The unrooted set covering connected subgraph problem differentiating between HIV envelope sequences

Stephen J Maher*[1,2] and John M Murray[2]

[1]Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany.

[2]School of Mathematics and Statistics, University of New South Wales, Sydney NSW 2052, Australia.

Abstract

This paper presents a novel application of operations research techniques to the analysis of HIV Env gene sequences, aiming to identify key features that are possible vaccine targets. These targets are identified as being critical to the transmission of HIV by being present in early transmitted (founder) sequences and absent in later chronic sequences. Identifying the key features of Env involves two steps: first, calculating the covariance of amino acid combinations and positions to form a network of related and compensatory mutations; and second, developing an integer program to identify the smallest connected subgraph of the constructed covariance network that exhibits a set covering property. The integer program developed for this analysis, labelled the unrooted set covering connected subgraph problem (USCCSP), integrates a set covering problem and connectivity evaluation, the latter formulated as a network flow problem. The resulting integer program is very large and complex, requiring the use of Benders' decomposition to develop an efficient solution approach. The results will demonstrate the necessity of applying acceleration techniques to the Benders' decomposition solution approach and the effectiveness of these techniques and heuristic approaches for solving the USCCSP.

Key words: OR in medicine, HIV Env sequence, Benders' decomposition, acceleration techniques

*Corresponding Author. Phone: +49 30 84185-252. Email: maher@zib.de

Introduction

Human immunodeficiency virus (HIV) currently infects approximately 40 million people worldwide and has resulted in over 20 million deaths. Although antiretroviral therapy has reduced mortality and morbidity from this disease in developed countries the large number of new infections each year demonstrates a need for more effective prevention programs. The most effective prevention will be provided by a vaccine, but to date no successful HIV vaccine has been developed. One of the difficulties of developing a vaccine is the high rate of mutation of HIV that results in shifting targets for the immune response. A feature of particular importance is the viral envelope gene Env that codes for the gp160 protein, which is cleaved into the gp120 and gp41 glycoproteins. gp120 buds from the virion surface and is responsible for binding to the CD4 receptor of immune cells, while gp41 is responsible for fusion with the target cell membrane and mediates the resulting infection of the cell.

The most exposed regions of gp120 are susceptible to antibody binding and clearance and hence can be highly variable. An effective vaccine must stimulate the expansion of antibodies to regions of gp120 that are necessary for viable virus and are also amenable to antibody contact. The evolving nature over the course of infection of Env and its glycoproteins gp120 and gp41 is evidence that antibodies against HIV envelope are generated and force the Env gene to mutate to avoid viral clearence. However, antibody development is too slow to inhibit infection of an individual or to eliminate infection once established. A vaccine needs to prime an antibody response to very early stages of HIV. The targets for such a vaccine are currently unclear, but may be elucidated by studying how Env changes from the early (founder) stage of infection to the chronic stage [11]. Features in the Env gene that are present in founder viruses but absent in chronic viruses may indicate aspects that are under eventual immune pressure, and are candidates for vaccine targets.

Initial investigations to identify amino acid features of Env between the two virus groups, founder and chronic, followed the approach of Murray *et al.* [17], which examined amino acid pairs in the hepatitis C virus (HCV) envelope that distinguished responders to antiviral therapy. The current investigation analyses 266 HIV Env protein sequences, 133 founder and 133 chronic, obtained from a previous study comparing glycosylation sites between founder and chronic individuals [15]. Since a total of 266 Env sequences in the current investigation, and in other studies [1, 13, 15], is small in comparison to the 858 amino acids (AA) in HIV Env there can

be many positions in the sequences that will express different AA for founders compared to chronics. Hence the underlying state space employed is not the individual AA, but rather covarying pairs of AA that achieve some minimal level of covariance. AA at a pair of positions are said to covary if the AA combinations observed at these positions are sufficiently different from random combinations. For example, considering the AA Phenylalanine (F), Lysine (K), Arginine (R) and Tyrosine (Y) if in 16 sequences at positions 223 and 432 there are 8 FK (F at 223 and K at 432) pairs and 8 YR pairs, then positions 223 and 432 covary since these observed pairs are sufficiently different to the random combinations of 4 FK, 4 FR, 4 YK and 4 YR pairs.

The approach developed by Murray *et al.* [17] identifies the fewest AA pairs that express particular amino acid combinations present in one group but not the other. The integer programming formulation and the resulting analysis identifies a set of separating pairs that are usually not connected. While these pairs identify important positions in Env the lack of connectivity reduces how useful the positions and the identified AA are in the development of a vaccine against HIV. The current analysis attempts to identify a set of covarying pairs that form a connected subgraph. This connected subgraph is desirable for several reasons. First, a number of compensatory mutations will be needed for Env to sufficiently change its structure to evade the immune response, identified by covarying positions. Second, an antibody stimulated by a vaccine will bind to between 5 and 8 AA of its antigen. Finally, a network of target AA may better describe underlying mechanisms by which HIV evades immune system clearance.

In this paper an integer programming model is developed that selects a set of covarying pairs with the following properties. First, the *most important* features of the sequence are identified. This is defined as the fewest number of pairs that express AA combinations that exist in some founder sequences but in no chronic sequences. Second, the selected covarying pairs and amino acid positions form a connected subnetwork of the original covariance network where feasible. The integer programming problem developed to achieve this is termed the unrooted set covering connected subgraph problem (USCCSP).

1 Literature Review

The connected subgraph problem describes a broad class of problems to which the USCCSP belongs. This problem class has applications to a variety of different research fields. Such areas of research include wildlife conservation [8, 9], network design [3], analysis of protein-protein

interactions [5, 10] and forest harvesting [6]. This paper represents the first application of this problem class to the analysis of mutational relationships in gene sequences. The fundamental aspect observed in applications of this problem class is the distribution of key features, either essential habitat regions or *important* covarying AA pairs, across a large network. The wildlife conservation applications attempt to identify contiguous regions that improve the mobility of threatened species between reserves [8, 9] or connect known habitats of a number of different species [18–20]. Identifying a connected subnetwork that describes these important features is extremely important to minimise the cost of conservation or, in our case, identify closely related features to aid vaccine development.

As stated previously, the USCCSP is a specific variant of the connected subgraph problem. This variant is distinguished by having no root or terminal nodes specified, implying that the connected subgraph may be found in any region of the underlying network. To the best of the authors knowledge, such a problem formulation has not been explicitly considered in the literature.

A typical example of the connected subgraph problem with multiple fixed terminal nodes is presented by Conrad *et al.* [7]. The specification of terminal nodes alters the problem formulation by providing a fixed set of vertices that must be included in the resulting subgraph. The solution approach employed by Conrad *et al.* [7] involves two key stages, i) identifying the nodes and edges to include in the subgraph and ii) checking the graph for connectivity. The latter of these stages is formulated as a network flow problem with a single source and multiple terminal locations. This work is extended by Gomes *et al.* [12], enhancing the solution approach with the introduction of a two-phase algorithm. The first phase of the algorithm presented in [12] solves a minimum Steiner tree problem to identify a feasible, but sub-optimal, connected subgraph solution. Given a feasible solution, the second phase then solves a mixed integer program to improve the solution quality.

The conservation reserve network problem considered by Önal and Briers [18, 19] and Önal and Wang [20] is similar to the problem presented in this paper. In particular, the selection of habitat regions is formulated as a set covering problem, without the specification of root or terminal nodes. However, the data used by Önal and Briers [18, 19] forces the inclusion of specific habitat sites. This requirement implicitly defines a set of terminal nodes for the resulting connected subgraph. A limitation of the work presented by Önal and Briers [18, 19] and Önal and Wang [20] is the use of a network based upon two dimensional data that is partitioned

into a regular square grid. Consequently, there is an upper bound of 8 on the degree of each node, which is significantly smaller than the node degree observed in the underlying graphs considered in this paper. A final limitation of these approaches [18–20] is the requirement that the resulting subgraph forms a spanning tree of the selected sites, preventing the possibility of cycles appearing in the optimal solution.

Connectivity in subgraphs has been evaluated using a variety of different modelling approaches. However, the permissible methods for connectivity evaluation are dependant on whether the subgraph is formed with the selection of nodes or edges. Forming subgraphs by selecting nodes is the most common approach employed, which permits the use of trees to construct connected subgraphs. The property that a tree is a completely connected graph is exploited by Önal and Briers [18,19] and Önal and Wang [20]. In addition, the Steiner tree problem is a very useful and closely related problem, which is employed by Dilkina and Gomes [9]. Alternatively, Gomes *et al.* [12], Conrad *et al.* [8] and Dilkina and Gomes [9] consider the single and multi-commodity flow problems as a method to impose connectivity constraints. Finally, the properties of node-cut sets are employed by Carvajal *et al.* [6] to impose connectivity constraints between two non-adjacent nodes that are selected in the subgraph. The connectivity evaluation approaches presented by Gomes *et al.* [12], Conrad *et al.* [8], Dilkina and Gomes [9] and Carvajal *et al.* [6] are also permissible for problems forming subgraphs through the selection of edges.

The contributions of this paper are twofold, the analysis of HIV Env sequences and the development of the unrooted set covering connected subgraph problem. The contribution to this application area is the development of a sophisticated integer programming problem to analyse HIV Env in regards to vaccine development. Second, this paper extends the connected subgraph problem class by presenting a general formulation of the unrooted set covering variant. The solution approaches presented in this paper have not been previously considered, and the novel implementation of Benders' decomposition is a contribution of this paper.

The exposition in this paper is presented with the following structure. Section 2 provides a description of the problem and presents key details related to the application. Section 3 describes the mathematical model for the general form of the USCCSP that is used to solve the problem presented in Section 2. An extension to the USCCSP that provides an alternative analysis of the HIV sequences is also given in Section 3. The application of Benders' decomposition to solve the connected subgraph problem is described in Section 4, including a description

of a trust region approach to accelerate the algorithm convergence. The computational results for the original problem formulation and extensions are presented in Section 5. The conclusions and possible future work are detailed in Section 6.

2 Problem description

The USCCSP attempts to identify a set covering of items by selecting edges of a graph, with those edges forming a connected subgraph. As input the problem receives an undirected graph $G = (V, E)$ and a set of items Q. Subsets of Q are observed on each edge contained in E. The objective of this problem is to select a subset of edges $\bar{E} \subseteq E$; such that, each item in Q appears on at least one edge contained in $\bar{E}$. This is a set covering problem for the items in Q on graph G. Additionally, the resulting subgraph $\bar{G} = (V(\bar{E}), \bar{E})$ must be connected, otherwise a penalty is applied for each additional edge from $E \backslash \bar{E}$ that is required to form a connected network.

The problem considered in this paper shares characteristics with many classical problems. First, a minimum spanning tree identifies a subset of edges $\bar{E}$ that connects all vertices V. While similar to the USCCSP, the constraints requiring that all vertices are connected and those prohibiting cycles are overly restrictive. Second, the Steiner tree problem aims to find a subset of vertices $\bar{V}$ and related edges $E(\bar{V})$ that connect a set of required vertices T. This is closely related to the USCCSP; however, the resulting graph is a tree and the set T must be known a priori. Finally, the maximum weight connected subgraph problem is solved on a graph, with both positive and negative vertex weights, to find $\bar{V} \subseteq V$ and subsequently $E(\bar{V})$ that form a connected subgraph with a minimum total vertex weight. Both the USCCSP and the maximum weight connected subgraph problem share the connected subgraph objective; however the former deviates from the latter by solving a set covering problem. As such, the selection of edges to satisfy a set covering problem and connectivity requirements separates the USCCSP from classical connected network problems.

Within the biology context, the solution to the USCCSP identifies the smallest set of covarying AA pairs forming a connected network that express combinations exhibited by the founder viruses but not the chronic. The graph G used to solve this problem is constructed using the following steps. First, a covariance network is constructed from both the founder and chronic sequences using the method presented by Aurora *et al.* [4]. The approach of [4] is basically a

variant of a chi-squared test to determine whether the AA combinations observed at a pair of positions across all sequences differ from what would be observed from random combinations. Covarying pairs are ones that achieve a specified cut-off value. The set Q consists only of the founder sequences, so the AA combinations on each covarying pair that are displayed by any chronic virus must be discarded. The resulting network (G) consists of Env positions (V) and covarying pairs (E) with AA combinations exhibited only by founder sequences (Q). This network is described as the founder sequence separating pairs network, hereafter the *separating pairs network*.

Methods identifying the smallest set of covarying AA pairs that separate groups of sequences are presented by [16,17]. These are formulated purely as set covering problems and hence these networks will not necessarily be connected. The USCCSP extends the *minimal separating pairs problem* [16, 17] by introducing constraints that enforce the connectivity between the selected pairs.

3 The unrooted set covering connected subgraph problem

Two important features of the USCCSP are the graph G and the set of sequences Q. The structure of G depends on the problem application. In regards to the HIV analysis application considered in this paper, two different constructions of G are possible. The different graph constructions lead to alternative problem formulations for the USCCSP. These alternative formulations will be presented in Sections 3.1 and 3.2.

The separating pairs network can be constructed in two different ways: a single edge between each pair of positions, which represents all AA combinations, or with an edge for each observed AA combination. An example of a pair of nodes in the two different graphs is displayed in Figure 1. The first graph $G = (V, E)$ is given by the separating pairs network where the vertices V are given by the set of AA positions observed on at least one separating pair and

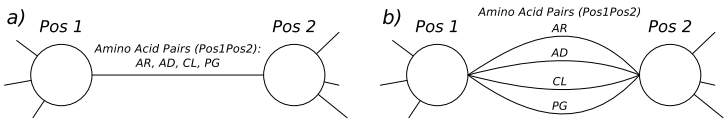


Figure 1: a) A single edge representing all separating covarying amino acid combinations between positions 1 and 2. b) An edge for each separating covarying amino acid combination between positions 1 and 2.

E is the set of separating pairs, which are denoted by (i, j), where $i, j \in V$. Figure 1b is an example of the second graph construction. Specifically, for each separating pair $(i, j) \in E$, the set of all observed amino acid combinations (m, n), where AA m is observed at i and n at j, are contained in the set A_{ij}. Extending this definition, the set B_i contains all AA that are observed at position i. The modified graph $G = (V, \hat{E})$ is given by $\hat{E}$ containing all separating pairs and related amino acid combinations, denoted by (i_m, j_n) where $(i, j) \in E$ and $(m, n) \in A_{ij}$.

3.1 Single edge for each separating pair

The single edge between each pair of positions represents all AA combinations exhibited only by the founder sequences. To formulate the set covering problem, all founder sequences q are contained in the set Q and the binary parameters a_{ijq} identify whether q is observed on the separating pair (i, j). The variables x_{ij} are defined to equal 1 if separating pair (i, j) is selected, 0 otherwise, at a cost of c_{ij}.

The connectivity evaluation is formulated as a network flow problem between each pair of nodes in the solution to the set covering problem. However, the AA positions selected in the set covering solution are unknown *a priori*. Consequently, a set of source-sink pairs is formed using all pairs of nodes in V. To properly model the connectivity requirements, the set S is defined to contain all source-sink pairs $(s, t), s \in V, t \in T^s$, where $T^s = \{t \in V | t > s\}$.

The network flow problem requires an additional set of variables for each source-sink pair contained in S. Flow variables are given by y_{ij}^{st} that equal 1 if a selected pair (i, j) is used in the path between source-sink pair (s, t), and 0 otherwise. To enforce connectivity of the set covering solution, a penalty is applied for each non-selected pair required for a unit of flow to pass from s to t. This penalty is applied with the introduction of variables z_{ij}^{st}. Through problem constraints, if $y_{ij}^{st} = 1$ and $x_{ij} = 0$, the value of z_{ij}^{st} is 1, otherwise $z_{ij}^{st} = 0$. A multiplicative parameter M is applied to the z_{ij}^{st} variables in the objective function to alter the potency of the connectivity evaluation in the USCCSP. There is no guarantee that the original graph G is connected. Hence, it may not be possible for flow to pass between each pair of nodes in the set covering solution. The variables ϵ_j^{st} and $\hat{\epsilon}_j^{st}$ are introduced to penalise any violation of the flow balance between a pair of nodes, with a cost N in the objective function.

While a network flow problem is formulated for each source-sink pair, it is only necessary to identify a path between nodes s and t if these nodes exist on at least one selected pair. The key features of the connectivity evaluation problem are explained with reference to a small graph

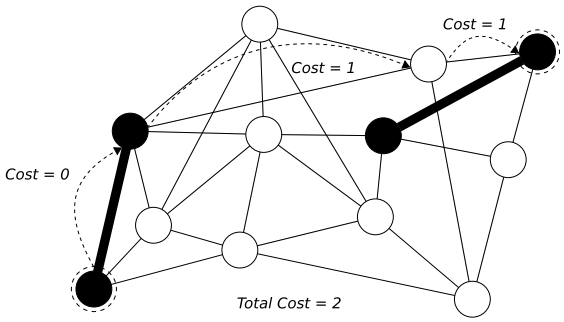


Figure 2: Given a feasible set covering solution, the connectivity evaluation problem *measures* the shortest distance between each selected pair of nodes. The shortest distance between the circled pair of nodes for this solution is two.

presented in Figure 2. In this example two edges are selected (highlighted in bold), resulting in a subgraph containing four nodes (filled circles). Given this solution to the USCCSP, a unit of flow must pass between each of the four selected nodes. In the model constraints, if $\max\{x_{si} + x_{jt} - 1 | i, j \in V\} = 1$ then flow must pass between source-sink pair (s, t). If an (s, t) pair does not satisfy this constraint, because at least one of s or t is not contained in the selected pairs, then the problem minimisation will lead to a zero flow between these nodes. As a result, the cost of passing a unit of flow between the two circled nodes in Figure 2 is two and the total cost of passing a unit of flow between all selected nodes is eight, since $t > s, \forall (s, t) \in S$.

The mathematical model of the USCCSP is given by,

$$\text{minimise} \quad \sum_{(i,j)\in E} c_{ij}x_{ij} + M \sum_{(s,t)\in S} \sum_{(i,j)\in E} z_{ij}^{st} + N \sum_{(s,t)\in S} \sum_{j\in V\setminus\{s,t\}} \left(\epsilon_j^{st} + \hat{\epsilon}_j^{st}\right), \tag{1}$$

$$\text{subject to} \quad \sum_{(i,j)\in E} a_{ijq}x_{ij} \geq 1 \quad \forall q \in Q, \tag{2}$$

$$y_{ij}^{st} - x_{ij} \leq z_{ij}^{st} \quad \forall s \in V, \forall t \in T^s, \forall (i,j) \in E, \tag{3}$$

$$\sum_{\substack{i\in V| \\ (i,j)\in E}} y_{ij}^{st} - \sum_{\substack{k\in V| \\ (j,k)\in E}} y_{jk}^{st} = \epsilon_j^{st} - \hat{\epsilon}_j^{st} \quad \forall s \in V, \forall t \in T^s, \forall j \in V\setminus\{s,t\}, \tag{4}$$

$$\sum_{\substack{k\in V| \\ (s,k)\in E}} y_{sk}^{st} \geq x_{si} + x_{jt} - 1 \quad \forall s \in V, \forall t \in T^s, \forall (s,i) \in E, \forall (j,t) \in E, \tag{5}$$

$$\sum_{\substack{k\in V| \\ (k,t)\in E}} y_{kt}^{st} \geq x_{si} + x_{jt} - 1 \quad \forall s \in V, \forall t \in T^s, \forall (s,i) \in E, \forall (j,t) \in E, \tag{6}$$

$$x_{ij} \in \{0, 1\} \quad \forall (i,j) \in E, \tag{7}$$

$$y_{ij}^{st} \in \{0, 1\} \quad \forall s \in V, \forall t \in T^s, \forall (i,j) \in E, \tag{8}$$

$$z_{ij}^{st} \in \{0,1\} \quad \forall s \in V, \forall t \in T^s, \forall (i,j) \in E, \tag{9}$$

$$\epsilon_j^{st}, \hat{\epsilon}_j^{st} \geq 0 \quad \forall s \in V, \forall t \in T^s, \forall j \in V \backslash \{s,t\}. \tag{10}$$

The solution of the USCCSP minimises the number of separating pairs required such that each founder sequence in Q is observed on at least one selected edge. Furthermore, this problem minimises the number of additional separating pairs that are required for flow to pass between each selected source-sink pair.

The set covering problem of the USCCSP is described by the first term in objective function (1) and constraints (2). Constraints (2) describe the requirement that each founder sequence $q \in Q$ must be observed on at least one selected pair ($x_{ij} = 1$). The connectivity evaluation problem is given by constraints (3)-(6) and the variables y_{ij}^{st}, z_{ij}^{st}, ϵ_{ij}^{st} and $\hat{\epsilon}_{ij}^{st}$. Constraints (3) impose a penalty each unselected pair ($x_{ij} = 0$) that is required for a unit of flow to pass between s and t. The flow balance at each node in the connected path between each (s,t) is given by constraints (4). If the flow balance can not be satisfied, then a penalty of N is applied. Constraints (5) and (6) are used to indicate whether a path with nonzero flow must be found between the source-sink pair (s,t), in which case the flow between s and t must be at least one.

The Separating Pairs Problem investigated in [16] is much simpler than the USCCSP by omitting the constraints enforcing connectivity. It consisted of the objective function $\sum_{(i,j)\in E} c_{ij} x_{ij}$ and constraints (2) and (7). The resulting problem is in the form of the classical set covering problem.

3.2 Multiple edges for each separating pair

It is common to observe numerous AA combinations on each separating pair. This results in a covariance network that exhibits multiple edges per pair of nodes. Identifying particular AA combinations per pair of positions may present more meaningful results than the positions themselves.

Multiple edges between each pair of nodes requires an alternative formulation of the USCCSP, which will be labelled the USCCSP-AA. To model the selection of multiple edges between each pair of nodes, the variables $w_{i_m j_n}$ are defined to equal one if AA combination (m,n) is selected on edge (i,j) and zero otherwise. In regards to the biology application, the selection of edges is restricted to at most one for each pair of nodes.

By focusing on the specific AA combinations at each pair of positions, it is possible that a

mismatch of amino acids at intermediate nodes can occur in the connectivity evaluation. For example, if separating pairs (i, j) and (j, k) are selected with the observed AA combinations (F, R) and (K, Y) respectively, the resulting connected subgraph has an AA mismatch at j. To thoroughly analyse the connected positions of HIV sequences, two formulations of the USCCSP-AA will be presented—one permitting AA mismatches and the other excluding them.

The formulation of the USCCSP-AA permitting AA mismatches employs an identical connectivity evaluation problem as that presented for the USCCSP. Using the notation presented in the previous section the formulation of the USCCSP-AA is given by,

$$\text{minimise} \quad \sum_{(i,j)\in E} c_{ij}x_{ij} + M \sum_{(s,t)\in S} \sum_{(i,j)\in E} z_{ij}^{st} + N \sum_{(s,t)\in S} \sum_{j\in V\setminus\{s,t\}} \left(\epsilon_j^{st} + \hat{\epsilon}_j^{st}\right), \tag{11}$$

$$\text{subject to} \quad \text{constraints (3)-(10)}, \tag{12}$$

$$\sum_{(i_m,j_n)\in \hat{E}} a_{i_m j_n q} w_{i_m j_n} \geq 1 \quad \forall q \in Q, \tag{13}$$

$$\sum_{(m,n)\in A_{ij}} w_{i_m j_n} = x_{ij} \quad \forall (i,j) \in E, \tag{14}$$

$$w_{i_m j_n} \in \{0,1\} \quad \forall (i_m, j_n) \in \hat{E}. \tag{15}$$

The most important difference between the USCCSP and USCCSP-AA is the addition of constraints (14). This set of constraints is required to ensure that at most one AA combination is selected per pair of positions.

The problem formulation excluding AA mismatches, labelled the USCCSP-AA', requires a modification to the flow balance constraints in the connectivity evaluation problem. Specifically, in constraints (12) all occurrences of E are replaced with $\hat{E}$ and the variables y_{ij}^{st} and z_{ij}^{st} are replaced by $y_{i_m j_n}^{st}$ and $z_{i_m j_n}^{st}$. Finally, to formulate the USCCSP-AA' constraints (12) are replaced by,

$$y_{i_m j_n}^{st} - w_{i_m j_n} \leq z_{i_m j_n}^{st} \quad \forall s \in V, \forall t \in T^s, \forall (i_m, j_n) \in \hat{E}, \tag{16}$$

$$\sum_{\substack{i\in V| \\ (i,j)\in E}} \sum_{\substack{l\in B_i| \\ (l,m)\in A_{ij}}} y_{i_l j_m}^{st} - \sum_{\substack{k\in V| \\ (j,k)\in E}} \sum_{\substack{n\in B_k| \\ (m,n)\in A_{jk}}} y_{j_m k_n}^{st} = \epsilon_{j_m}^{st} - \hat{\epsilon}_{j_m}^{st}$$

$$\forall s \in V, \forall t \in T^s, \forall j \in V\setminus\{s,t\}, \forall m \in B_j, \tag{17}$$

$$\sum_{\substack{k\in V| \\ (s,k)\in E}} \sum_{(g,h)\in A_{sk}} y_{s_g k_h}^{st} \geq w_{s_m i_n} + w_{j_u t_v} - 1 \quad \forall s \in V, \forall t \in T^s, \forall (s_m, i_n) \in \hat{E}, \forall (j_u, t_v) \in \hat{E}, \tag{18}$$

$$\sum_{\substack{k\in V|\\(k,t)\in E}} \sum_{(g,h)\in A_{kt}} y^{st}_{k_g t_h} \geq w_{s_m i_n} + w_{j_u t_v} - 1 \quad \forall s \in V, \forall t \in T^s, \forall (s_m, i_n) \in \hat{E}, \forall (j_u, t_v) \in \hat{E}, \quad (19)$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in E, \quad (20)$$

$$y^{st}_{i_m j_n} \in \{0,1\} \quad \forall s \in V, \forall t \in T^s, \forall (i_m, j_n) \in \hat{E}, \quad (21)$$

$$z^{st}_{i_m j_n} \in \{0,1\} \quad \forall s \in V, \forall t \in T^s, \forall (i_m, j_n) \in \hat{E}, \quad (22)$$

$$\epsilon^{st}_{j_m}, \hat{\epsilon}^{st}_{j_m} \geq 0 \quad \forall s \in V, \forall t \in T^s, \forall j \in V \backslash \{s,t\}, \forall m \in B_j. \quad (23)$$

The connectivity evaluation constraints (16)-(19) are modified to model the additional edges per pair of positions in the covariance network. In addition, the flow balance constraints (17) are defined for each AA/position combination to ensure the same AA is observed on all edges incoming and outgoing from each node. Finally, the connectivity infeasibility variables, $\epsilon^{st}_{j_m}, \hat{\epsilon}^{st}_{j_m}$, are defined with respect to the additional edges. Consequently, the objective function (11) must be replaced by

$$\sum_{(i,j)\in E} c_{ij} x_{ij} + M \sum_{(s,t)\in S} \sum_{(i_m,j_n)\in \hat{E}} z^{st}_{i_m j_n} + N \sum_{(s,t)\in S} \sum_{j\in V\backslash\{s,t\}} \sum_{m\in B_j} \left(\epsilon^{st}_{j_m} + \hat{\epsilon}^{st}_{j_m}\right). \quad (24)$$

4 Benders' decomposition

A feature of the problems presented Section 3 is the integration of the set covering and connectivity evaluation problems. While these individual problems can be solved using classical techniques, the integration of the two negatively impacts the problem tractability. Fortunately, the formulations of the USCCSP and USCCSP-AA are particularly suited for the application of Benders' decomposition. For conciseness and ease of exposition the application of Benders' decomposition will be described with reference to the USCCSP.

Benders' decomposition forms a master problem, labelled the BMP, and a series of subproblems to reduce the overall problem complexity. The decomposition of the USCCSP involves formulating a BMP as a set covering problem consisting of variables x_{ij} and constraints (2). The subproblems are formulated to evaluate the connectivity of the set cover solution to the BMP, which is provided as a fixed input to the subproblems. In iteration n, the set of solution values for the variables in the BMP is given by $\bar{\mathbf{x}}^n = \{\bar{x}^n_{ij}, (i,j) \in E\}$. The solution given by $\bar{\mathbf{x}}^n$ describes a set of separating pairs (i,j) that is a set cover of the founder sequences contained in Q. This set of edges forms a subgraph of the original separating pairs network.

An individual subproblem is formed for each source node $s \in V$. This is motivated by the prevalence of computationally efficient algorithms that find the shortest path from a single source to all other nodes. Thus, the primal Benders' subproblem for source node s, given selected pairs $\bar{\mathbf{x}}^n$, (PBSP-s) is formulated as,

$$\mu_s(\bar{\mathbf{x}}^n) = \text{minimise} \quad M \sum_{t \in T^s} \sum_{(i,j) \in E} z_{ij}^{st} + N \sum_{t \in T^s} \sum_{j \in V \backslash \{s,t\}} \left(\epsilon_j^{st} + \hat{\epsilon}_j^{st}\right), \tag{25}$$

$$\text{subject to} \quad y_{ij}^{st} - z_{ij}^{st} \leq \bar{x}_{ij}^n \quad \forall t \in T^s, \forall (i,j) \in E, \tag{26}$$

$$\sum_{\substack{i \in V| \\ (i,j) \in E}} y_{ij}^{st} - \sum_{\substack{k \in V| \\ (j,k) \in E}} y_{jk}^{st} = \epsilon_j^{st} - \hat{\epsilon}_j^{st} \quad \forall t \in T^s, \forall j \in V \backslash \{s,t\}, \tag{27}$$

$$\sum_{\substack{k \in V| \\ (s,k) \in E}} y_{sk}^{st} \geq \bar{x}_{si}^n + \bar{x}_{jt}^n - 1 \quad \forall t \in T^s, \forall (s,i) \in E, \forall (j,t) \in E, \tag{28}$$

$$\sum_{\substack{k \in V| \\ (k,t) \in E}} y_{kt}^{st} \geq \bar{x}_{si}^n + \bar{x}_{jt}^n - 1 \quad \forall t \in T^s, \forall (s,i) \in E, \forall (j,t) \in E, \tag{29}$$

$$y_{ij}^{st} \geq 0, \;\; z_{ij}^{st} \geq 0 \quad \forall t \in T^s, \forall (i,j) \in E, \tag{30}$$

$$\epsilon_j^{st}, \hat{\epsilon}_j^{st} \geq 0 \quad \forall t \in T^s, \forall j \in V \backslash \{s,t\}. \tag{31}$$

The connectivity evaluation problem given by the PBSP-s determines the fewest number of edges where $x_{ij} = 0$ that are required to form a path for a unit of flow to pass between s and all $t \in T^s$. While the PBSP-s does not display the classical form of a network flow problem, it is possible to demonstrate the similarities between these two problems by performing a few simple modifications.

The modifications of the PBSP-s rely on defining a fixed amount of flow through the network and using this to set the cost for passing along each edge. Since the paths for each source-sink pair are independent, the PBSP-s is separable by t and an individual network flow problem can be formulated for each $(s,t), t \in T^s$. Section 3 indicates that it is only necessary to identify a path between the source-sink pair (s,t) if there exists $i,j \in V, \bar{x}_{si}^n + \bar{x}_{jt}^n - 1 > 0$. This property can be used to determine the amount of flow to pass from s to t. Additionally, this is enforced by constraints (28) and (29) and the total amount of flow is given by $\psi^{st} = \max_{i,j \in V}\{\bar{x}_{si}^n + \bar{x}_{jt}^n - 1\}$. Finally, constraints (26) set the cost of pushing ψ^{st} amount of flow along edge (i,j).

From the previous observations, a classical network flow problem can be formed. Most importantly, it is possible to eliminate constraints (26) since the cost of pushing ψ^{st} amount of flow along edge (i,j) is fixed by the variables $\bar{\mathbf{x}}^n$. Using this fixed amount of flow, the

variable mapping $z_{ij}^{st} = y_{ij}^{st} \times \max\{\psi^{st} - \bar{x}_{ij}^{n}, 0\}$ can be applied to form the modified problem. Implementing these modifications results in the formulation of a classical network flow problem that can be efficiently solved using a variety of dedicated solution algorithms. Examples of appropriate network flow algorithms are described in Ahuja *et al.* [2].

4.1 Generating Benders' cuts

Solving the PBSP-s using a dedicated network flow algorithm significantly improves the efficiency of the Benders' decomposition solution process. Employing such an algorithm to solve the PBSP-s provides an optimal primal solution; however, no dual solution is readily available. Further, this optimal primal solution is for the modified problem, which needs to be mapped to the original formulation of the PBSP-s in order to generate cuts.

To aid the discussion in this section, the dual variable notation will be provided. The dual variables for the connection enforcement constraints (26) are described by $\boldsymbol{\lambda}^s = \{\lambda_{ij}^{st}, \forall t \in T^s, \forall (i,j) \in E\}$. The dual variables $\boldsymbol{\alpha}^s = \{\alpha_j^{st}, \forall t \in T^s, \forall j \in V\}$ are defined for the flow balance constraints (27). Finally, the dual variables for the source and sink node enforcement constraints (28) and (29) are given by $\boldsymbol{\delta}^s = \{\delta_{ij}^{st}, \forall t \in T^s, \forall (s,i) \in E, \forall (j,t) \in E\}$ and $\boldsymbol{\gamma}^s = \{\gamma_{ij}^{st}, \forall t \in T^s, \forall (s,i) \in E, \forall (j,t) \in E\}$ respectively.

An important observation of the PBSP-s is that the flow balance constraints (27) ensure the subproblem is feasible for all solutions $\bar{\mathbf{x}}$. Hence, only optimality cuts are generated. A Benders' optimality cut describes a feasible region extreme point from the dual of the PBSP-s. Since a dedicated solution algorithm is used to identify the optimal primal solution to the PBSP-s, an optimality cut is constructed by examining the reduced costs of the primal variables. There are six variable types in the PBSP-s, $y_{ij}^{st}, y_{sk}^{st}, y_{kt}^{st}, z_{ij}^{st}, \epsilon_j^{st}$ and $\hat{\epsilon}_j^{st}$, with their respective reduced cost functions given by,

$$\bar{c}_{ij}^{st} = \alpha_i^{st} - \alpha_j^{st} - \lambda_{ij}^{st} \quad \forall (i,j) \in E, \tag{32}$$

$$\bar{c}_{sk}^{st} = \alpha_s^{st} - \alpha_k^{st} - \sum_{(s,i)\in E} \sum_{(j,t)\in E} \delta_{ij}^{st} - \lambda_{sk}^{st} \quad \forall (s,k) \in E, \tag{33}$$

$$\bar{c}_{kt}^{st} = \alpha_k^{st} - \alpha_t^{st} - \sum_{(s,i)\in E} \sum_{(j,t)\in E} \gamma_{ij}^{st} - \lambda_{kt}^{st} \quad \forall (k,t) \in E, \tag{34}$$

$$\bar{d}_{ij}^{st} = M + \lambda_{ij}^{st} \quad \forall (i,j) \in E, \tag{35}$$

$$\bar{e}_j^{st} = N + \alpha_j^{st}, \forall j \in V \backslash \{s,t\}, \tag{36}$$

$$\bar{f}_j^{st} = N - \alpha_j^{st}, \forall j \in V \backslash \{s,t\}. \tag{37}$$

For the case that graph G is connected, the solution to the PBSP-s describes a collection of edges (i, j) that form a connected path p for a flow of ψ_{st} to pass from source node s to sink node t. Hence, the variables y_{ij}^{st}, where $(i, j) \in p$, are basic, implying that their reduced costs are zero. Additionally, the variables z_{ij}^{st} are set to one for each $(i, j) \in p$, if $\bar{x}_{ij} = 0$ and $y_{ij}^{st} = 1$ and zero otherwise, as given by the mapping described previously. Thus, the construction of a dual solution for the PBSP-s to generate Benders' cuts is performed using the following process.

The algorithm to construct an optimal dual solution initially sets the values for all dual variables to zero. Since the variables z_{ij}^{st} are basic if $\bar{x}_{ij} = 0$ and $y_{ij}^{st} = 1$, traversing through $(i, j) \in p$, the values of λ_{ij}^{st} are set by the reduced cost function (35). Then considering the source node and the connection $(s, k) \in p$, equation (33) states that it is valid to set $\alpha_k^{st} = -\lambda_{sk}^{st}$. It follows that for every connection $(i, j) \in p, j \neq t$ the related dual variables can be equated using (32), hence $\alpha_j^{st} - \alpha_i^{st} = -\lambda_{ij}^{st}$. Finally, for the sink node and the connection $(k, t) \in p$, equation (34) states that the expression $\sum_{(s,i)\in E} \sum_{(j,t)\in E} \gamma_{ij}^{st} - \alpha_k^{st} = -\lambda_{kt}^{st}$ is valid. A solution that satisfies this set of equations is given by setting α_k^{st}, for $k \neq t$, equal to the sum of λ_{ij}^{st} for all connections $(i, j) \in p$ from the source node to k. For $k = t$, $\sum_{(s,k)\in E} \sum_{(l,t)\in E} \gamma_{kl}^{st}$ is set to the sum of λ_{ij}^{st} for all connections $(i, j) \in p$. It is permissible to select any $l' \in V$ such that $\gamma_{kl'}^{st} \in \boldsymbol{\gamma^s}$ is positive, provided $\bar{x}_{l't} = 1$.

While an optimal objective value is given by setting the dual variables as described, this does not produce a valid Benders' cut. The added cut eliminates solutions by *transgresses* into the feasible region of the original problem. Such cuts will be termed *transgressing* Benders' cuts. Including the transgressing cut in the master problem penalises the disconnected source-sink pairs. This causes every possible solution including these source-sink pairs, except if the linking edges in the subproblem optimal path are selected, to be penalised. Hence, the values of λ_{ij}^{st} for all (i, j) where $\bar{x}_{ij} = 0$ must be set to $-M$ to produce a valid cut.

The above cut generation process does not hold for the case where graph G is not connected and s and t lie in two different connected components. In this situation no path exists between source s and sink t; as such, the only y_{ij}^{st} that may be set to one are those on an edge incident to the source or sink. Specifically, constraints (28) state that there exists exactly one j such that $y_{sj}^{st} = 1, (s, j) \in E$ and similarly constraints (29) ensure exactly one k exists such that $y_{kt}^{st} = 1, (k, t) \in E$. As a result, $\alpha_j^{st} = -N$ and $\alpha_k^{st} = N$, as given by equations (36) and (37) respectively. It follows from equations (33) and (34) that $\sum_{(s,i)\in E} \sum_{(j,t)\in E} \delta_{ij}^{st} = \sum_{(s,i)\in E} \sum_{(j,t)\in E} \gamma_{ij}^{st} = N$. The resulting solution, with all other dual

variables set to zero, is an optimal dual solution.

The addition of optimality cuts to the BMP also requires the introduction of the variables φ^s for each subproblem $s \in V$. These variables are bounded from below by the added cuts and provide the current lower bound on the objective value of the PBSP-s. The optimality cuts added to the BMP are of the form,

$$\varphi^s \geq \sum_{t \in T^s} \sum_{(s,i) \in E} \sum_{(j,t) \in E} \left(\delta_{ij}^{st} + \gamma_{ij}^{st}\right)\left(x_{si} + x_{jt} - 1\right) + \sum_{t \in T^s} \sum_{(i,j) \in E} \lambda_{ij}^{st} x_{ij}. \tag{38}$$

Cuts are continually added to the BMP until the gap between the upper and lower bounds, given by the solutions to the BMP and PBSP-s, $\forall s \in S$, respectively, reduces to a desired optimality gap.

While a transgressing Benders' cut is generated by the initial steps of the above algorithm, it is still possible to use this cut to generate upper bound solutions. A given disconnected solution is not eliminated, but penalised by this cut. Computational experience shows that searching in a neighbourhood of a previously found solution for alternative subgraphs can identify sub-optimal solutions, based on the added cuts, that are indeed connected. Applying transgressing cuts requires a method that forces the master problem to search for solutions in a neighbourhood of those found in previous iterations. One such method is to employ a trust region, which is described in Section 4.3. The approach using transgressing cuts and a trust region is a heuristic method that quickly identifies upper bound solutions. The produced solutions are of good quality for the HIV Env analysis application.

4.2 Benders' decomposition master problem

The BMP is formulated as a set covering problem with additional cuts to express the evaluation of the subgraph connectivity. The set Ω^s is introduced as an index set for the cuts added from the PBSP-s for source node s. Each cut generated from the solution to the PBSP-s is indexed by ω. The BMP is given by,

$$\text{minimise} \quad \Phi = \sum_{(i,j) \in E} c_{ij} x_{ij} + \sum_{s \in V} \varphi^s, \tag{39}$$

$$\text{subject to} \quad \sum_{(i,j) \in E} a_{ijq} x_{ij} \geq 1 \quad \forall q \in Q, \tag{40}$$

$$\varphi^s \geq \sum_{t \in T^s} \sum_{(s,i) \in E} \sum_{(j,t) \in E} \left(\delta_{ij}^{\omega st} + \gamma_{ij}^{\omega st}\right)\left(x_{si} + x_{jt} - 1\right)$$

$$+ \sum_{t \in T^s} \sum_{(i,j) \in E} \lambda_{ij}^{\omega st} x_{ij} \quad \forall s \in V, \forall \omega \in \Omega^s, \tag{41}$$

$$x_{ij} \in \{0,1\} \ \forall (i,j) \in E \quad \varphi^s \in \mathbb{R} \ \forall s \in S. \tag{42}$$

The objective value of the BMP in a given iteration provides a lower bound on the optimal solution to the USCCSP. In particular, the values of x_{ij} describe the best selection of separating pairs with the smallest "distance" between each of the identified nodes given the current evaluation information from the PBSP-s, $\forall s \in S$. Since the Benders' decomposition subproblems are always feasible, the solution to the BMP in each iteration is a feasible solution to the original problem.

4.3 Trust region method

The application of a trust region restricts the feasible region of the master problem to identify a solution close to that found in the previous iteration. Trust region approaches are employed for Benders' decomposition to focus the added cuts and improve the convergence of the algorithm. Examples of this approach include the addition of a regularisation term in the objective as presented by Ruszczyński [21] and the addition of a set of constraints as demonstrated by Linderoth and Wright [14] and Santoso *et al.* [22]. While both approaches have been demonstrated to improve the convergence of the solution process, the latter maintains the mixed integer programming structure of the master problem and hence it is more appropriate for the USCCSP and USCCSP-AA.

The constraints added to form the trust region restrict the distance between solutions in consecutive iterations. Using the notation presented at the start of this section, the solution values for the Benders' decomposition master problem in iteration n is given by $\bar{\mathbf{x}}^n$. Given the set of solution values $\bar{\mathbf{x}}^n$ it is possible to define $X^n = \{(i,j) | \bar{x}_{ij}^n = 1\}$ to contain the connections (i,j) related to the active variables. Thus, the implementation of a trust region for the USCCSP involves the addition of the following constraint to the BMP,

$$\sum_{(i,j) \notin X^{n-1}} x_{ij} + \sum_{(i,j) \in X^{n-1}} (1 - x_{ij}) \leq \Delta. \tag{43}$$

There are two different implementations of a trust region that can be employed using constraint (43). The first implementation models Δ as a constant and the second as a variable. Modelling Δ as a constant places a hard limit on the number of changes permitted in each

iteration. This approach is implemented by Santoso *et al.* [22] where it is stated that by using a fixed value of Δ the convergence of the algorithm is not guaranteed. Hence, it is necessary to increase the value of Δ throughout the solution process such that a non-redundant trust region is maintained. This approach is employed in applying a trust region to solve the USCCSP and USCCSP-AA.

A final consideration of the solution approach is the use of transgressing Benders' cuts. The use of transgressing cuts results in a heuristic solution approach. The addition of transgressing cuts in conjunction with a trust region approach is useful in identifying upper bound solutions. To avoid the elimination of the optimal solution by the addition of transgressing cuts, all cuts are removed each time an upper bound solution is identified. Additionally, constraints are included in the master problem to exclude all previously found upper bound solutions. Unfortunately this approach does not guarantee the optimal solution will be found. However, the solution elimination constraints and an increasing trust region aids the exploration of other parts of the feasible region to identify alternative and improved upper bound solutions.

5 Computational results

The computational experiments aim to demonstrate the ability of the USCCSP and USCCSP-AA to identify connected subgraph solutions. An analysis of the solution approach will be presented, discussing the performance improvements from applying Benders' decomposition and a trust region approach. Finally, a review of the connected subgraph solutions and their implications for the analysis of HIV Env sequences will be discussed.

The USCCSP and USCCSP-AA is presented in Section 3 with cost and penalty parameters in the objective function. The purpose of the cost parameter c_{ij} is to force the minimisation of the selected pairs. This is achieved by setting c_{ij} to 1. The penalty parameters, M and N, are included to enforce connectivity between the selected edges. As such, it is necessary to set the parameters M and N to a value sufficiently greater than the expected number of edges selected in the optimal solution. For the presented experiments the parameters are set to 100 and 10000 respectively. Computational experience indicates that varying the values for these parameters has little effect on the optimal solution provided M and N remain sufficiently greater than c_{ij}.

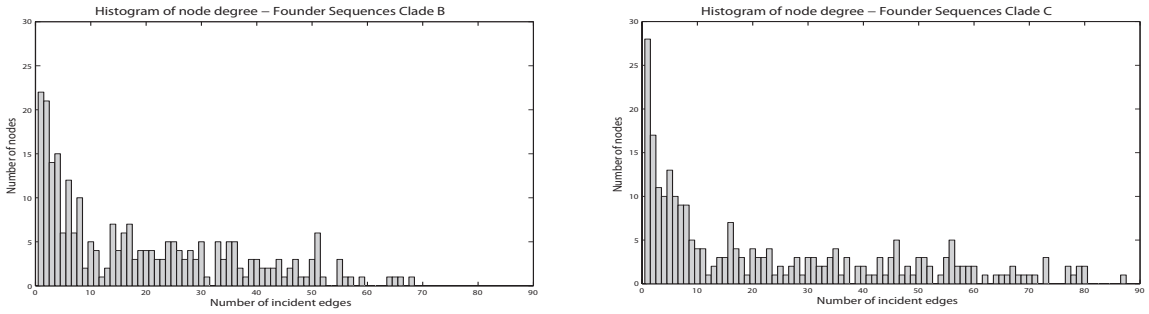
The USCCSP and USCCSP-AA are solved using Cplex 12.4 interfaced through Matlab 2014a using 12 cores with 15GB RAM.

5.1 Solving the USCCSP on the HIV separating pairs networks

The USCCSP is solved to identify a minimum number of covarying pairs forming a completely connected subgraph that separate the founder and chronic HIV Env sequences. Since each pair of positions in the network used to solve the USCCSP is connected by a single edge, the resulting connected subgraph provides a broad analysis of the separating pairs. Two different virus subtypes (clades) are investigated in this paper; as such, two distinct separating pairs networks are constructed.

The data collected for the clade B subtype includes 156 sequences, 78 for both founder and chronic groups. As explained in Section 3, the separating problem of the USCCSP involves selecting covarying pairs that cover the complete set of founder sequences. As such, 78 sequences are contained in the set Q and the resulting separating pairs network consists of 259 nodes and 2495 edges. There are significantly more edges than nodes in the covariance network, resulting in a high degree at each node, which is shown in Figure 3a. Figure 3a demonstrates that a large number of nodes have a small degree, but the majority of nodes have a degree greater than 10. The mean degree for the nodes in the clade B separating pairs network is 19.27 and the median is 15.

For the HIV clade C subtype, the data consists of 55 founder and chronic sequences and hence $|Q| = 55$. The clade C separating pairs network consists of 257 nodes and 3021 edges. Figure 3b presents similar node degree results for this separating pairs network, with the same median degree of 15, but a higher mean degree of 23.51.



(a) Clade B. (b) Clade C.

Figure 3: The number of nodes observed to have a given number of edges incident in the separating pairs network.

5.1.1 Clade B separating pairs network - runtime results

The formulation of the USCCSP using the clade B separating pairs network consists of 167 million variables and 104 million constraints. The vast majority of the variables and constraints are related to the connectivity evaluation. The large number of variables and constraints is a motivation for employing decomposition techniques to improve the problem tractability.

Table 1 presents the runtime results from solving the separating pairs set covering problem and the USCCSP using a standard Benders' decomposition implementation with full cuts and employing a trust region using transgressing cuts for clade B. The solution to the set covering problem provides a lower bound on the USCCSP, which is reported as consisting of 6 separating pairs. The total runtime for this problem is 0.67 seconds. The requirement of identifying a connected subgraph including only the selected separating pairs significantly increases the solution runtime compared to the separating pairs problem of [16]. The standard implementation of Benders' decomposition is unable to solve the USCCSP within 3600 seconds (1 hour). During the solution process, this implementation establishes a lower bound of 7 separating pairs in 1832.32 seconds, but this set of pairs is not connected. Comparatively, the trust region method establishes an upper bound of 7 separating pairs forming a connected subgraph in 54.36 seconds. Unfortunately, since the trust region approach is a heuristic it is not possible to prove the optimality of this solution. This demonstrates the ability of the trust region approach to identify upper bounds quickly and the difficulty in achieving a good lower bound.

		USCCSP	
Clade B Separating Pairs Network	Set Covering Problem	Standard Benders'	Trust Region
Best Lower Bound	6	7	-
Best Upper Bound	6	-	7
Time to Best Lower Bound (seconds)	0.67	1832.32	-
Time to Best Upper Bound (seconds)	0.67	>3600	54.36

Table 1: The best upper and lower bounds and the time to identify each for the clade B separating pairs network using the standard Benders' decomposition implementation and a trust region method.

5.1.2 Clade C separating pairs network - runtime results

While fewer clade C sequences were collected to construct the separating pairs network compared to clade B, the number of variables and constraints are greater with 198 million and 126 million respectively. This increase in problem size is the direct result of a larger number of edges in the separating pairs network and suggests that the many positions in the clade C virus subtype experience related mutations.

The runtime results for solving the set covering problem and the USCCSP for the clade C separating pairs network are presented in Table 2. While the runtime results presented in Table 2 are much shorter than those presented in Table 1, the comparative results are similar. In particular, there is a significant difference in the runtime to solve the set covering problem and the USCCSP. Further, the improvement in the solution runtime from employing the trust region method is also observed. The set covering problem, which involves solving the BMP without any added cuts, is solved to optimality in approximately 0.06 seconds and identifies 3 separating pairs. By contrast, the standard implementation of Benders' decomposition to solve the USCCSP finds the optimal separating pairs connected subgraph solution of 4 pairs in 14.54 seconds. This increased runtime is to be expected, since the complexity of the USCCSP is negatively affected by the inclusion of the connectivity constraints.

While the standard implementation of Benders' decomposition solves the USCCSP in very short runtimes, the results for the clade C separating pairs network still demonstrate the strength of the trust region method. By implementing the trust region for this problem, the first upper bound solution found contains 4 separating pair, which is found after approximately 5.53 seconds. This represents a significant reduction in the solution runtime for the USCCSP.

		USCCSP	
Clade C Separating Pairs Network	Set Covering Problem	Standard Benders'	Trust Region
Best Lower Bound	3	4	4
Best Upper Bound	3	4	-
Time to Best Lower Bound (seconds)	~ 0.06	7.87	-
Time to Best Upper Bound (seconds)	~ 0.06	14.54	5.53

Table 2: The best upper and lower bounds and the time to identify each for the clade C separating pairs network using the standard Benders' decomposition implementation and a trust region method.

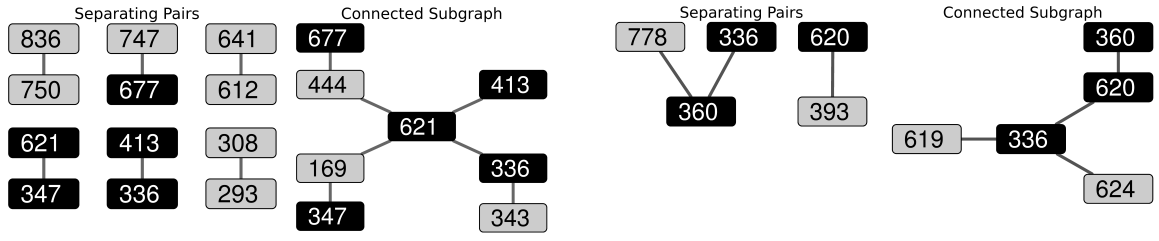
However, the magnitude of this result should not be extrapolated to problems using different networks.

5.1.3 Separating pairs and connected subgraph solutions

Figure 4 presents the separating pairs and connected subgraph solutions for clades B and C. The solutions presented in this figure are the best found solutions from the experiments performed in Sections 5.1.1 and 5.1.2.

The separating pairs problem and the USCCSP identify graphs containing 6 and 8 edges respectively, which are displayed in Figure 4a. This figure demonstrates the lack of connectivity between the pairs after solving the separating pairs problem of Murray *et al.* [17]. By comparison, the solutions to the USCCSP identifies a connected subgraph of the separating pairs network. Given the different problem definitions, it is unlikely that pairs will be common between the solutions to the separating pairs and connected subgraph problems, which is demonstrated in Figure 4a. However, there are three nodes common between the two figures (293, 336, 621), highlighted in black. It is interesting to observe that these three nodes form a connected subgraph in the solution to the USCCSP.

Optimal solutions for the separating pairs problem and the USCCSP for clade C are presented in Figure 4b. The separating pairs solution consists of three pairs, with two pairs connected through node 360. By contrast, the connected subgraph requires four pairs. Similar to the clade B solution, three nodes are common between the two graphs but there are no common separating pairs. Interestingly, it is possible to form an optimal connected subgraph from the separating pairs solution by including edge (336, 620). This suggests the existence of multiple optimal solutions to the separating pairs problem and the USCCSP. In a separating pairs network consisting of 3021 edges, it is likely there exists many subgraphs that solve the



(a) Clade B.

(b) Clade C.

Figure 4: The separating pairs and connected subgraph solutions for the clade B separating pairs network.

USCCSP to optimality.

5.2 Solving the USCCSP-AA

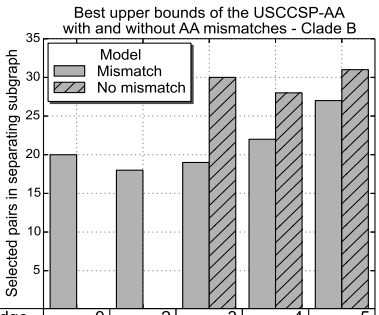
The USCCSP-AA involves selecting a set of separating pairs with only a single AA combination per pair. This problem formulation involves a different network construction compared to that used for the USCCSP. In particular, each of the amino acid combinations observed on a pair of positions must be represented by an edge. This graph construction contains the same number of nodes as the original network, but there is a large increase in the number of edges. Specifically, the number of edges in the multiple amino acid clade B separating pairs network is 12624 and for clade C there are a total of 15663 edges. This is a significant increase in comparison to the original networks containing 2495 and 3021 edges in the clade B and clade C networks respectively. The increase in the number of edges results in a much larger integer program.

It is observed in the computational experiments that the trust region identifies upper bounds very quickly, but very little improvement in the lower bound is achieved. As such, it is only possible to employ Benders' decomposition as an upper bounding heuristic. However, this heuristic approach still provides meaningful information in the pursuit of identifying a small set of important, connected features that separate founder and chronic sequences.

5.2.1 Analysis of USCCSP-AA formulations

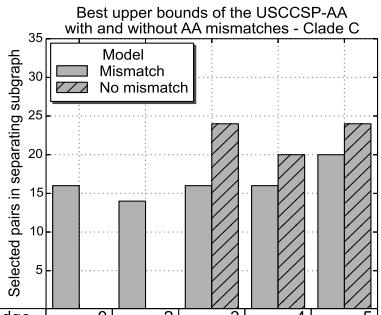
The USCCSP-AA is presented in Section 3.2 with two alternative formulations – permitting and excluding AA mismatches at positions in the subgraph solution. These two formulations are developed with alternative biological interpretations of the connected subgraph. However, the problem complexity is greatly increased by excluding AA mismatches. This is a consequence of identifying each AA combination per pair in the Benders' decomposition subproblem. As such, the added cuts are not as effective, which negatively affects the algorithms rate of convergence.

Figure 5 presents the best found upper bounds for the two alternative formulations and the runtime to achieve this. A maximum runtime of 18000 seconds is used for these experiments. The upper bounds solutions are found by using a trust region approach and applying transgressing cuts. Five different network constructions are used to evaluate the convergence of the algorithm. These are labelled by a value k that the network is constructed by retaining edges where at least k founder sequences are observed. Figure 5 presents experiments where $k \in \{0, 2, 3, 4, 5\}$–the full separating pairs network is given by $k = 0$. The results suggest an



Min seq per edge	0	2	3	4	5
Mismatch time	923.24	773.31	116.92	76.97	1278.1
No mismatch time	--	--	5084.0	532.24	388.52

(a) Clade B.



Min seq per edge	0	2	3	4	5
Mismatch time	56.95	231.41	685.8	20.02	26.74
No mismatch time	--	--	220.45	64.59	45.26

(b) Clade C.

Figure 5: The best upper bounds achieved for the USCCSP-AA either permitting or excluding AA mismatches at each node.

increased complexity when solving the USCCSP-AA by excluding AA mismatches. For both clades, the USCCSP-AA' is unable to identify upper bound connected subgraph solutions for the two largest network constructions. Surprisingly, neither of the problem formulations dominate the runtime to achieve the best upper bound solutions. This is a feature of the trust region approach using transgressing cuts where the upper bound solutions are dependent on previously found solutions.

Small differences are observed in the connected subgraph solutions to the two USCCSP-AA formulations. As presented in Figure 5, the total number of pairs in the upper bound solution is much greater when AA mismatches are excluded. However, there are many AA positions and pairs common between the solutions. Specifically, 44.92% and 41.43% of all positions and edges are observed in both solutions. Some pairs are more commonly observed than others, for example 750-836 is observed in the upper bound solutions for both formulations with networks constructed for clade B using a minimum of 3, 4 or 5 sequences. There are no pairs observed in all clade C solutions, which suggests a higher variability in the mutations. Given the similarities in the solutions between the two formulations the following results will focus primarily the USCCSP-AA with mismatches permitted.

5.2.2 Bound improvements of the USCCSP-AA

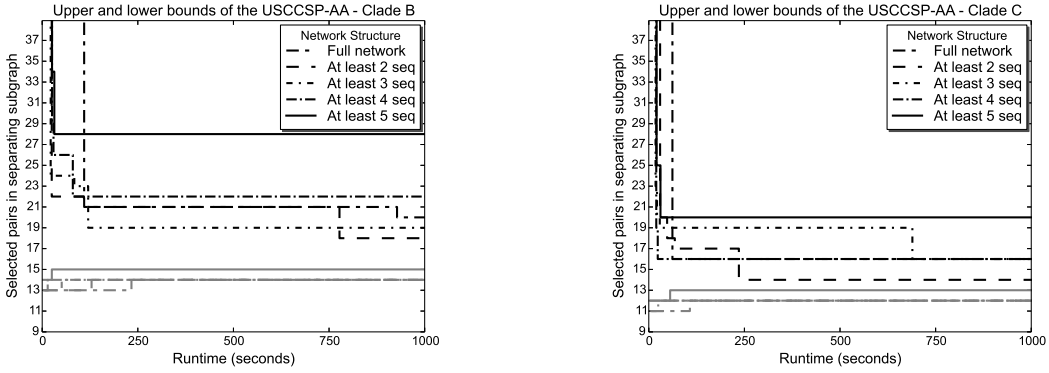
The application of a trust region aids the identification of upper bound solutions. While this is useful from a practical, application perspective, there is no guarantee of optimality. A lower bound solution is given by solving the USCCSP-AA without the trust region and applying full Benders' cuts, as described in Section 4.1.

The improvement in the upper (black) and lower (grey) bound for the USCCSP-AA is displayed in Figure 6. In this figure, the labels "At least k seq" indicates the minimum number of sequences in the network construction. The maximum solution runtime for the USCCSP-AA is set to 18000 seconds. Since no further bound improvements are observed after 1000 seconds for all experiments the time axes in Figure 6 are truncated to improve the presentation.

Figure 6 demonstrates that for all experiments the upper bound is improved very quickly. This illustrates the ability of the trust region approach to identify upper bound solutions in very short runtimes. It is expected that the number of pairs in the upper bound solutions will increase with the minimum number of observed sequences per edge. However, the networks constructed with a minimum of 2 or 3 sequences both achieve better upper bounds than that for the full network. This is also true for the clade C network with the best upper bound achieved with a minimum of 2 sequences per edge. This can be explained as a feature of the trust region approach, where identified upper bound solutions depend on the previously found solutions.

In comparison to the upper bound, very few improvements in the lower bound are achieved. In Figure 6, the lower bound increases by at most 2 pairs across all experiments. Surprisingly, the last improvement in the lower bound is achieved after 233.37 and 107.48 seconds for clade B and C respectively. This suggests that the full Benders' cut described in Section 4.1 is ineffective for this problem formulation. As such, the use of alternative Benders' decomposition acceleration techniques, such as applying Pareto optimal cuts, may be useful for improving the algorithm convergence.

Comparing the upper and lower bound solutions, it is clear that a large optimality gap exists



(a) Clade B. (b) Clade C.

Figure 6: Changes in the upper (black) and lower (grey) bounds for the USCCSP-AA employing a trust region method and network reduction.

for the identified connected subgraphs. This gap appears to be much larger for the clade B networks, compared to that presented for clade C. As such, it is possible that smaller connected subgraph solutions exist for the given covariance networks. While the smaller solutions may exist, the identified upper bound solutions are useful for identifying important features of HIV Env.

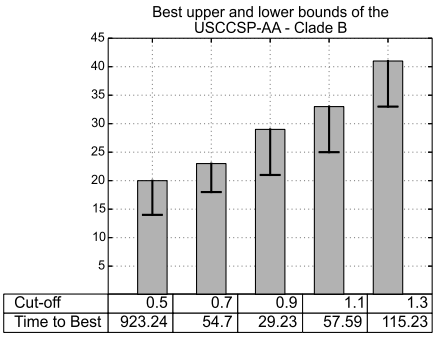
5.2.3 Solving the USCCSP-AA with different cut-off values

The efficacy of Benders' decomposition to identify optimal solutions is impacted by the structure of the USCCSP-AA. This is particularly evident in the results presented in Sections 5.2.1 and 5.2.2, where the optimal solution is not found even with different network constructions. Another network construction method that reduces the size of the graph is to vary the covariance cut-off value. The experiments conducted in the previous sections solve the USCCSP-AA on a network constructed using a cut-off value of 0.5. This value is employed by Aurora *et al.* [4] in constructing covariance networks to analyse hepatitis C virus. A higher cut-off value reduces the number of edges in the resulting network, which is demonstrated in Table 3.

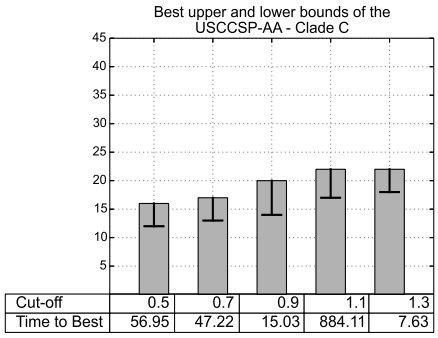
The upper and lower bounds achieve for clade B and C using the different cut-off values is presented in Figure 7. The upper bounds are given by the bars and the lower bound are presented by the lines at the top of each bar. It is observed that as the cut-off value increases, so does the size of the connected subgraph upper bound solution. This is the result of the elimination of more edges in the covariance network reducing the connectivity of the covariance network. Hence, it is less likely to identify edges satisfying the set covering solution in the same region of the graph. Similar to results in Sections 5.2.1 and 5.2.2, there appears to be little correlation between the time required to identify the best upper bound solution and the size of the covariance network. For example, it is observed in Figure 7b the shortest time to find the best upper bound is when a cut-off of 1.3 is used, while the longest is with a cut-off of 1.1. Finally, there appears to be a consistent absolute gap between the upper and lower bound

Cut-off Value	0.5	0.7	0.9	1.1	1.3
Clade B	12,624	5,576	2,544	1,355	779
Clade C	15,663	7,027	3,572	2,142	1,430

Table 3: The number of edges in the covariance graphs for clade B and C constructed using different cut-off values.



(a) Clade B.



(b) Clade C.

Figure 7: The best upper (bars) and lower (lines) bounds for the USCCSP-AA using different cut-off values.

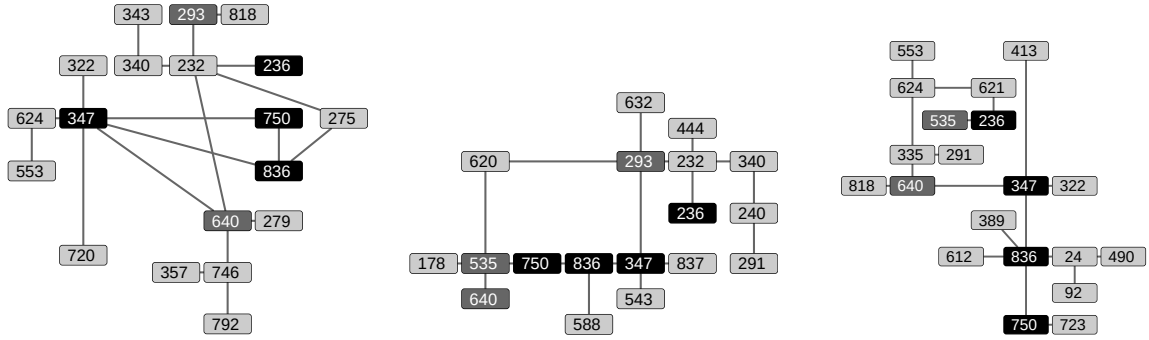
solutions across all experiments in Figure 7. This further emphasises the need for improved cut generation to aid the convergence of the solution approach.

The difficulty in the convergence of the algorithm can be explained by the large number of set covering solutions. Using a cut-off value of 1.3, on average each sequence is observed on 13.87 and 34.35 edges of the clade B and C covariance networks respectively. The prevalence of sequences throughout the network suggests that there are an enormous number of set covering solutions. As such, many Benders' cuts are required to cut off disconnected solutions and improve the lower bound. For comparison, using a cut-off value of 0.5 to construct the covariance networks, for clade B and C each sequence is observed on an average of 222.11 and 374.8 edges. Hence, the ability of the presented solution algorithm to identify good upper bound solution is valuable for large scale instances of the unrooted set covering connected subgraph problem.

5.2.4 Analysis of the separating pairs networks

Figure 6 presents five experiments performed for each clade using different settings for the network reduction. Each of these experiments provide an upper bound solution as a connected subgraph of separating pairs.

Three of the clade B connected subgraph solutions for the experiments presented in Figure 6a are given in Figure 8. The best found upper bound solution is given by the network constructed with a minimum of 2 sequences exhibited per edge, given in Figure 8b. The nodes of particular interest are 236, 347, 750 and 836, which are observed in all upper bounding networks generated for the clade B separating pairs network. In Figure 8 these nodes are highlighted in black. Interestingly, the commonly observed pairs of 347, 750 and 836 are connected in all subgraphs presented in Figure 8, while node 236 is disconnected. This suggests a particular importance

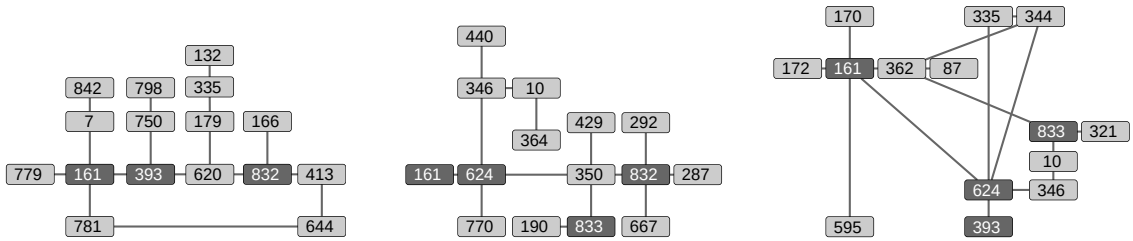


(a) Full network. (b) At least 2 sequences. (c) At least 3 sequences.

Figure 8: The best found separating pairs subgraphs for clade B using different network reduction parameters.

of positions 347, 750 and 836 and the related separating pairs. Expanding the analysis of the commonly selected nodes, the nodes that appear in at least four of the generated networks are highlighted with dark grey. Three grey highlighted nodes of interest are 239, 535 and 640. First, a similar relationship between the black and grey nodes is observed in Figures 8a and 8c. This is given by a direct connection is between 347 and 640 in both. Second, there is a much stronger relationship observed in Figure 8b where all black and grey nodes are exhibited. If node 232 is also considered, then the black and grey nodes form a smaller connected subgraph solution. All of the identified features of the separating pairs connected subgraphs potentially point to other important features of the clade B HIV sequence.

The separating pairs subgraph solutions identified for the clade C separating pairs network are presented in Figure 9. The degree of many of the nodes in the subgraphs is high, indicating the importance of particular AA positions. Surprisingly, there is no node selected in all of the subgraph solutions for the clade C separating pairs network. This has two alternative explanations, i) a more diverse range of mutations occur in the clade C Env sequence compared to clade B, or ii) there is no single position of high importance to the structure of the clade C virus. There are, however, five nodes – 161, 393, 624, 832 and 833 – that are selected in three of the four experiments. In Figure 9 these nodes are highlighted with dark grey, with at least three observed in each of the subgraphs in Figure 9. The connectivity between the grey nodes and the degree of these nodes identifies some interesting structures. First, positions 161 and 624 are directly connected in Figures 9b and 9c, with both having a degree of five in the latter. Further, the grey nodes appear to define a core structure of the subgraphs, particularly in Figure 9a. This structure is interesting given the many branches stemming from it. Further



(a) Full network. (b) At least 2 sequences. (c) At least 3 sequences.

Figure 9: The best found separating pairs subgraphs for clade C using different network reduction parameters.

review of these positions may identify important structures related to the transmission of the clade C virus.

6 Conclusions

This paper presents a novel application of operations research to the analysis of HIV Env sequences. The analysis attempts to construct a connected subgraph of covarying amino acid positions to identify key features related to the transmission of the virus. Two different mathematical models are presented, the second providing a more detailed view of the HIV Env sequences and related separating pairs networks. A critical feature of this analysis is the connectivity requirement between the selected covarying pairs. This requirement is modelled using a network flow problem, which is at the expense of producing very large problem formulations. The complexity of the resulting problem formulation is addressed by employing Benders' decomposition, along with enhancement techniques, to efficiently solve the USCCSP and reduce runtimes for the USCCSP-AA.

The results demonstrate the general performance of the Benders' decomposition solution approach to solve the USCCSP and USCCSP-AA. The models presented in this paper may be applied to any application requiring the construction of a set covering connected subgraph, in particular those without any specified terminal or root nodes. As such, these results aim to demonstrate the general performance of the developed solution approach.

This paper discusses the implication of identifying connected subgraphs in comparison to the set covering problem of Murray *et al.* [17] and the different results achieved using various enhancement techniques. It is demonstrated that the solution to the USCCSP and USCCSP-AA may identify important features for further research in the operation research and microbiology

context.

There are two key areas for further research regarding the work presented in this paper. In the operations research context, identifying other application areas that require the construction of a set covering connected subgraph would aid the further development of the presented approaches. In addition, the results present the use of Benders' decomposition as a heuristic approach to identify good upper bounds to the USCCSP-AA. This presents an area of further research to identify approaches that improve the solution process for the USCCSP-AA: Either through enhancements of the Benders' decomposition approach or the development of alternative techniques. Finally, the separating pairs connected subgraph results attempt to identify key features of the HIV Env sequence as possible vaccine targets. Further analysis of these connected subgraphs may yield more important features to aid the development of a vaccine preventing the transmission of HIV.

Acknowledgements

The authors thank the referees for their helpful suggestions, which improved the presentation of this manuscript. We thank Damian Purcell and Talia Mota for collecting and supplying the HIV sequence data. This research includes computations using the Linux computational cluster Katana supported by the Faculty of Science, UNSW Australia.

References

[1] M.-R. Abrahams, J. A. Anderson, E. E. Giorgi, C. Seoighe, K. Mlisana, L.-H. Ping, G. S. Athreya, F. K. Treurnicht, B. F. Keele, N. Wood, J. F. Salazar-Gonzalez, T. Bhattacharya, H. Chu, I. Hoffman, S. Galvin, C. Mapanje, P. Kazembe, R. Thebus, S. Fiscus, W. Hide, M. S. Cohen, S. A. Karim, B. F. Haynes, G. M. Shaw, B. H. Hahn, B. T. Korber, R. Swanstrom, C. Williamson, for the CAPRISA Acute Infection Study Team, and the Center for HIV-AIDS Vaccine Immunology Consortium. Quantitating the multiplicity of infection with human immunodeficiency virus type 1 subtype c reveals a non-poisson distribution of transmitted variants. *Journal of Virology*, 83(8):3556–3567, 2009.

[2] R. Ahuja, T. Magnanti, and J. Orlin. *Network flows: theory, algorithms, and applications*. Prentice Hall, 1993.

[3] E. Álvarez Miranda, I. Ljubić, and P. Mutzel. The rooted maximum node-weight connected subgraph problem. In C. Gomes and M. Sellmann, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 7874 of *Lecture Notes in Computer Science*, pages 300–315. Springer Berlin Heidelberg, 2013.

[4] R. Aurora, M. Donlin, N. Cannon, J. Tavis, et al. Genome-wide hepatitis C virus amino acid covariance networks can predict response to antiviral therapy in humans. *The Journal of clinical investigation*, 119(1):225–236, 2009.

[5] C. Backes, A. Rurainski, G. W. Klau, O. Müller, D. Stöckel, A. Gerasch, J. Küntzer, D. Maisel, N. Ludwig, M. Hein, A. Keller, H. Burtscher, M. Kaufmann, E. Meese, and H.-P. Lenhof. An integer linear programming approach for finding deregulated subgraphs in regulatory networks. *Nucleic Acids Research*, 40(6):e43, 2012.

[6] R. Carvajal, M. Constantino, M. Goycoolea, J. P. Vielma, and A. Weintraub. Imposing connectivity constraints in forest planning models. *Operations Research*, 61(4):824–836, 2013.

[7] J. M. Conrad, C. P. Gomes, W.-J. van Hoeve, A. Sabharwal, and J. F. Suter. Connections in networks: hardness of feasibility versus optimality. In P. Hentenryck and L. Wolsey, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 4510 of *Lecture Notes in Computer Science*, pages 16–28. Springer Berlin Heidelberg, 2007.

[8] J. M. Conrad, C. P. Gomes, W.-J. van Hoeve, A. Sabharwal, and J. F. Suter. Wildlife corridors as a connected subgraph problem. *Journal of Environmental Economics and Management*, 63(1):1–18, 2012.

[9] B. Dilkina and C. P. Gomes. Solving connected subgraph problems in wildlife conservation. In A. Lodi, M. Milano, and P. Toth, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 6140 of *Lecture Notes in Computer Science*, pages 102–116. Springer Berlin Heidelberg, 2010.

[10] M. T. Dittrich, G. W. Klau, A. Rosenwald, T. Dandekar, and T. Müller. Identifying functional modules in protein-protein interaction networks: an integrated exact approach. *Bioinformatics*, 24(13):i223–i231, 2008.

[11] S. Gnanakaran, T. Bhattacharya, M. Daniels, B. F. Keele, P. T. Hraber, A. S. Lapedes, T. Shen, B. Gaschen, M. Krishnamoorthy, H. Li, J. M. Decker, J. F. Salazar-Gonzalez, S. Wang, C. Jiang, F. Gao, R. Swanstrom, J. A. Anderson, L.-H. Ping, M. S. Cohen, M. Markowitz, P. A. Goepfert, M. S. Saag, J. J. Eron, C. B. Hicks, W. A. Blattner, G. D. Tomaras, M. Asmal, N. L. Letvin, P. B. Gilbert, A. C. DeCamp, C. A. Magaret, W. R. Schief, Y.-E. A. Ban, M. Zhang, K. A. Soderberg, J. G. Sodroski, B. F. Haynes, G. M. Shaw, B. H. Hahn, and B. Korber. Recurrent signature patterns in HIV-1 B clade envelope glycoproteins associated with either early or chronic infections. *PLoS Pathog*, 7(9):e1002209, 2011.

[12] C. P. Gomes, W.-J. van Hoeve, and A. Sabharwal. Connections in networks: a hybrid approach. In L. Perron and M. Trick, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 5015 of *Lecture Notes in Computer Science*, pages 303–307. Springer Berlin Heidelberg, 2008.

[13] B. F. Keele, E. E. Giorgi, J. F. Salazar-Gonzalez, J. M. Decker, K. T. Pham, M. G. Salazar, C. Sun, T. Grayson, S. Wang, H. Li, X. Wei, C. Jiang, J. L. Kirchherr, F. Gao, J. A. Anderson, L.-H. Ping, R. Swanstrom, G. D. Tomaras, W. A. Blattner, P. A. Goepfert, J. M. Kilby, M. S. Saag, E. L. Delwart, M. P. Busch, M. S. Cohen, D. C. Montefiori, B. F. Haynes, B. Gaschen, G. S. Athreya, H. Y. Lee, N. Wood, C. Seoighe, A. S. Perelson, T. Bhattacharya, B. T. Korber, B. H. Hahn, and G. M. Shaw. Identification and characterization of transmitted and early founder virus envelopes in primary hiv-1 infection. *Proceedings of the National Academy of Sciences*, 105(21):7552–7557, 2008.

[14] J. Linderoth and S. Wright. Decomposition algorithms for stochastic programming on a computational grid. *Computational Optimization and Applications*, 24(2-3):207–250, 2003.

[15] T. M. Mota, J. M. Murray, R. J. Center, D. F. J. Purcell, and J. M. McCaw. Application of a case-control study design to investigate genotypic signatures of HIV-1 transmission. *Retrovirology*, 9(1):1–12, 2012.

[16] J. M. Murray, S. J. Maher, T. M. Mota, R. J. Center, K. Suzuki, A. Kelleher, and D. F. J. Purcell. Differentiating founder and chronic HIV sequences. Sumbitted, 2015.

[17] J. M. Murray, R. Moenne-Loccoz, A. Velay, F. Habersetzer, M. Doffol, J.-P. Gut, I. Fofana, M. B. Zeisel, F. Stoll-Keller, T. F. Baumert, and E. Schvoerer. Genotype 1 hepatitis C virus envelope features that determine antiviral response assessed through optimal covariance networks. *PLoS ONE*, 8(6):e67254, 2013.

[18] H. Önal and R. A. Briers. Designing a conservation reserve network with minimal fragmentation: a linear integer programming approach. *Environmental Modeling & Assessment*, 10(3):193–202, 2005.

[19] H. Önal and R. A. Briers. Optimal selection of a connected reserve network. *Operations Research*, 54(2):379–388, 2006.

[20] H. Önal and Y. Wang. A graph theory approach for designing conservation reserve networks with minimal fragmentation. *Networks*, 51(2):142–152, 2008.

[21] A. Ruszczyński. A regularized decomposition method for minimizing a sum of polyhedral functions. *Mathematical Programming*, 35(3):309–333, 1986.

[22] T. Santoso, S. Ahmed, M. Goetschalckx, and A. Shapiro. A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, 167(1):96–115, 2005.