Dynamics using Geometric Constraint Decomposition

Meera Sitharam[12]
Mavis Agbandje-Mckenna[23]

Abstract

We develop a model for elucidating the assembly pathways by which an icosahedral viral shell forms from 60 identical constituent protein monomers. This poorly understood process a remarkable example of macromolecular self-assembly occuring in nature and possesses many features that are desirable while engineering self-assembly at the nanoscale.

The model uses static geometric and tensegrity constraints to represent the driving (weak) forces that cause a viral shell to assemble and hold it together. The goal is to answer focused questions about the structural properties of a successful assembly pathway. Pathways and their properties are carefully defined and computed using computational algebra and geometry, specifically state-of-art concepts in geometric constraint decomposition. The model is analyzable and refinable and avoids expensive dynamics. We show that it has a provably tractable and accurate computational simulation and that its predictions are roughly consistent with known information about viral shell assembly. Justifications for mathematical and biochemical assumptions are provided, and comparisons are drawn with other virus assembly models. A method for more conclusive experimental validation involving specific viruses is sketched. Overall the paper indicates a strong and direct, mutually beneficial interplay between (a) the concepts underlying macromolecular assembly; and (b) a wide variety of established as well as novel concepts from combinatorial and computational algebra, geometry and algebraic complexity.

Organization of Paper

1 Introduction and Motivation

Icosahedral viral shell assembly is an outstanding example of nanoscale, macromolecular self-assembly occuring in nature [81]. Mostly identical *coat protein* monomers assemble with high rate of efficacy into a closed icosahedral *capsid* or *shell*; onset and termination are spontaneous, and assembly is robust, rapid and economical. All of these requirements are both desirable and difficult to achieve when engineering macromolecular self-assembly. See Figures 1.

[1] corresponding author: sitharam@cise.ufl.edu; CISE department, CSE Bldg, PO Box 116120
[2] University of Florida, Gainesville, FL 32611, USA; supported in part by NSF grant EIA 0218435
[3] McKnight Brain Institute

However the viral assembly process - just like any other spontaneous macromolecular assembly process such as molecular crystal formation - is poorly understood. Answering focused questions about viral assembly pathways can help both to encourage macromolecular assemblies for engineering, biosensor and gene therapy applications, but and also discourage assembly for arresting the spread of viral infection.

This paper addresses the relevance of computational algebra and geometry to develop static, analyzable, refinable models and fast, accurate computational simulations for answering focused questions about virus assembly pathways. The paper also discusses the relevance of random walks on markov chains for randomly sampling symmetric algebraic structures. Specifically, we use the following. First, we use state of the art methods for decomposition of geometric constraint system [7, 10, 11, 9] [44, 42, 43, 100, 102, 32, 68], as well as solving and estimating number of solutions. These leverage both combinatorial approaches related to rigidity theory [17], as well as standard algebraic techniques for sparse elimination and solving [27, 38, 105, 33]. Second, we use random walks [1, 15, 70, 48, 40, 21, 80] to obtain statistically good samples for enumerating and counting substructures of symmetric algebraic structures that have an interpretation as decompositions of the underlying geometric constraint systems, and are combinatorially related to rigidity matroids [103, 101]. Existence of purely algebraic methods of estimating such statistics [24] is likely and would be useful to find.

1.1 Virus Preliminaries

The viral shell is important in that it packages viral "life" i.e, the genomic nucleic acid, which could be single stranded DNA (*ssDNA*), double-stranded DNA, or RNA. See Figure 1. However, in many cases, viral shell assembly occurs with no interference from the enclosed genetic material: empty shells, or shells packaging incomplete genomic material form with equal facility [61], a fact that simplifies the modeling. A symmetric shell [36] is a consequence of its consisting of (almost) identical monomers. The predominant structure of viral shells is icosahedral since the exact five-fold, three-fold and two-fold point-group symmetry of the icosahedron permits the *quasi-equivalent* symmetry [4] required to construct structures with a large number of monomers (see Figures 1, 8). The number of monomers for each vertex of each triangle of the (20-triangle) icosahedron is refered to as the (typically small) '*T*' number: a T=1 virus shell has 60 identical monomers, a T=7 virus shell has 420 monomers etc. See Figures 1, 8.

Virus assembly involves [37] highly specific monomer-monomer (protein-protein), - and possibly protein-genomic material *interactions*, all of which are governed by geometry or by weak forces that can be treated geometrically [13] (see Figure 2). More specifically, the final viral structure can be viewed formally as the solution to a system of geometric constraints that translate to algebraic equations and inequalities. **Note.** The model presented here directly applies (without any additional work) to viruses of any T-numbers, in fact, to any arbitrarily large macromolecular assembly formed from a fixed set of types of monomeric units and a fixed set of types of assembly-driving interactions between them. However, our experimental validation is based on ssDNA T = 1 viral shells, specifically those that assemble without interference from the genomic material or so-called chaperone or scaffolding proteins, since we do not model such interference. Furthermore, T=1 viral shells provide the most exacting validation for our model, since they are the most economical and precise assemblies. Finally, T=1 viral shells provide enough variety so that different viruses can be carefully chosen to test various aspects of our model, For this reason, and for simplicity of presentation, we present our model assuming T=1 rather than for a general T number.

1.2 State of the Art

While there is a well-developed structure theory of *complete* viral shells [36, 4], verified by X-ray crystallography and other experimental data, the *processes* of viral shell assembly are poorly understood. From an experimental point of view, this lack of understanding is due to the extreme rapidity of the assembly so that wet-lab snapshots of intermediate sub-assemblies are generally unsuccessful.

From a modeling point of view, this lack of understanding is due to the fact that existing computational models [52, 54, 55, 37, 58, 110, 29, 28, 25, 41] generally involve dynamics of (simplified versions) of virus assembly (further description of these approaches and comparison with our approach can be found in Section 5. Dynamics are currently used even when the assembly models only seek to elucidate the structure of *pathways*, See Section See Figures 12, 10. See Section 4 for definition. By carefully defining the probability space, we obtain the probabilities of pathway trees/dags *that are known to result in successful assemblies* using a purely static model based on geometric constraints.

Models whose output parameters are defined only as the end result of a dynamical process are computationally costly, often requiring oversimplifications to ensure tractability. In addition, such models are also not easily tunable or refinable since their input-to-output function is generally not analyzable and therefore do not provide a satisfactory conceptual explanation of the phenomenon being modeled.

2 List of Contributions

Contribution 1:
(Section 4) Development of a mathematical model of viral shell assembly whose *input parameters* are: information extracted from (a) the geometric structure of the coat protein monomer that forms the viral shell, including all relevant (rigid) conformations; (b) the geometric and weak-force interactions - between pairs of monomers - that drive assembly (see Figure 2); and (c) (optional) the neighborhood structure of the complete viral shell. The latter is crucial for a focused model that *only* deals with *those* pathways that are known *apriori* to lead to a complete viral shell. However, the model can be generalized to the case where (c) is not part of the input and unsuccessful assemblies are included. The imput utilized by the model is complete in that it comtains all of the information that is considered necessary to drive assembly.

The output information sought from the model: first, the probability of a specific successful assembly pathway that incorporates a specific subassembly and leads to the complete viral shell and has bounded total *effort*; in short, a probability distribution over successful, bounded effort assembly pathways that incorporate certain substructures; this has a straightforward generalization (Section 4.5) to a distribution over all possible assembly pathways (not necessarily successful) within an effort bound. The model satisfies the following requirements.

(i) (Section 4) The description of the model - i.e. the input-to-output function - is static, i.e. does not rely on dynamics of the assembly process. This is essential for forward analyzabilty.

(ii) The assumptions of the model are mathematically and biochemically justifiable. These justifications and comparisons of the model with existing models of viral shell assembly are given in Sections4.5, 5, and 8. The mathematical assumptions are clearly classified as pathway structure assumptions (concerning stability of subassemblies, validity and effort rating of pathways) and pathway probability space assumptions, and these assumptions are further categorized so that it will be clear that even if some of these assumptions were to be changed, other aspects of the model would still apply and the model would continue to be useful and novel.

(iii) (Section 6) The model is computationally tractable, i.e. is accompanied by an efficient algorithm for computing (a provably good approximation of) the pathway probability distribution. This is essential for backward analyzability which is needed for two reasons: first, for iteratively refining the model so that its output matches known biochemical information or experimental results; and second, for engineering a desired output, for example engineering the monomer structure to prevent/encourage certain subassemblies, inorder to force certain pathways to become more likely than others, or to prevent successful assembly.

Contribution 2:
(Section 7) Preliminary simulation results are given showing that, in principle, the model's predictions are qualitatively consistent with known studies of virus assembly.
(ii) (Section 8) For a more complete validation, we describe the process of designing the inputs to the computational model starting from real data from specific T=1, ssDNA viruses, in particular, from cryo electron microscopy and sequence mapping structure studies or X-ray structure of Murine Parvovirus minute virus of mice (MVM) and Maize streak virus (MSV), Figure 2. Input design for other viral structures such as AAV - Adeno-associated virus - are in process. We give a justification of the choice of viruses to test specific aspects of the model. We additionally sketch the experiment design. Conclusive experimental validation is in process.

Contribution 3:
Overall, the paper provides an indication of the direct, mutually beneficial interplay between (a) the concepts underlying macromolecular assembly and (b) established as well as novel concepts from combinatorial and computational algebraic geometry and algebraic complexity.

3 Geometric constraint solving and Tensegrity background

Geometric constraint systems arise in a wide variety of applications including robotics, mechanical computer aided design, and teaching geometry [108, 93, 107, 99, 91, 12, 19, 57] [26, 77, 31, 62, 60, 5] [7, 10, 11, 9] [16, 20, 9, 63, 64, 95] [96, 71, 94, 97, 59] [44, 42, 43, 100, 68, 102, 32]. More relevantly, geometric constraints are used in molecular conformational structure determination and representation [65], [76], [72]. For recent reviews of the extensive literature on geometric constraint solving see, e.g, [7, 90, 78]. Most of the constraint solvers so far deal with 2D constraint systems, although some of the newer approaches including [6, 8, 10, 11] [9, 71, 94] [44, 42, 43, 100, 68, 102, 32], extend to 3D constraint systems.

A *geometric constraint system* consists of a finite set of geometric objects and a finite set of constraints between them. See Figure 3. The constraints can usually be written as algebraic equations and inequalities whose variables are the coordinates of the participating geometric objects. For example, a distance constraint of d between two points (x_1, y_1) and (x_2, y_2) in 2D is written as $(x_2 - x_1)^2 + (y_2 - y_1)^2 = d^2$

A *solution or realization* of a geometric constraint system is the real algebraic variety or (set of) real zero(es) of the corresponding algebraic system. In other words, the solution is a class of valid real instantiations of (the position, orientation and any other parameters of) the geometric elements such that all constraints are satisfied. Here, it is understood that such a solution is in a particular geometry, for example the Euclidean plane, the sphere, or Euclidean 3 dimensional space. A constraint system can be classified as *overconstrained*, *well-constrained*, or *underconstrained*. Well-constrained systems have a finite, albeit potentially very large number of *rigid* solutions or *conformations*; their solution space is a zero-dimensional variety. Underconstrained systems have infinitely many solutions; their solution space is not zero-dimensional: it is called a *conformational or configuration space* A *roadmap* of this conformational set (space) – capturing connectivity and representing conformational regions of topologically distinct classes of configurations – is usually part of the realization. These classes also overlap with regions that are associated with fixed relative orientations of the set of geometric primitives and these regions can be "picked out" using so-called chirality constraints that are determinantal inequalities. Overconstrained systems do not have a solution unless they are *consistently overconstrained*. Well or overconstrained systems are called *rigid* systems.

The question of "to what extent can geometric constraint problems be approached combinatorially?" is important. Since a significant proportion of the results of this paper rely on combinatorial approaches, we discuss these and their limitations here.

3.1 Constraint Graphs and Degrees of Freedom

A geometric constraint graph $G = (V, E, w)$ corresponding to geometric constraint problem is a weighted graph with n vertices (representing geometric objects) V and m edges (representing constraints) E; $w(v)$ is the weight of vertex v and $w(e)$ is the weight of edge e, corresponding to the number of *degrees of freedom (dofs)* available to an object represented by v and number of degrees of freedom removed by a constraint represented by e respectively.

For example Figures 4 and 5, and Figure 3 show a 3D and 2D constraint graphs. All of these examples involve only points and distances: see [102, 100] for a variety of examples including other objects and constraints.

Note that the constraint graph could be a *hypergraph*, each hyperedge involving any number of vertices. A subgraph $A \subseteq G$ that satisfies

$$\sum_{e \in A} w(e) + D \geq \sum_{v \in A} w(v) \tag{1}$$

is called *dense*, where D is a dimension-dependent constant, to be described below. Function $d(A) = \sum_{e \in A} w(e) - \sum_{v \in A} w(v)$ is called *density* of a graph A.

The constant D is typically $\binom{d+1}{2}$ where d is the dimension. The constant D captures the degrees of freedom of a rigid body in d dimensions. For planar contexts and Euclidean geometry, we expect $D = 3$ and for spatial contexts $D = 6$, in general. If we expect the rigid body to be fixed with respect to a global coordinate system, then $D = 0$.

Next we give some purely combinatorial properties of constraint graphs based on density. These will be later shown to be related to properties of the corresponding constraint systems.

A dense graph with density strictly greater than $-D$ is called *overconstrained*. A graph that is dense and all of whose subgraphs (including itself) have density at most $-D$ is called *wellconstrained*. A graph G is called *well-overconstrained* if it satisfies the following: G is dense, G has atleast one overconstrained subgraph, and has the property that on replacing all overconstrained subgraphs by wellconstrained subgraphs, G remains dense. A graph that is wellconstrained or well-overconstrained is said to be *rigid* or a *cluster*. A dense graph is *minimal* if it has no dense proper subgraph. Note that all minimal dense subgraphs are clusters but the converse is not the case. A graph that is not a cluster is said to be *underconstrained*. If a dense graph is not minimal, it could in fact be an underconstrained graph: the density of the graph could be the result of embedding a subgraph of density greater than $-D$.

To discuss how the graph theoretic properties based on *degree of freedom (dof) analysis* described above relate to corresponding properties of the corresponding constraint *system*, we need to introduce the notion of genericity. Informally, constraint system is generically rigid if it rigid for most of choices of coefficients of system. More formally we use the notion of *genericity* of e.g, [30]. A property is said to hold *generically* for polynomials $f_1, \ldots, f_n$ if there is a nonzero polynomial P in the coefficients of the f_i such that this property holds for all $f_1, \ldots, f_n$ for which P does not vanish.

Thus the constraint system E is generically rigid if there is a nonzero polynomial P in the coefficients of the equations of E - or the parameters of the constraint system - such that E is solvable when P does not vanish. For example, if E consists of distance constraints, the parameters are the distances. Even if E has no overt parameters, i.e, if E is made up of constraints such as incidences or tangencies or perpendicularity or parallelism, E in fact has hidden parameters capturing the extent of incidence, tangency, etc., which we consider to be the parameters of E.

According to Laman's theorem [92] in 2D, if all geometric objects are points and all constraints are distance constraints between these points then any minimal dense cluster represents a generically rigid system. However, in 3D or in 2D with other constraints such as angle constraints, a generically rigid system always gives a cluster, but the converse is not always the case. In fact, there are well-constrained, even minimal dense clusters whose corresponding systems are not generically rigid and are in fact generically not rigid. A classic example is the so-called "bananas" problem in 3D, which can be detected as the root cause beneath large class of combinatorial misclassifications, although this detection is nontrivial. See [102, 68]. Here two clusters independently fix the distance between 2 points shared by them, causing an *algebraic overconstraint*. A combinatorial analysis will falsely report the pair as a well-constrained or rigid cluster, while in fact, the cluster is underconstrained (if the algebraic overconstraint is consistent) since the two clusters can rotate about the axis formed by the two shared points.

Another standard example, in 4 dimensions the graph $K_{7,6}$ representing distances is minimal dense, and hence a cluster, but it does not represent a generically rigid system.

In fact, there is no known, tractable characterization of generic rigidity of systems for 3 or higher dimensions, based purely properties of the constraint graph [17].

NOTE: Having noted these problems, we will nevertheless rely heavily on combinatorial dof analysis of constraint graphs - carefully augmented by some checks and corrections for algebraic dependences such as the "bananas" problem, given in [102, 68], to determine generic rigidity constraint systems; hence from now on we will use the terms *rigid system* and *cluster* interchangeably.

3.2 The need for decomposition: DR-plans and their properties

Now we describe a structure called the DR-plan which is crucial for our viral assembly pathway model. These structures are natural decompositions of geometric constraint systems and one of their many motivations (see [7, 10, 11, 42, 44, 102, 100, 32]) is that the overwhelming cost of solving a geometric constraint system is the size of the largest subsystem that is solved using a direct algebraic/numeric solver. This size dictates the practical utility of the overall constraint solver, since the time complexity of the constraint solver is at least *exponential* in the size of the largest such subsystem.

Therefore, an effective constraint solver should *combinatorially* develop a *plan* for (recursively) *decomposing* the constraint system into small subsystems, whose solutions (obtained from the algebraic/numeric solver) can be (recursively) *recombined* by solving other small subsystems. Such a recombination is straightforward, provided all the subsystems are generically rigid (have only finitely many solutions). The DR-planner is a graph algorithm that outputs a *decomposition-recombination plan (DR-plan)* of the constraint *graph.* In the process of combinatorially constructing the DR-plan in a bottom up manner, at stage i, it locates a wellconstrained subgraph or cluster S_i in the current constraint graph G_i, and uses an abstract *simplification* of S_i to to create a transformed constraint graph G_{i+1}.

Although recursive decompositions were used for geometric solving from the beginning, DR-plans and their properties were formally defined for the first time in [10]. Formally, a DR-plan of a constraint graph G is a directed acyclic graph (DAG) whose nodes represent clusters in G, and edges represent containment. The leaves or sinks of the DAG are all the vertices (primitive clusters) of G. The roots or sources are a complete set of maximal clusters of G. See Figures 3, 4 and 5. There could be many DR-plans for G. An *optimal* DR-plan is one that minimizes the maximum fan-in. The *size* of a cluster in a DR-plan is its fan-in (it represents the size of the corresponding subsystem, once its children are solved).

The DR-plan additionally incorporates another partial order called the *solving priority order*, which is consistent with the DR-plan's DAG order, but is more refined. This is particularly useful in 3D for correcting inaccurate dof analyses [68, 102]. The intent is that clusters that appear later in the order need to be solved after the clusters that appear earlier. In fact, the nodes in such a DR-plan may not be independent clusters that appear in the original constraint graph or constraint system. They become wellconstrained clusters only in the transformed constraint system (resp. graph) after earlier clusters in the solving order are already solved (resp. simplified).

A few other properties of DR-plans are of interest. We would like the *width* i.e, *number* of clusters in the DR-plan to be small, preferably linear in the size of G: this reflects the complexity of the planning process and affects the complexity of the solving process that is based on the DR-plan. Since non-minimal dense subgraphs could be misclassified as clusters, and for other reasons such as correcting misclassifications due to algebraic overconstraints, correcting combinatorial overconstraints, and for updating the constraint system, [42, 44, 102, 68], it is desirable for DR-plans to have the *cluster minimality* property: i.e., for any node in the DR-plan, no proper subset of its children induces a cluster. Another desirable property is that the DR-plan *incorporate an input partial decomposition*. I.e., given an input DAG P whose nodes are subgraphs of G and whose edges represent containment, a DR-plan of every node in P should be embedded in the output DR-plan for G.

All properties defined above for DR-plans transfer as performance measures of the *DR-planners* or DR-planning algorithms. It is shown in [42], that the problem of finding the optimal DR-plan of a constraint graph is NP-hard, and approximability results are shown only in special cases. Nonapproximability results are not known. However, most DR-planners make adhoc choices during computation (say the order in which vertices are considered) and we can ask of how well (close to optimal) the *best* computation path of such a DR-planner would perform (on the worst case input). We call this the *best-choice approximation factor* of the DR-planner.

3.3 Tensegrity

Tensegrity structures were invented by K. Snelson (popularized by Buckminster Fuller) [74]. The word *tensegrity* is obtained from the combination of the words tension and integrity [75], [79]. Tensegrity structures are very general spatial structures formed by a combination of rigid elements in compression (struts, or repulsive forces) and connecting elements that are in tension (ties or attractive forces). No pair of struts touch and the end of each strut is connected to non-coplanar ties [109]. The struts are usually, but not always, considered to have fixed length (or in a small interval) but the ties may or may not be compliant, with the force on them typically, but not necessarily, (inversely) proportional to the square of their length (as with tensile or electrostatic forces). The entire configuration stands by itself and maintains its form solely because of the internal arrangement of the struts and ties [106]. See Figure 6.

The development of tensegrity structures is relatively new and the works related have only existed for approximately twenty five years [87, 106, 109, 104, 89]. The method for solving the position solution of tensegrity systems is usually addressed in one of three ways. Force balance equations can be written for the node points of the system and the solution of this set of equations can yield the equilibrium position(s). In the second approach, equilibrium can be found by determining minimum potential energy configurations of the system. Thirdly, equilibrium can be determined by finding configurations where the virtual work done by an external force/moment is zero. These approaches have to date led to the closed-form determination of equilibrium configurations for special cases, such as when the tensegrity system is symmetric [104], which is sufficient for our virus application.

Our virus assembly model will leverage the fact that the first two methods of representing tensegrity structures can be directly formulated using abstract tensegrity *frameworks* [69], [73] – force balance equations are effectively geometric constraints. Thus general tensegrity systems are special cases of general geometric constraint systems and viceversa, common types of the latter, such as distance constraint systems are special cases of general tensegrity systems.

4 The Virus Assembly Model

The original architects of the so-called *quasi-equivalence* theory of viral structure and viral shell self-assembly, Caspar and Klug, [4], by their own admission, informally derived their inspiration from tensegrity structures. Since some of the aspects of the original quasi-equivalence model that were based on highly flexible and deformable subunits have been refuted, this has wrongly discredited tensegrity models based on the false assumption that tensegrity systems are necessarily highly flexible. To the contrary, general tensegrity systems include all (including rigid or nearly rigid) geometric constraint systems. In fact, *almost all* viral structure and assembly models (including those that use molecular dynamics or other dynamics simulations) are based entirely upon specific weak forces driving assembly and standard geometry (bond lengths, bond angles and torsion angles) of the monomers. This leads to the observation that all such models *informally and implicitly* treat viral shells as tensegrity systems within the general definition of tensegrity.

The main idea here is to use a geometric and tensegrity constraint formulation of viral structure to give a *formal, static* definition of a viral assembly pathway (i.e, a partial order of subassemblies), as a certain type of DR-plan of the underlying constraint system. We then give an effort rating for each pathway indicating the difficulty of subassembly formation along the pathway.

The input parameters of the viral assembly model are viewed as a geometric constraint system which is then represented as a geometric constraint graph based on degrees of freedom as described in Section 3. More specific definitions below.

4.1 Formal definition of the Model Input

Definition A *viral geometric constraint system* is specified in 4 parts. It is based on the assumption that the viral shell is icosahedral and made up of 60 identical protein molecules as in Figure 1. Hence both the *monomer structure constraints* and the *interface constraints* ((1) and (2) below) are specified just for a single reference monomer.

(1) The *monomer structure constraint system* (see Figure 6): the primitive objects in this system are points representing essential *atomic markers* and line segments representing essential *bonds* on the backbone and side chains of the protein monomer. The constraints in this system are of three distinct types.

(a) The *primary stucture* constraints consisting of distance, angle, torsion angle intervals between the points and line segments; these represent bond lengths, bond angles and torsion angles involving the corresponding atomic markers and bonds. These constraints typically form a polygonal chain with side chains.

(b) The *monomer weak force* constraints consisting of distance intervals and tensegrity forces between the points; these represent hydrogen bonds and other weak forces between atomic markers. This is also called the *monomer contact map*.

(c) Required relative orientation constraints on subsets of 4 points (see Section 3): these represent allowed chiralities of the corresponding atomic markers and are used pick out the allowed (rigid) conformations of the monomer. These constraints can be replaced in some cases by extra distance constraints between the relevant points (atomic markers).

(2) The *interface constraint system* (see Figures 6, 7) This consists of 3 constraint systems called A-P(A), A-T(A) and A-D(A), involving 4 monomers A, P(A), T(A) and D(A). Each constraint system is between the reference monomer A and one of its reference neighbors P(A), T(A) or D(A), across a pentamer, trimer or dimer interface respectively. The monomer A participates in two more symmetric pentamer and trimer interfaces P^{-1}(A)-A and T^{-1}(A)-A, with monomers P^{-1}(A) and T^{-1}(A), but these need not be specified as they can be inferred from the constraint systems A-P(A), A-T(A), where P^{-1}(A) (resp. T^{-1}(A)) takes the place of A, and A takes the place of P(A) (resp. T(A)). Each constraint system consists of distance intervals or tensegrity forces. Each of these constraints involves one point in A and one point in P(A), D(A) or T(A), depending on the interface. These represent the weak forces or interactions between the monomers that drive assembly. These are also called the *interface contact maps.* (3) The *neighborhood structure* (see Figures 8:) A regular, directed graph with labeled vertices representing the monomers and edges representing the icosahedral adjacency structure imposed by the interfaces. The labels are typically of the form Nx, where N is the label of one of the icosahedral triangular faces and x is one of the icosahedral vertices. Nx is the monomer closest to vertex x and face N. As described above, each vertex A has 3 outgoing edges A-P(A), A-T(A) and A-D(A), representing its forward pentamer, forward trimer and dimer interfaces and 2 incoming edges P^{-1}(A)-A and T^{-1}(A)-A, representing its backward pentamer and backward trimer interfaces (the backward dimer is generally irrelevant and is omitted).

(4) *Global chirality constraints:* these are not crucial, but can be used to specify relative orientations of sets of 4 atomic markers that do not belong to the same monomer. These could be icosahedrally symmetric, or not. in which case, it is sufficient to specify these for the reference monomer alone, i.e., at least one of the participating atomic markers in each constraint belongs to the reference monomer.

As in the case of the monomer chirality constraints, these constraints can be replaced in some cases by extra distance constraints between the relevant points (atomic markers). These constraints represent external restrictions on the conformations of the complete viral shell that are not captured by the input (1c) restrictions on monomer conformations. See Section 8 for the method by which these constraints are chosen. ♣ (end of definition)

4.2 Formal definition of Pathways and Effort ratings

Definition. A *pathway* P for a viral constraint graph G is a DR-plan for G (see Section 3 and Figures 8.4 3) that additionally satisfies 3 properties.
(i) No pair of clusters in the pathway intersect on a nontrivial cluster (for DR-plans, this invariant is slightly weaker – see below) unless one is contained in the other;
(ii) No cluster is formed entirely by overlap constraints between the children (i.e., only the overlap constraints between the children should be inadequate to form the parent cluster).
Instead of the cluster-minimality property of DR-plans, pathways should satisfy the following:

(iii) Every cluster in the pathway has the following *overlap-minimality* property. ♣ (end of definition).

In fact, Properties (i) and (ii) are inconsistent with the cluster-minimality property of DR-plans. Since nonminimal dense graphs may be misclassified as clusters, some form of cluster-minimality check has to be performed by the DR-planner before subgraphs can be correctly classified as clusters: however, due to Properties (i) and (ii) above, those clusters may have to be enlarged, before they are put into the pathway. See Section 6 for a description of this process. Intuitively, Properties (i) and (ii) assert that when two clusters overlap on a primitive object, then they are linked by that object. I.e., a biophysically valid decomposition cannot in general make "copies" of the variables corresponding object, treat them independently in separate clusters and then equate them as an overlap constraint, unless this overlap constraint is does not force the participating clusters to form a cluster. This is crucial in giving a biophysically valid measure of *effort* required form clusters defined below.

Definition. A cluster C with children $C_1, \ldots, C_m$ is *overlap-minimal* if for any cluster C' formed by a proper subset of child clusters $C_1, \ldots, C_k$, $1 < k < m$, it holds that: C' along with the remaining child clusters $C_{k+1}, \ldots, C_m$ form the cluster C using overlap constraints alone. See Figure 16. ♣(end of definition)

Notice that the Property (i) above guarantees the the *width* (see Section 3) of a pathway and in fact the total number of clusters in a pathway is roughly linear in the number of vertices in the viral constraint graph.

Assumption 1: Overlap-minimal clusters of a viral constraint graph G have a constant *size or fan-in* (see Section 3), i.e., number of children, which is independent of the number of vertices of G. This is based on observations of a large number of known viral structures. See Section 8 for detailed biochemical justifications of this and other model assumptions.

Notice that clusters could be constructed from parts of monomers. I.e., clusters are not necessarily *subassemblies,* i.e., consisting of whole monomers. This provides our pathways a sophistication that appears to be essential for making good predictions. See Figure 9.

Each cluster C does have a subassembly denoted *sub*(C) loosely associated with it, namely, that subassembly consisting of that set of monomers that the cluster overlaps.

A cluster C represents a (λ)-*stable* subassembly sub(C) only if a suficiently large fraction ($\geq \lambda$) of the points or atomic markers in sub(C) are actually present in C. A subassembly is *stabilizable* if a sufficiently large portion of it is a cluster. While a subassembly may be stabilizable, it may not be stable at a particular point in a pathway, i.e., when the corresponding cluster in the pathway does not encompass a large enough portion of the subassembly. Furthermore, by our definition of pathways, even a minimal stable subassembly (i.e., no subassembly contained in it is stable) may have several pathways to *its* formation.

Thus each pathway P embeds a unique coarser sub-pathway consisting of those clusters that represent stable subassemblies. We call this sub-pathway the embedded *stable subassembly pathway* in P. Figures 11, and 12 show a stable subassembly pathway based on 2 stable subassemblies: pentamers and trimers of pentamers. Figures 9, and 10 show a stable subassembly pathway based on 2 stable subassemblies: trimers and pentamers of trimers.

Note that a pathway could have single root or many roots or sources depending on whether the complete viral constraint system is (or forces the shell to be) rigid.

Assumption 2: The constraints in a viral constraint graph G are sufficient to enforce a stable entire assembly or viral shell; i.e., the pathways for G will contain one giant root or source cluster C whose associated subassembly sub(C) is the entire assembly. In other words, C contains the bulk (at least λ fraction) of the entire assembly or shell. The other sources of the pathway represent other smaller rigid components that appear within flexible sidechains and surface decorations of the shell. See Section 8 for justification of biochemical assumptions.

Note that Assumption 2 implies that for any pathway P of a viral constraint graph G, the embedded stable subassembly pathway has a single root.

We have just defined *labeled* pathways (i.e., where the leaves or geometric primitives are labeled). But we will be primarily interested in the probabilities associated with unlabeled pathways or *pathway isomorphism classes* for a viral constraint graph G. These are defined in the obvious manner as equivalence classes of the equivalence relation *perm* between pathways, induced by the automorphisms of G (i.e, those permutations of the vertices of G that preserve the geometric object types that the vertices represent; and additionally preserve the edges, including the geometric constraint types associated with the edges) More specifically, for two pathways P and Q of G are related by *perm*(P, Q) if P can be obtained from Q by applying some permutation - from the automorphism group of G - to the leaves or sinks, i.e., the primitive geometric objects of P.

Each pathway is assigned an *effort rating*, which will be formally based on (and inversely related to) the inherent algebraic complexity of the subsystems that appear in the pathway or alternately, the difficulty of formation of the clusters. We would like this to be a complexity measure satisfying several requirements. (i) It should be somewhat independent of the properties specific to known algorithms for solving the subsystems, or other arbitrary variables, for example elimination order (the complexity measure should, for example, assume the best order) (ii) Furthermore, we would like the value of this measure to be polynomial time computable, given a subsystem. (iii) On the other hand, it would be desirable to have an algorithm for solving the subsystem whose running time provides a reasonably close upper bound on the value of this measure atleast for a large, well-defined class of subsystems. (iv) Finally, as mentioned earlier in the context of overlap-minimality, the measure should have a biophysical justification, for example, it should be related to the energy barriers that are overcome in physically forming the cluster (solving the subsystem) or the geometric or topological precision of the motions of the child clusters that are necessary to form the cluster.

These considerations and the fact that viral geometric constraint systems are typically sparse, motivate us to use a refined version of the familiar BKK bound [22] as an appropriate measure. This bounds the number of solutions of a polynomial system, is usually superior to the Bezout bound for sparse systems, is computed as their *mixed volume* and is by now used as a standard tool in sparse polynomial resultant computation, elimination and root finding algorithms, both symbolic and semi-numeric [27], [38], [105], [33]. Although we are interested only in (isolated) real solutions, we find the BKK bound better for our purposes although it bounds the number of complex roots rather than the the Khovanskii fewnomial [88] bounds which actually estimate real roots. First some standard definitions.

The *Newton polytope* $New(P)$ of a k-variate polynomial P in variables $x_1, \ldots, x_k$ of total degree d, denoted as $P(x) = \sum\limits_{\alpha} a_\alpha x^\alpha$ (where $\alpha = \alpha_1, \ldots, \alpha_k$, $\sum_i \alpha_i \leq d$ and x^α denotes $x_1^{\alpha_1} \ldots x_k^{\alpha_k}$) is the convex hull of the points $\alpha \in \mathbb{Z}^k$ for which the coefficient a_α is nonzero. The *mixed volume* of a set S of polynomial equations $P = 0$ in k variables is defined as an alternating sum of the volumes of the Minkowski sums of subsets the Newton polytopes $New(P)$: $MV(S) = \sum\limits_{Q \subseteq S} (-1)^{|Q|} MS_{P \in Q}(New(P))$ where MS denotes the Minkowski sum. The *BKK* (Bernstein-Kuchirenko-Khovanskii) bound [22] says that the number of solutions to a system S of k polynomials in k variables is bounded by $MV(S)$ (this is an equality if the polynomials are generic, i.e., if their coefficients are algebraically independent).

Mixed volume computations are done using so-called *mixed subdivisions* of the underlying polytopes. Using these, algorithms that are polynomial in the degree and number of terms, and only singly exponential in the number of variables [27] are known for computing the resultant and solutions of sparse systems. Moreover, mixed subdivisions give rise to a class of numeric algorithms called *polyhedral homotopy* algorithms [105, 33] that find all roots, and which run for time roughly proportional to the mixed volume. Together with the fact that clusters in pathways (see definition above), are not underconstrained, i.e., they suit the BKK bound, and have only finitely many solutions, and the fact that they are overlap-minimal, along with Assumption 1 stating that overlap-minimal clusters of viral geometric constraint systems have constant size, our above Requirements (i), (ii) and (iii) are therefore met by the mixed volume and the BKK bound.

We now describe the effort rating of a pathway based on the BKK bound. The crucial measure is the effort rating of a cluster in a pathway and it incorporates three intuitive ideas. (a) Primary structure constraints are - in a physical sense - “solved” constraints, i.e, the underlying monomers and subassemblies inherently satisfy them; hence the effort in forming a cluster is based on the other unsolved (weak force) constraints both between its child clusters. (b) Overconstrained clusters offer different ways of resolution and the best choice is assumed in the computation of effort. (c) Although the clusters in pathways are overlap-minimal, they could be further *algebraically reduced* in well-defined ways that are biophysically justifiable: for example, as mentioned earlier under the context of overlap-minimality, the decomposition cannot indiscriminately "make copies" of the variables corresponding to overlapped objects.

One such method of algebraic reduction is shown in Figure 13 [32]. On Left: solving by first fixing 6 dofs (degrees of freedom) of 1 cluster resolving 2 of the overlaps; and then solving the algebraically reduced system consisting of 1 overlap (3 constraints) and 3 distance constraints simultaneously for all 3 rotational dofs for each of the other two clusters – *6 constraints, 6 variables.* On Right: first solving one algebraically reduced system of a triangle of distances, one from each cluster, between the overlapped points, fixing the 6 dofs of this new triangular cluster and resolving the 3 overlaps (this fixes all but 1 rotational dof for each of the 3 original triangular cluster); then solving the second algebraically reduced cluster 3 distance constraints simultaneously for 1 rotational dof per original triangular cluster – *only 3 constraints and 3 variables.*

Definition Let S be an overlap-minimal, algebraically-reduced cluster (subsystem) of a viral constraint system, and let S contain a set Q of primary structure constraints. Let $S_1, \ldots, S_k$. be the child clusters (subsystems) of S; hence k is the size or fan-in of S. S could be overconstrained, but any of its wellconstrained overlap-minimal subsystems must involve all of $S_1, \ldots, S_k$, due to overlap-minimality. The *effort rating* of S is denoted $Ef(S)$ and defined below. Here the minimum runs over all well-constrained overlap-minimal subsystems S' of S.

$$Ef(S) = w_1 2^{w_2 k} + \min_{S' \subseteq S} w_3 (\prod_i Sol(S_i))/Sol(S') + w_4 Sol(S'),$$

where $Sol(S') = MV(S') - MV(S' \cap Q)$, and $w_1, \ldots w_4$ are the viral model's tunable parameters. ♣ (end of definition) Here the first summand in $Ef(S)$ is simply the conjectured (and observed) time complexity of solving a cluster S of size or fan-in k, i.e., a sparse system S in $O(k)$ variables. The second summand represents the ratio of the size of the search space to the size of the solution space, and the second summand roughly represents the size of the solution space or the effort in listing all the solutions of S' assuming that the equations in $S' \cap Q$ (primary structure constraints) are already solved. This intuitively justifies the presence of both summands in the definition of $Ef(S)$.

4.3 The Pathway Probability Space

Our probability space is the set of all labeled pathways of a complete viral constraint graph. By definition of effort, and the isomorphism classes of pathways, isomorphic pathways have the same effort. For any fixed effort value, we assume a uniform probability distribution over all labeled pathways of that effort.

Assumption 3: An underlying natural assumption is an additional inverse relationship between a pathway's probability and its effort (see Section 8 for biochemical justifications of modeling assumptions). While this assumption intuitively motivates the results of this paper, they are not formally based on this assumption. For fixed effort bounds, we are generally interested in the probability of an unlabeled pathway or a pathway isomorphism class, which is proportional to the size of the isomorphism class. This type of probability and effort computation is sufficient to answer our focused questions about pathways as will be shown in Section 4.5.

4.4 A Restricted Model

While the formal definition of pathways and effort ratings given above are well-posed for the general viral constraint systems given above, here we consider only restricted viral constraint systems that include no inequalities and involve tensegrity forces in highly limited ways. Inorder to include these, we need a suitable generalization of clusters (and hence pathways) to include underconstrained systems. More importantly, the tractable and accurate computational simulation of the model given in Section 6 applies only to the restricted viral constraint systems.

4.5 Mathematical Assumptions and Justifications

We discuss two distinct types of assumptions: those concerning pathway structure and those concerning the pathway probability space. These are relatively independent and even if one set of assumptions changes, the other can still be applied to give a useful model.

Pathway structure.

The notions of a stable subassembly, effort of formation, and valid pathway are all defined geometrically. One justification for this has been mentioned earlier: assembly-driving interactions are, in effect, geometry-based and energy minimization in tensegrity systems can be expressed as the satisfaction of geometric constraints. Furthermore, the model offers candidate geometric properties for defining stability, effort etc. but offers a fairly wide scope for the exact choice of definition. In fact, the intent is to iteratively refine or tune these definitions (and hence the model) to fit experimental results. For example, as a first cut, stability of a subassembly is considered equivalent to approximate combinatorial rigidity. Firstly, this means that any apriori chosen level of underconstrainedness or flexibility can be incorporated into the notion of stability and the theory of DR-plans and pathways that has been developed in the previous sections still goes through. Secondly, and more importantly, the physical meaning of general combinatorial rigidity should not be confused with physical rigidity: it changes based on what the input geometric constraints are – as mentioned above, these could be chosen so that combinatorial rigidity would imply not physical rigidity, but rather energy-based stability.

Next, our *avoidance of dynamics* rests crucially on the ability to restrict our treatment to *monotonic* pathways, i.e, we assume that “disassembly” or the break-up of intermediate subassemblies - which is incorporated into well-developed formal molecular models of computation based on dynamical systems [23, 2] - can be avoided due to our focus on specific questions concerning successful pathways (see pathway probability space assumptions, below). Our pathway, by definition, only contains successful subassemblies i.e., ones that did not disassemble or break-up after they assembled. The effort rating assigned to a subassembly can be made to incorporate the various ways it can form from its successful constituents after potentially several intermediate assemblies and disassemblies; and we typically compute the probability of a pathway (isomorphism type) given a total effort bound. Similar care in designing the effort rating allows us to incorporate solvent interactions.

Pathway probability space. Recall from Section 4 that we are mainly interested in the probability of a pathway isomorphism type, for a complete viral constraint graph, i.e., for a successful assembly of a fixed labeled set of 60 monomers. Unsuccessful pathways and malformations do not enter the picture. One intuitive description: we start with a complete virus, and ask about the different pathways (partial order of subassemblies) in which its 60 labeled monomers in solution could have *monotonically* assembled successfully. We informally view these monomers as though they “know where they are going, are restricted to legal interactions with their neighbors in the complete viral structure, but are completely free to decide in what order they interact.” Thus, all labeled pathways of a given effort rating are considered equally probable and the probability of an (unlabeled) pathway (isomorphism type) of a given effort rating is directly proportional to the *size* of the isomorphism class. The difficulty of the interaction, stability of the resulting subassembly etc. are reflected in the effort rating, not in the probability of the pathways. The underlying understanding is that the probability of a pathway isomorphism type is, in addition, inversely related to the effort.

This type of probability space would give us enough information to conclude, for example, that an overwhelming number of low effort pathways rely on a building around a single pentamer nucleus or that most of the low effort pathways rely on trimeric subassemblies that further assemble using pentameric interactions etc. See Figures in 4. In particular, the symmetries of the final structure could force large isomorphism classes of pathways, so that sheer numbers of pathways could offset relatively large inaccuracies in estimation of effort ratings: i.e., the most likely pathway types remain the same over fairly large changes in the effort calculations.

Furthermore, relative *concentrations* of monomers and other subassemblies [58, 110] that occur during assembly would potentially affect the probability of pathways over a period of time (and viceversa). Even an equilibrium model where the concentration levels of monomers and final assemblies are fixed should be consistent with this. Our model seems to ignore these *kinetics*. The justification for this rests again on the careful definition of our probability space as explained at the beginning of this section, which is tailored to our focused questions. Hence the rates and concentrations need not enter these computations, beyond being directly incorporated into the effort ratings for subassemblies. However, viceversa, a rough estimate of rates and concentrations can be obtained simply from the average effort ratings of pathways that lead to specific subassemblies.

5 Novelty and Comparisons

The first novel aspect and the basis of the proposed approach is the following observation: by narrowly focusing only on the required output information - i.e. *successful* pathway likelihoods - we create a starting point for thinking about assembly while avoiding dynamics. Furthermore, the avoidance of dynamics permits more sophisticated - but essential - monomer structure and interaction information to be taken into consideration, while still remaining computationally tractable. Additionally, the restriction of effort to successful assemblies alone makes the assembly model significantly easier to develop, but at the same time is generalizable to a more extensive model of viral assembly.

The second new aspect of this approach to self-assembly is its basis on fully 3D geometric constraints (including tensegrity) and state-of-the-art concepts [7, 10, 11, 9] [44, 42, 43, 100, 102, 68, 32], and software [100] for decomposition and recombination of geometric constraint systems; as well as concepts and techniques for sparse elimination [27, 38]. These together permit a tractable and accurate simulation and visualization (see Section 6) of pathways. It should be noted that the use of so-called distance geometry is fairly established as one phase of (NMR) molecular structure determination [3], [65], [72], [76]. These however use very special types of geometric constraint systems – a complete set of distance constraints (usually a linear algebra based completion of a partial distance matrix) which gives the work a different character.

One key advantage of thus representing viral (sub)structures or any other geometric composites using geometric constraints is that their various conformations can be naturally obtained as the different solutions to the associated algebraic systems, described in Section 3. See Figure 13. It should be noted that conformational switching is a key ingredient of viral assembly. Many existing dynamic viral assembly models find it difficult to deal with this, since they do not have an inherent way of representing all possible conformations of subassemblies.

5.1 Comparisons with existing virus assembly models

We omit comparisons with self-assembly models for DNA computing, for example, [23, 2] which deal with dynamics issues discussed in Section 4.5. The geometry content crucial to our models is not relevant to theirs.

We only give initial comparisons of our virus assembly model with the closest existing computational virus model [56] based on the “local-rules” theory of [52, 54, 55, 53]. While we draw upon some of their ideas, there are several fundamental priniciples that differentiate our model from theirs. In addition, arguments that differentiate both our models from other relevant models [25, 110, 41, 29, 28] are given in [56]. First the superficial similarities. Since their model is not static, we draw the comparison with the generalization of our static model to a local-rules type of dynamical system, which can however be simply formulated as a system of stochastic differential equations that can be numerically solved to obtain the kinetics. This resembles their “local-rules” viral model. Both are overall consistent with the theme (not new, see e.g., [86, 85]) that the distributed protein molecules self-organize during virus assembly and crystallization, by behaving in an individually “selfish” manner, in response to local circumstances, without any programmed or prior knowledge of the global composite that they will assemble into.

Another point of similarity is that both our models use (probability spaces based on) relatively few protein subunits as a function of the number that constitute a virus. This contrasts both of our models from those such as [110].

The fundamental difference stems from the [56] model relying crucially on the following. (i) Full-blown dynamic simulation (their approach has no static analogy for analyzing successful pathways alone, which is the thrust of of this paper). (ii) Simple polygonal subcomposites consisting of only a handful of geometric primitives and an explicitly specified set of their rigid conformations. (iii) Simplified geometric interactions explicitly and procedurally specified as local rules. These provided just the necessary level of detail to answer the kinds of questions about (nucleation limited) viral shell assembly that they were interested in.

In contrast, we begin with the thesis that complex (time, phase or any continuous parameter) dynamics are not necessary for our precise and focused questions about viral assembly, [66], even those assemblies that involve “acrobatic” interactions that could be forged even as the constituent protein monomers complete their folding. We maintain that our “no dynamics” model is consistent with these suggestions.

In particular, due to our geometry and tensegrity based algebraic representation that captures conformations naturally, we have a relatively simply described configuration space. While key factors such as current conformation geometry of the composite, and other factors such as current position interference from neighboring molecules, solvent interaction etc. are strongly factored into the definition of the effort rating of subassemblies and pathways, these factors have static definitions.

In a rough sense, our “no dynamics” model adequately captures the complexity of the dynamical simulations by other viral models such as [56] because of the carefully defined probability space, the focused nature of the questions being answered, and the appropriate increase in the (algebraic) complexity of: the representation of individual monomers, interactions, and the algebraic functions that determine the effort ratings of subassemblies and pathways.

6 Tractable and Accurate Computational Simulation of the Model

It is not viable to give a combinatorial enumeration of pathways of viral constraint graphs – even those that satisfy the restrictions of Section 4.4 – using generating functions or in any manner give a closed form analysis of the probability distribution over pathways of bounded total effort. Also, despite the overall icosahedral symmetry, the number of possible pathway isomorphism classes is prohibitively large, and hence an exhaustive enumeration and over the pathways is not tractable.

We use the following approach. We
(1) use the fact that pathways for viral constraint graphs satisfying the restrictions of Section 4.4 are a modified DR-plan (see Section 3);
(2) modify and randomize an existing DR-planner called the *Frontier vertex* algorithm (FA) that runs in time cubic in the number of vertices of the input constraint graph, so that it generates a random valid pathway that contains a given subassembly or sub-pathway, and computes the pathway's rating efficiently;
(3) use the fact that viral constraint systems inherit the icosahedral symmetries :
(4) use the fact that DR-plans have a correspondence with distinct bases of the underlying rigidity matroid (see [67], [82], [17]);
(5) and the fact that matroid bases form a markov chain on which random walks converge in polynomial time to stationary distributions that permit random sampling and approximate counting [1, 15, 70, 48, 21, 40, 80, 103] and
(6) use (3), (4) and (5) to argue that the algorithm in (2) generates a representative sample of pathways that approximates their true probability distribution well.

6.1 Generating Random Pathways using the Frontier Vertex (FA) DR-Planner

DR-planners based on geometric constraint graphs have been proposed since the early 90's for restricted classes of graphs that are decomposable simply by detecting certain patterns such as triangles ("triangle decomposable") [18, 95, 96, 94] [12, 19]; and based on Maximum Matching [97, 59, 12, 90], and rigidity matroids (for 2D points and distances) [67, 82]. However, prior to [10], the DR-planning problem and appropriate (and strongly competing) desirable properties for DR-planners were not formally defined or motivated. That paper also gives a table comparing 3 main types of DR-planners, with respect to these performance measures including those in Section 3. These performance measures were optimized by the Frontier vertex DR-plans and the corresponding DR-planner (FA DR-planner) described very briefly below [11, 9, 42, 43, 44, 100, 102, 32].

6.2 The Frontier Vertex DR-plan (FA DR-plan)

Intuitively, an FA DR-plan is built by following two steps repeatedly:

1. *Isolate* a cluster-minimal dense subgraph C in the current graph G_i (which is also called the *cluster graph* or *flow graph* for reasons that will be clear below). Check for algebraic dependencies that could cause a dof misclassification.

2. *Simplify* C into $T(C)$, transforming G_i into the next cluster graph $G_{i+1} = T(G_i)$ (the recombination step).

6.2.1 Isolating Clusters

The isolation algorithm, first given in [5, 7] is a modified incremental network maximum flow algorithm. The key routine is the *distribution* of an edge (see the DR-planner pseudocode in the Appendix of Part II) in the constraint graph G. For each edge, we try to *distribute* the weight $w(e) + D + 1$ to one or both of its endpoints as *flow* without exceeding their weights, referred to as "distributing the edge e." See *DistributeEdge* in the pseudocode in Part II, Appendix. This is best illustrated on a corresponding bipartite graph G^*: vertices in one of its parts represent edges in G and vertices in the second part represent vertices in G; edges in G^* represent incidence in G. As illustrated by Figure 14, we may need to redistribute (find an augmenting path).

If we are able to distribute all edges, then the graph is not dense. If no dense subgraph exists, then the flow based algorithm will terminate in $O(n(m+n))$ steps and announce this fact. If there is a dense subgraph, then there is an edge whose weight plus $D+1$ cannot be distributed (edges are distributed in some order, for example by considering vertices in some order and distributing all edges connecting a new vertex to all the vertices considered so far). It can be shown that the search for the augmenting path while distributing this edge marks the required dense graph. It can also be shown that if the found subgraph is not overconstrained, then it is in fact minimal. If it is overconstrained, [5, 7] give an efficient algorithm to find a minimal cluster inside it.

The found cluster is then checked (and corrected) for possible dof misclassification due to the presence of algebraic overconstraints described in Section 3 (especially in 3D or in the presence of angle constraints), or *geometric* overconstraints such as rotational symmetry.

6.2.2 Cluster Simplification

This simplification was given in [11, 9]. The found cluster C interacts with the rest of the constraint graph through its *frontier vertices*; i.e., the vertices of the cluster that are adjacent to vertices not in the cluster. The vertices of C that are not frontier, called the *internal vertices*, are contracted into a single *core* vertex. This core is connected to each frontier vertex v of the simplified cluster $T(C)$ by an edge whose weight is the the sum of the weights of the original edges connecting internal vertices to v. Here, the weights of the frontier vertices and of the edges connecting them remain unchanged. The weight of the core vertex is chosen so that the density of the simplified cluster is $-D$, where D is the geometry-dependent constant. This is important for proving many properties of the FA DR-plan: even if C is overconstrained, $T(C)$'s overall weight is that of a wellconstrained graph, (unless C is rotationally symmetric - in this case, it is simplified to a cluster of weight $D-1$). Technically, $T(C)$ may not be wellconstrained in the precise sense: it may contain an overconstrained subgraph consisting only of frontier vertices and edges, but its overall dof count is that of a wellconstrained graph.

Figure 15 illustrates this iterative simplification process ending in the final DR-plan of Figure 8.4.

The challenge met by the FA DR-planner is that it provably meets several competing requirements. See Section 3 for definitions.
(a) It deals with key problems of algebraic overconstraints and rotational symmetries and hence rectifies misclassification of clusters for a large class of 2D constraint graphs containing angle and incidence constraints as well as 3D constraint graphs.
(b) For wellconstrained graphs it outputs a DR-plan with a single root representing the entire graph as a cluster: for underconstrained graphs - it outputs a complete set of maximal clusters as sources of the DR-plan.
(c) It controls the width of the DR-plan to ensure a polynomial time algorithm.
(d) It ensures the cluster-minimality of clusters in the DR-plan.
(e) It outputs DR-plans consistent with input partial decompositions.

The graph transformation performed by the FA cluster simplification is described formally in [11, 9] that provide the vocabulary for proving the properties of FA that follow directly from this simplification. However, other properties of FA require details of the actual DR-planner that ensures them [42, 43, 44, 100, 102, 68].

Note. A detailed pseudocode of the FA DR-planner can be found in [42, 100]. Properties (a) and (e) above are proven in [102, 68] and Properties (b), (c), (d) are proven in [42].

6.3 Crucial Modifications for obtaining Random Pathways

The randomization procedure is straightforward and yet performs well in providing a representative sample of pathways (see discussion on simulation accuracy in Section 6.3.1). The exact randomization cannot be formally described without refering in detail to the FA pseudocode in [42, 100]. However, it is conceptually simple and easy to describe intuitively: everywhere that the FA DR-planner performs operations on vertices, edges or clusters in *lexicographic* order or *queue* order, this is replaced by a *random* order.

Recall from Section 4, that the additional requirements that a pathway has to meet 3 properties beyond the DR-plan properties.

Property (i) is ensured by modifying the FA DR-planner as follows.

FA achieves a linear bound on DR-plan width by maintaining the following invariant of the cluster or flow graph: *every pair of clusters in the flow graph (top level of the DR-plan) at any stage intersect on at most a rotationally symmetric subgraph.* FA does this by performing 2 operations after a new potential cluster-minimal cluster is isolated by the cluster-minimality routine (see [42, 102]). Note that this routine is required even though we do not require cluster-minimality for pathways. It is required to ensure that the found cluster has not been misclassified due to combinatorial and algebraic overconstraints, i.e., to ensure Property (a) of the FA DR-planner (see [42, 102, 68]).

The first operation is an *enlargement* of the found cluster. In general, a new found cluster N is enlarged by any cluster D_1 currently in the flow graph (top level of the DR-plan), if their nonempty intersection is *not* a rotationally symmetric or trivial subgraph. In this case, N does not enter the top level of the DR-plan Only $N \cup D_1$ enters the DR-plan, as a parent of both D_1 and the other children of N. It is easy to see that the *sizes* of the subsystems corresponding to both $N \cup D_1$ and N are the same, since D_1 would already be solved.

For the example in Figure 16, when the DR-plan finds the cluster C_2 after C_1, the DR-planner will find that C_1 can be enlarged by C_2 The DR-planner forms a new cluster C_4 based on C_1 and C_2 and puts C_4 into the top level of the DR-plan, instead of C_2.

The second operation is to iteratively *combine* $N \cup D_1$ with any clusters $D_2, D_3, \ldots$ based on a nonempty overlap that is not rotationally symmetric or trivial. In this case, $N \cup D_1 \cup D_2$, $N \cup D_1 \cup D_2 \cup D_3$ etc. enter the DR-plan as a staircase, or chain, but only the single cluster $N \cup D_1 \cup D_2 \cup D_3 \cup \ldots$.. enters the top level of the DR-plan.

Ofcourse, both of these processes are distinct from the original flow distribution process that *locates* clusters.

Now, in order to ensure that the DR-plan has the Property (i) of pathways mentioned above, the enlargement process subsumes the combining process. The cluster N is iteratively enlarged by the clusters D_i so that only $N \cup D_1 \cup D_2 \ldots$ enters both the DR-plan as a common parent of N and the D_i's; and it enters the cluster graph, after removing all D_i.

Property (ii) is ensured by checking each newly found cluster N to see if it can be formed entirely by overlap constraints between the children. This requires a careful dof count using inclusion-exclusion. If overlap constraints are sufficient to form the cluster N, then the last found child cluster C of N is removed from the DR-plan and its children are directly made children of N (leaving out any children that are trivial clusters in the intersection of C and one of the other original children of N). See Figure 16.

To ensure Property (iii) or overlap-minimality of clusters, they are treated as follows after the combining step has found the cluster C that can no longer be combined with any of the clusters in the flow graph, i.e., top level clusters of the current DR-plan. Recall that overlap-minimality of C with children $C_1, \ldots, C_m$ requires that for any cluster C' formed by a proper subset of child clusters $C_1, \ldots, C_k$, it holds that: C' along with the other child clusters $C_{k+1}, \ldots, C_m$ form the cluster C using overlap constraints alone. If C does not satisfy overlap-minimality, then it is decomposed into a sub-DR-plan consisting of overlap-minimal clusters. This is done by a recursive method *Overlapmin* similar to the cluster-minimality routine of [42]. This method takes as arguments the subgraph S of the cluster graph, (S does not have to be a cluster) and the set $S = \{C_1, \ldots, C_m\}$) of the overlap-minimal clusters that constitute it (we identify the set of clusters with the subgraph of the cluster graph induced by them). The base case is when S consists of only 2 clusters C_1 and C_2. In this case, if C_1 and C_2 have a nontrivial overlap, then S is returned, since it is itself overlap-minimal. If not, C_1 and C_2 are returned as the maximal overlap-minimal clusters within S. The recursion is acheived by first locating a proper subgraph S' of S that overlaps the remaining set of clusters in $S \setminus S'$ only on a trivial subgraph. This is done by running a minimum vertex separator (or cut) algorithm such as [84] on the *bipartite* graph obtained from the standard *overlap hypergraph* of the clusters in S. In the hypergraph, clusters are vertices and each overlap vertex is an edge; in the bipartite graph, one side consists of vertices representing vertices of the hypergraph and the other consists of vertices representing edges of the hypergraph. See Figure 17.

If no such subset is located, then C is overlap minimal. If such a subset S' is located, then *Overlapmin* calls itself twice. The first call is with arguments: the subgraph induced by S' and the set of clusters $\{C_1, \ldots, C_k\}$ within S'. The method generates a sub-DR-plan of S' whose leaves are the clusters $C_1, \ldots, C_k$ and in which each intermediate cluster is overlap-minimal. If S' is itself a cluster, then it is the single root or source of this sub-DR-plan. If not, a complete set T of *maximal* clusters within S' appear as the sources. The second call is with arguments: S and the union of the set T and the remaining clusters $\{C_{k+1} \ldots, C_m\}$ in $S \setminus S'$.

It can be formally proven that the above modifications ensure the additional Properties (i), (ii) and (iii) of pathways and do not affect the Properties (a), (b), (c), (e) guaranteed by the FA DR-planner ([42] and [102, 68]). In particular, the Property (e) can be leveraged by adding a particular (labeled) subassembly or even a particular (labeled) sub-pathway as an input partial decomposition, so that the output pathway will contain them. Moreover, the pathway's effort can be computed in time directly proportional to the number of nodes in the pathway, since by the assumption on viral constraint graphs in Section 4, the number of children of a overlap-minimal cluster of a viral constraint graph is bounded by a constant and hence the effort rating of such a cluster can be computed in constant time, independent of the size of the viral constraint graph (number of atomic markers). Finally, since the FA-DR-planner's property I.e., we obtain the following theorem.

Theorem 6.1 *On an input viral constraint graph G and a labeled subassembly or sub-pathway P the FA-DR-planner - as modified above - outputs a valid assembly pathway A for G that contains P, as well as A's effort rating, within time cubic in the number of vertices (or essential atomic markers) of G.*

Note The FA DR-planner is incorporated into the (opensource, available from GNU) FRONTIER geometric constraint software [100] developed at the university of Florida. This provides a hands-on efficient computational simulation and visualization tool for viral pathway simulation as seen in the screenshot figures in Sections 4 and 7.

6.3.1 Proving Accuracy of Simulation

The above theorem proves the efficiency or tractability of the simulation. However, it is still necessary to show that the randomization procedure mentioned above outputs a truly random pathway; i.e., it outputs a representative sample of pathways (under a given effort bound) that accurately reflects the probability distribution over all successful assembly pathways (under the same effort bound) predicted by the formal model in Section 4. For example, it is necessary to show that bounded effort pathways obtained over m trials of the simulation accurately (with error decreasing, say, proportionally to $m^{1/k}$ for some k) represent the probability distribution over the entire set of successful, bounded effort pathways. I.e., we need to prove the following type of theorem.

Definition. Let M be a multiset of successful pathways of effort at most b that contain a particular (labeled) subassembly or subpathway P for some viral constraint graph G. Let $0 < \epsilon < 1$ be a desired accuracy. Let m_T be the number of pathways in M that are in the same isomorphism class T. Let p_T be the probability that a b-effort pathway of G that contains P belongs to the isomorphism class T. We say that M is an *ϵ-representative* sample (of b-effort pathways for G containing P) if for every pathway isomorphism class T, $|m_T/|M| - p_T| \leq \epsilon$. ♣ (end of definition).

Theorem 6.2 *There is a fixed k such that for any $0 < \epsilon < 1$ the following holds. Let M be any set of b-effort pathways containing a (labeled) subpathway P output over independent runs by the the randomized, modified FA-DR-planner (described above) on an input viral constraint graph G. Then M is an ϵ-representative sample provided $M \geq (1/\epsilon)^k$. This holds for effort bounds b that include a reasonably large number of pathways and for subpathways P of constant bounded size which includes the case when P is a single subassembly.*

In the general case of viral constraint graph G of Section 4.4, this theorem is still a conjecture. However, the observed accuracy of the sample obtained by the above randomization is supported by the fact that the theorem has been proven in [103] for a special class of of viral constraint graph G in which each pathway corresponds to a distinct, well-defined basis class of the rigidity matroid that is obtained from G (see [67], [82], [17]). The proof further leverages (1) the symmetry of the complete icosahedral shell which intuitively makes counting easier and (2) the fact that distinct matroid bases form a markov chain on which random walks converge rapidly (in time polynomial in the size of the original graph and in $1/\epsilon$) to stationary distributions that provide ϵ-representative samples. Various results of this type are shown in [48], [80], [40] for so-called balanced matroids. The proof requires the extension of these results to (1) a class of rigidity matroids; (2) where each element in the markov chain is a pathway, i.e, a distinct well-defined basis class of the underlying rigidity matroid, as opposed to a distinct basis. Finally, the proof shows that the simple randomization of the FA-DR-planner described above corresponds in a natural manner to a random walk on the above markov chain.

7 Preliminary Validation of the Model

For viral constraint systems obeying the restrictions of Section 4.4 (no inequality constraints and hence no chirality constraints, and no tensegrity forces) a simulation and visualization study was performed using the FRONTIER geometric constraint solver and interactive visualization software [100] with the randomization and modifications described in Section 6. The monomer constraint system that was used for the simulations contained 12-15 atomic markers (see Figure 6) related by primary structure constraints that specified distances and angles and weak force constraints specifying distances. The interface constraints were purely distance constraints. These constraints were modified over the different runs in order to simulate various aspects of the model. For example, the constraints were engineered so that: monomers were underconstrained, wellconstrained, or overconstrained; the pathway contained small clusters that included parts of several monomers as in Figure 9, or in contrast, all overlap-minimal clusters were forced to include entire subassemblies (i.e., all overlap-minimal clusters represent stable subassemblies). Similarly, the constraints were engineered so that only a desired combination of subassemblies were stabilizable and others were not. For example pentamer (resp. trimer assemblies) Figure 18 are stabilizable but dimers are not, etc. A complete assembly obtained during the simulation is shown in Figure 17. All the pathways shown in Figure 10 and Figures 11, 12 were obtained during the simulation. (The visualization software is best used interactively, with clicking and zooming, especially for pathways with more than 10 vertices, we have therefore used hand-drawn figures instead of screenshots to illustrate the pathways).

Next, we indicate some basic questions that can be asked and predictions that can be made by the simulation results.

Prediction 1. We first asked the question: *Assuming that the smallest stabilizable (polymeric) subassemblies are pentamers or trimers, which (successful) stable-subassembly pathways are most likely: (a) those that are based on a pentamer nucleations that assemble further via trimeric (and dimeric) interactions first forming a trimer of pentamers and then adding single pentamers, such as in Figures 11, 12, or (b) those that are based on trimer nucleations followed by pentameric interactions first forming a pentamer of trimers and then adding on more trimers such as in Figures 9, 10?*

We asked the same question using standard cluster pathways which are easier to analyze than stable-subassembly pathways, by engineering the monomer and interface constraints in such a way that all overlap-minimal clusters are in fact stable subassemblies. The overall qualitative trend - of the pathway probabilities and effort ratings obtained by the running several trials of the simulation algorithm - can be described as follows. We use rough calculations that are directly based on the model definition in Section 4. The model (and simulation) predict - given an effort bound - that pentamer nucleation based pathways are overall more likely, since both the number and size of the isomorphism classes of such pathways is larger. A rough calculation also shows that the former pathways of type (a) in fact involve less effort than the latter type (b), on average. Pentamer based pathways of type (a) as in Figures 11 , 12, use only 12 pentamer clusters followed by one cluster of fanin 3 and nine subsequent clusters of fanin 2. In contrast, trimer based pathways of type (b) as in Figure 10, use 20 trimer clusters as in Figure 9 followed by one cluster of fanin 5, one cluster of fanin 4, four clusters of fanin 3 and 4 clusters of fanin 2. For simplicity, we assume rigid monomers, (i.e. monomers are clusters), and that the number of monomer conformations (size of the solution space of the monomer constraint system), while nontrivial (our example monomers typically have about 8 conformations caused by 3 independent reflections), is however significantly smaller than the number of pentamer and trimer conformations (size of pentamer and trimer solution spaces). We assume the latter two to be equal (typically about 24 in our examples).

Using these natural assumptions, the effort rating calculation (assuming all weights w_i are 1) shows that trimer based pathways of type (b) involve effort that is atleast a factor of 24 (number of trimer conformations) larger than the pentamer based pathways of type (a).

Note that the equality between the number of trimer and pentamer conformations is justified since, due to symmetry, pentamers tend to become overconstrained as soon as they are forced to be wellconstrained and hence have relatively fewer conformations (if monomers are clusters with 6 degrees of freedom, we require the pentamer interfaces to remove 5 degrees of freedom each inorder to obtain the pentamer as a 1-stable subassembly, i.e., as a cluster; however, now the pentamer has only 5 degrees of freedom, i.e., it is overconstrained).

The above analysis also answers the question: *Under what natural conditions would trimer based pathways of type (b) be more likely?* Clearly, if trimers are forced to have small numbers of conformations (typically enforced by overconstraints) compared to pentamers, then the above trend in effort calculations can be reversed, making type (b) pathways have significantly lower effort and hence more likely, but they would have to be severely overconstrained to offset the small size of their isomorphism classes. In fact, in this situation, by the above arguments, the most likely pathways would still be those that would use a combination of trimer and pentamer nucleations.

These predictions seem to fit structural studies of common T=1 viruses: those viruses such as MVM that exhibit highly overconstrained trimers (highly interdigitated trimer interfaces with large proportion of buried surfaces, which can be obtained using the CNS software [72]) are generally believed to use trimer nucleation based pathways, while others with less interdigitated trimers such as MSV are believed to use pentamer nucleation based pathways.

In fact, 2 even simpler arguments based on our model support these and structural studies.

Prediction 2. Assume monomers and pentamers are stable subassemblies (for simplicity, we assume 1-stable subassemblies, i.e., they are exactly clusters and hence have 6 degrees of freedom), but trimers are not. Assume as well that dimer interfaces remove 4 degrees of freedom *dofs*. Then a triple of pentamers as in Figure 11 interacting via 3 dimer interface constraints alone (no trimer interface constraints required) has (6*3 - 4*3 =) 6 dofs, making it a cluster as well. Thus we can achieve a pentamer based pathway as in Figures 11 and 12 using pentamer and dimer constraints alone. In fact, clusters further up the pathway are even *overconstrained* (again without using trimer interface constraints; just using dimer interface constraints alone) and hence their effort rating is significantly lower as discussed in Section 4, increasing the likelihood of such pathways.

In contrast, if monomers and trimers were stable subassemblies (clusters with 6 dofs) and pentamers were not, and if dimer interfaces again removed 4 dofs, and pentamer interface constraints were analogously not relied upon, the quintuple of trimers as in Figure 9 has (6*5 - 4*5 =) 10 dofs and is hence underconstrained, not a cluster or a stable subassembly. In fact, the smallest stable assembly (beyond the trimers) would involve the entire shell (each of the 20 trimers has 6 dofs and each of the 30 dimers removes 4 dofs)! This pathway's isomorphism class has size 1 (there is a single labeled pathway of this type) and it has a very large effort (the top cluster of the pathway has a fanin of 20!) and is hence very unlikely.

Prediction 3. Even if pentamers are *not* 1-stabilizable subassemblies - assume they have 7 dofs - a triple of pentamers is still fully stabilizable using 5-dof-removing dimer interfaces (7*3 - 5*3 =6; no trimer interfaces required), and thus a pathway based on triples of pentamers such as in Figures 11 and 12 would be valid when the bottom two levels are merged to form one cluster of fanin 15 and others of fanin 6 or less. While it has relatively high effort, the isomorphism class of such pathways is still fairly large and hence the likelihood of such pathways is nontrivial.

In contrast, trimers with 7 dofs and 5-dof-removing dimer interfaces again permit only the single highly unlikely trivial pathway of fanin 20, since the entire shell is again the minimal stable assembly. In comparison, highly underconstrained pentamers with even 10 dofs and only 4-dof-removing dimer interfaces or pentamers with 7 dofs and very loose 3-dof-removing dimer interfaces (no trimer interfaces required in either case) will still result in a stable entire shell (10*12 = 4*30; 7*12 ¡ 3*30).

These predictions using our model indicate that for viral constraint systems to significantly favor trimer based pathways such as Figures 9, 10 as opposed to pentamer based pathways such as Figures 11 and 12 a set of competing, stringent conditions have to be met.
(a) Such pathways typically have to depend on pentamer interface constraints as well (not just dimer interfaces, as shown above).
(b) At the same time, these pentamer interface constraints should nevertheless be inadequate (should not remove too many dofs) to stabilize pentamers, in order to hinder pentamer based pathways.
(c) To keep effort ratings manageable, the trimer interfaces should remove enough dofs (be sufficiently interdigitated) to make the trimers highly overconstrained.

Prediction 4. Another prediction by our model supports structural studies of the geminate "two headed" MSV virus. See Figure 2 (right). This virus is made up of 2 single heads, each missing a pentamer where they join. Thus each head consists of 11 pentamers. The viral monomers sometimes form single heads [51] but not 3 or multiple heads. These are indicated by our model as follows. Using the example in (3) above, pentamers with 7 dofs along with 3-dof-removing dimer interfaces achieve 1-stability with exactly 11 pentamers that use the corresponding 25 dimer interfaces (7*11 - 3*25 ¡ 6). The 5 *dimer-like* interfaces - that (require a conformational switch to) join the 2 heads - need only remove a total of

6 dofs to stabilize the entire geminate shell. However, to form a 3 headed “peanut” is significantly more difficult, since the central head will consist of 10 pentamers and only 20 dimer constraints, causing it to be highly underconstrained (10 dofs) and it will have to be stabilized by its 5 dimer-like interfaces with each of the 2 neighboring heads. This will permit a single unlikely pathway (isomorphism class of size 1) with two clusters of fanin 11 joining with 10 monomers to give the entire shell as a cluster of fanin 12. The pathway would have low probability additionally due to high effort caused by the lack of overconstraints and due to the large solution spaces of the clusters involved.

8 Conclusive Validation of the Model

The model will be tested using real input data from MVM, MSV, AAV and other viruses. The simulation predictions will be verified using carefully directed and designed experiments. For example, if the simulation predicts high likelihood of pentamer and dimer based assembly pathways for MSV, then site directed mutagenesis to hinder either the formation or continuation of pentamers could be used to verify the prediction. A brief description of these and other experiments along with justifications of biochemical assumptions can be found below.

8.1 About the choice of viruses

The Murine Parvovirus minute virus of mice (MVM) and the Maize Streak Virus (MSV) were chosen carefully so that specific aspects of the model can be experimentally tested.

Some of the reasons for their choice are the following. Both are T=1 viruses, i.e, icosahedral viruses containing the smallest possible number of monomers and requiring highly economical and precise assembly. MSV is a geminivirus, dumbell shaped, with 2 icosahedral heads, apparently requiring a key conformational switch for the 2 heads to join. See Figure 2. While both viruses have a T=1 capsid architecture, MVM represents the prototypical icosahedral capsid made up of 60 monomeric subunits [14], while MSV is atypical in that capsids contain 110 monomers, with a geminate arrangement of two incomplete icosahedral capsids (the two

heads), each missing a pentamer at the interface [51]. They additionally appear to be limited to 2 heads, not 3 or more. The MVM capsid is able to assemble in the absence of genetic material [83], and without the involvement of scaffolding proteins which is important since our formal model does not consider the involvement of such material. ”Single” head MSV capsids were also recently isolated [51] that contain incomplete genetic material. Finally, structural studies of the geminate capsid and the single head particles clearly shows that a conformational switch of the coat protein monomer is required to enable the two heads to join. Thus MVM and MSV single heads provide an example of capsid assembly driven mainly by protein-protein interactions and geminate MSV an example in which structural switching is essential.

8.2 Designing the input parameters for 2 specific viruses

8.2.1 Designing model input

The capsid structures of MVM and MSV have been very recently mapped either by X-ray crystallography [14] or by cryo-electron microscopy, image reconstruction and homologous modeling [51]. Significantly, high resolution 3D data available for other members of the Parvoviridae such as AAV has recently shown that members with as little as 15-18% sequence identity have the same overall capsid topology as MVM, built from conserved amino acid sequences present at the N- and C-terminal ends of the primary amino acid sequence. In addition, our recent sequence analysis of geminiviruses from the four known genera also indicates conservation of capsid architecture and core residues.

Only such *conserved* interactions, based on structural observations, will be used to design the i input parameters of the proposed model. This similarity to other viruses additionally allows a fairly high confidence in isolating both the essential features of monomer structure and the driving biochemical interactions (interface constraints) between the constituent monomers.

Designing the model's input, i.e., a viral geometric constraint system (see Section 4) based on MVM, MSV, AAV or any of the planned T=1 viruses involves the choice of relatively few (20 -100) *atomic markers* (points) - which represent an average position of a group of essential residues - on the coat protein monomer backbone and inportant side chains. These are chosen either because they are crucial for specifying the monomer structure or bacause they are crucial in the interface constraints that drive assembly. These can be first found using the CONTACT subroutine in the CNS program package [76] [72] followed by selection of *average* atomic positions, based on groups of interacting atoms in the program O [34] and - most importantly - abbreviated using conservation of residue type in related viruses as discussed in the previous paragraph.

As defined in Section 4, the monomer structure primitives additionally include bonds (line segments) incident on the atomic markers, representing averaged bonds between the corresponding essential residues. The monomer structure constraints are of two types: first the primary structure constraints obtained as average values of distance, angle and torsion angle intervals, representing the bond length, angle, and torsion angle predictions from first priniciples. Second, the weak force constraints (these constitute the so-called *contact map* of the monomer): hydrogen bond lengths and other weak forces represented as distance intervals and tensegrity force constraints between the atomic markers. These are also obtained from NMR or X-ray structure data and predictions both from biochemical first principles, and mechanical significance using broad tensegrity principles. The monomer chirality constraints - representing relative orientations of atomic markers which restrict the monomer conformations - are obtained using X-ray and Cryo-EM structure data.

The 3 types of interface constraints are similarly generated by a simple averaging calculation of inter-atomic hydrogen bond and weak force interactions at buried dimer, trimer and pentamer interfaces (these constitute the so-called *interface contact maps*). These are again found using the CONTACT subroutine in the CNS program package [76] [72] followed by selection of *average* forces and distances between the chosen atomic markers using the program O [34] and biochemical as well as tensegrity principles. The third and fourth model input are the standard icosahedral neighborhood topology (see Figure in Section 4) of the monomers, and the global chirality constraints representing relative orientations of subsets of 4 atomic markers on different monomers: these could arise, for example, due to crucial external restrictions such as interactions with the viral core of genetic material and are obtained using X-ray and Cryo-EM structure data. These could be replaced by extra distance constraints as mentioned in Section 4. Together, all of these will determine the complete viral geometric constraint system that serves as the input to the model and to its computational simulation of Sections 4 and 6. A preliminary input for MVM for the restricted model of Section 4.4 has been generated by the above-described process

consisting of *completely rigid monomers* and the interface constraints have been expressed as purely distance constraints between atomic markers. The monomers are further restricted to *one* fixed conformation and hence the monomer structure constraint system is subsumed by specifying explicit positions of the atomic markers in its local coordinate system. This type of input still retains adequate complexity in the interface constraints in order to obtain meaningful predictions output by the model and its computational simulation of Sections 4 and 6, which can then be experimentally tested. See Figure 19.

8.3 Justification of Biochemical Assumptions and Experiment Design

The *first* key assumption is that the complete assembled structure together with sequence comparison with similar viruses can be used in determining the pre-assembly monomer structure and driving interactions. Our justification for this is based on the following observations.

(a) For the Parvoviridae, our 3D structures show that members with as little as 15% sequence identity are built from the same arrangement of core beta-strands which contain conserved residues at both the N- and C-terminal regions of the primary sequence. These strands knit together to form a core shell onto which surface decorations - which It is therefore reasonable to assume the conserved residues, at monomer-monomer interfaces, are playing a role in capsid assembly.

(b) Structures determined for fully assembled virus capsids have been predictive in our analysis of assembly enabling interactions for the adeno-associated parvoviruses (AAVs). We can correlate disruption of monomer-monomer interactions with mutations that result in no capsid phenotypes [49](Agbandje-McKenna and Muzyczka, unpublished data).

(c) There are numerous examples of structural studies where low resolution cryo-EM structures of viruses and their complexes with receptor or antibody molecules are interpreted based high resolution structures of component proteins. These are placed as rigid bodies inside the cryo-EM envelope. Sometimes adjustments are required to satisfy the density, but for the most part, the high resolution structures fit well without adjustments (Reviewed (for viruses) in [39] [50]). Antibodies that are generated against assembled capsids have been shown to recognize subassemblies for MVM [35] and other parvoviruses. These recognize the capsid surface not interfaces where the interactions happen.

(d) Our sequence analysis of geminiviruses from the four known genera (MSV is the type species of the Mastrevirus genus) show that the core beta-strands that form the capsid are more conserved than the variable loops that differentiates members of the different genera (Faulker et al, in preparation).

A *second* significant assumption made by our assembly model is that no involvement by extraneous genomic nucleic acids or scaffolding or chaperone proteins is present during assembly. While this is justified for MVM and probably for the single head MSV's, there is evidence that the characteristic geminate capsid of MSV is only able to form when genomic DNA is present [47];[98]. Furthermore, it is not known whether chaperones are needed in MSV assembly.

8.4 Experimental validation

The model will be refined by testing the output prediction of the simulation against experiments on MSV briefly described below. For MSV, we have identified what we believe are key subassemblies (pentameric capsomers, "single" head and geminate capsids) for MSV from wild type maize leaf infections using biochemical techniques and it seems that concentrations of subassembly formations are observable. They are able to express single mutant monomers, altered to arrest assembly.

Well-established experimental protocols will be utilized to validate the essential subassemblies predicted by our computational simulation. For example, if pentamers are theoretically predicted as having the highest concentrations during the assembly of the T=1 capsids, we are able to alter conserved monomer-monomer interactions in the 3D structures by *site-directed mutagenesis* to either arrest assembly either *prior* to pentamer formation or stop growth *beyond* assembled pentamers by mutagenesis. We would generate mutant clones to express the MSV coat protein, bearing the appropriate amino acid changes, to be produced in E.coli, as has been done previously to express an N-terminal truncated coat protein mutant [47] and also into the available MSV infectious clone for infection of maize leaves. Then we would use established methods to purify subassemblies from E.coli expression or maize leaf infection and determine their concentration. Further experimental goals are the following.

First to and refine our model inputs (the current inputs on MSV are obtained from cryo-EM and sequence mapping structure) by performing complete X-ray crystallographic structural analyses of isolated subassemblies, for comparison with our available structures of ”single” head and geminate capsids [51] (Zhang et al., unpublished data), using standard procedures [45], [46]. It should be noted that obtaining structural data on assembly naive components is never trivial (as evidence by the fact that there is no such data for any ssDNA virus) and is often precluded by a shift in the equilibrium from monomers to assembled capsid due to accumulation and subsequent nucleation. These studies are aimed at visualizing snapshots of intermediates that give rise to assembled topologies. They will enable us to analyze structural re-arrangements the drive assembly and provide refined input data for computational simulations. CD and DLS will be used to identify candidate intermediates that are correctly folded and mono-disperse for crystallization studies.

Second, to try to set up an in vitro assembly system for T=1 viruses by which more precise control of rates and concentrations would become viable.

Third, to test our model on other T=1 parvoviruses besides MVM and MSV, specifically starting with AAV2, by all of the above methods.

References

[1] D. Aldous. The random walk construction for spanning trees and uniform labeled trees. *SIAM J. of Disc. Math.*, pages 450–465, 1990.

[2] L Adleman and A Goel and Q Chenge and M D Huang and H Wasserman. Linear self assemblies: equilibria, entropy and convergence rates. *Sixth international conference on difference equations and applications*, 2001.

[3] D. J. Jacobs and A. J. Rader and L. A. Kuhn and M. F. Thorpe. Protein flexibility predictions using graph theory. *Proteins: Structure, Function, and Genetics*, 44:150–165, 2001.

[4] D Caspar and A Klug. Physical principles in the construction of regular viruses. *Cold Spring Harbor Symp Quant Biol*, 27:1–24, 1962.

[5] C. M. Hoffmann and A. Lomonosov and M. Sitharam. Finding solvable subsets of constraint graphs. In *Constraint Programming ’97 Lecture Notes in Computer Science 1330, G. Smolka Ed., Springer Verlag*, Linz, Austria, 1997.

[6] C. M. Hoffmann and A. Lomonosov and M. Sitharam. Finding solvable subsets of constraint graphs. In Smolka G., editor, *Springer LNCS 1330*, pages 463–477, 1997.

[7] C. M. Hoffmann and A. Lomonosov and M. Sitharam. Geometric constraint decomposition. In Bruderlin and Roller Ed.s, editor, *Geometric Constraint Solving*. Springer-Verlag, 1998.

[8] C. M. Hoffmann and A. Lomonosov and M. Sitharam. Geometric constraint decomposition. In Bruderlin B. and Roller D., editor, *Geometric Constr Solving and Appl*, pages 170–195, 1998.

[9] C. M. Hoffmann and A. Lomonosov and M. Sitharam. Planning geometric constraint decompositions via graph transformations. In *AGTIVE ’99 (Graph Transformations with Industrial Relevance), Springer lecture notes, LNCS 1779, eds Nagl, Schurr, Munch*, pages 309–324, 1999.

[10] C. M. Hoffmann and A. Lomonosov and M. Sitharam. Decomposition of geometric constraints systems, part i: performance measures. *Journal of Symbolic Computation*, 31(4), 2001.

[11] C. M. Hoffmann and A. Lomonosov and M. Sitharam. Decomposition of geometric constraints systems, part ii: new algorithms. *Journal of Symbolic Computation*, 31(4), 2001.

[12] R. Latham and A. Middleditch. Connectivity analysis: a tool for processing geometric constraints. *Computer Aided Design*, 28:917–928, 1996.

[13] P Ceres and A Zlotnick. Weak protein-protein interactions are sufficient to drive assembly of hepatitis b virus capsids. *Biochemistry*, 41:11525–11531, 2002.

[14] M Agbandje-McKenna and AL Llamas-Saiz and F Wang and P Tattersall and MG Rossmann. Functional implications of the structure of the murine parvovirus, minute virus of mice. *Structure*, 6:1369–1381, 1998.

[15] Yossi Azar and Andrei Z. Broder and Alan M. Frieze. On the problem of approximating the number of bases of a matroid. *Information Processing Letters*, 50(1):9–11, 1994.

[16] C. M. Hoffmann and B. Yuan. On spatial constraint solving approaches. In *Proc. ADG 2000, ETH Zurich*, page in press, 2000.

[17] Jack E. Graver and Brigitte Servatius and Herman Servatius. *Combinatorial Rigidity*. Graduate Studies in Math., AMS, 1993.

[18] I. Fudos and C. M. Hoffmann. A graph-constructive approach to solving systems of geometric constraints. *ACM Transactions on Graphics*, 16:179–216, 1997.

[19] A. Middleditch and C. Reade. A kernel for geometric features. In *ACM/SIGGRAPH Symposium on Solid Modeling Foundations and CAD/CAM Applications*. ACM press, 1997.

[20] C. M. Hoffmann and C. S. Chiang. Variable-radius circles in cluster merging, Part I: translational clusters. *CAD*, 33:in press, 2001.

[21] James Gary Propp and David Bruce Wilson. How to get a perfectly random sample from a generic markov chain and generate a random spanning tree of a directed graph. *Journal of Algorithms*, 27:170–217, 1998.

[22] Bernstein and David Naumovich. The number of roots of a system of equations. *Functional Analysis and its Applications (translated from Russian)*, 9(2):183–185, 1975.

[23] P Rothemund and E Winfree. The program size complexity of self-assembled squares. *ACM STOC*, 2000.

[24] Giovanni Pistone and Eva Riccomagno and Henry P. Wynn. *Algebraic Statistics: Computational Commutative Algebra in Statistics*. CRC Press, December 2000.

[25] V S Reddy and H A Giesing and R T Morton and A Kumar and C B Post and C L Brooks and J E Johnson. Energetics of quasiequivalence: computational analysis of protein-protein interactions in icosahedral viruses. *Biophys*, 74:546–558, 1998.

[26] W. Bouma and I. Fudos and C. M. Hoffmann and J. Cai and R. Paige. A geometric constraint solver. *CAD*, 27:487–501, 1995.

[27] John F. Canny and Ioannis Z. Emiris. A subdivision-based algorithm for the sparse resultant. *Journal of the ACM (JACM)*, pages 417–451, May 2000.

[28] J E Johnson and J A Speir. Quasi-equivalent viruses: a paradigm for protein assemblies. *J. Mol. Biol.*, 269:665–675, 1997.

[29] D Rapaport and J Johnson and J Skolnick. Supramolecular self-assembly: molecular dynamics modeling of polyhedral shell formation. *Comp Physics Comm*, 1998.

[30] D. Cox and J. Little and D. O'Shea. *Using algebraic geometry*. Springer-Verlag, 1998.

[31] C. M. Hoffmann and J. Peters. Geometric constraint for CAGD. In T. and Schumaker L. Daehlen, M. and Lyche, editor, *Mathematical Methods for Curves and Surfaces*, pages 237–254, 1995.

[32] M Sitharam and J Peters and Y Zhou. Solving minimal, wellconstrained, 3d geometric constraint systems: combinatorial optimization of algebraic complexity. *Automated deduction in Geometry (ADG) 2004, available upon request*, 2004.

[33] A. Sommese and J. Verschelde. Numerical homotopies to compute generic points on positive dimensional algebraic sets. *Journal of Complexity*, 16(3):572–602, 1999.

[34] T Jones and J Y Zou and S W Cowan and M Kjeldgaard. Improved methods for building protein models in electron density maps and the location of errors in these models. *Acta Crystallographica*, A97:110–119, 1991.

[35] E Lombardo and JC Ramrez and M Agbandje-McKenna and JM Almendral. A β-stranded motif drives capsid protein oligomers of the parvovirus minute virus of mice into nucleus for viral assembly. *Journal of Virology*, 74:3804–3814, 2000.

[36] FHC Crick and JD Watson. Structure of small viruses. *Nature*, 177:473–475, 1956.

[37] A Zlotnick and JM Johnson and PW Wingfield and SJ Stahl and D Endres. A theoretical model successfully identifies features of hepatitis b virus capsid assembly. *Biochemistry*, 38:14644–14652, 1999.

[38] Ioannis Emiris and John Canny. A practical method for the sparse resultant. In *International Conference on Symbolic and Algebraic Computation, Proceedings of the 1993 international symposium on Symbolic and algebraic computation*, pages 183–192, 1993.

[39] Baker T. S and Johnson J. E. Low resolution meets high: towards a resolution continuum from cells to atoms. *Current Opinions in Structural Biology*, 6:585–594, 1996.

[40] Ravi Montenegro and Jung-Bae Son. Edge isoperimetry and rapid mixing on matroids and geometric markov chains. In *ACM Symposium on Theory of Computing*, pages 704–711, 2001.

[41] C J Marzec and L A Day. Pattern formation in icosahedral virus capsids: the papova viruses and nudaurelia capensis β virus. *Biophys*, 65:2559–2577, 1993.

[42] A. Lomonosov and M. Sitharam. Graph algorithms for geometric constraint solving. In *submitted*, 2004.

[43] J. J. Oung and M. Sitharam and B. Moro and A. Arbree. Frontier: fully enabling geometric constraints for feature based design and assembly. In *abstract in Proceedings of the ACM Solid Modeling conference*, 2001.

[44] C Hoffman and M Sitharam and B Yuan. Making constraint solvers more useable: the overconstraint problem. *to appear in CAD*, 2004.

[45] Bubb M. R. L. Govindasamy L. Yarmola E. G.Vorobiev S. M. Almo S. C. Somasundaram T. Chapman M. S. Agbandje-McKenna. M. and McKenna R. Polylysine induces an antiparallel actin dimer that nucleates filament assembly. *Journal of Biological Chemistry*, 277:20999–21006, 2002.

[46] Wu D. Govindasamy L. Lian W. Gu Y. Kukar T. Agbandje-McKenna M. and McKenna R. Structure of human carnitine acetyltransferase:molecular basis for fatty acyl transfer. *Journal of Biological Chemistry*, 278:13159–13165, 2003.

[47] H Liu and MI Boulton and JW Davies. Maize streak virus coat protein binds single and double stranded dna in vitro. *Journal of General Virology*, 78:1265–1270, 1997.

[48] Tomas Feder and Milena Mihail. Balanced matroids. In *Annual ACM Symposium on Theory of Computing, Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 26–38, 1992.

[49] Wu P. Xiao W. Conlon T. Hughes J. Agbandje-McKenna M. Ferkol T. Flotte T. and Muzyczka N. Mutational analysis of the adeno-associated virus type 2 (aav2) capsid gene and construction of aav2 vectors with altered tropism. *Journal of Virology*, 74:8635–8647, 2000.

[50] TS Baker and NH Olson and SD Fuller. Adding the third dimension to virus life cycles: Three-dimensional reconstruction of icosahedral viruses from cryo-electron micrographs. *Microbiology Molecular Biology Reviews*, 63:862–922, 1999.

[51] W Zhang and NH Olson and TS Baker and L Faulkner and M Agbandje-McKenna and MI Boulton and RH Davies and R McKenna. Structure of the maize streak virus geminate particle. *Virology*, 279:471–477, 2001.

[52] B. Berger and P. Shor and J. King and D. Muir and R. Schwartz and L. Tucker-Kellogg. Local rule-based theory of virus shell assembly. *Proc. Natl. Acad. Sci. USA*, 91:7732–7736, 1994.

[53] R Schwartz and PE Prevelige and B Berger. Local rules modeling of nucleation-limited virus capsid assembly. *Technical report, MIT-LCS-TM-584*, 1998.

[54] B Berger and PW Shor. On the mathematics of virus shell assembly. 1994.

[55] B Berger and PW Shor. Local rules switching mechanism for viral shell geometry. *Technical report, MIT-LCS-TM-527*, 1995.

[56] R Schwartz and PW Shor and PE Prevelige and B Berger. Local rules simulation of the kinetics of virus capsid self-assembly. *Biophysical journal*, 75:2626–2636, 1998.

[57] N. Sridhar and R. Agrawal and G. L. Kinzel. An active occurrence-matrix-based approach to design decomposition. *CAD*, 25:500–512, 1993.

[58] A Zlotnick and R Aldrich and J M Johnson and P Ceres and M J Young. Mechanisms of capsid assembly for an icosahedral plant virus. *Virology*, 277:450–456, 2000.

[59] S. Ait-Aoudia and R. Jegou and D. Michelucci. Reduction of constraint systems. In *Compugraphics*, pages 83–92, 1993.

[60] C. M. Hoffmann and R. Joan-arinyo. Symbolic constraints in geometric constraint solving. *J. for Symbolic Computation*, 23:287–300, 1997.

[61] M Agbandje and R McKenna and MG Rossmann and ML Strassheim and PR Parrish. Structure determination of feline panleukopenia virus empty particles. *Proteins*, 16:155–171, 1993.

[62] C. M. Hoffmann and R. Vermeer. Geometric constraint solving in R^2 and R^3. In Du D. Z. and Hwang F., editor, *Computing in Euclidean Geometry*, pages 266–298, 1995.

[63] X. S. Gao and S. C. Chou. Solving geometric constraint systems. I. a global propagation approach. *CAD*, 30:47–54, 1998.

[64] X. S. Gao and S. C. Chou. Solving geometric constraint systems. II. a symbolic approach and decision of rc-constructibility. *CAD*, 30:115–122, 1998.

[65] G. Crippen and T. Havel. *Distance Geometry and Molecular Conformation*. John Wiley & Sons, 1988.

[66] J E Johnson and W R Wikoff. Macromolecular assembly: Chainmail stabilization of a viral capsid. *Current Biology*, 8:R914–R917, 1998.

[67] T. Tay and W. Whiteley. Generating isostatic frameworks. *Topologie Structurale*, 11:21–69, 1985.

[68] M Sitharam and Y Zhou. A tractable, approximate, combinatorial 3d rigidity characterization. *Automated Deduction in Geometry (ADG) 2004*, 2004.

[69] W Whiteley B Roth. Tensegrity frameworks. *Transactions of the AMS*, 265:419–446, 1981.

[70] A.A. Broder. Generating random spanning trees. In *FOCS*, pages 442–447, 1989.

[71] B. Bruderlin. Constructing three-dimensional geometric object defined by constraints. In *ACM SIGGRAPH*. Chapel Hill, 1986.

[72] CNS. http://cns.csb.yale.edu/v1.0/. In *NMR structure software*.

[73] Robert Connelly. Tensegrity structures: why are they stable? *Rigidity Theory and Applications*, pages 47–54, 1998.

[74] BG DeVarco. *Invisible architecture: Nanoworld of Buckminster Fuller*. John Wiley & Sons, 1988.

[75] A Edmondson. *A Fuller Explanation: The Synergetic Geometry of R. Buckminster Fuller*. Birkhauser Verlag, 1987.

[76] A Bringer et al. Crystallography and nmr systems: a new software suite for macromolecular structure determination. *Acta Crystallographica*, D54:905–921, 1998.

[77] I. Fudos. *Constraint solveing for computer aided design*. Ph.D. thesis, Dept. of Computer Sciences, Purdue University, August 1995.

[78] I. Fudos. *Geometric Constraint Solving*. PhD thesis, Purdue University, Dept of Computer Science, 1995.

[79] R. Fuller. *Synergetics: Explorations in the Geometry of Thinking*. MacMillan Publishing Co., Inc., New York, 1975.

[80] Anna Gambin. On approximating the number of bases of exchange preserving matroids. In *Mathematical Foundations of Computer Science*, pages 332–342, 1999.

[81] D Brenner S Lyshefski G Iafrate WA Goddard. *Handbook of nanoscience engineering and technology*. CRC press, 2002.

[82] B. Hendrickson. Conditions for unique graph realizations. *SIAM J. Comput.*, 21:65–84, 1992.

[83] I. Agbandje-McKenna M. and Almendral J. M. Hernando E. Llamas-Saiz A. L. Foces-Foces C. McKenna R. Portman. Biochemical and physical characterization of parvovirus minute virus of mice virus-like particles. *Virology*, 267:299–309, 2000.

[84] David R. Karger. Minimum cuts in near-linear time. *Journal of the ACM (JACM)*, 47(1):46–76, 2000.

[85] S A Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.

[86] S A Kauffman. *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity*. Oxford University Press, 1995.

[87] H. Kenner. *Geodesic Math and How to Use It*. University of California Press, Berkeley and Los Angeles, CA, 1976.

[88] A. Khovanskii. Fewnomials. *Trans. of Math. Monographs*, 88, 1991.

[89] B.F. Knight. *Deployable Antenna Kinematics using Tensegrity Structure Design*. PhD thesis, University of Florida, Gainesville, FL, 2000.

[90] G. Kramer. *Solving Geometric Constraint Systems*. MIT Press, 1992.

[91] G. Kramer. *Solving geometric constraint systems: a case study in kinematics*. MIT Press, 1992.

[92] G. Laman. On graphs and rigidity of plane skeletal structures. *J. Engrg. Math.*, 4:331–340, 1970.

[93] G. J. Nelson. A costraint-based graphics system. In *ACM SIGGRAPH*, pages 235–243, 1985.

[94] J. Owen. www.d-cubed.co.uk/. In *D-cubed commercial geometric constraint solving software*.

[95] J. Owen. Algebraic solution for geometry from dimensional constraints. In *ACM Symp. Found. of Solid Modeling*, pages 397–407, Austin, Tex, 1991.

[96] J. Owen. Constraints on simple geometry in two and three dimensions. In *Third SIAM Conference on Geometric Design*. SIAM, November 1993. To appear in Int J of Computational Geometry and Applications.

[97] J.A. Pabon. Modeling method for sorting dependencies among geometric entities. In *US States Patent 5,251,290*, Oct 1993.

[98] Citovsky V. and Gafni Y. Palanichelvam K., Kunik T. The capsid protein of tomato yellow leaf curl virus binds cooperatively to single-stranded dna. *Journal of General Virology*, 79:2829–2833, 1998.

[99] D. Roller. An approach to computer-aided parametric design. *CAD*, 23:303–324, 1991.

[100] M. Sitharam. Frontier, opensource gnu geometric constraint solver: Version 1 (2001) for general 2d systems; version 2 (2002) for 2d and some 3d systems; version 3 (2003) for general 2d and 3d systems. In *http://www.cise.ufl.edu/∼sitharam, http://www.gnu.org*, 2004.

[101] M Sitharam. A game-based decomposition of a class of real algebraic varieties. *preliminary manuscript, available upon request*, 2004.

[102] M Sitharam. Graph based geometric constraint solving: problems, progress and directions. In Dutta and Janardhan and Smid, editor, *To appear in AMS-DIMACS volume on Computer Aided Design*, 2004.

[103] M Sitharam. Markov sampling of algebraic structures related to rigidity matroid bases. *preliminary manuscript, available upon request*, 2004.

[104] I.P. Stern. Development of design equations for self-deployable n-strut tensegrity systems. Master's thesis, University of Florida, Gainesville, FL, 1999.

[105] B. Huber AND B. Sturmfels. A polyhedral method for solving sparse polynomial system. *Math. Comp.*, 64:1541–1555, 1995.

[106] R.S. Tobie. A report on an inquiry into the existence, formation and representation of tensile structures. Master's thesis, Pratt Institute, New York, 1976.

[107] P. Todd. A k-tree generalization that characterizes consistency of dimensioned engineering drawings. *SIAM J. Discrete Mathematics*, 2:255–261, 1989.

[108] D. C. Gossard and R. Light V. Lin. Variational geometry in computer-aided design. In *ACM SIGGRAPH*, pages 171–179, 1981.

[109] J. and Crane C. Yin J. Duffy. An analysis for the design of self-deployable tensegrity and reinforced tensegrity prisms with elastic ties. *International Journal of Robotics and Automation, Special Issue on Compliance and Compliant Mechanisms*, 17, 2002.

[110] A Zlotnick. To build a virus capsid: an equilibrium model of the self assembly of polyhedral protein complexes. *J. Mol. Biol.*, 241:59–67, 1994.

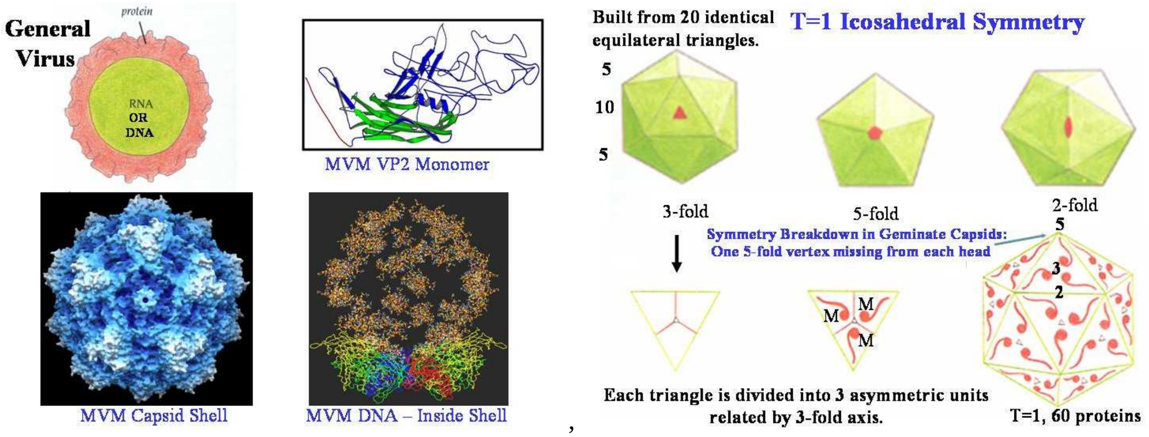
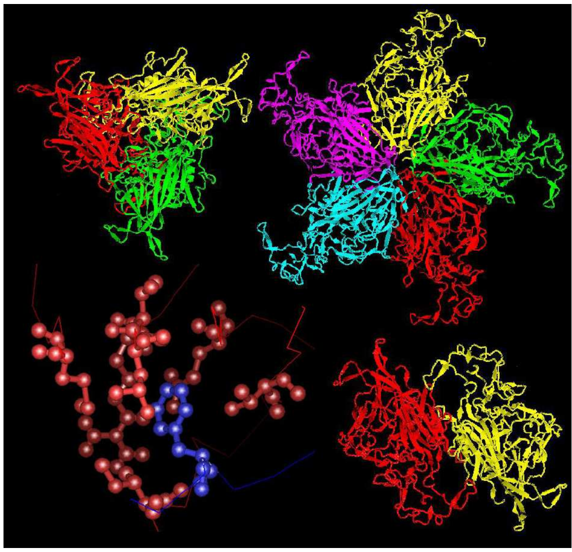

General Virus
protein
RNA
OR
DNA
MVM VP2 Monomer
MVM Capsid Shell
MVM DNA – Inside Shell
Built from 20 identical equilateral triangles.
T=1 Icosahedral Symmetry
5
10
5
3-fold
5-fold
2-fold
Symmetry Breakdown in Geminate Capsids: One 5-fold vertex missing from each head
5
3
2
M
M
M
Each triangle is divided into 3 asymmetric units related by 3-fold axis.
T=1, 60 proteins

Figure 1: Basic Viral Structure. See Figure 2 for geminate shell structure



,

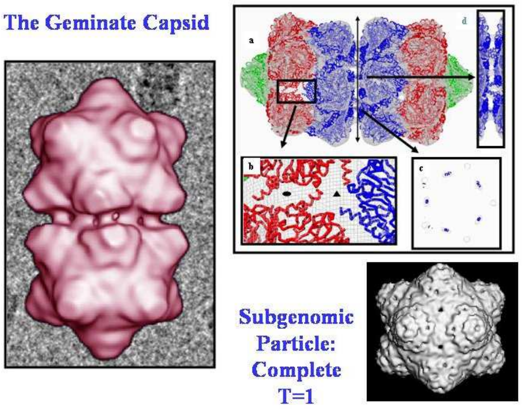


Figure 2: Dimer, trimer and pentamer interactions and close-up of atomic interaction (Left). Geminivirus MSV

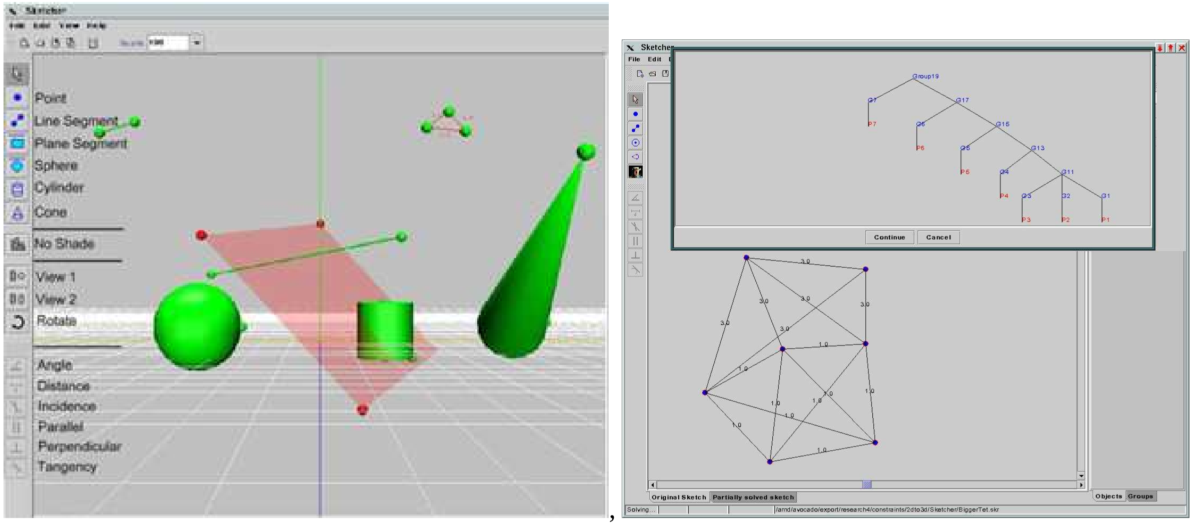
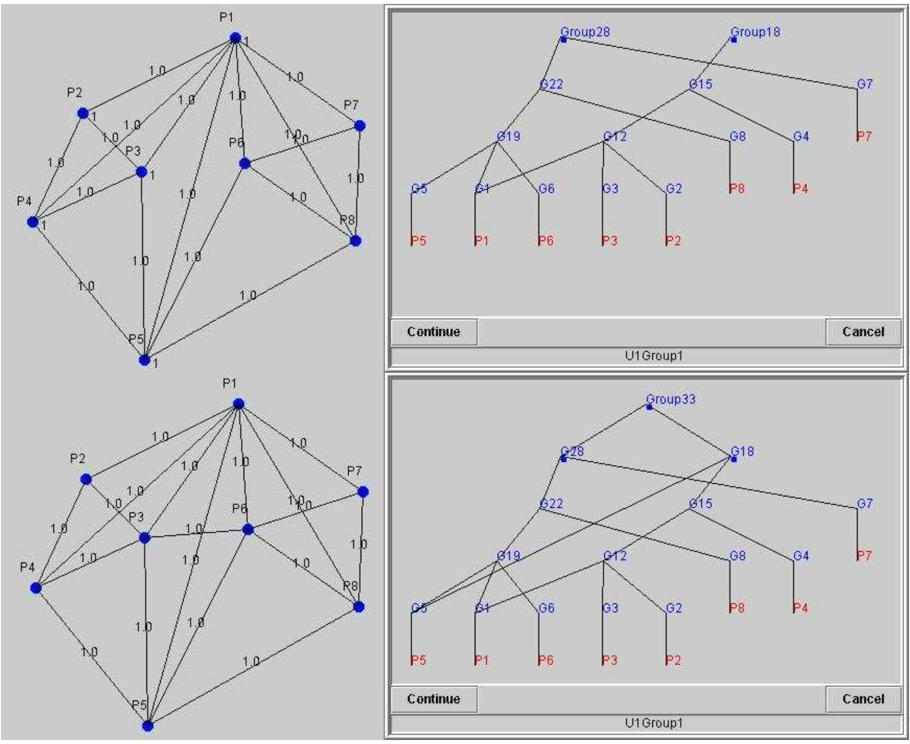


Figure 3: (Left)FRONTIER [100] 3D geometric constraint interface: icons show repertoire of object and constraint types. (Right) Input 3D constraint system, output decomposition or DR plan



,

Figure 4: Combinatorially well constrained (DR-plan has single source, top) and underconstrained (DR-plan has many sources, bottom) 3D constraint system.

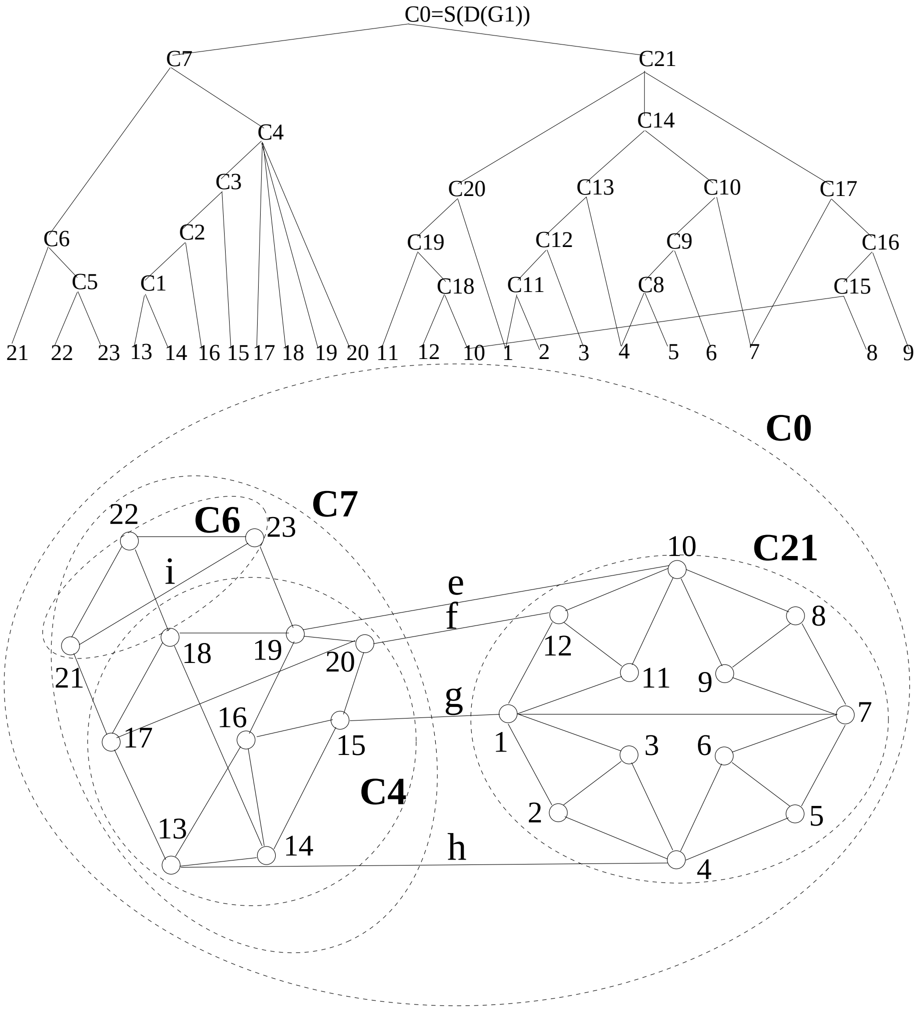
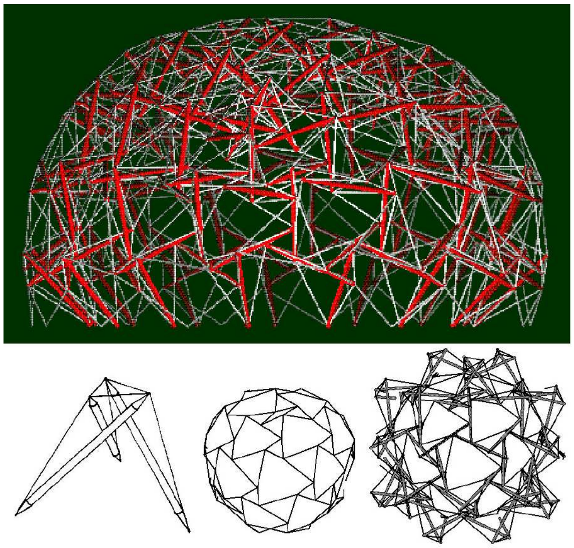
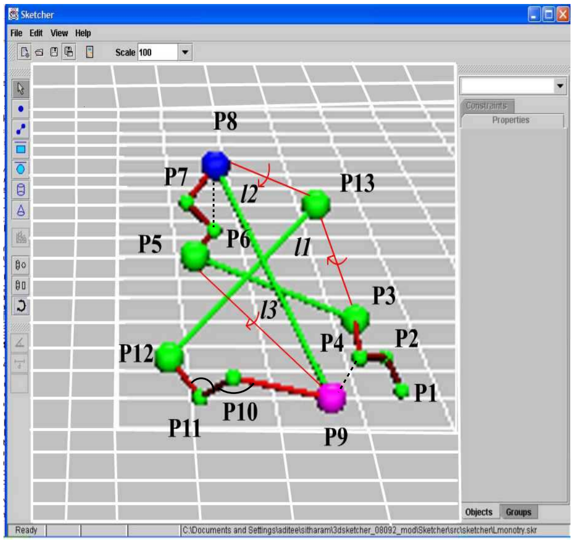


Figure 5: 2D constraint graph G1 and DR-plan; all vertices are points and edges are distance constraints





Ready

C:\Documents and Settings\aditee\sitharam\3dsketcher_08092_mod\Sketcher\src\sketcher\Lmonotry.skr

Figure 6: (Left) Tensegrity systems. (Right) Example monomer primitives and constraints. Balls (points) - atomic markers; Green line segments - variable length bonds; Arrows - torsion angles between green line segments (primary structure) Red - distances representing fixed length bonds (primary structure), Arcs – angles (primary structure), Dotted lines – distances (weak force) (using FRONTIER [100])

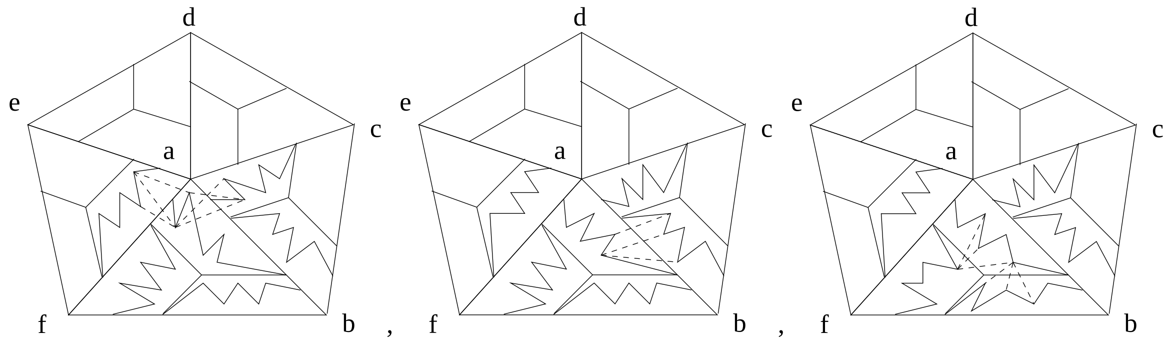
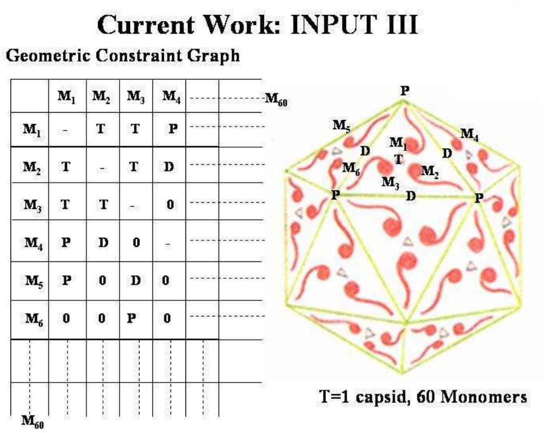


Figure 7: 3 types of interface constraints



Current Work: INPUT III

Geometric Constraint Graph

	M_1	M_2	M_3	M_4	---------- M_{60}
M_1	-	T	T	P	----------
M_2	T	-	T	D	----------
M_3	T	T	-	0	----------
M_4	P	D	0	-	----------
M_5	P	0	D	0	----------
M_6	0	0	P	0	----------
M_{60}					

T=1 capsid, 60 Monomers

,

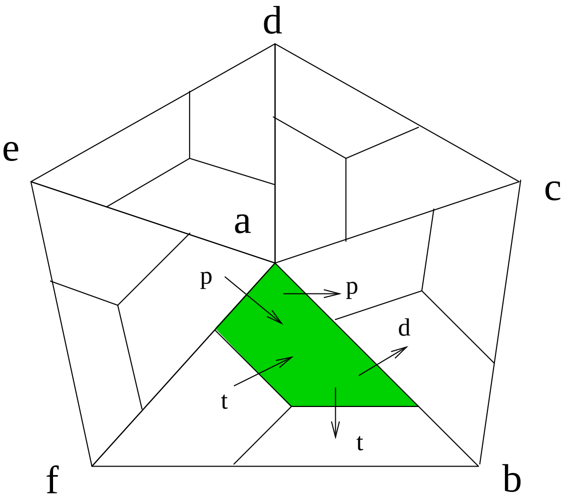


Figure 8: Icosahedral neighborhood graph - the 3rd input parameter

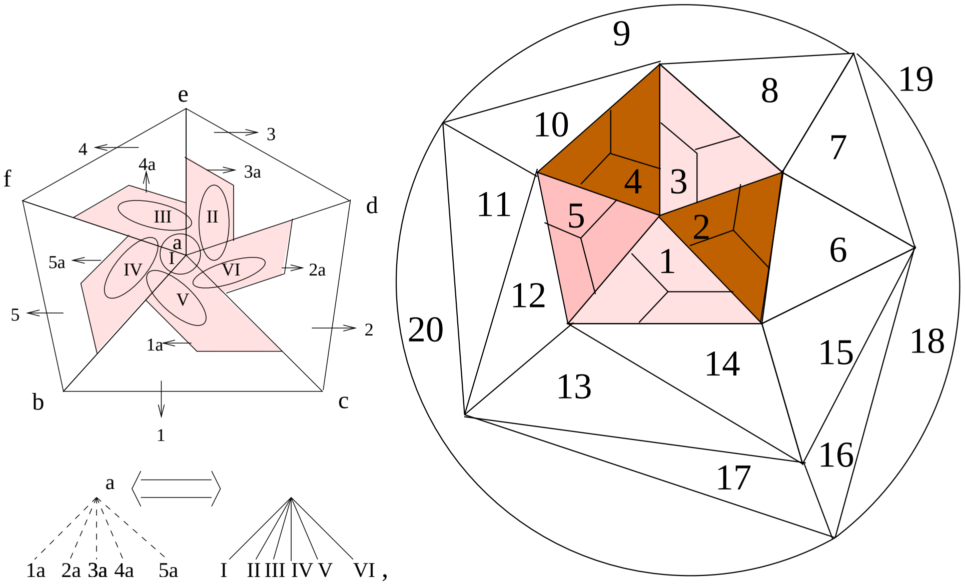


Figure 9: (Right) Icosahedral face numbers and pentamer of trimers. (Left) Clusters that overlap parts of monomers; decomposition of stable pentamer a into non-monomer clusters

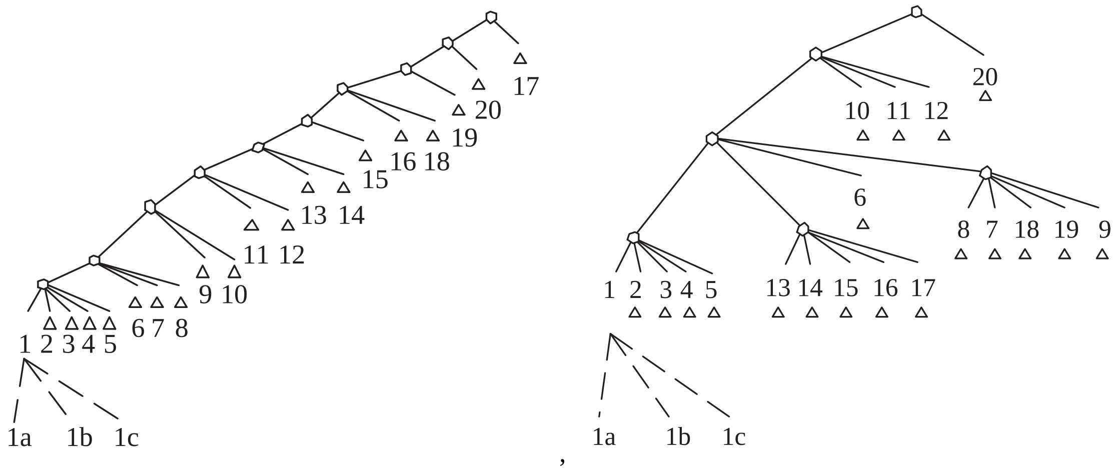


Figure 10: Pathways based on trimeric stable subassemblies

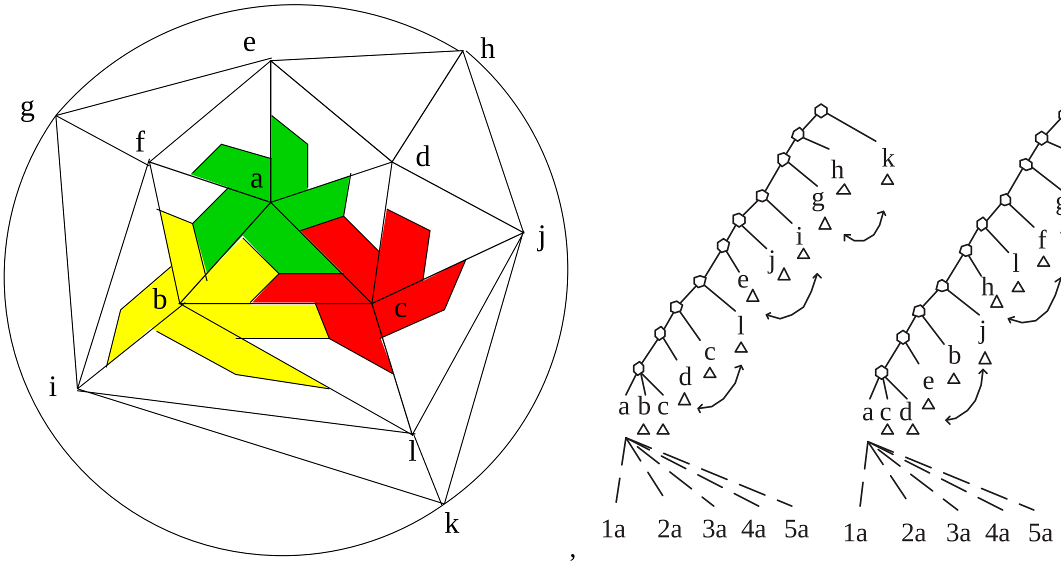


Figure 11: (Left) Icosahedral vertex numbers: trimer of pentamers. (Right) pathway isomorphism classes based on pentamer stable subassemblies

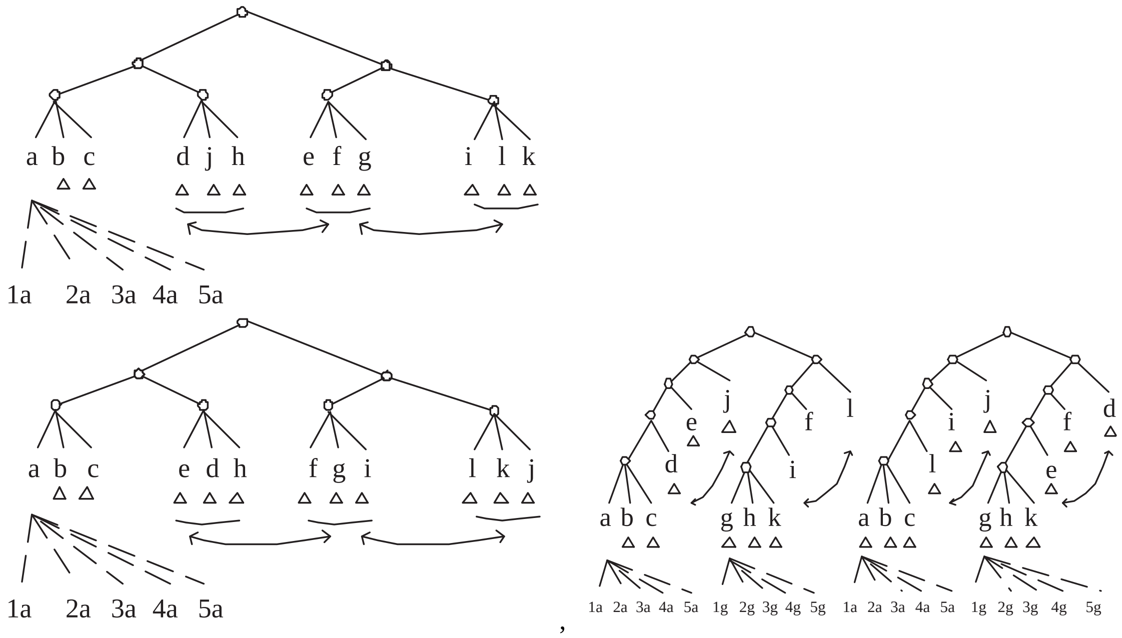
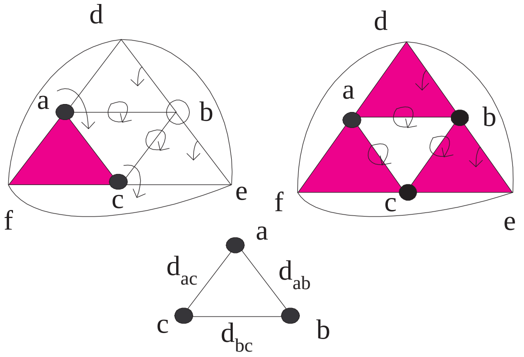


Figure 12: More pathway isomorphism classes based on pentameric stable subassemblies



,

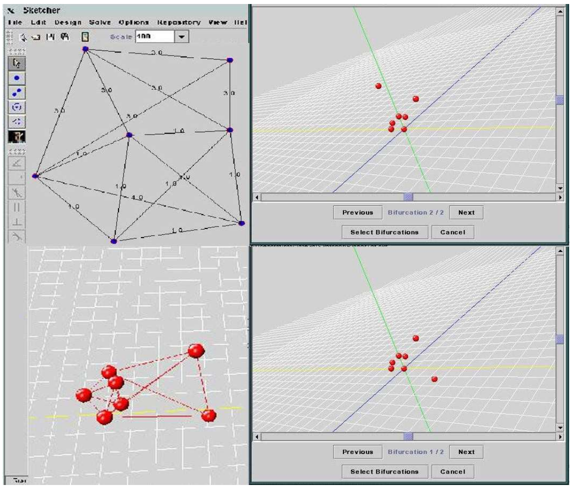


Figure 13: (Left) Example of algebraic reduction: Octahedron split into 3 triangle clusters joined by 3 overlap points and 3 distance constraints – see description in text. (Right) FRONTIER: conformational navigation, final chosen realization for constraint system in Figure 3

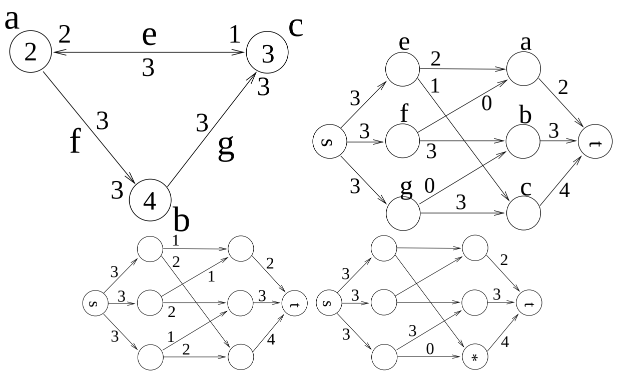


Figure 14: From Left. Constraint graph G with edge weight distribution. D is assumed to be 0 (system fixed in coordinate system); A corresponding flow in bipartite G^*. Another possible flow. Initial flow assignment that requires redistribution

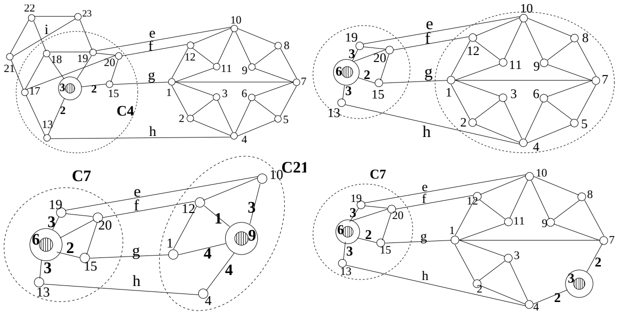


Figure 15: From left: FA's simplification of graph giving DR-plan in Figure 8.4; clusters are simplified in their numbered order: C4 is simplified before C7 etc.

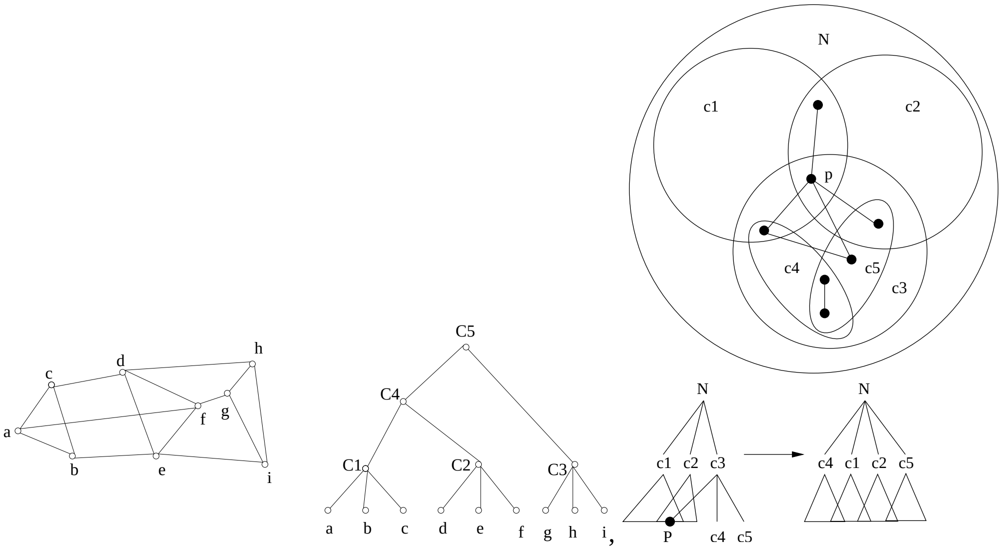


Figure 16: (Left) Maintaining linear width of DR-plan: see text. (Right) Ensuring Property (ii): C_5 and C_4 are directly made children of C, destroying C_3; the operation is performed recursively on C_4 and C_5.

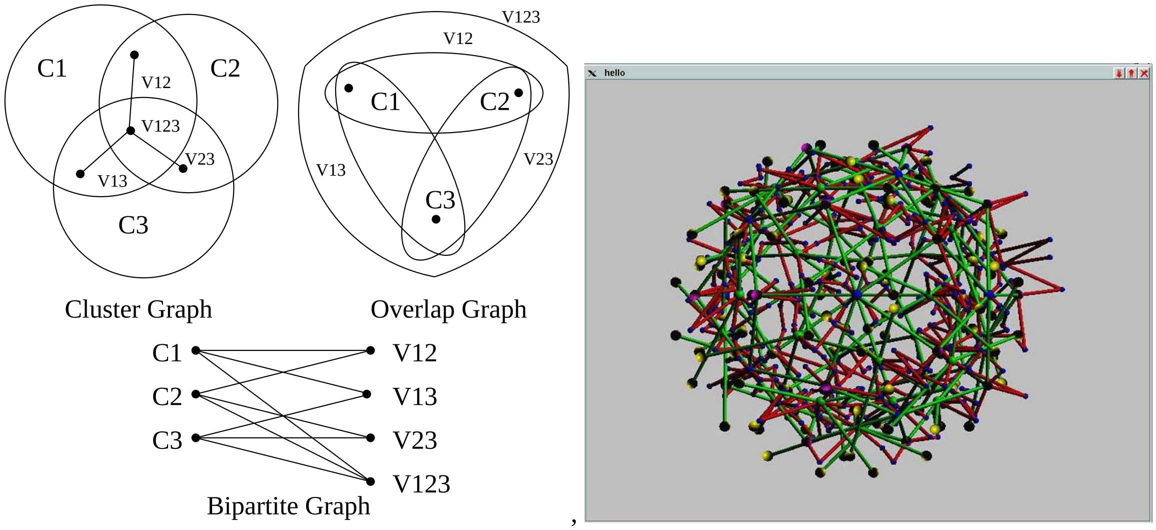
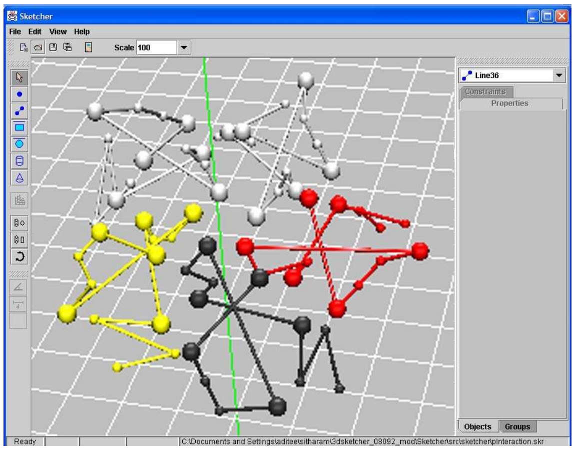


Figure 17: (Left) Ensuring Overlap minimality – finding minimum vertex separator on bipartite graph obtained from the overlap graph of the original cluster graph. (Right) The assembled viral shell: the solved viral geometric constraint graph of Figures 6, 7 and 8


Sketcher
File
Edit
View
Help
Scale
100
Line36
Constraints
Properties
Objects
Groups
Ready
C:\Documents and Settings\aditee\sitharam\3dsketcher_08092_mod\Sketcher\src\sketcher\pInteraction.skr

,

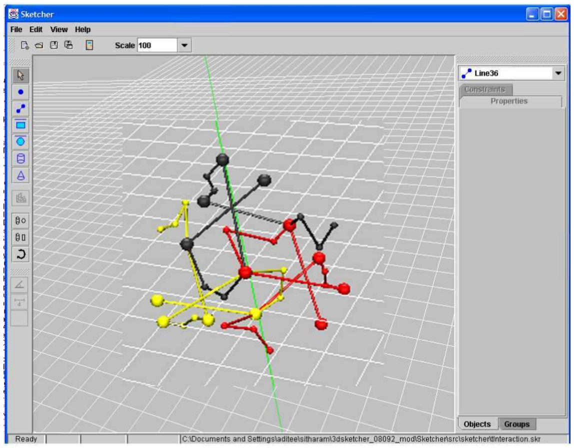
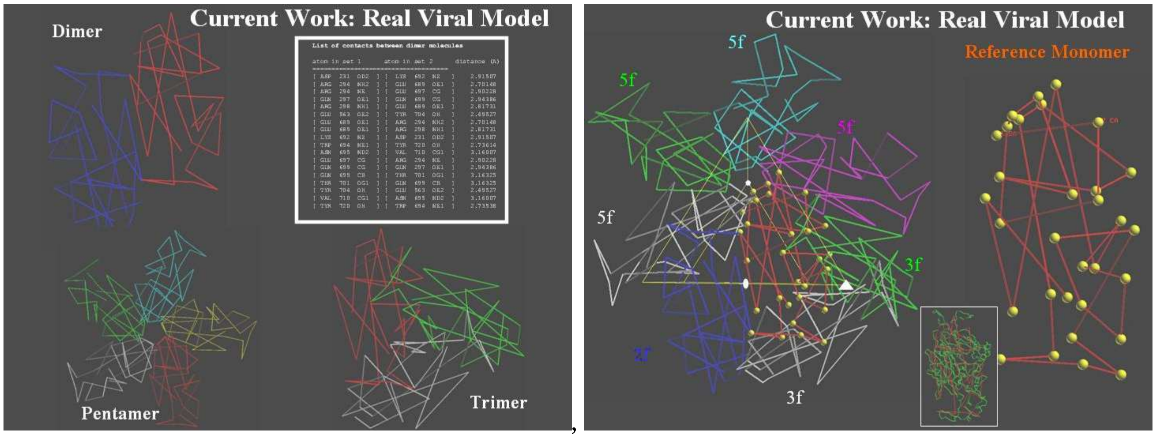

Sketcher
File
Edit
View
Help
Scale
100
Line36
Constraints
Properties
Objects
Groups
Ready
C:\Documents and Settings\aditee\sitharam\3dsketcher_08092_mod\Sketcher\src\sketcher\tInteraction.skr

Figure 18: Pentamer and Trimer stable subassemblies of viral geometric constraint graphs obtained from Figures 6, 7 and 8


Dimer
Current Work: Real Viral Model
Pentamer
Trimer
Current Work: Real Viral Model
5f
5f
5f
Reference Monomer
5f
3f
2f
3f

```
List of contacts between dimer molecules

atom in set 1        atom in set 2        distance (A)
======================================
[ ASP  231  OD2  ] [  LYS  692  NZ   ]     2.91587
[ ARG  294  NH2  ] [  GLU  689  OE1  ]     2.78148
[ ARG  294  NE   ] [  GLU  697  CG   ]     2.90228
[ GLN  297  OE1  ] [  GLN  699  CG   ]     2.94386
[ ARG  298  NH1  ] [  GLU  689  OE1  ]     2.81731
[ GLU  563  OE2  ] [  TYR  704  OH   ]     2.45527
[ GLU  689  OE1  ] [  ARG  294  NH2  ]     2.78148
[ GLU  689  OE1  ] [  ARG  298  NH1  ]     2.81731
[ LYS  692  NZ   ] [  ASP  231  OD2  ]     2.91587
[ TRP  694  NE1  ] [  TYR  720  OH   ]     2.73614
[ ASN  695  ND2  ] [  VAL  710  CG1  ]     3.16087
[ GLU  697  CG   ] [  ARG  294  NE   ]     2.90228
[ GLN  699  CG   ] [  GLN  297  OE1  ]     2.94386
[ GLN  699  CB   ] [  THR  701  OG1  ]     3.16325
[ THR  701  OG1  ] [  GLN  699  CB   ]     3.16325
[ TYR  704  OH   ] [  GLU  563  OE2  ]     2.45527
[ VAL  710  CG1  ] [  ASN  695  ND2  ]     3.16087
[ TYR  720  OH   ] [  TRP  694  NE1  ]     2.73538
```

Figure 19: Real viral (MVM) data: atomic markers, rigid monomers and simplified distance-based interactions (preliminary computational model input)