Feasibility of Clustered Spanning Trees by Trees for Domino Intersection Graphs

Avi Levin

Under the supervision of Dr. Nili Beck and Prof. Michal Stern

March 6, 2020

Contents

1 Introduction

Let $H = < V, S >$ be a hypergraph, where V is a set of vertices $\{v_1, \ldots, v_n\}$ and $\mathcal{S} = \{S_1, \ldots, S_m\}$ a set of not necessarily disjoint clusters, $S_i \subseteq V$, $\forall\ 1 \leq i \leq m$. The Clustered Spanning Tree problem, denoted by *CST*, is to find whether a spanning tree exists on V such that each cluster induces a subtree.

If no feasible solution tree exists, we consider on adding vertices to clusters from $\mathcal{S}$ in order to gain feasibility, by finding feasible addition lists with minimum cardinality.

We find a minimum feasible list by looking at the intersection graph of H. The research focuses on intersection graphs with special characteristics, where it is easy to show that there is no feasible solution for the given hypergrah.

The research starts by looking at an intersection graph which is a single chordless cycle problem, then a hypergraph H with a separating edge or a separating path of size three. A significant part of the research is where the intersection graph of H is a d-level t-domino graph, initially treated a smaller domino structured problem up to solving the general case.

An algorithm is provided that finds a possible feasible addition list for the intersection graph of H, denoted by L, whose cardinality is minimum. By adding L to H we can provide a feasible solution tree for the hypergraph.

An important and essential question is whether a feasible solution tree exists for a given instance of the *CST* problem.

Theorem 1.0.1. *([6], [7], [18]) A hypergraph $H = < V, \mathcal{S} >$ has a feasible solution tree if and only if it satisfies the Helly property and its intersection graph is chordal.*

Another approach for the feasibility question of the *CST* problem is presented in [10], where they introduce a technique that requires $O(n^2m)$ time complexity and can handle every instance hypergraph. On the first stage of the algorithm a new weighted graph is constructed, where the weight of each edge in the graph is equal to the number of clusters containing both endpoints of this edge. Next, a maximum spanning tree for this graph is found. A feasible solution for the *CST* problem exists if and only if the weight of this tree is $\sum_{i=1}^{m} |S_i| - m$. Furthermore, the maximum spanning tree offers a feasible solution, when it exists.

[10] introduces feasibility decisions using information derived from the hypergraph and the corresponding intersection graph. When H has a feasible

solution tree T^H, then subtrees of T^H are feasible solution trees for the corresponding induced subproblems, summarized in the following theorem.

Theorem 1.0.2. *([10]) Let $H = < V, \mathcal{S} >$ be a hypergraph with a connected intersection graph and a feasible solution tree T^H. If the intersection graph of $\mathcal{S}' \subset \mathcal{S}$ is connected, then $T^H[V(\mathcal{S}')]$ is a feasible solution tree for $H[\mathcal{S}']$.*

This also proves that when an induced hypergraph does not have a feasible solution tree, neither does H. When the intersection graph contains a cut-edge, [10] prove that deciding whether a feasible solution exists can be based on the decision made independently for each component of the intersection graph after removing the cut-edge. The feasible solution tree for the given hypergraph is constructed using the feasible solution subtrees created for the corresponding subproblems and thus may significantly reduce the required complexity. For the special case where every vertex in V is contained in at most 2 clusters from S, the CST problem has a feasible solution if and only if the corresponding intersection graph is a tree. In all cases a feasible solution tree is offered when it exists.

There are not many studies dealing with the CST problem when there is no feasible solution. In [10], for those instances where no feasible solution tree exists, the paper characterize when adding vertices to exactly one cluster will gain feasibility. This approach finds the appropriate cluster and the vertices that should be added.

Related problems consider different structures of the required solution and different structures of the induced clusters induced.

An important known and most restricted case is where the solution tree is required to be a path, such that every cluster induces a subpath in the solution path. A solution to this problem is testing for the Consecutive Ones Property, denoted by *COP*. A binary matrix has the *COP* when there is a permutation of its rows that gains the 1's consecutive in every column. In [3] Booth and Lueker introduce a data structure called a PQ-tree. PQ-trees can be used to represent the permutations of V in which the vertices of each cluster of S are required to occur consecutively.

Considering the optimization CST problem, where the edges of E have weights and the objective is to find a feasible solution tree with minimum weight. This problem was solved by Korach and Stern in [12] where an optimum solution is found in $O(n^4m^2)$ time complexity, when a feasible solution exists. In addition, an abstraction of the problem using matroids is presented. For the restricted case where each cluster contains at most three vertices, there is a linear time algorithm and a polyhedral description of all feasible solutions.

A special case of the optimization *CST* problem, where the optimum spanning tree solution is required to induce a complete star for each cluster, is presented in [13]. A structure theorem which describes all feasible solutions and a polynomial algorithm for finding an optimum solution are presented, when the intersection graph is connected.

Another related optimization problem, called the clustered-*TSP*-path, arises when the optimum solution tree is required to be a *TSP*-path. The clustered-*TSP*-path is proven to be NP-hard in [11].

A lot of research has been done on the clustered *TSP*-path, where the clusters are disjoint. An heuristic for this problem is presented in [4], a branch and bound algorithm for solving this problem is presented in [14] and bounded-approximation algorithms are presented in [2] and [9]. In [1] the ordered disjoint clustered *TSP* is considered and an approximation algorithm is offered. In [16] a genetic algorithm for solving this problem is presented.

Our problem has many possible applications. One possible application is from the field of bioinformatics. An evolutionary tree, defined in [8], is a tree graph showing the evolutionary relationships among various biological species or other entities (their phylogeny) based upon similarities and differences in their physical or genetic characteristics. Trees are useful in fields of biology such as bioinformatics, systematics and phylogenetics. Each vertex in the tree graph represents one of the species or species with a common origin. Each cluster in $\mathcal{S}$ represents a common feature (e.g. a shared gene or protein). The problem is to find the evolutionary tree, under the assumption that each cluster will create a connected subtree solution. Note that in these trees, the leaves are species that exist today, while some of the internal vertices cannot be directly observed. Using the solution we propose, adding vertices to clusters means adding a certain attribute to one of the species, which is very likely for an unknown species. The purpose of the additions is to find a consistent evolutionary tree with consistency of all the features described by all clusters.

Another possible application is from the field of information security in organizational networks. Analysis of organizational networks [17] is a fairly new area that has been gaining momentum in recent years. This field started with social network analysis. There are many insights that can be learned about the organization from the analysis of the work relations network and its hierarchy. The findings can provide insights about the organizational culture, cooperation between employees and departments, internal cooperation within departments and more. The idea is to look at the organizational network as a graph. Vertices in V represent the people working in the various departments of the organization. A cluster describes groups of people assigned to certain projects. Our goal is to enable a fast and secure flow of information inside each cluster. Therefore, we demand that each subgraph induced by a cluster has to be a subtree. When the problem is not feasible, adding a vertex to a cluster is interpreted as adding an employee to a particular project. We allow such additions to gain a structure of a tree.

2 Definitions

The following definitions will be used throughout the paper.

Definition 2.0.1. *Let $\mathcal{S} = \{S_1, \ldots, S_p\}$ be a family of subsets. S satisfies the* ***Helly Property*** *if the following holds: For every $\mathcal{S}' \subseteq \mathcal{S}$, if each pair members of $\mathcal{S}'$ intersect, then all the members of $\mathcal{S}'$ have a common element. In other words, if every $S_i, S_j \in \mathcal{S}'$ satisfy $S_i \cap S_j \neq \emptyset$ then $\bigcap_{S_i \in \mathcal{S}'} S_i \neq \emptyset$.*

Definition 2.0.2. *Given a hypergraph $H =< V, \mathcal{S} >$ where V is a set of vertices and $\mathcal{S}$ is a set of not necessarily disjoint clusters $\{S_1, \ldots, S_p\}$ of V. The* ***intersection graph of*** $\boldsymbol{\{S_1, \ldots, S_p\}}$*, denoted by* $\boldsymbol{G_{int}(\{S_1, \ldots, S_p\})}$*, is defined to be a graph whose set of nodes is $\{s_1, \ldots, s_p\}$, where s_i corresponds to S_i, and an edge (s_i, s_j) exists if $S_i \cap S_j \neq \phi$. See [15] for more details.*

Definition 2.0.3. *Let $H =< V, \mathcal{S} >$ be a hypergraph where V is a set of vertices and $\mathcal{S}$ is a set of not necessarily disjoint clusters $\{S_1, \ldots, S_p\}$ of V. Let $\mathcal{S}' \subset \mathcal{S}$ be a set of clusters. We define* $\boldsymbol{H[\mathcal{S}']}$ *to be the hypergraph whose vertex set is* $\boldsymbol{V(\mathcal{S}')} = \bigcup_{S_i \in \mathcal{S}'} S_i$ *and its clusters set is $\mathcal{S}'$.*

Definition 2.0.4. *Let $H =< V, \mathcal{S} >$ be a hypergraph where V is a set of vertices and $\mathcal{S}$ is a set of not necessarily disjoint clusters $\{S_1, \ldots, S_p\}$ of V. Let $\mathcal{S}' \subset \mathcal{S}$ be a set of clusters and let $G_{int}(\mathcal{S})$ be the intersection graph of H. The* **induced graph** *$G_{int}(\mathcal{S})[(\bigcup_{S_i \in \mathcal{S}'} S_i)]$ is the intersection graph of $H[\mathcal{S}']$ and therefore can be denoted as $G_{int}(\mathcal{S}')$.*

Definition 2.0.5. *(v_1, v_t) is a* **separating edge** *of a connected graph $G = (V, E)$ if G contains an edge (v_1, v_t) and by removing both vertices v_1 and v_t from G disconnects G into two connected components, whose vertex sets are*

V_a, V_b such that $V = V_a \cup V_b \cup \{v_1, v_t\}$ with $V_a \cap V_b = \emptyset$. However, G remains connected if we remove only one of v_1 or v_t.

Definition 2.0.6. *$(v_1, \ldots, v_t)$, for $t \geq 2$, is a* **separating path** *of a connected graph $G = (V, E)$ if G contains edges (v_i, v_{i+1}) for $i = 1, \ldots, t-1$ and by removing the vertices $\{v_1, \ldots, v_t\}$ from G disconnects G into two connected components, whose vertex sets are V_a, V_b such that $V = V_a \cup V_b \cup \{v_1, \ldots, v_t\}$ with $V_a \cap V_b = \emptyset$. However, G remains connected if we remove only one of v_1 or v_t. A separating path is a general case of a separating edge, where $t \geq 3$.*

Now we focus on definitions regarding feasible addition lists.

Definition 2.0.7. *Let $H = <V, \mathcal{S}>$ be a hypergraph. L is an* **addition list of *H*** *if L is a list of pairs $\{(v_1, S_{i_1}), \ldots, (v_k, S_{i_k})\}$ with $v_j \notin S_{i_j}$, such that adding every vertex v_j to cluster S_{i_j}, creates a new instance of the hypergraph denoted by $H + L$. If the new hypergraph $H + L$ has a feasible solution tree we say that L is a* **feasible addition list of *H***.

Notation 2.0.8. *Let $H = <V, \mathcal{S}>$ be a hypergraph. We denote L' to be an addition list for the induced cluster set $\mathcal{S}' \subseteq \mathcal{S}$. If the induced hypergraph $H[\mathcal{S}'] + L'$ has a feasible solution tree we say that L' is a* **feasible addition list of $H[\mathcal{S}']$**.

Definition 2.0.9. *Let $H = <V, \mathcal{S}>$ be a hypergraph and $L = \{(v_1, S_{i_1}), \ldots, (v_k, S_{i_k})\}$ an addition list. Denote $\boldsymbol{L[\mathcal{S}']}$ to contain all pairs (v_j, S_{i_j}) such that $(v_j, S_{i_j}) \in L, v_j \in V(\mathcal{S}'), S_{i_j} \in \mathcal{S}'$. Also denote $L(S_i) = \{v | (v, S_i) \in L\}$.*

Definition 2.0.10. *Let $H = <V, \mathcal{S}>$ be a hypergraph. Define $\boldsymbol{ML(H)} = argmin\{|L(H)| | L$ is a feasible addition list$\}$ to be a* ***feasible addition list whose cardinality is minimum*** *for H.*

Let C be a chordless cycle of size $m \geq 4$. Define $\boldsymbol{ML(C)} = argmin\{|L(C)| | L$ is a feasible addition list$\}$ to be a ***feasible addition list whose cardinality is minimum*** *for C.*

Definition 2.0.11. *Let $G = (V, E)$ be a chordless cycle. Define* ***CLE(G), chord addition list of edges****, to be a feasible addition list whose addition to G achieves chordality.*

Definition 2.0.12. *Let $G = (V, E)$ be a chordless cycle, define* ***MCLE(G)*** *$= argmin\{|CLE(G)| | CLE(G)$ is a chord addition list of edges$\}$ to be a* ***minimum cardinality chord addition list of edges*** *for G.*

3 Chordless cycles

3.1 Core Properties

The following theorems are regarding to minimum cardinality of feasible addition lists within chordless cycles.

Theorem 3.1.1. *Let C be a chordless cycle of size $m \geq 3$, then $|MCLE(C)| = m-3$, which correlates with the number of created triangles in the cycle which is $m-2$.*

Proof. We prove the theorem by induction on m.

A cycle of size $m = 3$ is also a triangle. In this case, the number required chord addition is zero and the result is exactly one triangle. When C is a cycle of size $m = 4$, we add one chord to split C into two triangles.

Assume the theorem is correct for cycles of $size < m$ and let C be a chordless cycle of size m. By adding a chord into C we split the cycle into two secondary cycles, denoted C_1 and C_2. We denote by t and $m+2-t$ the number of nodes in C_1 and C_2 respectively.

According to the induction hypothesis, for C_1 we have to add $t-3$ chords and get $t-2$ triangles. For C_2 we have to add $m+2-t-3$ chords and get $m+2-t-2$ triangles. The total amount of chords addition is $1+(t-3)+(m+2-t-3) = m-3$. Furthermore, we have created $(t-2)+(m+2-t-2) = m-2$ triangles. □

Property 3.1.2. *A triangle in G_{int} indicates an intersction between every two clusters within the triangle. To satisfy the Helly Property there must be at least one vertex in the intersection of all three clusters. Thus, every created triangle requires addition of at least one vertex to one of the clusters triangle.*

Theorem 3.1.3. *Let $H =< V, \mathcal{S} >$ be a hypergraph whose intersection graph $G_{int}(H)$ is a chordless cycle of size $m \geq 4$, denoted as C, then $|ML(C)| \geq$ the number of created triangles by adding $MCLE(G_{int}(H))$ to $G_{int}(H)$.*

Proof. According to Theorem 3.1.1, the number of created triangles is $m-2$. Since every created triangle must satisfy the Helly Property, according to Property 3.1.2, for every created triangle, at least one vertex must be added to one of the clusters within the triangle. Therefore, $|ML(C)| \geq m-2$. □

Theorem 3.1.4. *Let $H =< V, \mathcal{S} >$ be hypergraph whose intersection graph $G_{int}(H)$ is a chordless cycle of size $m \geq 4$, denoted as C, then $|ML(C)| = m - 2$.*

Proof. According to Theorem 3.1.3, $|ML(C)| \geq m - 2$. To prove equality, consider adding the following addition: choose a vertex $v^* = S_1 \cap S_2$ and add it to clusters $\{S_3, \ldots, S_m\}$. Initially the vertex $v^* \notin \{S_3, \ldots, S_m\}$ since $G_{int}(H)$ is a chordless cycle. In this case, the total vertices additions is $m-2$. A star spanning V whose centre is v^* is a feasible solution tree for H. Since $m - 2$ additions are enough then $|ML(C)| = m - 2$. □

Theorem 3.1.5. *Let C be a chordless cycle of size $m \geq 4$, in $MCLE(C)$ there are no overlapping chords and hence it induces a planar graph.*

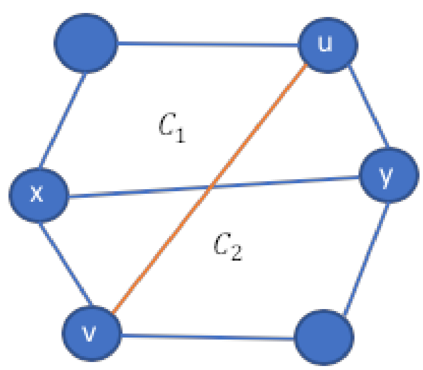

Figure 1: A cycle graph with overlapping chords

Proof. Suppose by contradiction that $MCLE(C)$ contains overlapping chords, denoted (x, y) and (u, v), as the overlapping chords in C, see figure 1.

Chord (x, y) splits C into two secondary cycles, denoted C_1 and C_2. Let m_1 and m_2 be the number of nodes in C_1 and C_2, respectively, not including nodes x and y. Hence, the total number of nodes in C is $m = m_1 + m_2 + 2$.

Chord (u, v) does not split C_1 or C_2 and hence, according to Theorem 3.1.1, we need to add at least $m_1 + 2 - 3$ and $m_2 + 2 - 3$ chords for C_1 and C_2, respectively, to achieve chordality. Therefore the total number of edges we add are $(m_1 + 2 - 3) + (m_2 + 2 - 3) + 2 = m_1 + m_2 = |C| - 2$, which contradicts Theorem 3.1.1. □

3.2 Separating edges

Theorem 3.2.1. *Let $H =< V, \mathcal{S} >$ be a hypergraph whose intersection graph $G_{int}(H)$ contains a separating edge (s_1, s_t) whose removal creates two connected components corresponding to the clusters collections $\mathcal{S}_a, \mathcal{S}_b$.*

If $H_a = H[\mathcal{S}_a \cup \{S_1, S_t\}]$ and $H_b = H[\mathcal{S}_b \cup \{S_1, S_t\}]$ have feasible solution tree then H has a feasible solution tree.

Proof. Since $H_a = H[\mathcal{S}_a \cup \{S_1, S_t\}]$ has a feasible solution tree, $G_{int}(H_a)$ is chordal and H_a satisfies the Helly Property. Similarly, $H_b = H[\mathcal{S}_b \cup \{S_1, S_t\}]$ has a feasible solution trees, therefore $G_{int}(H_b)$ is chordal and H_b satisfies the Helly Property.

Assume that $G_{int}(H)$ contains a chordless cycle, denoted by C, then there are three options for the location of C:

- C is contained in $G_{int}(H_a)$,
- C is contained in $G_{int}(H_b)$,
- C is contained in $G_{int}(H_a) \cup G_{int}(H_b)$.

Since $G_{int}(H_a)$ and $G_{int}(H_b)$ are chordal, the only possible option is that C is contained in $G_{int}(H_a) \cup G_{int}(H_b)$. Since $G_{int}(H_a) \cup G_{int}(H_b)$ contains (s_1, s_t) as a chord, C is not chordless, a contradiction to the assumption. Hence, $G_{int}(H) = G_{int}(H_a) \cup G_{int}(H_b)$ is chordal.

Assume there is a subset of clusters $\{S_{i,1}, \ldots, S_{i,e}\} \subseteq \mathcal{S}$ such that $S_{i,j} \cap S_{i,k} \neq \emptyset$ for every $j \neq k$. Since (s_1, s_t) is a separating edge in $G_{int}(H)$, $\mathcal{S}_a \cap \mathcal{S}_b = \emptyset$ and therefore either $\{S_{i,1}, \ldots, S_{i,e}\} \subseteq \mathcal{S}_a \cup \{S_1, S_t\}$ or $\{S_{i,1}, \ldots, S_{i,e}\} \subseteq \mathcal{S}_b \cup \{S_1, S_t\}$. Since H_a and H_b satisfy the Helly Property, hence $S_{i,1} \cap \ldots \cap S_{i,e} \neq \emptyset$. Therefore, H satisfies the Helly Property.

Since $G_{int}(H)$ is chordal and H satisfies the Helly Property, according to Theorem 1.0.1, H has a feasible solution tree. □

Theorem 3.2.2. *Let $H =< V, \mathcal{S} >$ be a hypergraph whose intersection graph $G_{int}(H)$ contains a separating edge (s_1, s_t) whose removal creates two connected components corresponding to the clusters collections $\mathcal{S}_a, \mathcal{S}_b$.*

If L_a is a feasible addition list for $H_a = H[\mathcal{S}_a \cup \{S_1, S_t\}]$ and L_b is a feasible addition list for $H_b = H[\mathcal{S}_b \cup \{S_1, S_t\}]$, then $L_a \cup L_b$ is a feasible addition list for H.

Proof. Since L_a and L_b are feasible addition lists for H_a and H_b respectively, then $H_a + L_a$ and $H_b + L_b$ have feasible solution trees. According to Theorem 3.2.1, $H + (L_a \cup L_b)$ has a feasible solution tree and therefore $L_a \cup L_b$ is a feasible addition list for H. $\square$

Remark 3.2.3. *Let $H =< V, \mathcal{S} >$ be a hypergraph whose intersection graph $G_{int}(H)$ contains a separating edge (s_1, s_t) whose removal creates two connected components corresponding to the clusters collections $\mathcal{S}_a, \mathcal{S}_b$. For every $\{S_{i,1}, \ldots, S_{i,e}\} = \mathcal{S}' \subseteq \mathcal{S}$ a subset of clusters, then $(H + L)[\mathcal{S}'] = H[\mathcal{S}'] + L[\mathcal{S}']$ since the clusters in $(H+L)[\mathcal{S}']$ are equal to $\{S_{i_1} \cup L(S_{i,1}), S_{i,2} \cup L(S_{i,2}), \ldots, S_{i,e} \cup L(S_{i,e})\}$.*

Theorem 3.2.4. *Let $H =< V, \mathcal{S} >$ be a hypergraph whose intersection graph $G_{int}(H)$ contains a separating edge (s_1, s_t) whose removal creates two connected components corresponding to the clusters collections $\mathcal{S}_a, \mathcal{S}_b$.*

If H does not have a feasible solution tree and $L = L[\mathcal{S}]$, then $L[\mathcal{S}']$ is a feasible addition list for $H[\mathcal{S}']$, for any $\mathcal{S}' \subseteq \mathcal{S}$.

Proof. Since L is a feasible addition list for H, then $H + L$ has a feasible solution tree. According to remark 3.2.3, $(H+L)[S'] = H[S'] + L[S']$. Hence, according to Theorem 1.0.2, $(H+L)[S']$ has a feasible solution tree, therefore $L[\mathcal{S}']$ is a feasible addition list for $H[\mathcal{S}']$. $\square$

Lemma 3.2.5. *Let $H =< V, \mathcal{S} >$ be a hypergraph whose intersection graph $G_{int}(H)$ contains a separating edge (s_1, s_t) whose removal creates two connected components corresponding to the clusters collections $\mathcal{S}_a, \mathcal{S}_b$.*

If H does not have a feasible solution tree and $L = ML(\mathcal{S})$, $L_a = L[\mathcal{S}_a \cup \{s_1, s_t\}]$, $L_b = L[\mathcal{S}_b \cup \{s_1, s_t\}]$, then $L_a \cap L_b = \emptyset$

Proof. Assume by contradiction that there exists a pair $(v^*, S^*) \in L_a \cap L_b$. Since $(v^*, S^*) \in L_a$ then $v^* \in V(\mathcal{S}_a) \cup S_1 \cup S_t$ and $S^* \in \mathcal{S}_a \cup \{S_1, S_t\}$. Similarly, $v^* \in V(\mathcal{S}_b) \cup S_1 \cup S_t$ and $S^* \in \mathcal{S}_b \cup \{S_1, S_t\}$.

Since (s_1, s_t) is a separating edge, $\mathcal{S}_a \cap \mathcal{S}_b = \emptyset$ and hence $S^* \in \{S_1, S_t\}$. In addition, since $V(\mathcal{S}_a) \cap V(\mathcal{S}_b) = \emptyset$ we can conclude that $v^* \in S_1 \cup S_t$.

In this case, we are supposed to add a vertex from S_1 to S_t or from S_t to S_1. But this is not an addition of an edge to $G_{int}(H)$, which contradicts the existance of (v^*, S^*) and therefore $L_a \cap L_b = \emptyset$.

$\square$

Lemma 3.2.6. *Let $H =< V, \mathcal{S} >$ be a hypergraph whose intersection graph $G_{int}(H)$ contains a separating edge (s_1, s_t) whose removal creates two connected components corresponding to the clusters collections $\mathcal{S}_a, \mathcal{S}_b$.*

If H does not have a feasible solution tree, then $|ML(\mathcal{S})| \geq |ML(\mathcal{S}_a \cup \{s_1, s_t\})| + |ML(\mathcal{S}_b \cup \{s_1, s_t\})|$.

Proof. Let $L = ML(\mathcal{S})$ be a minimum cardinality feasible addition list for H. Then, according to Theorem 3.2.4, $L_a = L[\mathcal{S}_a \cup \{s_1, s_t\}], L_b = L[\mathcal{S}_b \cup \{s_1, s_t\}]$ are feasible addition lists for $H_a = H[S_a \cup \{s_1, s_t\}], H_b = H[S_b \cup \{s_1, s_t\}]$, respectively. Hence, $|L| \geq |L_a \cup L_b| = |L_a| + |L_a| - |L_a \cap L_b|$. According to Lemma 3.2.5, $L_a \cap L_b = \emptyset$ and therefore $|L| \geq |L_a| + |L_a|$.

According to definition 2.0.10, $|L_a| \geq |ML(\mathcal{S}_a \cup \{s_1, s_t\})|$ and $|L_b| \geq |ML(\mathcal{S}_b \cup \{s_1, s_t\})|$. Thus $|ML(\mathcal{S})| \geq |L_a| + |L_b| \geq |ML(\mathcal{S}_a \cup \{s_1, s_t\})| + |ML(\mathcal{S}_b \cup \{s_1, s_t\})|$. □

Lemma 3.2.7. *Let $H =< V, \mathcal{S} >$ be a hypergraph whose intersection graph $G_{int}(H)$ contains a separating edge (s_1, s_t) whose removal creates two connected components corresponding to the clusters collections $\mathcal{S}_a, \mathcal{S}_b$.*

If H does not have a feasible solution tree, then $|ML(\mathcal{S})| \leq |ML(\mathcal{S}_a \cup \{s_1, s_t\})| + |ML(\mathcal{S}_b \cup \{s_1, s_t\})|$.

Proof. Let $H_a = H[\mathcal{S}_a \cup \{s_1, s_t\}]$ and $H_b = H[\mathcal{S}_b \cup \{s_1, s_t\}]$. Choose $L_a = ML[\mathcal{S}_a \cup \{s_1, s_t\}]$ and $L_b = ML[\mathcal{S}_b \cup \{s_1, s_t\}]$. According to Theorem 3.2.2, $L_a \cup L_b$ is a feasible addition list for H. According to Lemma 3.2.5, $L_a \cap L_b = \emptyset$ and therefore $|L_a \cup L_b| = |L_a| + |L_b| - |L_a \cap L_b| = |L_a| + |L_b|$.

According to definition 2.0.10, $|ML(\mathcal{S})| \leq |L_a \cup L_b| = |L_a| + |L_b|$. Hence $|ML(\mathcal{S})| \leq |ML(\mathcal{S}_a \cup \{s_1, s_t\})| + |ML(\mathcal{S}_b \cup \{s_1, s_t\})|$. □

Corollary 3.2.8. *Let $H =< V, \mathcal{S} >$ be a hypergraph such that $G_{int}(H)$ contains a separating edge (s_1, s_t) whose removal creates two connected components corresponding to the cluster collections $\mathcal{S}_a, \mathcal{S}_b$.*

If H does not have a feasible solution tree, then $|ML(\mathcal{S})| = |ML(\mathcal{S}_a \cup \{s_1, s_t\})| + |ML(\mathcal{S}_b \cup \{s_1, s_t\})|$

3.3 Separating Paths of Size Three

Let $G =< V, E >$ be a graph with a separating path (v_1, v_2, v_3), whose removal creates two connected components corresponding to vertices sets V_a and V_b. Denote $G_a = G[V_a \cup \{v_1, v_2, v_3\}]$, $G_b = G[V_b \cup \{v_1, v_2, v_3\}]$, $E_a =$

$CLE(G_a)$, $E_b = CLE(G_b)$ and $E_G = CLE(G)$. Note that E_a, E_b, E_G may be empty.

Lemma 3.3.1. *Let $G =< V, E >$ be a graph with a separating path (v_1, v_2, v_3), whose removal creates two connected components corresponding to vertices sets V_a and V_b, then $|E_a \cap E_b| \leq 1$.*

Proof. Since $V_a \cap V_b = \emptyset$ and E_a, E_b touches only vertices from $V_a \cup \{v_1, v_2, v_3\}$ and $V_b \cup \{v_1, v_2, v_3\}$ respectively, then $E_a \cap E_b$ touches only vertices from $\{v_1, v_2, v_3\}$.

According to the definition of a separating path, G includes edges (v_1, v_2) and (v_2, v_3), therefore $E_a \cap E_b$ may only include edge (v_1, v_3). □

Theorem 3.3.2. *Let $G =< V, E >$ be a graph with a separating path (v_1, v_2, v_3) whose removal creates two connected components corresponding to vertices sets V_a, V_b, then $E_a \cup E_b \cup (v_1, v_3)$ is a chord addition list of edges for G.*

Proof. Assume by contradiction that after adding $E_a \cup E_b \cup (v_1, v_3)$ to G it contains a chordless cycle C, there are three options for the location of *C:*

- *C is contained in $V_a \cup \{v_1, v_2, v_3\}$, which contradicts the definition of E_a.*
- *C is contained in $V_b \cup \{v_1, v_2, v_3\}$, which contradicts the definition of E_b.*
- *C is contained in $V_a \cup V_b \cup \{v_1, v_2, v_3\}$ with at least one node in V_a and at least one node in V_b. Hence, C contains at least 2 vertices from (v_1, v_2, v_3). In this case, one of the edges $(v_1, v_2), (v_2, v_3)$ or (v_1, v_3) is a chord in C. Contradicting the assumption that C is chordless.*

□

Lemma 3.3.3. *Let $G = (V, E)$ and let $E_G = CLE(G)$, then for $V' \subseteq V$, $E_G[V'] = \{e = (v_i, v_j) | e \in E_G, v_i, v_j \in V'\}$ is a chord addition list for $G[V']$.*

Proof. Suppose by contradiction that $E_G[V']$ is not a $CLE(G[V'])$. This means that after adding $E_G[V']$ to $G[V']$, $G[V']$ contains a chordless cycle C *whose vertices are contained in V'.*

Since $E_G[V'] \subseteq E_G[C]$ then G contains a chordless cycle C. Since $E_G = CLE(G)$, by adding $E_G[C]$ to C it induces a chordal graph on C, contradicting the assumption that C is a chordless cycle in $G[V']$. □

Lemma 3.3.4. *Let $G =< V, E >$ be a graph with a separating path (v_1, v_2, v_3) whose removal creates two connected components corresponding to vertices set V_a, V_b and let $E_G = CLE(G)$, then $|E_G| \geq |MCLE(G_a)| + |MCLE(G_b)| - 1$.*

Proof. Let $E_{G_a} = \{e = (v_i, v_j) | e \in E_G, v_i, v_j \in V_a \cup \{v_1, v_2, v_3\}\}$ and $E_{G_b} = \{e = (v_i, v_j) | e \in E_G, v_i, v_j \in V_b \cup \{v_1, v_2, v_3\}\}$.

According to lemma 3.3.3, $E_{G_a} = CLE(G_a)$ and $E_{G_b} = CLE(G_b)$. In this case $E_G \supseteq E_{G_a} \cup E_{G_b}$, therefore, $|E_G| \geq |E_{G_a} \cup E_{G_a}| = |E_{G_a}| + |E_{G_a}| - |E_a \cap E_b|$. According to lemma 3.3.1, $|E_{G_a} \cap E_{G_b}| \leq 1$ and therefore, $|E_G| \geq |E_{G_a}| + |E_{G_b}| - 1$. According to Definitions 2.0.11 and 2.0.12, $|E_{G_a}| \geq |MCLE(G_a)|$ and $|E_{G_b}| \geq |MCLE(G_b)|$. Hence $|E_G| \geq |MCLE(G_a)| + |MCLE(G_b)| - 1$.

□

Theorem 3.3.5. *Let $G =< V, E >$ be a chordless graph with a separating path (v_1, v_2, v_3) whose removal creates two connected components corresponding to vertices set V_a and V_b and let $E_a = CLE(G_a)$ and $E_b = CLE(G_b)$ such that $(v_1, v_3) \in E_a \cap E_b$ then $|MCLE(G)| = |MCLE(G_a)| + |MCLE(G_b)| - 1$*

Proof. According to Theorem 3.3.2, $E_a \cup E_b \cup (v_1, v_3)$ is a chord addition list of edges for G. Hence, $|MCLE(G)| \leq |E_a \cup E_b \cup (v_1, v_3)|$.

Since $(v_1, v_3) \in E_a \cap E_b$, $|MCLE(G)| \leq |E_a \cup E_b| = |E_a| + |E_b| - 1 \leq |MCLE(G_a)| + |MCLE(G_b)| - 1$.

According to lemma 3.3.4, $|MCLE(G)| \geq |MCLE(G_a)| + |MCLE(G_b)| - 1$.

Hence $|MCLE(G)| = |MCLE(G_a)| + |MCLE(G_b)| - 1$.

□

4 Domino Graphs

Definition 4.0.1. *A* **d-level t-domino** *(see figure 2), is a family of graphs that has d uncontained chordless cycles, denoted by $C_1, \ldots, C_d$, for $d \geq 1$.*

The graph satisfies:

- *$C_i \cap C_{i+1}$ for $i = 1, \ldots, d-1$, is a path which contains t nodes, for $t \geq 2$.*
- *$C_i \cap C_j = \emptyset$ if $|i - j| > 1$, for $i, j = 1, \ldots, d$.*
- *$|C_i| \geq 2t$, for $i = 1, \ldots, d$.*

*We denote the nodes in $C_i \cap C_{i+1}$ **path nodes** and mark them $s_{i,k}$, for $i = 1, \ldots, d-1$ and $k = 1, \ldots, t$, representing clusters $S_{i,k}$.*

For simplicity of notation, for $t \geq 3$, we mark two additional paths $\{S_{0,1}, \ldots, S_{0,t}\}$ in $C_1 \backslash C_2$ and $\{S_{d,1}, \ldots, S_{d,t}\}$ in $C_d \backslash C_{d-1}$.

All the other nodes are denoted as regular nodes, $r_{i,k}$, for $i = 1, \ldots, d$ and $k = 1, \ldots, K_i$, where cycle C_i contains K_i regular nodes, representing clusters $R_{i,k}$. The amount of regular nodes is $K_i = |C_i| - 2t$, for $i = 1, \ldots, d$.

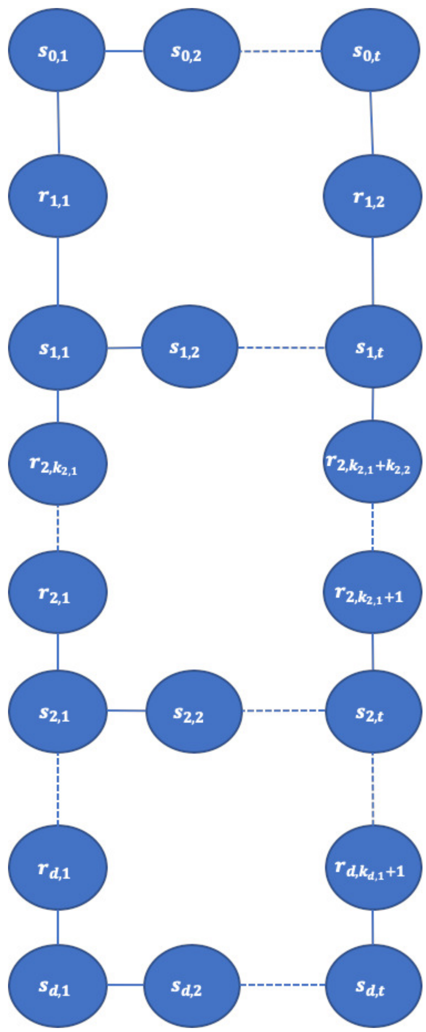

Figure 2: A d-level t-domino graph

4.1 A d-level 2-domino graph

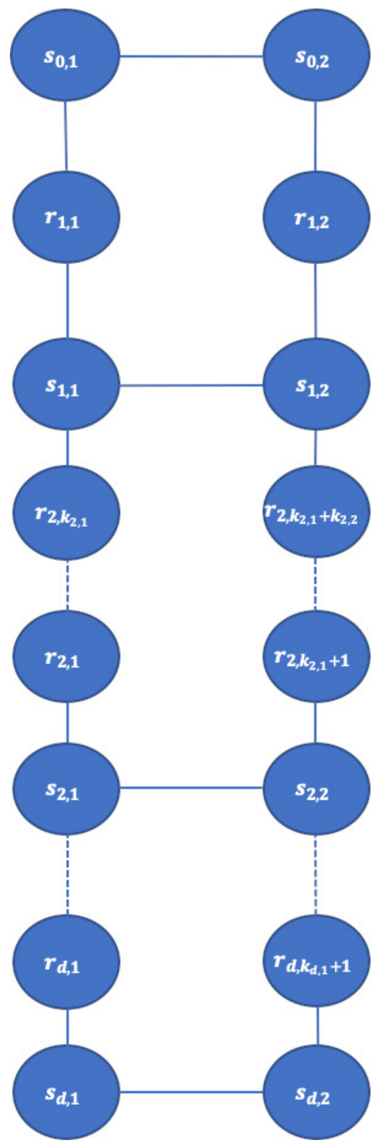

Figure 3: A d-level 2-domino graph

Theorem 4.1.1. *Let $H = < V, \mathcal{S} >$ be a hypergraph with a d-level 2-domino intersection graph $G_{int}(H)$, with $C_1, \ldots, C_d$, then $|ML(H)| = \sum_{i=1}^{d}(|C_i| - 2)$.*

Proof. We prove the theorem by induction on d.

For d=1, $G_{int}(H)$ is a single cycle C_1, according to Theorem 3.1.4, $|ML(H) = |C_1| - 2$.

Assume by induction that the theorem's claim is correct for $d-1$, and provide it for d.

Let $H' = H[C_i \cup \cdots \cup C_{d-1}]$, its intersection graph is a $(d-1) - level$ 2-domino graph and according to the induction hypothesis $|ML(H')| = \sum_{i=1}^{d-1}(|C_i| - 2)$. According to Theorem 3.1.4, $|ML(C_d)| = |C_d| - 2$. Edge (s_{d_1}, s_{d_t}) is a separating edge, dividing H to H' and C_d. According to corollary 3.2.8, $|ML(H)| = |ML(H')| + |ML(C_d)| = \sum_{i=1}^{d}(|C_i| - 2)$.

□

4.1.1 Algorithm d-level 2-domino

In this section we propose an algorithm that finds a minimum cardinality fesiable addition list for $C_1, \ldots, C_d$ in a d-level 2-domino graph. We denote $s_{i,1}$ and $s_{i,2}$ the nodes in $C_i \cap C_{i+1}$, for $i = 1, \ldots, d$.

Given a hypergraph H whose intersection graph $G_{int}(H)$ is a d-level 2-domino, denote $CL = \{C_1, \ldots, C_d\}$, a list of chordless cycles such that $C_i \cap C_{i+1} \neq \emptyset$, for $i = 1, \ldots, d-1$.

.

The algorithm is based on 4 main parts:

- Algorithm: FindFeasibleSetForIntersectionGraph, described in figure 4.
  Input: A hypergraph H and its intersection graph $G_{int}(H)$.
  Output: $ML(H)$.
  Description: Given a hypergraph H, whose intersection graph $G_{int}(H)$ is a d-level 2-domino graph, with no feasible solution tree. The function finds L, a minimum cardinality feasible addition list.

- Procedure: FindDominoCyclesOrder described in figure 5.
  Input: A hypergraph H and its intersection graph $G_{int}(H)$.
  Output: $\{C_1, \ldots, C_d\}$ - list of ordered cycles.
  Description: Given a hypergraph H, whose intersection graph $G_{int}(H)$ is a d-level 2-domino. The function returns a list of chordless cycles $C_1, \ldots, C_d$, such that $C_i \cap C_{i+1} \neq \emptyset$.

- Procedure: ChordlessCycles described by Dias,Castonguay,Longo and Jradi as described in [5]
  Input: A hypergraph H and its intersection graph $G_{int}(H)$
  Output: A list of all chordless cycles in $G_{int}(H)$.

- Procedure: FindFeasibleSetForCycle described in figure 6.
  Input: $\{s_{i,1}, s_{i,2}\}$ and $\{r_{i,1}, \ldots, r_{i,k}\}$, the nodes of the chordless cycle.
  Output: $ML(C_i)$.
  Description: Given a chordless cycle C with two path nodes $\{s_{i,1}, s_{i,2}\}$ and regular nodes $\{r_{i,1}, \ldots, r_{i,k}\}$. The function return $L(C)$ a minimum cardinality feasible addition list for C.

FindFeasibleSetForIntersectionGraph

input

A hypergraph H and its intersection graph G_{int}
which is a d-level 2-domino.

returns

A minimum cardinality feasible addition list L.

begin

$CL = FindDominoCyclesOrder(G_{int}(H))$, *in Figure 5.*

Initialize L to be empty and d to be $|CL|$.

for $(i \in 1, \ldots, d-1)$

  if $i \geq 2$

   then $\{S_{i-1,1}, S_{i-1,2}\} = CL_{i-1} \cap CL_i$

   else *Choose 2 continuous clusters* $\{S_{i-1,1}, S_{i-1,2}\}$

  end if

  $\{S_{i,1}, S_{i,2}\} = CL_i \cap CL_{i+1}$

  $\{R_{i_1}, \ldots, R_{i,k_{i,1}+k_{i,2}}\} = CL_i \setminus CL_{i+1}$

  such that:

   $R_{i,1} \cap S_{i,1} \neq \emptyset$

   $R_{i,k_{i,1}+1} \cap S_{i,2} \neq \emptyset$

   $R_{i,k_{i,1}} \cap S_{i-1,1} \neq \emptyset$

   $R_{i,k_{i,1}+k_{i,2}} \cap S_{i-1,2} \neq \emptyset$

  $L = L \cup findFeasibleSetForCycle(\{S_{i-1,1}, S_{i-1,2}\}, \{S_{i,1}, S_{i,2}\},$
   $\{R_{i,1}, \ldots, R_{i,k_i}\})$, *in Figure 6.*

end for

$\{S_{d-1,1}, S_{d-1,2}\} = CL_{d-1} \cap CL_d$

Choose 2 continuous clusters $\{S_{d,1}, S_{d,t}\}$

$\{R_{d,1}, \ldots, R_{d,k_{d,1}+k_{d,2}}\} = CL_d \setminus (CL_{d-1} \cup \{S_{d,1}, S_{d,t}\})$

such that:

  $R_{d,1} \cap S_{d,1} \neq \emptyset$

  $R_{d,k_{d,1}+1} \cap S_{d,2} \neq \emptyset$

  $R_{d,k_{d,1}} \cap S_{d-1,1} \neq \emptyset$

  $R_{d,k_{d,1}+k_{d,2}} \cap S_{d-1,2} \neq \emptyset$

$L = L \cup findFeasibleSetForCycle(\{S_{d,1}, S_{d,2}\}, \{S_{d-1,1}, S_{d-1,2}\},$
  $\{R_{d,1}, \ldots, R_{d,k_d}\})$, *in Figure 6.*

return L

end *FindFeasibleSetForIntersectionGraph*

Figure 4: Algorithm FindFeasibleSetForIntersectionGraph

FindDominoCyclesOrder

input

$G_{int}(H)$ *an intersection graph*

returns

CL *a list of ordered cycles*

begin

Let VD *be the set of nodes in* $G_{int}(H)$ *with rank 3.*

Let CC = *ChordlessCycles(*$G_{int}(H)$*), described in [5].*

Find $C_1 \in CC$ *such that:*

  $|C_1 \cap VD| = 2$

  $CL = \{C_1\}$

Let $(X_{cur}, Y_{cur}) = C_1 \cap VD$

for *(*$i \in 2, \ldots, |CC| - 1$*)*

  Find $C_i \in CL$ *such that:*

    $X_{cur}, Y_{cur} \in C_i$

    There are $X_{new}, Y_{new} \in VD$ *such that* $X_{new}, Y_{new} \in C_i$

    $CL = CL \cup \{C_i\}$

  Set $(X_{cur}, Y_{cur}) = (X_{new}, Y_{new})$

end for

Find $C_d \in CC$ *which satisfies* $|C_d \cap VD| = 2$ *such that:*

  $C_d \notin CL$ *and* $\{X_{cur}, Y_{cur}\} \in C_d$

  $CL = CL \cup \{C_d\}$

end *FindDominoCyclesOrder*

Figure 5: Procedure FindDominoCyclesOrder

findFeasibleSetForCycle

input

$\{S_{1,1}, S_{1,2}\}, \{S_{2,1}, S_{2,2}\}$ *and* $\{R_1, \ldots, R_{k_1+k_2}\}$,

the clusters of a chordless cycle.

such that:

$R_1 \cap S_{2,1} \neq \emptyset$

$R_{k_1+1} \cap S_{2,2} \neq \emptyset$

$R_{k_1} \cap S_{1,1} \neq \emptyset$

$R_{k_1+k_2} \cap S_{2,2} \neq \emptyset$

returns

A feasible addition list L.

begin

Initialize L to be empty.

for *(*$j \in 1, \ldots, k_1 - 1$*)*

Choose $v \in R_j \cap R_{j+1}$ *and add* $(v, S_{2,1})$ *to L.*

end for

for *(*$j \in k_1 + 1, \ldots, k_1 + k_2 - 1$*)*

Choose $v \in R_j \cap R_{j+1}$ *and add* $(v, S_{2,1})$ *to L.*

end for

Choose $v \in R_{k_1} \cap S_{1,1}$ *and add* $(v, S_{2,1})$ *to L.*

Choose $v \in S_{2,2} \cap R_{k_1+1}$ *and add* $(v, S_{2,1})$ *to L.*

Choose $v \in R_{k_1+k_2} \cap S_{1,2}$ *and add* $(v, S_{2,1})$ *to L.*

return L

end *findFeasibleSetForCycle*

Figure 6: Procedure FindFeasibleSetForCycle

4.1.2 Testing example

In this section, we solve an example of a 3-level 2-domino intersection graph problem shown in Figure 7, by using the algorithm described in Section 4.1.1.

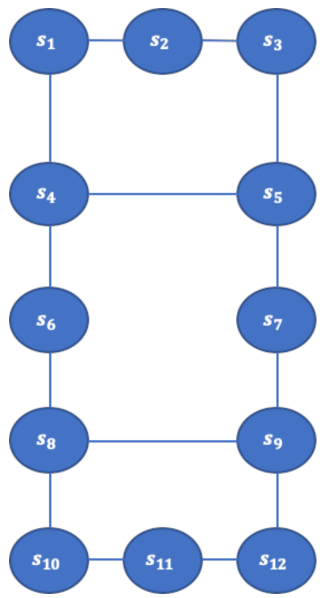

Figure 7: A 3-level 2-domino example

$H = <V, \mathcal{S}>$, $V = \{1, \ldots, 14\}$, $\mathcal{S} = \{S_1, \ldots, S_{12}\}$,$S_1 = \{1,2\}$, $S_2 = \{2,3\}$, $S_3 = \{3,4\}$, $S_4 = \{1,5,6\}$, $S_5 = \{4,5,8\}$, $S_6 = \{6,7\}$, $S_7 = \{8,9\}$, $S_8 = \{7,10,11\}$, $S_9 = \{9,10,14\}$, $S_{10} = \{11,12\}$, $S_{11} = \{12,13\}$, $S_{12} = \{13,14\}$.

Algorithm FindFeasibleSetForIntersectionGraph, presented in figure 4, receives $G_{int}(H)$ and returns a minimum cardinality feasible addition list for H.

The first step of the algorithm is to find all chordless cycles in $G_{int}(H)$ as a list of ordered cycles and add them to CL.

The Procedure FindDominoCyclesOrder, as presented in Figure 5, starts by finding all nodes in $G_{int}(H)$ with rank 3, $VD = \{4,5,8,9\}$. By using the algorithm ChordlessCycles [5], we get a list of random chordless cycles $CC = \{C_3, C_1, C_2\}$. The chordless cycles are $C_1 = \{S_1, S_2, S_3, S_4, S_5\}$, $C_2 = \{S_4, S_5, S_6, S_7, S_8, S_9\}$, $C_3 = \{S_8, S_9, S_{10}, S_{11}, S_{12}\}$. For this example, we assume C_1 contains the following clusters $\{S_1, S_2, S_3, S_4, S_5\}$. By iterating over CC, we find the following order for the cycles:

- C_1 contains path clusters: $\{S_4, S_5\}$ and regular clusters: $\{S_1, S_2, S_3\}$ and add to CL.
- C_2 contains path clusters: $\{S_8, S_9\}$ and regular clusters: $\{S_6, S_4, S_5, S_7\}$ and add to CL.
- C_3 contains path clusters: $\{S_8, S_9\}$ and regular clusters: $\{S_{10}, S_{11}, S_{12}\}$ and add to CL.

By using Procedure FindFeasibleSetForCycle, presented in Figure 6, we add for each cycle a minimum cardinality feasible addition list to L. The flow is presented as follows:

- i=1: input $\{S_1, S_2\}$, $\{S_4, S_5\}$, $\{S_3\}$, output: $\{(2, S_4), (3, S_4), (4, S_4)\}$
- i=2: input $\{S_4, S_5\}$, $\{S_8, S_9\}$, $\{S_6, S_7\}$, output: $\{(6, S_8), (5, S_8), (8, S_8), (9, S_8)\}$
- i=3: input: $\{S_{10}, S_{11}\}$, $\{S_8, S_9\}$, $\{S_{12}\}$, output: $\{(12, S_8), (13, S_8), (14, S_8)\}$

In conclusion we return:
$L = \{(2, S_4), (3, S_4), (4, S_4), (6, S_8), (5, S_8), (8, S_8), (9, S_8), (12, S_8), (13, S_8), (14, S_8)\}$

The clusters after the vertex addition $H + L$ are:
$S_1 = \{1, 2\}$, $S_2 = \{2, 3\}$, $S_3 = \{3, 4\}$, $S_4 = \{1, 2, 3, 4, 5, 6\}$, $S_5 = \{4, 5, 8\}$, $S_6 = \{6, 7\}$, $S_7 = \{8, 9\}$, $S_8 = \{5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$, $S_9 = \{9, 10, 14\}$, $S_{10} = \{11, 12\}$, $S_{11} = \{12, 13\}$, $S_{12} = \{13, 14\}$.

Figure 8 presents the intersection graph after the vertex addition. Figure 9 presents a feasible solution tree after the vertex addition.

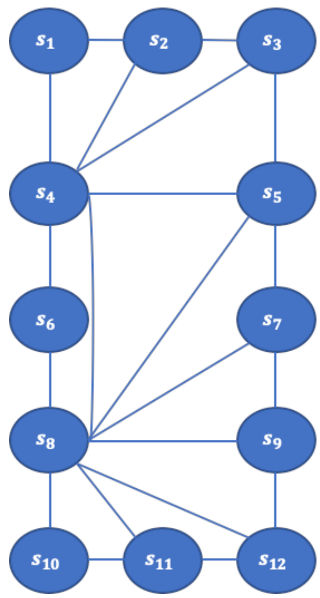

Figure 8: The intersection graph after the vertex addition for Example 4.1.2

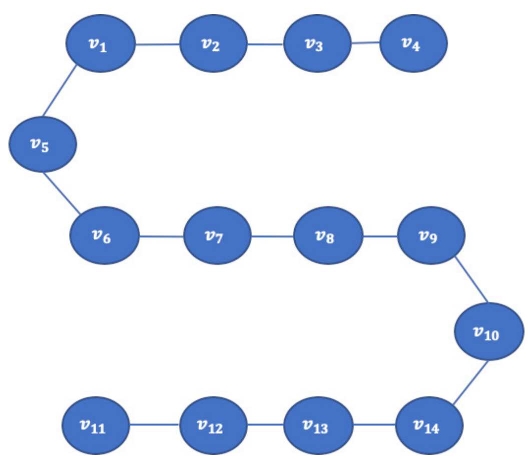

Figure 9: A feasible solution tree for Example 4.1.2

4.2 A d-level 3-domino

A d-level 3-domino, see figure 10, is a private case of a d-level t-domino graph, as presented in definition 4.0.1, where $i = 0, \ldots, d$ and $t = 3$.

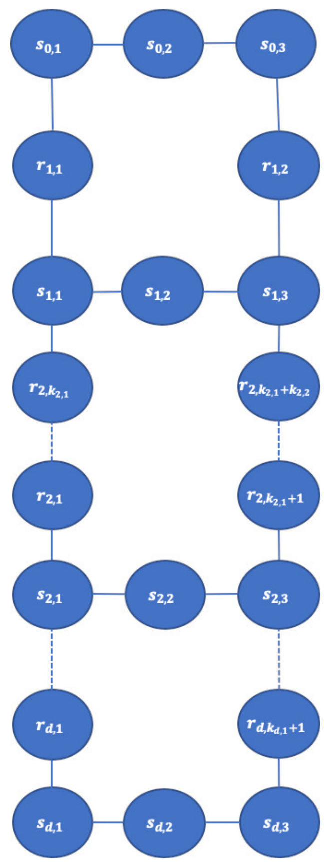

Figure 10: A d-level-3-domino graph

Definition 4.2.1. *Let $H = < V, \mathcal{S} >$ be a hypergraph whose intersection graph $G_{int}(H)$ is a d-level 3-domino graph and whose cycles are $C_1, C_2, \ldots, C_d$. Define **FAEL**, Feasible Addition Edges List, to be a chord addition list which contains the following edges:*

- $(s_{i,1}, s_{i,3})$, *for* $i = 0, \ldots, d$
- $(s_{i,1}, r_{i,p_i})$, *for* $i = 1, \ldots, d$ *and* $p_i = 2, \ldots, k_i$
- $(s_{i,1}, s_{i-1,1})$, *for* $i = 1, \ldots, d$
- $(s_{i,1}, s_{i-1,3})$, *for* $i = 1, \ldots, d$

Note that $G_{int}(H)$ contains edge $(s_{i,1}, r_{i,1})$ for every $i = 1, \ldots, d$.

Theorem 4.2.2. *Let $H = < V, \mathcal{S} >$ be a hypergraph whose intersection graph $G_{int}(H)$ is a d-level 3-domino graph. $FAEL$ defined in Definition 4.2.1 is a chord addition list for $G_{int}(H)$.*

Proof. We prove the theorem by induction on d.

For d=1, $G_{int}(H)$ is a single cycle, according to Theorem 3.1.4, $FAEL$ is a chord addition list.

Assume by induction the theorem's claim is correct for $d-1$, and consider $H = < V, \mathcal{S} >$ whose intersection graph is a d-level 3-domino.

Assume by contradiction $G_{int}(H + FAEL)$ contains a cycle C' such that $|C'| \geq 4$.

According to the induction assumption, $FAEL$ is a chord addition list for the first $d-1$ levels, hence there are no cycles of size greater or equal to 4 in the intersection graph of $(H+FAEL)[\bigcup_{i=1}^{d-1} C_i]$. $FAEL$ contains also a chord addition list for the last cycle C_d, as proven in the base part of the induction. Thus C' has to be partially in C_d and partially in $(H+FAEL)[\bigcup_{i=1}^{d-1} C_i]$ and therefore contains $(s_{d-1,1}, s_{d-1,3})$.

Since $FAEL$ contains edge $(s_{d-1,1}, s_{d-1,3})$, it is a chord in C', contradictioning the assumption.

□

Theorem 4.2.3. *Let $H = < V, S >$ be a hypergraph, whose intersection graph $G_{int}(H)$ is a d-level 3-domino graph, the cardinality of $FAEL$ defined in Definition 4.2.1 is $\sum_{i=1}^{d} (|C_i|) - 4d + 1$.*

Proof. $FAEL$ contains the following edges:

- The set of edges $(s_{i,1}, s_{i,3})$, for $0 \leq i \leq d$, which contains $d + 1$ edges.
- The set of edges $(s_{i,1}, r_{i,p})$, for $1 \leq i \leq d$ and $p = 2, \ldots, k_i$, which contains $\sum_{i=1}^{d} (K_i - 1)$ edges.
- The set of edges $(s_{i,1}, s_{i-1,1})$ and $(s_{i,1}, s_{i-1,3})$, for $1 \leq i \leq d$, which contains $2d$ edges.

According to Definition 4.0.1, for $t = 3$, $K_i = |C_i| - 2t = |C_i| - 6$ for $i = 1, \ldots, d-1$.

By summing the above, the cardinality of $FAEL$ is:

$$(d+1)+\sum_{i=1}^{d}(K_i-1)+(2d)=$$

$$\sum_{i=1}^{d}(|C_i|-6-1)+3d+1=$$

$$\sum_{i=1}^{d}(|C_i|)-7d+3d+1=$$

$$\sum_{i=1}^{d}(|C_i|)-4d+1$$

□

Theorem 4.2.4. *Let $H = <V, \mathcal{S}>$ be a hypergraph, whose intersection graph $G_{int}(H)$ is a d-level 3-domino graph, $FAEL$ defined in Definition 4.2.1 is a $MCLE(G_{int}(H))$.*

Proof. We prove the theorem by induction on d.

For d=1, $G_{int}(H)$ is a single cycle, $|FAEL(H)| = \sum_{i=1}^{1}(|C_i|) - 4d + 1 = |C_1| - 4 + 1 = |C_1| - 3$. According to Theorems 3.1.1 and 4.2.2, $FAEL(H)$ is a $MCLE(G_{int}(H))$.

Assume by induction the theorem's claim is correct for $d-1$ and consider $H = <V, \mathcal{S}>$ whose intersection graph is a d-level 3-domino.

Denote $H' = H[C_1 \cup \cdots \cup C_{d-1}]$, according to Theorem 4.2.3, $|FAEL[H']| = \sum_{i=1}^{d-1}(|C_i|) - 4(d-1) + 1$ and it is a $MCLE$ of H'. For the last cycle C_d, as proven in the base induction, $FAEL[C_d] = |C_d| - 3$ and it is a $MCLE$ of C_d.

Since $FAEL[H'] \cap FAEL[C_d] = (s_{d-1,1}, s_{d-1,3})$, according to Theorem 3.3.5, $|MCLE(H)| = |MCLE(H')| + |MCLE(C_d)| - 1 = \sum_{i=1}^{d-1}(|C_i|) - 4(d-1) + 1 + |C_d| - 3 - 1 = \sum_{i=1}^{d}(|C_i|) - 4d + 1$.

Since edge $(s_{d-1,1}, s_{d-1,3})$ is counted in both $FAEL[H']$ and $FAEL[C_d]$, then $|FAEL| = |FAEL[H']| + |FAEL[C_d]| - (s_{d-1,1}, s_{d-1,3}) = \sum_{i=1}^{d-1}(|C_i|) - 4(d-1) + 1 + |C_d| - 3 - 1 = \sum_{i=1}^{d}(|C_i|) - 4d + 1$, thus $FAEL$ is also a $MCLE$. □

4.3 d-level t-domino

Theorem 4.3.1. *Let $C = \{v_1, \dots, v_k\}$ be a chordless cycle graph of size $k \geq 4$ and let $E_C = MCLE(C)$ which includes edge (v_{k_1}, v_{k_2}), for $k_1 < k_2$ and let $E_{k_1 k_2} = \{(v_{p_1}, v_{p_2}) | (v_{p_1}, v_{p_2}) \in E_C, k_1 \leq p_1 < p_2 \leq k_2\} \setminus (v_{k_1}, v_{k_2})$.*

Then $E'_C = (E_C \setminus E_{k_1 k_2}) \cup \{(v_{k_1}, v_{k_1+2}), (v_{k_1}, v_{k_1+3}), \dots, (v_{k_1}, v_{k_2-1})\}$ is a chord addition list for C.

Proof. An example for E_C is shown in Figure 11 (left) and example for $E_{C'}$ is shown in Figure 11(right).

Assume that after adding E'_C to C it contains a chordless cycle C^*, as shown in Figure 12, there are three options for the location of C^*:

- C^* is contained in $V \setminus \{v_{k_1+1}, \dots, v_{k_2-1}\}$, in this case C^* is also in $C + E_C$ which contradicts the assumption that $E_C = MCLE(C)$.

- C^* is contained in $\{v_{k_1}, \dots, v_{k_2}\}$, this contradicts the fact that we added edges $\{(v_{k_1}, v_{k_1+2}), (v_{k_1}, v_{k_1+3}), \dots, (v_{k_1}, v_{k_2-1})\}$.

- C^* contains a node from $V \setminus \{v_{k_1+1}, \dots, v_{k_2-1}\}$ and a node from $\{v_{k_1+1}, \dots, v_{k_2-1}\}$, In this case C^* must contain both nodes v_{k_1} and v_{k_2}, and therfore edge (v_{k_1}, v_{k_2}) is a chord in C^*, a contradiction.

Hence, E'_C is a chord addition list for C. □

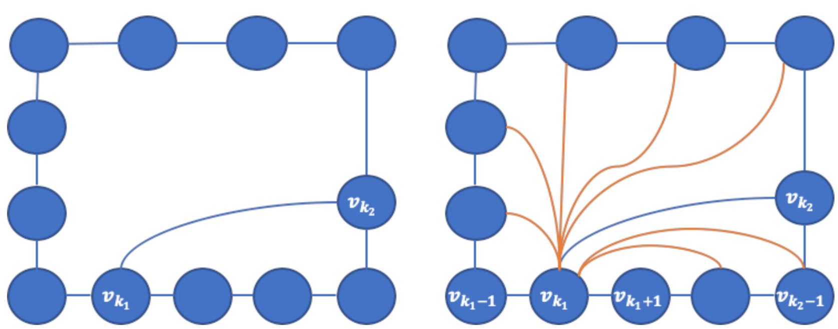

Figure 11: Example for Theorem 4.3.1

Lemma 4.3.2. *Let $C = \{v_1, \dots, v_k\}$ be a chordless cycle graph of size $k \geq 4$ and let $E_C = MCLE(C)$ which includes edge (v_{k_1}, v_{k_2}), for $k_1 < k_2$, and let $E_{k_1 k_2} = \{(v_{p_1}, v_{p_2}) | (v_{p_1}, v_{p_2}) \in E_C, k_1 \leq p_1 < p_2 \leq k_2\} \setminus (v_{k_1}, v_{k_2})$, then $|E_{k_1 k_2}| \leq k_2 - k_1 - 2$.*

Proof. According to Theorem 4.3.1, $E'_C = (E_C \setminus E_{k_1 k_2}) \cup \{(v_{k_1}, v_{k_1+2}), (v_{k_1}, v_{k_1+3}), \ldots, (v_{k_1}, v_{k_2-1})\}$ is a chord addition list for C.

$|E'_C| = |E_C| - |E_{k_1 k_2}| + (k_2 - k_1 - 2)$. Since $E_C = MCLE(C)$ then $|E'_C| \geq |E_C|$. Hence, $|E_C| - |E_{k_1 k_2}| + (k_2 - k_1 - 2) \geq |E_C|$ and therefore $|E_{k_1 k_2}| \leq k_2 - k_1 - 2$. □

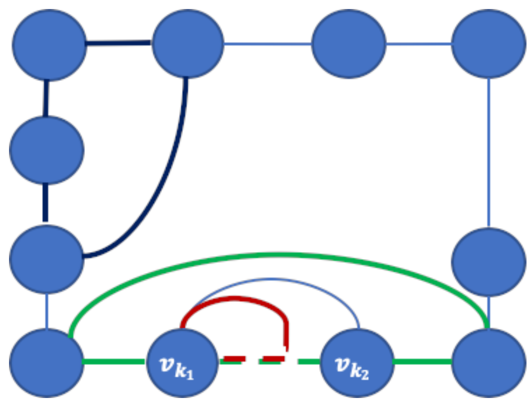

Figure 12: Possible locations of C^* in Proof of Theorems 4.3.1

Lemma 4.3.3. *Let $C = \{v_1, \ldots, v_k\}$ be a chordless cycle graph of size $k \geq 4$ and let $E_C = MCLE(C)$.*

Then for every $k_1 < k_2$ such that $|k_2 - k_1| \geq 2$, E_C contains at least one edge (p_1, p_2) such that $k_1 \leq p_1 < p_2 \leq k_2$.

Proof. Without loss of generality, suppose that $k_2 > k_1$, in this case $k_2 \geq k_1 + 2$.

Assume by contradiction that E_C does not contain any edge (p_1, p_2), such that $k_1 \leq p_1 < p_2 \leq k_2$.

Denote $(a, b) \in E_C$, the edge which is the closest edge to (k_1, k_2), such that $a \leq k_1$ and $b \geq k_2$ and $b - a$ is minimum. Possible locations of (a, b) are demonstrated in Figure 13.

According to the assumption E_C, does not contain any other edge (p_1, p_2) such that $a \leq p_1 < p_2 \leq b$. In this case, the following cycle is created $a, \ldots, k_1, \ldots, k_2, \ldots, b, a$. This is a chordless cycle which contains at least $k_2 - k_1 + 1 + 1$. And since $k_2 \geq k_1 + 2$, the cycle contains at least 4 nodes, which contradicts the definition of E_C. □

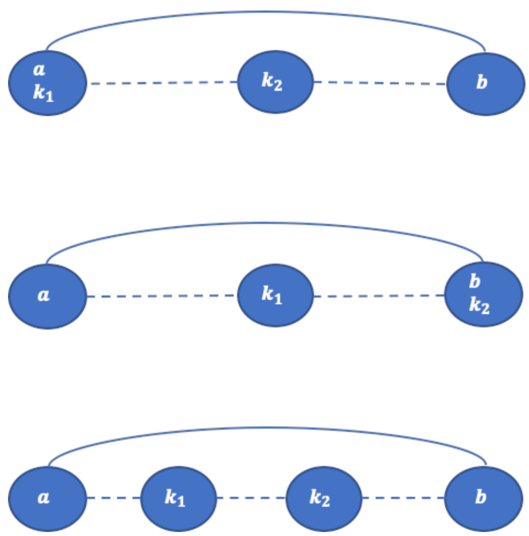

Figure 13: Possible locations of (a, b) in the proof of Lemma 4.3.3

Theorem 4.3.4. *Let C be a chordless cycle which contains a path $v_1, \ldots, v_t$ such that $t \geq 3$, $E_C = MCLE(C)$ and let $E_P = \{(v_{k_1}, v_{k_2}) | (v_{k_1}, v_{k_2}) \in E_C, \{v_{k_1}, v_{k_2}\} \subseteq \{v_1, \ldots, v_t\}]\}$ then $|E_P| \leq t - 2$.*

Proof. We prove the theorem by induction on t.

For $t = 3$ the path contains the following vertices $\{v_1, v_2, v_t\}$. Since edges $\{(v_1, v_2), (v_2, v_3)\}$ already exists, then E_P may only contain (v_1, v_t) and in any case $|E_p| \leq 1$.

Assume by induction the theorem's claim is correct for a path of size lower than t and prove it for t.

If $t \geq 3$, according to Lemma 4.3.3, E_C contains at least one edge (v_{k_1}, v_{k_2}), such that $(v_{k_1}, v_{k_2}) \subseteq \{v_1, v_2, \ldots, v_t\}$ and since $E_C = MCLE(C)$ then $k_2 \geq k_1 + 2$.

Cycle C and edge (v_{k_1}, v_{k_2}) are demonstrated in Figure 14.

E_C contains the following edges:

- Edges from $E_{k_1 k_2} = \{(v_{p_1}, v_{p_2}) | (v_{p_1}, v_{p_2}) \in E_C, k_1 \leq p_1 < p_2 \leq k_2\} \backslash (v_{k_1}, v_{k_2})$.
- Edges from $E_C \setminus E_{k_1 k_2}$.
- Edge (v_{k_1}, v_{k_2}).

Consider cycle $C' = (C \setminus \{(v_{k_1+1}, v_{k_1+2}), (v_{k_1+2}, v_{k_1+3}), \ldots, (v_{k_2-1}, v_{k_2})\}) \cup (v_{k_1}, v_{k_2})$. C' is a chordless cycle which contains path $P' = v_1, \ldots, v_{k_1}, v_{k_2}, \ldots v_t$. Define $E_{P'} = \{(v_{p_1}, v_{p_2}) | (v_{p_1}, v_{p_2}) \in E_C, \{v_{k_1}, v_{k_2}\} \subseteq P'\}$. According to the induction, $|E_{P'}| \leq t - (k_2 - k_1 - 1) - 2 = t - (k_2 - k_1) + 1 - 2$

According to Lemma 4.3.2, $|E_{k_1 k_2}| \leq k_2 - k_1 - 2$.

Hence $|E_P| = |E'_P \cup (v_{k_1}, v_{k_2}) \cup E_{k_1 k_2}| \leq (t - (k_2 - k_1) + 1 - 2) + 1 + (k_2 - k_1 - 2) = t + 1 - 2 + 1 - 2 = t - 2$.

□

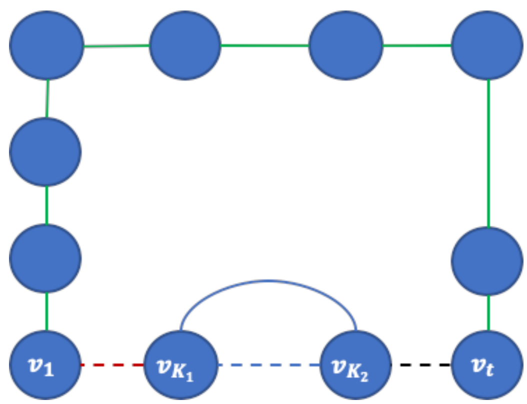

Figure 14: A graph after adding edge (v_{k_1}, v_{k_2})

Corollary 4.3.5. *Let $H = < V, S >$ be a hypergraph, whose intersection graph $G_{int}(H)$ is a 2-level t-domino graph, with a separating path $(s_1, \ldots, s_t)$ whose removal creates two connected components S_a and S_b. Denote $C_a = G[S_a \cup \{s_1, \ldots, s_t\}]$, $C_b = G[S_b \cup \{s_1, \ldots, s_t\}]$.*

Let $E_{C_a} = MCLE(C_a)$ and $E_{C_b} = MCLE(C_b)$ then $E_{C_a} \cap E_{C_b}$ contains only edges whose both endpoints are in $\{s_1, \ldots, s_t\}$ and according to Theorem 4.3.4, $|E_{C_a} \cap E_{C_b}| \leq t - 2$.

Theorem 4.3.6. *Let $H = < V, S >$ be a hypergraph whose intersection graph $G_{int}(H)$ is a 2-level t-domino graph, with a separating path $(s_1, \ldots, s_t)$ whose removal creates two connected components S_a and S_b. Denote $C_a = G[S_a \cup \{s_1, \ldots, s_t\}]$, $C_b = G[S_b \cup \{s_1, \ldots, s_t\}]$.*

Let $E_{C_a} = MCLE(C_a)$ and $E_{C_b} = MCLE(C_b)$ such that $\{(s_1, s_k) | 3 \leq k \leq t\} \subseteq E_{C_a} \cap E_{C_b}$, then $E_{C_a} \cup E_{C_b}$ is a chord addition list of edges for G.

Proof. Assume that after adding $E_{C_a} \cup E_{C_b}$ to $G_{int}(H)$, it contains a chordless cycle C. There are three options for the location of C:

- C is contained in $C_a \cup E_{C_a}$, which contradicts the definition of E_a.
- C is contained in $C_b \cup E_{C_b}$, which contradicts the definition of E_b.
- C contains at least one node in $C_a \setminus \{s_1, \ldots, s_t\}$ and one node in $C_b \setminus \{s_1, \ldots, s_t\}$. Hence, the cycle contains s_1 and s_t. Since $E_{C_a} \cap E_{C_b}$ contains (s_1, s_t) which acts as a chord in C, contradicting the assumption that C is chordless.

□

Theorem 4.3.7. *Let $H = < V, S >$ be a hypergraph whose intersection graph $G_{int}(H)$ is a 2-level t-domino graph, with a separating path $(s_1, \ldots, s_t)$ whose removal creates two connected components S_a and S_b. Denote $C_a = G[S_a \cup \{s_1, \ldots, s_t\}]$, $C_b = G[S_b \cup \{s_1, \ldots, s_t\}]$ and let E_G be a feasible chord addition list for $G_{int}(H)$, then $|E_G| \geq |MCLE(C_a)| + |MCLE(C_b)| - (t-2)$.*

Proof. Let $E_{C_a} = E_G[C_a]$ and $E_{C_b} = E_G[C_b]$ a feasible chord addition list for C_a and C_b respectively, in this case $E_G \supseteq E_{G_a} \cup E_{G_b}$, then $|E_G| \geq |E_{C_a}| + |E_{C_b}| - |E_{C_a} \cap E_{C_b}|$.

According to Corollary 4.3.5, $|E_{C_b} \cap E_{C_b}| \leq t-2$ and therefore, $|E_G| \geq |E_{C_a}| + |E_{C_b}| - (t-2)$.

According to Definitions 2.0.11 and 2.0.12, $|E_{C_a}| \geq |MCLE(C_a)|$ and $|E_{C_b}| \geq |MCLE(C_b)|$. Hence $|E_G| \geq |MCLE(C_a)| \cup |MCLE(C_b)| - (t-2)$.
□

Definition 4.3.8. *Let $H =< V, S >$ be a hypergraph, whose intersection graph $G_{int}(H)$ is a d-level t-domino graph, with cycles $C_1, C_2, \ldots, C_d$, Define **FAEL** to be a $CLE(G_{int}(H))$ which contains the following edges:*

- $(s_{i,1}, s_{i,k})$ *for* $i = 0, \ldots, d$ *and* $k = 3, \ldots, t$
- $(s_{i,1}, r_{i,p_i})$ *for* $i = 1, \ldots, d$ *and* $p_i = 2, \ldots, k_i$
- $(s_{i,1}, s_{i-1,1})$ *for* $i = 1, \ldots, d$
- $(s_{i,1}, s_{i-1,t})$ *for* $i = 1, \ldots, d$

Theorem 4.3.9. *Let $H = < V, \mathcal{S} >$ be a hypergraph, whose intersection graph $G_{int}(H)$ is a d-level t-domino graph. $FAEL$ defined in Definition 4.3.8 is a chord addition list for $G_{int}(H)$.*

Proof. Similar to the proof of Theorem 4.2.2. □

Theorem 4.3.10. *Let $H = < V, \mathcal{S} >$ be a hypergraph, whose intersection graph $G_{int}(H)$ is a d-level t-domino graph. The cardinality of $FAEL$ is $\sum_{j=1}^{d}(|C_i|) - d(t+1) + t - 2$*

Proof. $FAEL$ contains the following edges:

- There are $(t-2)(d+1)$ edges $(s_{i,1}, s_{i,k})$ for $i = 0, \ldots, d$ and $k = 3, \ldots, t$.
- There are $\sum_{j=1}^{d}(K_i - 1)$ edges $(s_{i,1}, r_{i,p})$, for $i = 1, \ldots, d$ and $p = 2, \ldots, K_i$.
- There are d edges $(s_{i,1}, s_{i-1,1})$, for $i = 1, \ldots, d$.
- There are d edges $(s_{i,1}, s_{i-1,t})$, for $i = 1, \ldots, d$.

Since $K_i = |C_i| - 2t$, for $i = 1, \ldots, d$, the sum of all edges in $FAEL$ is:

$$
\begin{aligned}
|FAEL| = (t-2)(d+1) + \sum_{j=1}^{d}(K_i - 1) + d + d = \\
= (td + t - 2d - 2) + (\sum_{j=1}^{d}(|C_i| - 2t - 1) + (2d) \\
= (td + t - 2d - 2) + \sum_{j=1}^{d}(|C_i|) - 2td - d + (2d) \\
= \sum_{j=1}^{d}(|C_i|) - td - d + t - 2 \\
= \sum_{j=1}^{d}(|C_i|) - d(t+1) + t - 2
\end{aligned}
$$

□

Theorem 4.3.11. *Let $H =< V, S >$ be a hypergraph whose intersection graph $G_{int}(H)$ is a d-level t-domino graph, then $FAEL(G_{int}(H)) = MCLE(G_{int}(H))$.*

Proof. We prove the theorem by induction on d.

For d=1, according to Theorem 4.3.10, $|FAEL| = |C_1| - (t+1) + t - 2 = |C_1| - 3$, which is the size of $MCLE(C_1)$ according to Theorem 3.1.1.

Denote $H^{d-1} = \bigcup_{i=1}^{d-1} C_i$ and $FAEL^{d-1} = FAEL[H^{d-1}]$ and $E_d = MCLE(H[C_d])$.

Since H^{d-1} is a $(d-1)$-level t-domino, according to the induction hypothesis $FAEL^{d-1} = MCLE(H^{d-1})$.

According to Theorem 4.3.10, $|FAEL^{d-1}| = \sum_{j=1}^{d-1}(|C_i|) - (d-1)(t+1) + t - 2$.

According to Theorem 3.1.4 $|MCLE(C_d)| = |C_d| - 3$.

Since $G_{int}(H^{d-1}) \cap G_{int}(H[C_d]) = \{s_{d-1_1}, \ldots, s_{d-1_t}\}$, according to Theorem 4.3.7, $|MCLE(G_{int}(H))| \geq |MCLE(H^{d-1})| \cup |MCLE(H[C_d])| - (t-2)$.

Therefore, $|MCLE(G_{int}(H))| \geq \sum_{j=1}^{d-1}(|C_i|) - (d-1)(t+1) + (t-2) + |C_d| - 3 - (t-2) = \sum_{j=1}^{d}(|C_i|) - d(t+1) + (t+1) - 3 = \sum_{j=1}^{d}(|C_i|) - d(t+1) + (t-2)$.

According to Theorem 4.3.10, $|FAEL(G_{int}(H))| = \sum_{j=1}^{d}(|C_i|) - d(t+1) + t - 2$. Therefore, according to Theorem 4.3.11, $|FAEL(G_{int}(H))| = |MCLE(G_{int}(H))|$, and $FAEL(G_{int}(H)) = MCLE(G_{int}(H))$.

□

Theorem 4.3.12. *Let $H =< V, S >$ be a hypergraph, whose intersection graph $G_{int}(H)$ is a d-level t-domino graph, then the total number of created triangles by adding $FAEL$ to $G_{int}(H)$ is $\sum_{i=1}^{d}(|C_i|) - dt + t - 2$*

Proof. Denote $K_{i,1}$ and $K_{i,2}$ the amount of regular nodes in cycle C_i, in a d-level t-domino, such that $K_{i,1}$ ($K_{i,2}$) is the number of regular nodes between $S_{i,1}$ and $S_{i-1,1}$ ($S_{i,t}$ and $S_{i-1,t}$), respectively, as shown in Figure 15. The total number of created triangles, by adding $FAEL$ to $G_{int}(H)$, is shown in Figure 16 and can be calculated as follows:

Each cycle i, for $1 \leq i \leq d$, contains 5 types of triangles:

- $t-2$ triangles of the type:

$$\{s_{i-1,1}, s_{i-1,2}, s_{i-1,3}\}, \{s_{i-1,1}, s_{i-1,3}, s_{i-1,4}\}, \ldots, \{s_{i-1,1}, s_{i-1,t-1}, s_{i-1,t}\}$$

- $t-2$ triangles of the type:

$$\{s_{i,1}, s_{i,2}, s_{i,3}\}, \{s_{i,1}, s_{i,3}, s_{i,4}\}, \ldots, \{s_{i,1}, s_{i,t-1}, s_{i,t}\}$$

- $k_{i,1}$ triangles of the type:

$$\{s_{i,1}, r_{i,1}, r_{i,2}\}, \{s_{i,1}, r_{i,2}, r_{i,3}\}, \ldots, \{s_{i,1}, r_{i,k_{i,1}}, s_{i-1,1}\}$$

- $K_{i,2} + 1$ triangles of the type:

$$\{s_{i,1}, s_{i,t}, r_{i,k_{i,1}+1}\}, \{s_{i,1}, r_{i,k_{i,1}+1}, r_{i,k_{i,1}+2}\}, \{s_{i,1}, r_{i,k_{i,1}+2}, r_{i,k_{i,1}+3}\}, \ldots, \{s_{i,1}, r_{i,k_{i,1}+K_{i,2}}, s_{i-1,t}\}$$

- One more triangle $\{s_{i,1}, s_{i-1,1}, s_{i-1,t}\}$

Denote all the created triangles for a cycle i by TC_i. Therefore, $|TC_i| = 2(t-2) + k_{i,1} + k_{i,2} + 1 + 1$. Since $k_{i,1} + k_{i,2} = |C_i| - 2t$, then $|TC_i| = 2(t-2) + |C_i| - 2t + 1 + 1 = |C_i| - 2$. (Note that this result agrees with Theorem 3.1.1).

Every two consecutive chordless cycles in $G_{int}(H)$ share $t-2$ triangles, therefore, $|TC_i \cap TC_{i+1}| = t-2$.

Hence, the total number of created triangles by $FAEL$ is: $|\bigcup_{i=1}^{d} TC^i| = \sum_{i=1}^{d} |TC^i| - \sum_{i=1}^{d-1} |TC^i \cap TC^{i+1}| = \sum_{i=1}^{d}(|C_i| - 2) - (d-1)(t-2) = \sum_{i=1}^{d}(|C_i|) - 2d - (d-1)(t-2) = \sum_{i=1}^{d}(|C_i|) - 2d - dt + 2d + t - 2 = \sum_{i=1}^{d}(|C_i|) - dt + t - 2$.

□

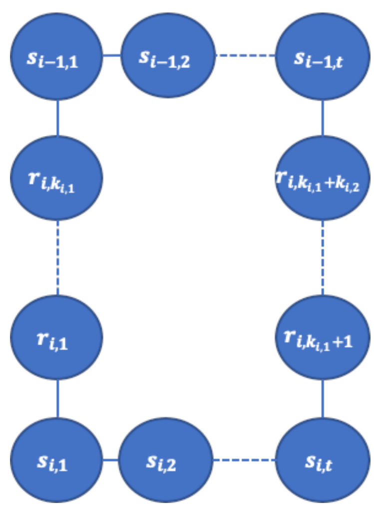

Figure 15: Cycle i in a d-level t-domino

Corollary 4.3.13. *Let $H = < V, S >$ be a hypergraph whose intersection graph $G_{int}(H)$ is a d-level t-domino graph. According to Theorem 3.1.4, the amount of vertex additions is greater or equal to the amount of created triangles. To show equality, we find a feasible vertex addition list whose cardinality is equal to the amount of created triangles.*

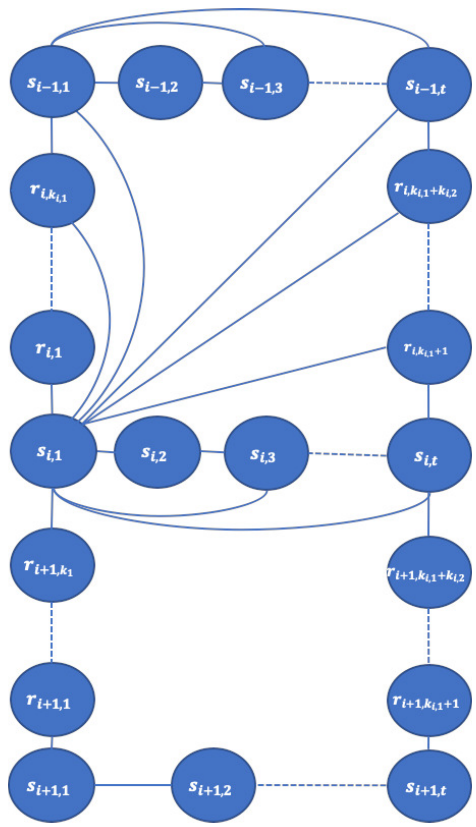

Figure 16: Cycle i edges addition

To show equality between the amount of vertex additions and created triangles, we describe how for each created triangle we add only one vertex. Describes in the proof of Theorem 4.3.12, we use the 5 types of triangles, to define the following vertex addition list, and prove that it is a feasible addition list for H.

Definition 4.3.14. *Let $H =< V, S >$ be a hypergraph whose intersection graph $G_{int}(H)$ is a d-level t-domino graph, with cycles $C_1, C_2, \ldots, C_d$. We define L_{FAEL} to be a vertex addition list for H, which contains the following additions, for $i = 1, \ldots, d$:*

- *Add a vertex from each of the following intersections to cluster $S_{i-1,1}$ ($t-2$ vertices additions):*

$$S_{i-1,2} \cap S_{i-1,3},\ S_{i-1,3} \cap S_{i-1,4}\ ,\ \ldots\ ,\ S_{i-1,t-1} \cap S_{i-1,t}$$

- *Add a vertex from each of the following intersections to cluster $S_{i,1}$ ($t-2$ vertices additions):*

$$S_{i,2} \cap S_{i,3}\ ,\ S_{i,3} \cap S_{i,4}\ ,\ \ldots\ ,\ S_{i,t-1} \cap S_{i,t}$$

- *Add a vertex from each of the following intersections to cluster $s_{i,1}$ ($k_{i,1}$ vertices additions):*

$$R_{i,1} \cap R_{i,2}\ ,\ R_{i,2} \cap R_{i,3}\ ,\ \ldots\ ,\ R_{i,k_{i,1}} \cap S_{i-1,1}$$

- *Add a vertex from each of the following intersections to cluster $s_{i,1}$ ($k_{i,2}+1$ vertices additions):*

$$S_{i,t} \cap R_{i,k_{i,1}+1}\ ,\ R_{i,k_{i,1}+1} \cap R_{i,k_{i,1}+2}\ ,\ R_{i,k_{i,1}+2} \cap R_{i,k_{i,1}+3}\ ,\ \ldots\ ,\ R_{i,k_1+k_2} \cap S_{i-1,t}$$

- *In a previews step we already added a vertex from $S_{i-1,t-1} \cap S_{i-1,t}$ to $S_{i-1,1}$. Add the same vertex to $S_{i,1}$.*

Theorem 4.3.15. *Let $H =< V, S >$ be a hypergraph, whose intersection graph $G_{int}(H)$ is a d-level t-domino graph, then $|L_{FAEL}| = \sum_{i=1}^{d}(|C_i|) - dt + t - 2$*

Proof. Since every created triangle must satisfy the Helly Property, according to Property 3.1.2, for every created triangle in FAEL, at least one vertex must be added to one of the clusters within the triangle.

According to Definition 4.3.14, the total number of vertex additions for a cycle C_i is $2(t-2) + K_{i,1} + (K_{i,2}+1) + 1$. Since the number of vertex addition is equal to the number of triangles, computed in Theorem 4.3.12, we have $|L_{FAEL}| = \sum_{i=1}^{d}(|C_i|) - dt + t - 2$. □

Theorem 4.3.16. *Let $H =< V, S >$ be a hypergraph, whose intersection graph $G_{int}(H)$ is a d-level t-domino graph, then L_{FAEL} is a feasible addition list of H.*

Proof. To prove that L_{FAEL} is a feasible addition list we will construct a feasible solution tree. Figure 17 describes a part of a feasible solution tree for $H + L_{FAEL}$. This is the part that contains all the clusters of cycle C_i and some clusters from C_{i+1}, such that the tree is created for the clusters in cycle i, including some clusters from cycle $i+1$. A blue circle refers to a group of nodes, each node can be shared between clusters or it can be only in one cluster, denoted by $S^*_{i,j}$, where $i = 1, \ldots, d$ and $j = 1, \ldots, t$. A blue square refers to a single shared node. A blue continuous line represents direct neighbour relation and a blue dash line refers to a sequence of the same structure.

To prove the feasibility of the tree we will demonstrate the subtree induced on $S_{i,1}, S_{i,t}$ and $R_{i,1}$. Similar subtree can be presented in C_i for $i = 1, \ldots, d$. Therefore, the tree is a feasible solution tree for $H + L_{FAEL}$ and L_{FAEL} is a feasible addition list.

The graph in Figure 18, presents the subtree induced for $S_{i,1}$. It induces the vertices in $S^*_{i,1} \cup (S_{i,1} \cap R_{i,k_{i,1}} \cap S_{i-1,1}) \cup (S_{i,1} \cap R_{i,k_{i,1}-1} \cap R_{i,k_1}) \cup \cdots \cup (S_{i,1} \cap R_{i,2} \cap R_{i,3}) \cup (S_{i,1} \cap R_{i,1} \cap R_{i,2}) \cup (S_{i,1} \cap R_{i,1}, S_{i,1} \cap R_{i,2}) \cup (S_{i,1} \cap R_{i,k_{i,1}+k_{i,2}} \cap S_{i-1,t}) \cup \cdots \cup (S_{i,1} \cap R_{i,k_{i,1}+1} \cap R_{i,k_{i,1}+2}) \cup (S_{i,1} \cap S_{i,t} \cap R_{i,k_{i,1}+1}) \cup (S_{i,1} \cap S_{i,t-1} \cap S_{i,t}) \cup \cdots \cup (S_{i,1} \cap S_{i,2} \cap S_{i,3}, S_{i,1} \cap S_{i,2}) \cup (S_{i,1} \cap S_{i,t-1} \cap S_{i,t} \cap S_{i+1,1}) \cup (S_{i,1} \cap R_{i+1,k_{i+1,1}} \cap S_{i+1,1}\})$. Moreover, in Figure 18, continuous and dash red bold lines present the subtree induced for $S_{i,1}$. Note that $S^*_{i,1}, S_{i,1} \cap S_{i,2}, S_{i,1} \cap R_{i,1}$ and $S_{i+1,1} \cap R_{i+1,K_{i+1,1}} \cap S_{i,1}$ are all connected in this subtree.

The graph in Figure 19 presents the subtree induced on $S_{i,t}$. It includes the vertices in $S^*_{i,t} \cup (S_{i,t} \cap R_{i,k_{i,1}+1}) \cup (S_{i,1} \cap S_{i,t} \cap R_{i,k_{i,1}+1}) \cup (S_{i,1} \cap R_{i,t-1} \cap S_{i,t}) \cup (S_{i,t-1} \cap S_{i,t}) \cup (S_{i,1} \cap S_{i,t-1} \cap S_{i,t} \cap S_{i+1,1}) \cup (S_{i+1,1} \cap R_{i+1,k_{i+1,1}+k_{i+1,2}} \cap S_{i,t})$. Note that $S^*_{i,t}, S_{i,t} \cap S_{i,t-1}, S_{i,t} \cap R_{i,K_1+K_2}$ and $S_{i+1,1} \cap R_{i+1,K_{i+1,1}+K_{i+1,2}} \cap S_{i,t}$ are all connected in this subtree.

The graph in Figure 20 presents the subtree induced on $R_{i,1}$. It includes vertices in $R^*_{i,1} \cup (R_{i,1} \cap R_{i,2}) \cup (S_{i,1} \cap R_{i,1} \cap R_{i,2}) \cup (S_{i,1} \cap R_{i,1})$. Note that $R^*_{i,1}, R_{i,1} \cap R_{i,2}$ and $R_{i,1} \cap S_{i,1}$ are all connected in this subtree.

□

Theorem 4.3.17. *Let* $H =< V, S >$ *be a hypergraph, whose intersection graph* $G_{int}(H)$ *is a d-level t-domino graph, then* $ML(H) = L_{FAEL}$ *and* $|ML(H)| = |L_{FAEL}| = \sum_{i=1}^{d}(|C_i|) - dt + t - 2$.

Proof. According to Theorem 4.3.15, $|L_{FAEL}| = \sum_{i=1}^{d}(|C_i|) - dt + t - 2 = |FAEL(H)|$. Since according to Theorem 4.3.11 $FAEL(H) = MCLE(H)$, then $L_{FAEL} = ML(H)$. □

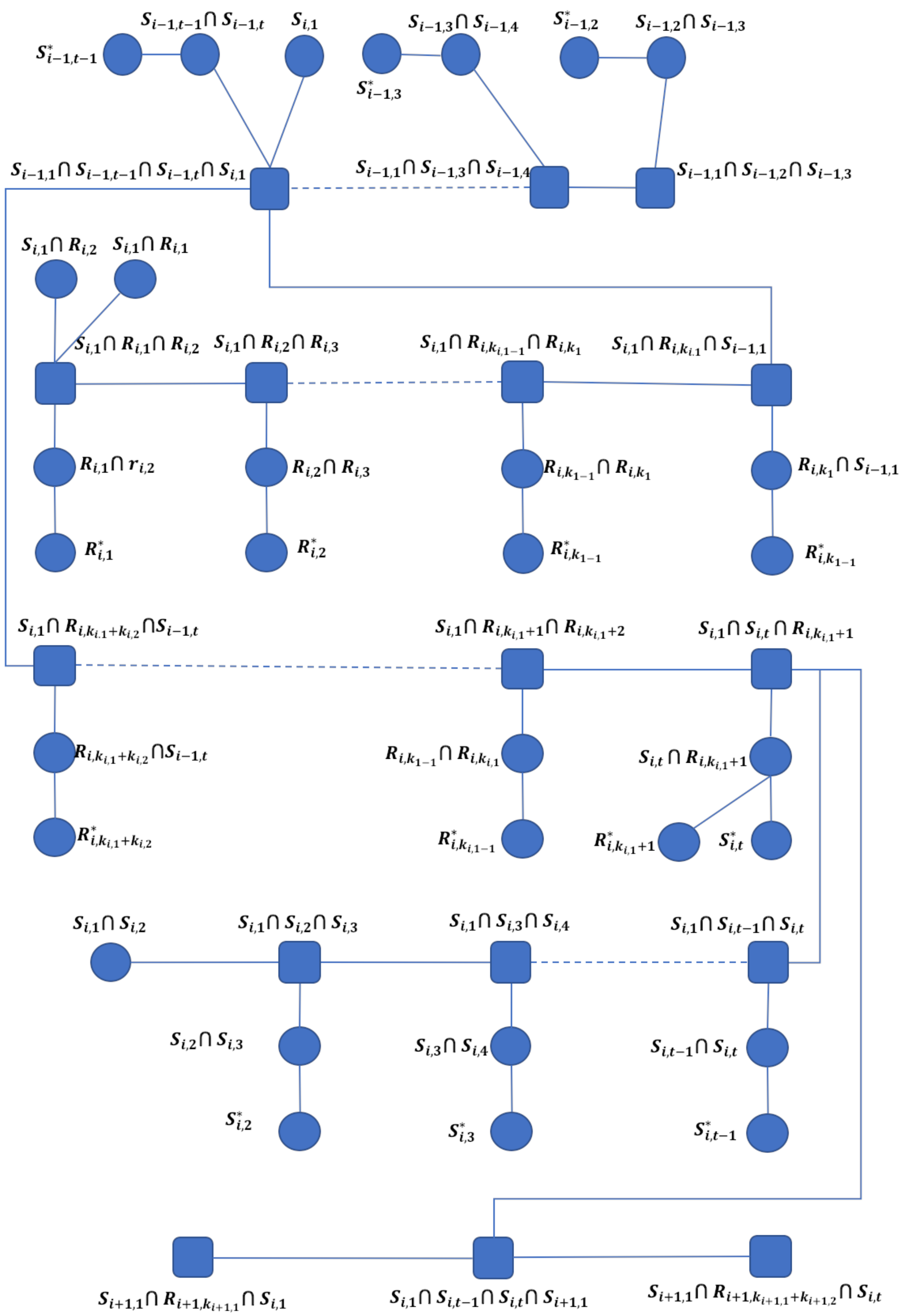

Figure 17: The tree created for the clusters in cycle i, including some clusters from cycle $i+1$.

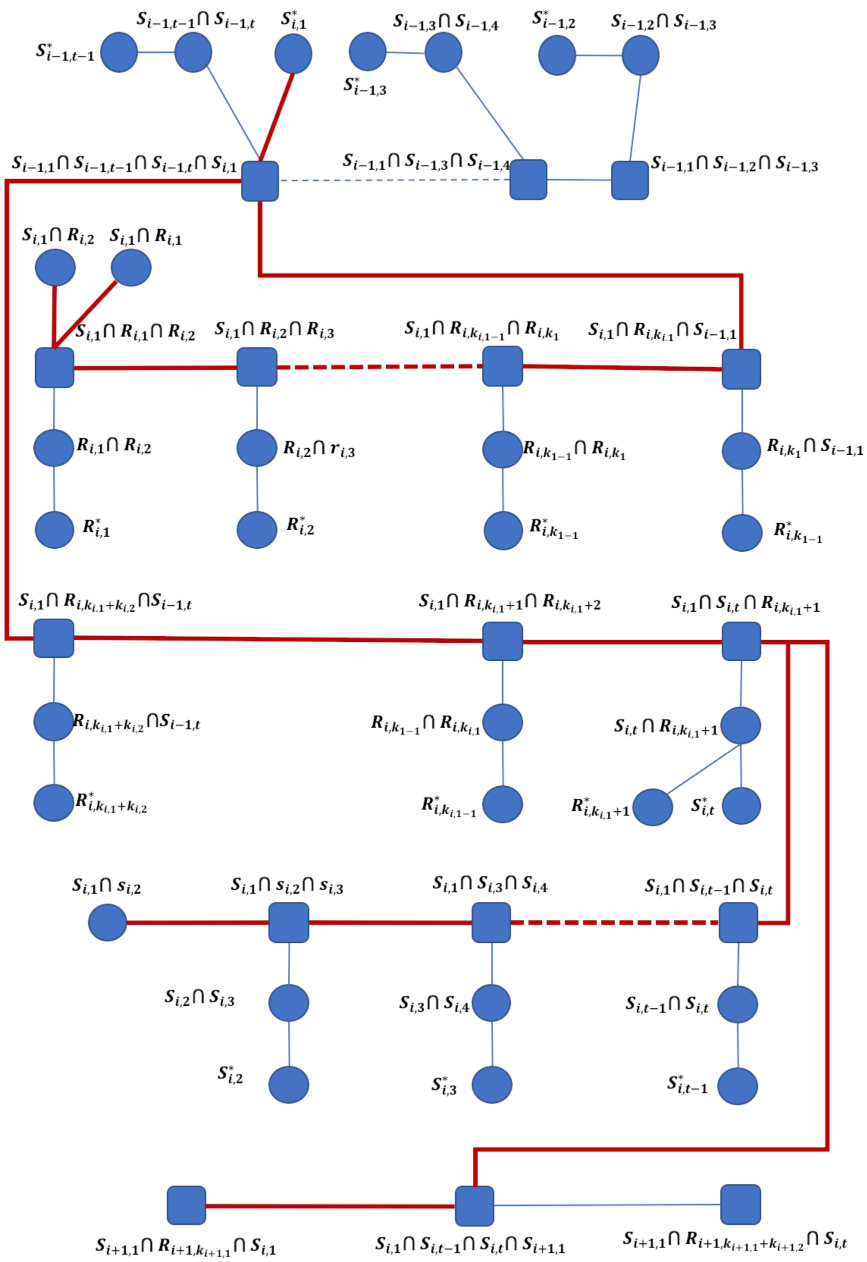

Figure 18: The subtree induced on $S_{i,1}$.

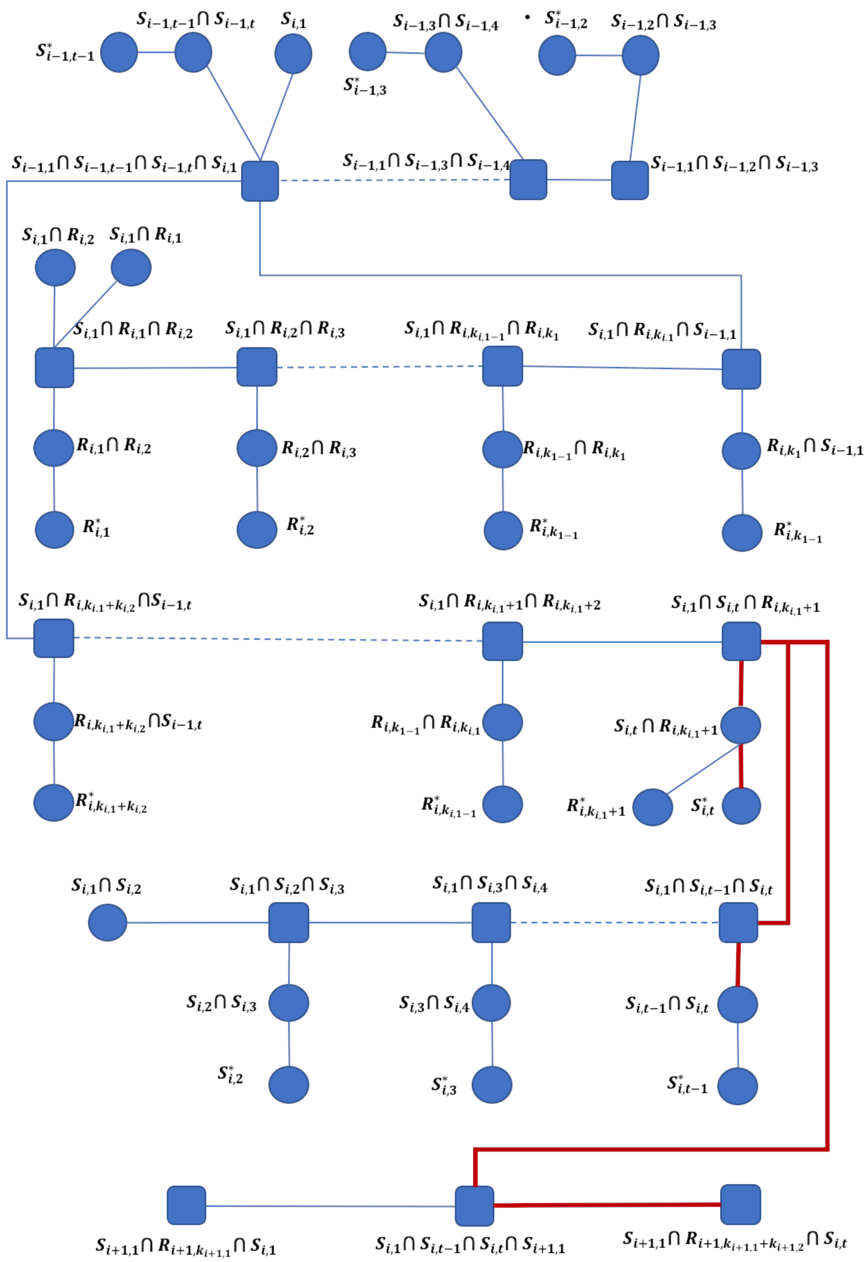

Figure 19: The subtree induced on $S_{i,t}$.

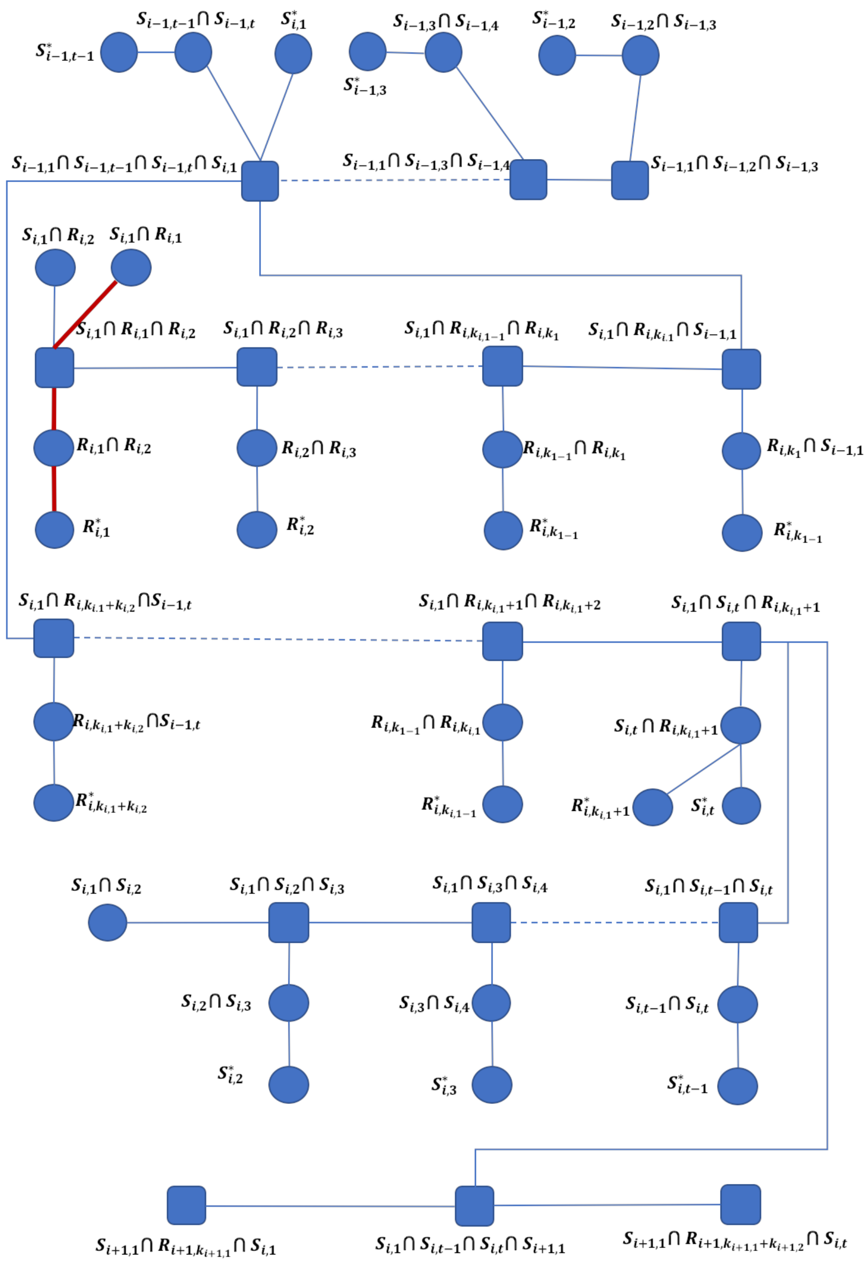

Figure 20: The subtree induced on $R_{i,1}$.

4.3.1 Algorithm d-level t-domino

In this section we propose an algorithm that finds a minimum cardinality fesiable addition list for $C_1, \ldots, C_d$ in a d-level t-domino graph.

Given a hypergraph H, whose intersection graph $G_{int}(H)$ is a d-level t-domino, denote $CL = \{C_1, \ldots, C_d\}$, a list of chordless cycles such that $C_i \cap C_{i+1} \neq \emptyset$, for $i = 1, \ldots, d$. As described in Definition 4.3.14, the algorithm finds L_{FAEL} which is a feasible addition list of H.

The algorithm is based on 4 main parts:

- Algorithm: FindFeasibleSetForIntersectionGraph, described in figure 21.
  Input: A hypergraph H and its intersection graph $G_{int}(H)$.
  Output: $ML(H)$.
  Description: Given a hypergraph H, whose intersection graph $G_{int}(H)$ is a d-level t-domino graph, with no feasible solution tree. The function finds L, a minimum cardinality feasible addition list.

- Procedure: FindDominoCyclesOrder described in figure 22.
  Input: A hypergraph H and its intersection graph $G_{int}(H)$.
  Output: $\{C_1, \ldots, C_d\}$ - list of ordered cycles.
  Description: Given a hypergraph H, whose intersection graph $G_{int}(H)$ is a d-level t-domino. The procedure returns a list of chordless cycles $C_1, \ldots, C_d$, such that $C_i \cap C_{i+1} \neq \emptyset$.

- Procedure: ChordlessCycles described by Dias,Castonguay,Longo and Jradi as described in paper [5]
  Input: A hypergraph H and its intersection graph $G_{int}(H)$
  Output: A list of all chordless cycles in $G_{int}(H)$.

- Procedure: FindFeasibleSetForCycle described in Figure 23.
  Input: $\{s_{i-1,1}, \ldots, s_{i-1,t}\}, \{s_{i,1}, \ldots, s_{i,t}\}$ and $\{r_{i,1}, \ldots, r_{i,k}\}$, the nodes of the chordless cycle.
  Output: $ML(C_i)$.
  Description: Given a chordless cycle C with path nodes $\{s_{i-1,1}, \ldots, s_{i-1,t}\}$, $\{s_{i,1}, \ldots, s_{i,t}\}$ and regular nodes $\{r_{i,1}, \ldots, r_{i,k_i}\}$. The procedure return $L(C)$ a minimum cardinality feasible addition list for C.

FindFeasibleSetForIntersectionGraph

input

A hypergraph H and its intersection graph $G_{int}(H)$ *which is a d-level t-domino.*

returns

A minimum cardinality feasible addition list L.

begin

$CL = FindDominoCyclesOrder(G_{int}(H))$, *in Figure 22.*

Initialize L to be empty and $d = |CL|$.

for $(i \in 1, \ldots, d-1)$

  if $i \geq 2$

    then $\{S_{i-1,1}, \ldots, S_{i-1,t}\} = CL_{i-1} \cap CL_i$

    else *Choose t continuous clusters* $\{S_{i-1,1}, \ldots, S_{i-1,t}\}$

  end if

  $\{S_{i,1}, \ldots, S_{i,t}\} = CL_i \cap CL_{i+1}$

  $\{R_{i,1}, \ldots, R_{i,k_{i,1}+k_{i,2}}\} = CL_i \setminus CL_{i+1}$

  such that:

    $R_{i,1} \cap S_{i,1} \neq \emptyset$

    $R_{i,k_{i,1}+1} \cap S_{i,t} \neq \emptyset$

    $R_{i,k_{i,1}} \cap S_{i-1,1} \neq \emptyset$

    $R_{i,k_{i,1}+k_{i,2}} \cap S_{i-1,t} \neq \emptyset$

  $L = L \cup FindFeasibleSetForCycle(\{S_{i-1,1}, \ldots, S_{i-1,t}\},$
    $\{S_{i,1}, \ldots, S_{i,t}\}, \{R_{i,1}, \ldots, R_{i,k_{i,1}+k_{i,2}}\})$, *in Figure 23.*

end for

$\{S_{d-1,1}, \ldots, S_{d-1,t}\} = CL_{d-1} \cap CL_d$

Choose t continuous clusters $\{S_{d,1}, \ldots, S_{d,t}\}$

$\{r_{d,1}, \ldots, r_{d,k_{d,1}+k_{d,2}}\} = CL_d \setminus CL_{d-1}$

such that:

  $R_{d,1} \cap S_{d,1} \neq \emptyset$

  $R_{d,k_{d,1}+1} \cap S_{d,t} \neq \emptyset$

  $R_{d,k_{d,1}} \cap S_{d-1,1} \neq \emptyset$

  $R_{d,k_{d,1}+k_{i,2}} \cap S_{d-1,t} \neq \emptyset$

$L = L \cup FindFeasibleSetForCycle(\{S_{d,1}, \ldots, S_{d,t}\},$
  $\{S_{d-1,1}, \ldots, S_{d-1,t}\}, \{R_{d,1}, \ldots, R_{d,k_{d,1}+k_{d,2}}\})$, *in Figure 23.*

return L

end *FindFeasibleSetForIntersectionGraph*

Figure 21: Algorithm FindFeasibleSetForIntersectionGraph

FindDominoCyclesOrder

input

$G_{int}(H)$ *an intersection graph*

returns

CL *a list of ordered cycles*

begin

Initialize CL *to be empty*

let VD *be the set of vertices in* $G_{int}(H)$ *with rank 3*

let $d = (|VD|/2) + 1$ *be the level of the domino*

let $CC = FindChordlessCycles(G_{int}(H))$*, described in [5].*

Find $C_1 \in CC$ *which satisfies* $|C_1 \cap VD| = 2$, $CL = CL \cup \{C_1\}$

let $\{X_{cur}, Y_{cur}\} = C_1 \cap VD$

for $(i \in 2, \ldots, d-1)$

Find $C_i \in CC \setminus CL$ *such that:*

$|C_i \cap VD| = 4$

$\{X_{cur}, Y_{cur}\} \in C_i$

Let $\{X_{new}, Y_{new}\} = (C_i \cap VD) \setminus \{X_{cur}, Y_{cur}\}$

$CL = CL \cup \{C_i\}$

Set $(X_{cur}, Y_{cur}) = (X_{new}, Y_{new})$

end for

Find $C_d \in CC \setminus CL$ *which satisfies* $|C_d \cap VD| = 2$

and $\{X_{cur}, Y_{cur}\} \subseteq C_d$

$CL = CL \cup \{C_d\}$

end *FindDominoCyclesOrder*

Figure 22: Procedure FindDominoCyclesOrder

FindFeasibleSetForCycle

input

$\{S_{1,1}, \ldots, S_{1,t}\}, \{S_{2,1}, \ldots, S_{2,t}\}, \{R_1, \ldots, R_{k_1+k_2}\}$

the clusters inducing a chordless cycle.

Such that:

$R_1 \cap S_{2,1} \neq \emptyset$

$R_{k_1+1} \cap S_{2,t} \neq \emptyset$

$R_{k_1} \cap S_{1,1} \neq \emptyset$

$R_{k_1+k_2} \cap S_{1,t} \neq \emptyset$

returns

A feasible addition list L.

begin

Initialize L to be empty.

for *(*$j \in 2, \ldots, t$*)*

Choose $v \in S_{1,j} \cap S_{1,j+1}$ *and add* $(v, S_{1,1})$ *to L.*

Choose $v \in S_{2,j} \cap S_{2,j+1}$ *and add* $(v, S_{2,1})$ *to L.*

end for

for *(*$j \in 1, \ldots, k_1 - 1$*)*

Choose $v \in R_j \cap R_{j+1}$ *and add* $(v, S_{2,1})$ *to L.*

end for

Choose $v \in R_{k_1} \cap S_{1,1}$ *and add* $(v, S_{2,1})$ *to L.*

for *(*$j \in k_1 + 1, \ldots, k_1 + k_2 - 1$*)*

Choose $v \in R_j \cap R_{j+1}$ *and add* $(v, S_{2,1})$ *to L.*

end for

Choose $v \in S_{2,t} \cap R_{k_1+1}$ *and add* $(v, S_{2,1})$ *to L.*

Choose $v \in R_{k_1+k_2} \cap S_{1,t}$ *and add* $(v, S_{2,1})$ *to L.*

Choose $v \in S_{1,t-1} \cap S_{1,t}$ *and add* $(v, S_{2,1})$ *to L.*

return L

end *FindFeasibleSetForCycle*

Figure 23: Procedure FindFeasibleSetForCycle

5 Summary and further research

Given a hypergraph, the research focuses on intersection graphs with special characteristics, where it is easy to show that there is no feasible solution for the given hypergrah.

The research starts by looking at an intersection graph which is a single chordless cycle, then on a hypergraph H with a separating edge or a separating path of size three. A significant part of the research is where $G_{int}(H)$ is a d-level t-domino graph, initially treated a smaller domino structured problem up to solving the general case.

We provide an algorithm that finds a possible feasible addition list for $G_{int}(H)$, denoted by L, whose cardinality is minimum. By adding L to H we can provide a feasible solution tree for the given hypergraph.

We plan to extend the generalization of the d-level t-domino solution to a d-level domino, where on each level of the domino the cardinality of the intersection path nodes is different. In addition, we plan to investigate more complex structures of intersection graphs, find their minium cardinality feasible addition lists, to gain feasibility for the given hypergraph.

References

[1] S. Anily, J. Bramel, and A. Hertz. A $\frac{5}{3}$-approximation algorithm for the clustered traveling salesman tour and path problems. *Oper. Res. Lett.*, 24(1-2):29–35, 1999.

[2] E. M. Arkin, R. Hassin, and L. Klein. Restricted delivery problems on a network. *Networks*, 29(4):205–216, 1997.

[3] K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms. *Journal of Computer and System Sciences*, 13(3):335 – 379, 1976.

[4] J. A. Chisman. The clustered traveling salesman problem. *Computers & Operations Research*, 2(2):115–119, 1975.

[5] E. S. Dias, D. Castonguay, H. J. Longo, and W. A. R. Jradi. Efficient enumeration of all chordless cycles in graphs. *CoRR*, abs/1309.1051, 2013.

[6] P. Duchet. Propriété de Helly et problémes de représentation. *Colloqu. Internat. CNRS, Problemes Conbinatoires et Theorie du Graphs, Orsay, France*, 260:117–118, 1976.

[7] C. Flament. Hypergraphes arborés. *Discrete Math.*, 21(3):223–227, 1978.

[8] D. Gusfield. *Algorithms on strings, trees and sequences: computer science and computational biology.* Cambridge university press, 1997.

[9] N. Guttmann-Beck, R. Hassin, S. Khuller, and B. Raghavachari. Approximation algorithms with bounded performance guarantees for the clustered traveling salesman problem. *Algorithmica*, 28(4):422–437, 2000.

[10] N. Guttmann-Beck, Z. Sorek, and M. Stern. Clustered spanning tree - conditions for feasibility. *Discrete Mathematics & Theoretical Computer Science*, vol. 21 no. 1, ICGT 2018, Aug 2019.

[11] R. M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[12] E. Korach and M. Stern. The clustering matroid and the optimal clustering tree. *Mathematical Programming*, 98(1-3):385–414, 2003.

[13] E. Korach and M. Stern. The complete optimal stars-clustering-tree problem. *Discrete Applied Mathematics*, 156(4):444 – 450, 2008.

[14] F. C. J. Lokin. Procedures for travelling salesman problems with additional constraints. *European Journal of Operational Research*, 3(2):135–141, 1979.

[15] T. A. McKee and F. R. McMorris. *Topics in Intersection Graph Theory*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.

[16] J. Y. Potvin and F. Guertin. A genetic algorithm for the clustered traveling salesman problem with a prespecified order on the clusters. In *Advances in computational and stochastic optimization, logic programming, and heuristic search*, volume 9 of *Oper. Res./Comput. Sci. Interfaces Ser.*, pages 287–299. Kluwer Acad. bl., Boston, MA, 1998.

[17] J. Scott. *Social network analysis*. Sage, 2017.

[18] P. J. Slater. A characterization of soft hypergraphs. *Canad. Math. Bull.*, 21(3):335–337, 1978.