Restrictions of graph partition problems.
Part I

Hans L. Bodlaender, Klaus Jansen

RUU-CS-91-44
November 1991

Utrecht University

Department of Computer Science

Padualaan 14, P.O. Box 80.089,
3508 TB Utrecht, The Netherlands,
Tel. : ... + 31 - 30 - 531454

Restrictions of graph partition problems.
Part I

Hans L. Bodlaender, Klaus Jansen

Technical Report RUU-CS-91-44
November 1991

Department of Computer Science
Utrecht University
P.O.Box 80.089
3508 TB Utrecht
The Netherlands

ISSN: 0924-3275

Restrictions of graph partition problems. Part I

Hans L. Bodlaender* Klaus Jansen†

Abstract

In this paper, the problems to partition a given graph into k independent sets or cliques of bounded size k' are analysed for several classes of graphs. We investigate the computational complexity of both problems restricted to cographs, split graphs, bipartite graphs and interval graphs given general or constant k and k'. It is shown, that the assignment problem for operations in a branching flow graph to processors, each with a limit on the number of executable operations, equals the problem to partition into independent sets, restricted to cographs. In addition a job-assignment problem given intervals for each job and k machines, each executing at most k' jobs, equals the problem to partition into independent sets restricted to interval graphs. It is shown, that both problem are NP-complete.

1 Introduction

In this paper the problems to partition a given graph into a bounded number of cliques or independent sets of bounded size are studied. The motivation of this analysis of graph partition problems is given by practical problems. One application is an assignment problem of operations given in a flow graph to processors. A flow graph is an acyclic digraph with operation nodes and independent branching nodes where the operations are executed in a given time interval. Depending on the control of the branchings only a subset of all operations must be executed. For this flow graphs we identify the set of fork nodes by F, the set of join nodes by J and the set of operation nodes by Op. To allow a branch in the flow graph we give the edges $e = (f, v)$ going from the fork nodes $f \in F$ away additionally a weight $w_e \in \{0, 1\}$.

*Department of Computer Science, Utrecht University, P.O.Box 80.089, 3508 TB Utrecht, The Netherlands. The work of this author was partially supported by the ESPRIT II Basic Research Actions Program of the EC under contract no. 3075 (project ALCOM).

†Fachbereich IV, Mathematik und Informatik, Universität Trier, Postfach 3825, W-5500 Trier, Germany.

Definition 1.1 *Each branching flow graph can be represented recursively in the following form:*

- *A digraph $D = (\{v\}, \emptyset)$ with $F = J = \emptyset$ and $Op = \{v\}$ is a flow graph.*
- *A digraph $D = (V, E)$ given by the union of two disjoint flow graphs $D_i = (V_i, E_i)$, $i \in \{0,1\}$ and by adding a subset of edges of the set*

$$\{(v,w) | d_{out}(v) = d_{in}(w) = 0, v \in V_i, w \in V_k, i \neq k\}$$

which generates no cycle is a flow graph.

- *A digraph $D = (V, E)$ given by the union of two disjoint flow graphs $D_i = (V_i, E_i)$, $i \in \{0,1\}$, a new fork node f, a new join node j and additionally edges $\{(f,v) | v \in V_i, d_{in}(v) = 0\}$ with weight i and edges $\{(v,j) | v \in V_i, d_{out}(v) = 0\}$ for each i is a flow graph.*

In addition there are execution times for the operations v given by intervals I_v such that for each pair of operations $v \neq w$ with directed path from v to w in the digraph and $x \in I_v, y \in I_w$, $x < y$.

We note that the constructed flow graph is acyclic and that the sizes of F and J are equal. Using the first graph operation each partial order can be constructed. The second operation constructes a branch with a fork node f at the top and a join node j at the bottom. We see also that each node in this digraph lies on a directed path from f to j. Each flow graph has m pairs of fork and join nodes specified by the definition. Using such a pair we can divide the flow graph into different parts. Let $V(f,j)$ be the vertices v lying on a directed path from f to j, and let $V_i(f,j)$ be the operations in $V(f,j)$ which can reached over a i-weighted edge going away from the fork node f. Using a control function $\psi : F \to \{0,1\}$ for the fork nodes, the set of executed operations for ψ is defined by

$$Op_\psi = Op \setminus \bigcup_{f \in F} V_{(1-\psi(f))}(f,j).$$

For these flow graphs an incompatibility graph can be defined with an edge between two operations if they are executed at a common time slot and if they can be executed depending on the control of the branching nodes together.

Definition 1.2 *Two operations v, w of a flow graph are compatible, if one of next conditions is satisfied:*

- *$I_v \cap I_w = \emptyset$.*
- *there is no control function $\psi : F \to \{0,1\}$ and $\{v,w\} \subset Op_\psi$.*

In [5] it was shown that the incompatibility graphs can be classified as the intersection of a cograph and an interval graph and that the coloring problem restricted to these graphs remains NP-complete. Since compatible operations which can be assigned to the same processor form a clique in the complement graph, the problem of finding an assignment with a minimum number of processors is NP-complete.

If we have no branching nodes we get the interval graphs as incompatibility graphs and if all operations have unit-time length we get the cographs. For these graph classes there are linear time algorithms [4, 9] for the coloring problem. But if we allow that only a given number of operations can be assigned to each processor, we get a coloring problem for cographs or interval graphs such that for each color, there are at most k' operations with that color. A solution of this problem gives us an assignment such that each processor must execute only a bounded number of operations.

For the interval graphs there is another application. Let T be a set of jobs with start-time $s(t)$ and end-time $e(t)$ of the execution and let M be a set of k machines. We search for an assignment of jobs to machines where each job is executed by exactly one machine and where each machine can only execute one job at the same time. If each machine can execute only a bounded number of jobs we get the same coloring problem for interval graphs.

If we have jobs with unit time-length equal one, a partial order $P = (T, A)$ for the set of jobs, m machines and a deadline D and if we ask for a m machine schedule for T that meets the deadline D and obeys the precedence constraints, we get the classical PRECEDENCE CONSTRAINED SCHEDULING problem. We can show that if the partial order is an interval order, we get the partition into bounded cliques problem, restricted to interval graphs.

2 Definition of the problem

An important combinatorial problem is the coloring problem of an undirected graph $G = (V, E)$. A k-coloring is a mapping $f : V \to \{1, \ldots, k\}$ with for all edges $\{v, w\} \in E$, $f(v) \neq f(w)$. A set U is called independent if each pair $v, w \in U$ with $v \neq w$ is not connected by an edge. The k-coloring problem corresponds to the problem of finding a partition of the vertices into k independent sets. Karp [7] has shown that the coloring problem is NP-complete for general undirected graphs and up to this time no polynomial-time algorithm is known for this problem.

However, the coloring problem becomes much easier, when we restrict the inputs to certain special graph classes. For example there are efficient algorithms for interval graphs [4], cographs [9] and split graphs [3].

A graph $G = (V, E)$ is an interval graph, iff to each vertex $v \in V$, a closed interval I_v in the real line can be associated, such that for each pair of vertices $u, v \in V$, $u \neq v$, $\{u, v\} \in E$, if and only if $I_u \cap I_v \neq \emptyset$. The complement G^c of an interval graph G can be transitively oriented with $(u, v) \in A$ iff $(x \in I_u, y \in I_v \to x < y)$.

This orientation A induces a partial order $P = (V, A)$. Partial orders obtained in this way are called interval orders.

Cographs are graphs without a path of length four as induced subgraph [9]. These graphs can be generated by disjoint union and join operations on graphs starting with single-vertex graphs and can be represented about these operations. For graphs $G_i = (V_i, E_i)$ with $V_1 \cap V_2 = \emptyset$ the union of G_1 and G_2, $\cup(G_1, G_2)$ is given by $(V_1 \cup V_2, E_1 \cup E_2)$. The join of G_1 and G_2, denoted by $+(G_1, G_2)$ is obtained by first taking the union of G_1 and G_2, and then adding all edges $\{v_1, v_2\}$ with $v_i \in V_i$. The join of three or more graphs $G_1, \ldots, G_r$ is obtained similary: take the disjoint union, and add all edges between vertices in different graphs G_i.

To each cograph G one can associate a corresponding rooted binary tree T, called a *cotree* of G, in the following way. Each non-leaf node in the tree is labeled with either $\cup$ (union-nodes) or $+$ (join nodes). Each non-leaf node has exactly two children. Each node of the cotree corresponds to a cograph and a leaf node to a single-vertex graph. We remark that the usual definition of cotrees allows for arbitary degree of internal nodes. However, it is easy to see that both definitions have the same power and that arbitrary cotrees can be transformed to cotrees with two children per internal node. In [1] it is shown that one can decide in $O(n + e)$ time, whether a graph is a cograph, and build a corresponding cotree.

A graph $G = (V, E)$ is called bipartite if there is a partition of the vertices V into two disjoint set V_1, V_2 where the set of edges E forms a subset of $\{\{v, w\} | v \in V_1, w \in V_2\}$. A graph $G = (V, E)$ is a split graph if there is partition of the vertices $V = U \cup C$ into an independent set U and a clique C. There is no restriction on edges between vertices of U and C. Another characterization of the split graphs is the condition that G and the complement $\overline{G}$ are both chordal graphs. A chordal graph can be represented by an intersection graph of a family of subtrees of a tree. Thus, each interval graph is chordal.

For an overview to the coloring problem restricted to different graph classes, we refer to [6]. In this paper, we consider two problems, related to the coloring problem. The first we consider, is the problem, given a graph $G = (V, E)$, and two integers k, k', to determine whether G can be partitioned into k independent sets of size at most k'. In other words, we search for a coloring of G with at most k colors, such that for each color, there are at most k' vertices with that color. We call this problem PARTITION INTO BOUNDED INDEPENDENT SETS. We also consider this problem on the complement of G. Then it becomes the following problem:

Problem: PARTITION INTO BOUNDED CLIQUES
Input: Undirected graph $G = (V, E)$, $k, k' \in \mathbb{N}$.
Question: Is there a partition of V into cliques $C_1, \ldots, C_k$ with $|C_i| \leq k'$ for $1 \leq i \leq k$?

This problem is NP-complete, because it contains the problem PARTITION INTO CLIQUES [7]. We denote with $\chi(G, k')$ the minimum number of independent sets of size at most k' that cover G and we denote with $\kappa(G, k')$ the minimum number of cliques of size k' that cover G.

In this paper, we have analysed the PARTITION INTO BOUNDED CLIQUES (INDEPENDENT SETS) problems for cographs, split graphs, bipartite graphs and interval graphs.

3 Cographs

Since the complement of a cograph is again a cograph, the same results hold for the complexity of PARTITION INTO BOUNDED CLIQUES and PARTITION INTO BOUNDED INDEPENDENT SETS when restricted to cographs.

Theorem 3.1 *The problems* PARTITION INTO BOUNDED CLIQUES (INDEPENDENT SETS) *remain NP-complete for cographs.*

Proof:
We proof the result for PARTITION INTO BOUNDED CLIQUES. Clearly the problem is in NP. To prove NP-hardness, use a transformation from the BIN-PACKING problem to the PARTITION INTO BOUNDED CLIQUES problem on cographs. An instance of the bin-packing problem is given by numbers $a_1, \ldots, a_n \in \mathbb{N}$, by $K \in \mathbb{N}$ bins and by a bin-capacity $B \in \mathbb{N}$ with $B > a_i$ and $K > 1$. The question is to decide whether there exists a partition of the set $\{1, \ldots, n\}$ in sets $I_1, \ldots, I_K$ with $\sum_{i \in I_j} a_i < B$ for each $1 \leq j \leq K$. This problem is NP-complete, see [2]. We may assume that the bin-capacity B is greater than the number n, because we can multiply the capacity B and the numbers a_i with the value n and get an equivalent problem.

For every number a_i, we construct a graph G_{a_i} in the following way. Take a complete graph with $B \cdot a_i$ vertices, $C_{B \cdot a_i}$, and take the union of this complete graph with $K-1$ independent vertices. Let $G_{a_1,\ldots,a_n}$ be the graph, obtained by taking the join of all G_{a_i} $(1 \leq i \leq n)$. We note that a maximal clique in this graph can be represented by an index set $I \subset \{1, \ldots, n\}$ such that the vertices of this clique are given by the $B \cdot a_i$ vertices in G_{a_i} for each $i \in I$ and by one of the $K-1$ independent vertices in G_{a_i} for each $i \notin I$. Now we can prove the following equivalence.

There is a partition $I_1, \ldots, I_K$ of $\{1, \ldots, n\}$ with $\sum_{i \in I_j} a_i < B$ for $1 \leq j \leq K$ iff the graph $G_{a_1,\ldots,a_n}$ has a partition into K cliques each of size at most $B^2 - B + n$.

First suppose we have a partition $I_1, \ldots, I_K$ of $\{1, \ldots, n\}$, with $\sum_{i \in I_j} a_i < B$ for $1 \leq j \leq K$. We take for each index set I_i the corresponding clique C_i described above. Then the cliques $C_1, \ldots, C_K$ form a partition into cliques and the sizes $|C_j|$ can be bounded by

$$\sum_{i \in I_j} B \cdot a_i + (n - |I_j|) \leq B(B-1) + n.$$

Now, suppose we have a partition of $G_{a_1,\ldots,a_n}$ into cliques $C_1, \ldots, C_K$ such that each clique has at most $B(B-1)+n$ vertices. By the construction of the graph each clique C_j must be a maximal clique; otherwise we can not cover the whole graph. Therefore the cliques C_j can be represented by their index sets I_j. Now take these index sets as solution of the bin packing problem. To prove that we obtain a correct solution of the bin packing problem in this way, consider the following inequality.

$$|C_j| = \sum_{i \in I_j} B{\cdot}a_i + (n - |I_j|) \leq B(B-1) + n.$$

By using $|I_j| \leq n < B$ we get:

$$\sum_{i \in I_j} a_i \leq \frac{B(B-1) + |I_j|}{B} < \frac{B^2}{B} = B.$$

□

Since the problems to partition into k cliques or independent sets of size at most k' are NP-complete, we look at instances with constant k or k'. When $k' = 3$ and $k = |V|/3$, we get the problem PARTITION INTO TRIANGLES. For constant k' we can use the recursive structure of the cograph.

We consider vectors $x = (x_1, \ldots, x_{k'})$ where x_i gives the number of cliques of size i used for the cograph. We call such a vector $x = (x_1, \ldots, x_{k'})$ *feasible with respect to* G, if there is a partition of G into cliques where x_i cliques have size i for $1 \leq i \leq k'$. For every graph G, let $L(G)$ be the set of feasible vectors. We compute a partition into a minimum number of cliques of size at most k' by computing sets $L(G)$. If $V = \{v\}$ then the set $L(G)$ consists of one vector $(1, 0, \ldots, 0)$. The generation of these vector sets can be done recursively on the cograph.

Lemma 3.2 *Let* $G = (V, E)$ *be a cograph. If* $G = \cup(G_1, G_2)$ *then*

$$L(G) = \{x + y | x \in L(G_1), y \in L(G_2)\}.$$

For $G = +(G_1, G_2)$, $z \in L(G)$, *if and only if there exist* $x \in L(G_1)$, $y \in L(G_2)$ *and a mapping* $f : \{(i,j) \mid i, j \geq 0,\ 1 \leq i + j \leq k'\} \to N_0$, *such that:*

$$\begin{array}{lll}(1) & \sum_{j | i+j \leq k'} f(i,j) = x_i & 1 \leq i \leq k' \\ (2) & \sum_{i | i+j \leq k'} f(i,j) = y_j & 1 \leq j \leq k' \\ (3) & \sum_{i,j | i+j=h} f(i,j) = z_h & 1 \leq h \leq k'.\end{array}$$

Proof:
For the union the assertion is clear. For the join let $x \in L(G_1)$ and $y \in L(G_2)$ and let $C_1, \ldots, C_k$ be a partition into cliques of size at most k'. A clique in G of size k' is given either by a clique of size k' in G_1 or G_2 or is given by a clique of size i with $1 \leq i < k'$ in one of both graphs and of size $k' - i$ in the other. For a clique of size less than k' we have a similar representation. Using x_i for the numbers of cliques of size i we can describe an assignment of cliques in G_1, G_2 by such a mapping f. □

We note that we must only store $k'-1$ of the components of the vector, because the last component is given by the number of vertices and the other components. Therefore only $O(n^{k'-1})$ vectors are possible for each cograph.

Theorem 3.3 *The problems* PARTITION INTO BOUNDED CLIQUES (INDEPENDENT SETS) *of at most a constant size k' can be computed in polynomial time.*

Proof:
We consider PARTITION INTO BOUNDED CLIQUES; for the other problem PARTITION INTO BOUNDED INDEPENDENT SETS, consider the complement of the input graph. For every node of the cotree, we compute the set $L(H)$, where H is the cograph, associated to the node. Note that the number of feasible vectors in a set $L(G)$ is polynomial, for constant k'. The computation for the set $L(G)$, when G is obtained by the union of G_1 and G_2, can be done in $O(n^{2(k'-1)})$ steps, given the sets $L(G_1)$ and $L(G_2)$. Since the set $\{(i,j)|i,j \in \mathbb{N}_0, 1 \leq i+j \leq k'\}$ has only a constant number $\frac{k' \cdot k' + 3k'}{2} \leq k' \cdot k' + 1$ of elements, for each vector pair x, y there are at most polynomial $O(n^{k' \cdot k' + 1})$ many feasible mappings f. Therefore, if G is obtained by the join of G_1 and G_2, $L(G)$ can be determined in polynomial, namely $O(n^{k'(k'+2)-1})$ many steps. In this way, we can compute $L(H)$ for every cograph H, associated with a node of the cotree. At the end, we can choose a feasible vector $x \in L(G)$ with a minimum number of cliques, i.e. with $\sum_{i=1}^{k} x_i$ minimum over all $x \in L(G)$.
□

A similar approach we can use for the problems where the number of cliques k is constant. For these problems we describe partitions into cliques $C_1, \ldots, C_k$ by the sizes $|C_1|, \ldots, |C_k|$ and consider sequences of sizes instead of subsets of the vertices V. We call a sequence of sizes *feasible*, if it corresponds to a partition. Since k is constant the number of different feasible sequences for a graph can be bounded by a polynom $O(n^k)$. Using that one component is given by the size $|V|$ and the other components, we must store only $O(n^{k-1})$ sequences. Now we give a recursive formula for the sets $S(G)$ of feasible sequences and a cograph G. If $V = \{v\}$ then $S(G) = \{1\}$. Let $\circ$ denote the concatenation of two sequences and let $\ell(s)$ denote the length of a sequence s. Given a sequence s with $\ell(s) \leq k$ we denote with $\overline{s}$ the sequence of length k, obtained from s by adding zero or more zeros at the end of the sequence s.

Lemma 3.4 *Let $G = (V, E)$ be a cograph and let k be a positive integer. If $G = \cup(G_1, G_2)$ then*

$$S(G) = \{t = s \circ s' | s \in S(G_1), s' \in S(G_2), \ell(t) \leq k\}.$$

If $G = +(G_1, G_2)$ then

$$\begin{aligned} S(G) = \{w_1 + u_1 \ldots w_k + u_k |\ & w = \overline{s}, s \in S(G_1), t \in S(G_2), \\ & w_i + u_i \leq k' \text{ for } 1 \leq i \leq k, \\ & \text{and } u \text{ is a permutation of } \overline{t}\}. \end{aligned}$$

Proof:
For the union the assertion is clear. Let us consider the disjoint join of two graphs $G_i = (V_i, E_i)$. Let $C_1, \ldots, C_{k_1}$ and $C'_1, \ldots, C'_{k_2}$ be partitions of G_1 and G_2 into cliques with $|C_i|, |C'_i| \leq k'$ and with $k_1, k_2 \leq k$. By adding to the collections $C_1, \ldots, C_{k_1}$, and $C'_1, \ldots, C'_{k_2}$ a number of empty sets, we can assume that $k_1 = k_2 = k$. Now for every permuation π of $\{1, \ldots, k\}$, we have that $C_1 \cup C'_{\pi(1)}, \ldots, C_k \cup C'_{\pi(k)}$ is a partition of G into cliques. If the sizes are bounded by k' we get a solution of $S(G)$.

For the other direction let $C_1, \ldots, C_h$ be a partition into $h \leq k$ cliques of size $|C_i| \leq k'$. By intersection of these cliques with vertices V_1 and V_2 we get partitions of G_1 and G_2 with the properties described above. □

Theorem 3.5 *The problems* PARTITION INTO K BOUNDED CLIQUES (INDEPENDENT SETS) *of size at most k' with constant k can be solved in polynomial time for cographs.*

Proof:
We use that each set $S(G)$ contains at most $O(n^{k-1})$ sequences. Compute for each node of the cotree the set $S(H)$ with H the cograph corresponding to that node, after these sets have been computed for the children of the node. From lemma 3.4 it follows that these computations cost at most $O(k \cdot k! \cdot n^{2(k-1)})$ time. Hence, the total time for the algorithm is bounded by $O(n^{2(k-1)+1})$ time, given a constant k. □

4 Split graphs

In this section we consider the problem PARTITION INTO BOUNDED CLIQUES for split graphs. The same results hold for PARTITION INTO BOUNDED INDEPENDENT SETS for split graphs, as the complement of a split graph is again a split graph. We transform the problem to a covering problem, which may — at first sight — seem a hard problem. However, we show that it can be solved efficiently, using a maximum flow algorithm. We use the following problem.

Problem: P_1
Input: Sets $S_1, \ldots, S_m \subset \{1, \ldots, n\}$, $\ell, h \in \mathbb{N}$
Question: Do there exist subsets $A_i \subset S_i$ with $|A_i| \leq \ell$ and $|\cup_{i=1}^m A_i| \geq h$?

Lemma 4.1 *The problem* PARTITION INTO BOUNDED CLIQUES *for split graphs is equivalent to P_1.*

Proof:
Consider the vertices $u_i \in U$ of the independent set and the neighbors $\Gamma(u_i) \subset C$. Define $n = |C|$, $m = |U|$, $S_i = \{j | v_j \in \Gamma(u_i)\}$ for $i \in \{1, \ldots, m\}$. Note that each clique can contain at most one vertex in U. An optimal solution of the partition

problem is obtained by finding the largest h for which this instance of P_1 has a solution, and then taking cliques $\{u_i\} \cup A_i$ of one vertex in U and at most $k' - 1$ vertices in C. The remaining vertices in C' can be covered in an optimal way.

It is not hard to see, that the construction above can be reversed. Hence, a solution of the partition problem on split graphs can also be transformed to a solution of P_1. □

Lemma 4.2 *Problem P_1 can be solved in polynomial time.*

Proof:
Define the following digraph

$$\begin{aligned} D = (\quad & \{s,t\} \cup \{S_1, \ldots, S_m\} \cup \{1, \ldots, n\}, \\ & \{(s, S_i) | 1 \le i \le m\} \cup \{(j,t) | 1 \le j \le n\} \cup \{(S_i, j) | j \in S_i\}). \end{aligned}$$

with capacities $0 \le c(s, S_i) \le min(|S_i|, \ell)$, $0 \le c(S_i, j), c(j,t) \le 1$. Then a maximum flow in this digraph corresponds with the largest h, for which P_1 has a solution. □

Theorem 4.3 *The problems* PARTITION INTO BOUNDED CLIQUES (INDEPENDENT SETS) *for split-graphs can be solved in polynomial time using a max flow algorithm.*

5 Bipartite graphs

Cliques in a bipartite graph have at most the size two and therefore the problem PARTITION INTO BOUNDED CLIQUES can be solved in polynomial time using a matching algorithm. But for independent sets the situation is more difficult.

Theorem 5.1 *The problem* PARTITION INTO THREE BOUNDED INDEPENDENT SETS *remains NP-complete for bipartite graphs.*

Proof:
We give a transformation from CLIQUE to this partition problem on bipartite graphs. Let $G = (V, E)$ be a graph and $k > 5$. We assume that the number of edges is bounded by: $|E| < |V| + k(k-2)$. It is not hard to see that CLIQUE remains NP-complete under this restriction. To prove this, take the graph $G' = (V \cup V', E)$, obtained from G by adding a disjoint set of isolated vertices V' with $|V'| = |E|$. Then G' has a clique of size k iff G has a clique of size k and $|E(G')| = |E| < |V| + |E| + k(k-2) = |V(G')| + k(k-2)$.

Suppose we have a graph $G = (V, E)$ with $|E| < |V| + k(k-2)$. We define a bipartite graph $G_B = (V_1 \cup V_2, E_B)$ with $V_1 = V \cup A$, $V_2 = E \cup B$. The number of vertices in A is $|V| - 2k + \frac{k(k-1)}{2}$ and the number of vertices in B is $|V| + k(k-2) - |E|$. Since we have the inequalities $|E| < |V| + k(k-2)$ and $k > 5$ the number of vertices in A and B are positive. As edges we take all pairs $\{a, v_2\}, \{v_1, b\}$ with $a \in A, b \in B$ and $v_i \in V_i$ and for an edge $e = \{v, w\} \in E$ we take $\{v, e\}$ and $\{w, e\}$.

Now we can prove the following equivalence: G has a clique of size k iff G_B has a partition into three independent sets of size $k' = |V| - k + \frac{k(k-1)}{2}$.

Let C be a clique of size k and let E_C be the edges with both endpoints in C. Define as independent sets: $U_1 = A \cup C$, $U_2 = (V \setminus C) \cup E_C$ and $U_3 = B \cup (E \setminus E_C)$. The union of these sets are the vertices of the bipartite graph and the sizes of the independent sets are $|U_1| = |U_2| = |U_3| = k'$.

Now let $V_B = U_1 \cup U_2 \cup U_3$ be a partition into independent sets where each independent set is bounded by k'. Since the number of vertices equals $3k'$, all independent sets must have the same size k'. At first consider the set A. If the set A is distributed in two different sets U_1 and U_2, the vertices in B and E must lie in the last set U_3. Since the size $|U_3| \geq |B| + |E| = k' + \frac{k(k-1)}{2} > k'$ is to large in this case, the set A is a subset of one of the independent sets. The same can be proved for B. W.l.o.g. assume $A \subset U_1$ and $B \subset U_3$. To get k' independent vertices in U_1 and U_2 we can only add k vertices from V to U_1 and $|E| - \frac{k(k-1)}{2}$ vertices from E to U_3. The remaining vertices from V and from E must form the independent set U_2. Define the set C as the k vertices from V in U_1. Then this set forms a clique of the graph G. Otherwise U_2 can not be independent. □

Now we analyse the problem PARTITION INTO BOUNDED INDEPENDENT SETS of at most constant size k'.

Lemma 5.2 *Let $G = (V, E)$ be a bipartite graph with partition $V = V_1 \cup V_2$, $E \subset \{\{v_1, v_2\} | v_i \in V_i\}$ and let $a = |V_1| mod(k') + |V_2| mod(k')$ and $b = \frac{|V_1|+|V_2|-a}{k'}$. Then we get*

$$\chi(G, k') \begin{cases} = & b & \text{for } a = 0 \\ \in & \{b+1, b+2\} & \text{for } 0 < a \leq k' \\ = & b+2 & \text{otherwise.} \end{cases}$$

If exactly one of the terms $|V_i| mod(k') = 0$, we get $\chi(G, k') = b + 1$.

Proof:
We get the assertion by using the following inequality for a bipartite graph

$$\lceil \frac{|V_1| + |V_2|}{k'} \rceil \leq \chi(G, k') \leq \lceil \frac{|V_1|}{k'} \rceil + \lceil \frac{|V_2|}{k'} \rceil$$

The first inequality is satisfied for each graph if we replace $|V_1| + |V_2|$ by $|V|$. The second holds, because we can cover both independent sets V_i with at most $\lceil \frac{|V_i|}{k'} \rceil$ independent sets of size k'. □

Now we consider the case that $a = |V_1| mod(k') + |V_2| mod(k') = k'$. In this case we need either $\frac{|V_1|+|V_2|}{k'}$ independent sets of size k' or one set more. The other cases with $0 < a \leq k'$ can be transformed by adding some isolated vertices to this problem.

We must decide whether $\chi(G,k') = b+1$ or $\chi(G,k') = b+2$, (with b as in lemma 5.2). We transform this decision problem to a problem of finding a sequence of independent sets in G. For example we get $b+1$ if we have an independent set $U = U_1 \cup U_2$ with $U_i \subset V_i$ and $|U_i| = |V_i| mod(k')$. But in general it is possible to have several independent sets which eliminate the overhanging $|V_i| mod(k')$ in both sides.

Lemma 5.3 *Let $G = (V, E)$ be a bipartite graph with partition $V = V_1 \cup V_2$, $E \subset \{\{v_1, v_2\} | v_i \in V_i\}$ and let $a = |V_1| mod(k') + |V_2| mod(k') = k'$.*

Then $\chi(G,k') = \frac{|V_1|+|V_2|}{k'}$ if and only if there is a sequence of $t > 0$ pairwise disjoint independent sets $U_{a_i,b_i} = U_{a_i} \cup U_{b_i}$ of size k' which satisfies the following conditions:

$$\begin{array}{ll} (1) & U_{a_i} \subset V_1, 0 < |U_{a_i}| = a_i < k' \\ (2) & U_{b_i} \subset V_2, 0 < |U_{b_i}| = b_i < k' \\ (3) & (\sum_{i=1}^{t} a_i) mod(k') = |V_1| mod(k') \\ (4) & (\sum_{i=1}^{t} b_i) mod(k') = |V_2| mod(k'). \end{array}$$

Proof:
By simple calculation. □

We can omit condition (4), as it follows from condition (1) – (3), because all independent sets have size k' and because $|V_1| mod(k') + |V_2| mod(k') = k'$.

Theorem 5.4 *The problem* PARTITION INTO BOUNDED INDEPENDENT SETS *of size at most k' with constant k' can be solved in polynomial time for bipartite graphs.*

Proof:
By using of Lemma 5.3. We search for sequences of independent sets of size k' with the given conditions. We can show at first that the length t of these sequences can be bounded by $k'-1$.

Consider $c = |V_1| mod(k') > 0$ and sequences of t pairwise disjoint independent sets U_{a_i,b_i} of size k' with $(\sum_{i=1}^{t} a_i) mod(k') = c$. We must only consider sequences where no subsequence satisfies these conditions. Using this fact the values $(\sum_{i=1}^{t'} a_i) mod(k')$ with $1 \le t' \le t$ must all be different and must lie between 1 and $k'-1$. Therefore we can bound t by $k'-1$.

Since the values $a_i \in \{1, \ldots, k'-1\}$ we have only a constant number of these sequences which we must consider. For each such sequence $a_1 \ldots a_t$ we can test in polynomial time whether G has such a sequence of disjoint independent sets, because the size of these sets can be bounded by the constant $(k'-1)^{k'-1}$. □

6 Interval graphs

In this section we show that the problem PARTITION INTO BOUNDED CLIQUES can be solved in linear time. For the other problem we can show the NP-completeness even if the sizes of independent sets are bounded by a constant. At first we give a relation between interval orders and interval graphs. This relation can be used for the problem PARTITION INTO BOUNDED CLIQUES.

Lemma 6.1 *Let $G = (V, E)$ be an interval graph, let $P = (V, A)$ be the corresponding interval order and let $k, k' \in \mathbb{N}$. Then the following equivalence is satisfied: There is a partition of G into k cliques of size at most k' iff there is a feasible schedule of P with unit-times which needs at most k' machines and k time steps.*

Proof:
Let $C_1, \ldots, C_k$ be a partition of G into cliques with $|C_i| \leq k'$. For each clique C_i there is at least one point x on the real line with $x \in \bigcap_{v \in C_i} I_v$. We assume that the cliques are ordered according to these points on the real line. Define $T(v) = i$ if $v \in C_i$. We now prove that T gives a feasible schedule. Let $v, w \in V$ with $(v, w) \in A$. Since the interval I_v lies on the left side of I_w the corresponding cliques C_i with $v \in C_i$ and C_j with $w \in C_j$ satisfy $i < j$ and therefore we have $T(v) = i < j = T(w)$. The number of vertices at each time step is less or equal k' and the number of steps is k.

Now let $T : V \to \{1, \ldots, k\}$ be a feasible schedule where for each $1 \leq i \leq k$ we have $|\{v | T(v) = i\}| \leq k'$. Define $C_i = \{v | T(v) = i\}$. If C_i is not a clique there are vertices $v, w \in C_i$ with $\{v, w\} \notin E$. This means that the intervals $I_v \cap I_w = \emptyset$. Therefore I_v lies to the left or to the right side of I_w. In both cases we have an arc in P and hence, we have not a feasible schedule. Therefore each set C_i is a clique and we get a partition into k cliques with $|C_i| \leq k'$. □

Theorem 6.2 *The problem* PARTITION INTO BOUNDED CLIQUES *can be solved in linear time on interval graphs.*

Proof:
Apply lemma 6.1 and solve the scheduling problem for interval orders using a linear time algorithm by Papadimitriou and Yannakakis [8]. □

We get a different result, for the problem to partition into independent sets, on interval graphs.

Theorem 6.3 *The problem* PARTITION INTO BOUNDED INDEPENDENT SETS *remains NP-complete for interval graphs.*

Proof:
We can use basically the same transformation as in the proof of theorem 3.1. (Use that the complement of $G_{a_1,\ldots,a_n}$ is an interval graph.) □

For the problem with constant number k of independent sets we can use the same approach as for the cographs. A partition $U_1, \ldots, U_h$ with $h \leq k$ for an interval graph can be identified with a sequence of sizes $|U_1|, \ldots, |U_h|$ and the last endpoint $x_i = max_{v \in U_i} max(I_v)$ on the real line for each set U_i. Using dynamic programming we can generate all feasible sequences.

Theorem 6.4 *Given a constant k the problem* PARTITION INTO K BOUNDED INDEPENDENT SETS, *each of size at most k' can be solved in polynomial time for interval graphs.*

The complexity of PARTITION INTO BOUNDED INDEPENDENT SETS each of size at most k' is open for $k' = 3$. This problem (for $k' = 3$) contains the problem PARTITION INTO TRIANGLES for the complement of interval graphs (namely when $k = |V|/3$).

Theorem 6.5 *The problem* PARTITION INTO BOUNDED INDEPENDENT SETS *each of size at most four remains NP-complete for interval graphs.*

Proof:
We give a transformation from NUMERICAL 3-DIMENSIONAL MATCHING [2] to the partition problem. An instance of numerical 3-dimensional matching is given by disjoint sets W, X and Y each containing m elements, a size $s(a) \in \mathbb{N}$ for each element $a \in W \cup X \cup Y$ and a bound Z such that $\sum_{a \in W \cup X \cup Y} s(a) = mZ$. The question is to decide whether $W \cup X \cup Y$ can be partitioned into m disjoint sets A_i such that each A_i contains exactly one element from each of W, X and Y and such that for $1 \leq i \leq m$, $\sum_{a \in A_i} s(a) = Z$. This problem remains NP-complete if we require that $s(a) < \frac{Z}{2}$ for all $a \in W \cup X \cup Y$. This can be proved by transforming the original problem in one where this assumption holds, by adding the value Z to each $a \in W \cup X \cup Y$ and by setting $Z' = 4Z$.

Now we give the construction of a set of intervals. The interval graph that is modeled by this set of intervals forms the instance for the partition problem. Write $W = \{w_1, \ldots, w_m\}$, $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_m\}$.

1. take for each $w_i \in W$ an interval $a_i = [0, w_i]$.

2. take for each $w_i \in W, x_j \in X$ an interval $b_{i,j} = [w_i + 1, w_i + x_j + (j\,Z)]$.

3. take for each $x_j \in X, y_k \in Y$ an interval $c_{j,k} = [(j+1)Z - y_k + 1, (m+1)Z + k]$.

4. take for each $1 \leq k \leq m$ an interval $d_k = [(m+1)Z + k + 1, (m+3)Z + 1]$.

5. take for each $w_i \in W$ $(m-1)$ intervals $e_{i,\ell} = [1, w_i]$ and $(m-1)$ intervals $f_{i,\ell} = [0, 0]$.

6. take for each $1 \leq j \leq m$ $(m-1)$ intervals $g_{j,\ell} = [(j+1)Z, (m+3)Z + 1]$ and $(m-1)$ intervals $h_{j,\ell} = [0, jZ]$.

7. take for each $1 \leq k \leq m$ $(m-1)$ intervals $p_{k,\ell} = [(m+1)Z + k + 1, (m+3)Z]$ and $q_{k,\ell} = [(m+3)Z + 1, (m+3)Z + 1]$.

We give an example in Figure 1. In this example, we have $w_1 = 1$,$w_2 = 2$,$x_1 = 1$, $x_2 = 1$, $y_1 = 2$, $y_2 = 3$ and $Z = 5, m = 2$.

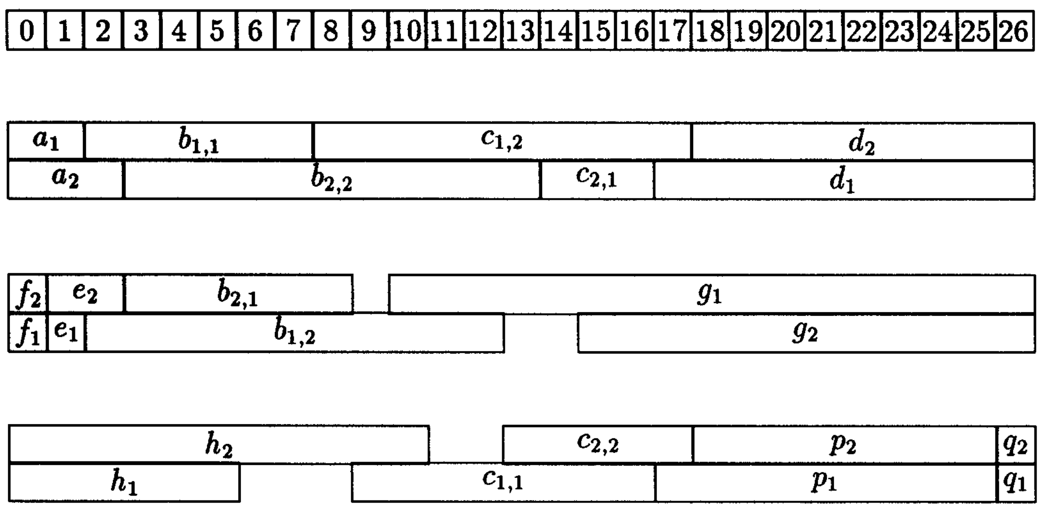

Figure 1: Example for the transformation

Denote the set of all intervals a_i $(1 \leq i \leq m)$ by A. In a similar way, define sets B, C, D, E, F, G, H, P, Q; each of these sets contains all intervals denoted with the same letter. At first let us consider which sets of vertices form a clique. These are $A \cup F \cup H$, $A \cup E \cup H$, $B \cup H$, $C \cup G$, $P \cup G \cup D$ and $P \cup Q \cup D$ and some other which depend on the instance. We note that each independent set in the interval graph has size at most five.

The sizes of the sets are $|A| = |D| = m$, $|B| = |C| = m^2$ and the other sets E, F, G, H, P, Q have size $m(m-1)$. In total, this are $8m^2 - 4m$ vertices. We consider the problem to partition the interval graph, corresponding to the set of intervals into $2m^2 - m$ independent sets of size at most four. Note that each independent set must have size exactly four.

Let $h \in H$ and consider an independent set U of size four which contains h. Then the only possibility is to choose one vertex $c \in C$, one vertex $p \in P$ and one vertex $q \in Q$ for the set U. For a vertex $g \in G$ and an independent U which contains g we can only take one vertex $b \in B$, one vertex $e \in E$ and one vertex $f \in F$. If we delete these vertices, we have only m elements of A, B, C and D.

We now study 'cuts' between two sets of vertices in the interval graph. Consider the following cuts:

(1)	$A \cup E$	and	B
(2)	$B \cup H$	and	$C \cup G$
(3)	C	and	$P \cup D$.

We consider first the last of these three cuts. We see that the sizes $|C|$ and $|P \cup D|$ are equal to m^2. Since G has $m^2 - m$ vertices and since $C \cup G$ and $P \cup D \cup G$ are cliques, we must choose for the m^2 independent sets one vertex of C and one of $P \cup D$.

Now we prove that for each vertex $c_{j,k}$ there is a vertex $p_{k,\ell} \in P$ or a vertex $d_k \in D$ such that both are together in one of the independent sets. This means that there is no independent set U with $\{c_{j,k}, p_{k',\ell}\} \subset U$ or $\{c_{j,k}, d_{k'}\} \subset U$ if $k \neq k'$. Assume that this is not the case. We have m^2 independent sets where each contains exactly one element of C and one of P or D. Let $c_{j,k}$ a vertex with minimum k which lies in an independent set with a vertex $p_{k',\ell}$ or $d_{k'}$ for $k \neq k'$. If $k > k'$ the intervals overlap and therefore this case is not possible. But if $k < k'$, the vertices with second index less than k are correctly connected. Therefore at least one of the vertices in $\{p_{k,\ell} | 1 \leq \ell \leq m-1\} \cup \{d_k\}$ must be connected to a vertex $c_{j',k''}$ with $k'' > k$. Since the corresponding intervals overlap, we get a contradiction.

Let us consider the second cut with $B \cup H$ on the left and with $C \cup G$ on the right side. Since we have $2m^2 - m$ vertices in both sets, and since both sets are cliques, each independent sets must have one element from $B \cup H$ and one from $C \cup G$. We see that $w_i + x_j + jZ < (j+1)Z$ and that the interval $c_{j,k}$ with left endpoint $(j+1)Z - y_k + 1$ intersect with $h_{j+1,\ell}$. Therefore we have the same situation as above and can prove in similar way that there is no independent set U with $\{b_{i,j}, c_{j',k}\} \subset U$ or with $\{b_{i,j}, g_{j',\ell}\} \subset U$ or with $\{h_{j,\ell}, c_{j',k}\} \subset U$ for $j \neq j'$.

For the first cut $A \cup E$ and B we get with the same argument that only vertices of $a_i, e_{i,\ell}$ are together with vertices $b_{i',j}$ if $w_i = w_{i'}$. It is possible to swap elements between independent sets, such that each a_i and each $e_{i,\ell}$ lies in an independent set together with one $b_{i,j}$.

¿From this analyse of cuts, we have that we may assume that the independent sets contain pairs of intervals, illustrated in the following table. In other words, there is for example no independent set which contains $\{a_i, b_{i',\ell}\}$ or $\{e_{i,\ell}, b_{i',\ell'}\}$ for $i \neq i'$.

first interval	second interval
a_i or $e_{i,-}$	$b_{i,-}$
$b_{-,j}$ or $h_{j,-}$	$c_{j,-}$ or $g_{j,-}$
$c_{-,k}$	d_k or $p_{k,-}$.

Now consider the interval graph after deleting all independent sets U which contain $h \in H$ or $g \in G$. We now have m independent sets U_i which cover the vertices in A, D and the remaining vertices in B, C. Using that each $g_{j,-}$ is connected to one $b_{-,j}$ and that each $h_{j,-}$ is connected to one $c_{j,-}$ we have for each j exactly one vertex $b_{-,j}$ in the rest of B and one $c_{j,-}$ in the rest of C. Therefore each independent set U_i has the form $U_i = \{a_i, b_{i,j}, c_{j,k}, d_k\}$.

Now we can prove that there is a partition of $W \cup X \cup Y$ into sets A_i with exactly one element of W, X and Y and with $\sum_{a \in A_i} s(a) = Z$ iff the constructed interval graph has a partition into $2m^2 - m$ independent sets of size at most four.

Let $U_1, \ldots, U_{2m^2-m}$ be such a partition. From the analyse above, it follows that we may assume w.l.o.g. the first m independent sets have the form $U_i = \{a_i, b_{i,j}, c_{j,k}, d_k\}$ such that the sets $\{j | b_{i,j} \in U_i, 1 \leq i \leq m\}$, $\{k | c_{j,k} \in U_i, 1 \leq i \leq m\}$ are equal to $\{1, \ldots, m\}$. Using that U_i is an independent set, we have $w_i + x_j + jZ < (j+1)Z - y_k + 1$ and therefore we get $w_i + x_j + y_k \leq Z$. Since each index appears exacly once, we have

$$\sum_{i=1}^{m} w_i + \sum_{j=1}^{m} x_j + \sum_{k=1}^{m} y_k = mZ.$$

Therefore $w_i + x_j + y_k = Z$ and the sets $A_i = \{w_i, x_j, y_k\}$ given by the intervals U_i solve the matching problem.

To prove the equivalence in other direction, let $A_i = \{w_i, x_j, y_k\}$ be the sets with $\sum_{a \in A_i} s(a) = Z$. As the first m sets we choose

$$U_i = \{a_i, b_{i,j}, c_{j,k}, d_k\}.$$

The interval a_i lies on the left side to $b_{i,j}$ and the interval d_k lies on the right side to $c_{j,k}$. To prove that $b_{i,j}$ lies on the left side to $c_{j,k}$ we compare the right endpoint of $b_{i,j}$ and the left endpoint of $c_{j,k}$. Using that $w_i + x_j + y_k = Z$, we get $w_i + x_j + (jZ) < (j+1)Z - y_k + 1$. Therefore the set U_i is independent.

Let $B' \subset B$ be the set of vertices which are not covered and construct iteratively independent sets. Let $b_{i,j} \in B'$. Then take vertices $e_{i,\ell}, f_{i,\ell}, g_{j,\ell'}$ which are not covered and put them together in one set U. Clearly, this set is independent. The construction is correct, because each index i and j appeares only $(m-1)$ times in B'. Now consider the set $C' \subset C$ of vertices which are not covered and construct in a similar way independent sets. For these take for each $c_{j,k} \in C'$ vertices $h_{j,\ell}, p_{k,\ell'}, q_{k,\ell'}$ which are not covered. After these steps all vertices are covered and we have $2m^2 - m$ independent sets. □

7 Conclusion and Applications

In the following two tables we give an overview about the results for the problems PARTITION INTO K BOUNDED CLIQUES and PARTITION INTO K BOUNDED INDEPENDENT SETS each of size at most k'. An entry NPc means that the problem is NP-complete and an entry P that the problem can be solved in polynomial time. The complexity of PARTITION INTO BOUNDED INDEPENDENT SETS is open for interval graphs for $k' = 3$.

graph class	general	constant k	constant k'
cographs	NPc	P	P
split graphs	P	P	P
bipartite graphs	P	P	P
interval graphs	P	P	P

Table 1: Complexity for bounded clique partition

graph class	general	constant k	constant k'
cographs	NPc	P	P
split graphs	P	P	P
bipartite graphs	NPc	NPc	P
interval graphs	NPc	P	NPc ($k' \geq 4$)

Table 2: Complexity for bounded independent set partition

We conclude the paper with the complexity of the applications mentioned in the introduction. The first problem we consider is to find an assignment of operations in a branching flow graph to a minimum number of processors.

Theorem 7.1 *The problem to decide whether there is an assignment of the unit-time operations in a branching flow graph to k processors where each processor can only execute at most k' operations is NP-complete. If one of the integers k or k' is constant, the problem can be solved in polynomial time.*

We get also a consequence for the job-assignment problem where each machine has a limit of licences.

Theorem 7.2 *Given a set T of jobs with interval times and k machines, where each machine can execute only k' jobs. The problem to find an assignment of the jobs to the machines, where each machine executes at most one job per time, is NP-complete even for constant k'. It can be solved in polynomial time if the integer k is constant.*

References

[1] D.G. Corneil, Y. Perl, and L.K. Stewart, A linear recognition algorithm for cographs, SIAM J. Comput. **4** (1985), pp. 926 – 934.

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

[3] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, London, 1980.

[4] U.I. Gupta, D.T. Lee and J.Y.-T. Leung, Efficient algorithms for interval graphs and circular arc graphs, Networks **12** (1982), pp. 459 – 467.

[5] K. Jansen, The processor optimization problem, to appear in Theor. Comp. Science.

[6] D.S. Johnson, The NP-completeness column: an ongoing guide, J. Algorith. **6** (1985), pp. 434 – 451.

[7] R.M. Karp, Reducibility among combinatorial problems, in: Miller and Thatcher: Complexity of Computer Computations, Plenum Press (1972), pp. 85 – 104.

[8] C.H. Papadimitriou and M. Yannakakis, Scheduling interval-ordered tasks, SIAM J. Comp. **8** (1979), pp. 405 – 409.

[9] D. Seinsche, On a property of the class of n-colorable graphs, J. Comb. Theory **B 16** (1974), pp. 191 – 193.