Computational complexity of partitioning a graph into a few connected subgraphs

J. Bensmail[1]

1: LaBRI, Bordeaux University
Talence, France

GT G&A, LaBRI
November 9th, 2012

Part 1: Partitioning a graph into a few connected subgraphs

Part 2: Partitioning a graph following vertex prescriptions

Part 3: Partitioning a graph into arbitrarily many connected subgraphs

Part 4: Conclusions and open questions

Let us consider the following definition...

Def. *Realizable sequence - Realization*
Let G be a graph. A sequence $\tau = (n_1, ..., n_p)$ of positive integers summing up to $|V(G)|$ is *realizable in* G if there exists a partition $(V_1, ..., V_p)$ of $V(G)$ such that every V_i has size n_i and induces a connected subgraph of G. The partition $(V_1, ..., V_p)$ of $V(G)$ is a *realization of* τ *in* G.

... and the associated decision problem.

REALIZABLE SEQUENCE - REALSEQ
Instance: A graph G and a sequence τ.
Question: Is τ realizable in G?

On the complexity of REALSEQ

It is already known that REALSEQ is an NP-complete problem even when:

- $\tau = (k, \ldots, k)$, where $k \geq 3$ is a divisor of $|V(G)|$ [DF85];
- G is a tree with maximum degree 3 [BF06].

These results were proved by reduction from the PLANAR 3-DIMENSIONAL MATCHING and EXACT COVER BY 3-SETS problems, respectively.

It is already known that REALSEQ is an NP-complete problem even when:

- $\tau = (k, ..., k)$, where $k \geq 3$ is a divisor of $|V(G)|$ [DF85];
- G is a tree with maximum degree 3 [BF06].

These results were proved by reduction from the PLANAR 3-DIMENSIONAL MATCHING and EXACT COVER BY 3-SETS problems, respectively.

However, in any instance of REALSEQ resulting from one of these reductions, the size of τ is polynomial in the size of the original instance. Therefore, these reductions do not involve the existence of a constant threshold $t \geq 1$ such that the following problem

> REALIZABLE SEQUENCE WITH SIZE k - k-REALSEQ
> Instance: A graph G and a sequence τ with size k.
> Question: Is τ realizable in G?

is in P when $k \leq t - 1$ and NP-complete otherwise.

On the complexity of k-RealSeq

Since partitioning G into one single connected component is possible iff G is connected, we have $t \geq 2$.

We here prove that $t = 2$ as follows.

1. First, we show that 2-RealSeq is NP-complete.
2. We then explain how to generalize the reduction used to k-RealSeq for any $k \geq 3$.

Since partitioning G into one single connected component is possible iff G is connected, we have $t \geq 2$.

We here prove that $t = 2$ as follows.

1. First, we show that 2-REALSEQ is NP-complete.
2. We then explain how to generalize the reduction used to k-REALSEQ for any $k \geq 3$.

Let us first show that 2-REALSEQ is NP-complete by reduction from

> 1-IN-3 SAT
> Instance: A 3CNF formula F over variables $X = \{x_1, ..., x_n\}$.
> Question: Is F satisfiable in a 1-in-3 way, that is in such a way that each of its clauses has exactly one true literal?

where a *3CNF formula* is a CNF formula whose clauses have exactly three literals.

2-RealSeq is complete in NP

Thm. *B. - 2012*
2-RealSeq is NP-complete.

Proof.
First notice that k-RealSeq is in NP for every $k \geq 2$. One can provide a satisfying realization R of τ in G to an algorithm that makes sure that R is a partition of $V(G)$, and that the parts of R have the correct sizes regarding τ and induce connected subgraphs of G. This can be done in polynomial time.

2-RealSeq is complete in NP

Thm. *B. - 2012*
2-RealSeq is NP-complete.

Proof.
First notice that k-RealSeq is in NP for every $k \geq 2$. One can provide a satisfying realization R of τ in G to an algorithm that makes sure that R is a partition of $V(G)$, and that the parts of R have the correct sizes regarding τ and induce connected subgraphs of G. This can be done in polynomial time.

We now show that 1-in-3 SAT $\leq_p$ 2-RealSeq. From a given 3CNF formula F over variables $\{x_1, \dots, x_n\}$ and clauses $\{C_1, \dots, C_m\}$ we construct a graph G_F and a sequence $\tau = (n_1, n_2)$ with $n_1, n_2 \geq 2$ such that

$$F \text{ is satisfiable in a 1-in-3 way} \\ \Leftrightarrow \\ \tau \text{ is realizable in } G_F.$$

Reduction from 1-IN-3 SAT to 2-REALSEQ

Thm. *B. - 2012*
2-REALSEQ is NP-complete.

We may suppose that every literal appears in F - if x_i does not appear in F, then

$$F' = F \wedge (x_i \vee \overline{x_i} \vee x_{n+1}) \wedge (x_{n+1} \vee \overline{x_{n+1}} \vee x_{n+1})$$

is satisfiable in a 1-in-3 way iff F is satisfiable in this way too.

Thm. *B. - 2012*
2-REALSEQ is NP-complete.

We may suppose that every literal appears in F - if x_i does not appear in F, then

$$F' = F \wedge (x_i \vee \overline{x_i} \vee x_{n+1}) \wedge (x_{n+1} \vee \overline{x_{n+1}} \vee x_{n+1})$$

is satisfiable in a 1-in-3 way iff F is satisfiable in this way too.

We first construct the *clause subgraph* of G_F:

- with each literal l_i in F is associated a *literal vertex* v_{l_i} in G_F;
- every pair of literal vertices $\{v_{l_i}, v_{\overline{l_i}}\}$ is linked to the root of a star S^i with n vertices of degree 1;
- a pair of literal vertices $\{v_{l_i}, v_{l_j}\}$ is linked to the root of a star $S^{i,j}$ with n vertices of degree 1 if l_i and l_j appear in a same clause of F;
- all the literal vertices of G_F are linked to the root of a new star S^c with n vertices of degree 1.

Thm. *B. - 2012*
2-REALSEQ is NP-complete.

Let n_2 be the number of vertices of the clause subgraph. Then

$$n_2 \leq 2n + n(n+1) + 3m(n+1) + n + 1$$
$$n_2 \leq n(n + 3m(1 + 1/n) + 4) + 1.$$

Thm. *B. - 2012*
2-REALSEQ is NP-complete.

Let n_2 be the number of vertices of the clause subgraph. Then

$$n_2 \leq 2n + n(n+1) + 3m(n+1) + n + 1$$
$$n_2 \leq n(n + 3m(1 + 1/n) + 4) + 1.$$

G_F is finally augmented with a *base subgraph* as follows:

- for each clause C_i in F, we add a new *clause vertex* v_{C_i} to G_F;
- each vertex v_{C_i} is linked to $n_2 - n$ vertices of degree 1;
- for each $i \in \{1, ..., m-1\}$, we add $v_{C_i}v_{C_{i+1}}$ to $E(G_F)$;
- if $C_i = (l_{i_1} \vee l_{i_2} \vee l_{i_3})$, then we add $v_{C_i}v_{l_{i_1}}$, $v_{C_i}v_{l_{i_2}}$ and $v_{C_i}v_{l_{i_3}}$ to $E(G_F)$.

We added $n_1 = m(n_2 - n + 1)$ vertices to G_F, and we thus have $|V(G_F)| = n_1 + n_2$.

Thm. B. - 2012
2-REALSEQ is NP-complete.

Let us consider $\tau = (n_1 + n, n_2 - n)$.

Observe that if a part U of a realization of τ in G_F contains the root of an induced star, then U also has to cover all the vertices of degree 1 of that star. Hence, in a realization (V_1, V_2) of τ in G_F, the base subgraph has to be covered by the part V_1 of size $n_1 + n$.

Once the base subgraph is covered by V_1, this part is missing n additional vertices from the clause subgraph of G_F. Because of the structure of the clause subgraph, we may only pick up some literal vertices. It has to be done in such a way that the clause subgraph remains connected.

Reduction from 1-in-3 SAT to 2-RealSeq

Thm. *B. - 2012*
2-RealSeq is NP-complete.

Let us consider $\tau = (n_1 + n, n_2 - n)$.

Observe that if a part U of a realization of τ in G_F contains the root of an induced star, then U also has to cover all the vertices of degree 1 of that star. Hence, in a realization (V_1, V_2) of τ in G_F, the base subgraph has to be covered by the part V_1 of size $n_1 + n$.

Once the base subgraph is covered by V_1, this part is missing n additional vertices from the clause subgraph of G_F. Because of the structure of the clause subgraph, we may only pick up some literal vertices. It has to be done in such a way that the clause subgraph remains connected.

Choosing a literal vertex v_{l_i} to belong to V_1 is like setting l_i true. In particular:

- two covered literal vertices cannot be both linked to a same clause vertex;
- two covered literal vertices cannot be related to a variable of F and its negation.

Finally, a realization of τ in G_F exists iff F is satisfiable in a 1-in-3 way. Moreover, G_F has a polynomial number of vertices regarding the size of F. Thus, this reduction can be performed in polynomial time. ∎

Reduction from 1-IN-3 SAT to 3-REALSEQ

We now explain how to modify our reduction from 1-IN-3 SAT to 2-REALSEQ so that we get a reduction from 1-IN-3 SAT to k-REALSEQ for any $k \geq 3$.

Thm. *B. - 2012*
k-REALSEQ is NP-complete for every $k \geq 3$.

Proof.
k-REALSEQ is in NP for every $k \geq 3$ as claimed before. As an illustration of our statement above, we here only show that 3-REALSEQ is NP-complete by reduction from 1-IN-3 SAT.

We now explain how to modify our reduction from 1-IN-3 SAT to 2-REALSEQ so that we get a reduction from 1-IN-3 SAT to k-REALSEQ for any $k \geq 3$.

Thm. *B. - 2012*
k-REALSEQ is NP-complete for every $k \geq 3$.

Proof.
k-REALSEQ is in NP for every $k \geq 3$ as claimed before. As an illustration of our statement above, we here only show that 3-REALSEQ is NP-complete by reduction from 1-IN-3 SAT.

Given a 3CNF formula F we construct a graph G_F and a sequence $\tau = (n_1, n_2, n_3)$ with $n_1, n_2, n_3 \geq 2$ such that

$$F \text{ is satisfiable in a 1-in-3 way} \\ \Leftrightarrow \\ \tau \text{ is realizable in } G_F.$$

Thm. *B. - 2012*
k-REALSEQ is NP-complete for every $k \geq 3$.

By performing the reduction from 1-IN-3 SAT to 2-REALSEQ, we get, from F, a graph G'_F and a sequence $\tau' = (n'_1, n'_2)$ with $n'_1, n'_2 \geq 2$ such that F is satisfiable in a 1-in-3 way iff τ' is realizable in G'_F.

The graph G_F is then obtained as follows:

- consider the disjoint union of G'_F and a star $S_{n'_1+n'_2+1}$ whose root is denoted by r;
- add an edge between r and an arbitrary vertex v of G'_F.

Finally, let $\tau = (n'_1 + n'_2 + 1, n'_1, n'_2)$.

Reduction from 1-IN-3 SAT to 3-REALSEQ

Thm. *B. - 2012*
k-REALSEQ is NP-complete for every $k \geq 3$.

Obviously, if a realization (V_1, V_2) of τ' in G'_F exists, then (U, V_1, V_2), where U contains all the vertices from the star subgraph of G_F, is a correct realization of τ in G_F. Conversely, in a realization (U, V_1, V_2) of τ in G_F, all the vertices of the new star subgraph have to be contained in U since otherwise $G_F - U$ would contain too many small connected components. Therefore, (V_1, V_2) is a realization of τ' in G'_F.

Hence, we get that τ is realizable in G_F iff τ' if realizable in G'_F. By transitivity, we get that τ is realizable in G_F iff F is satisfiable in a 1-in-3 way. ■

Thm. *B. - 2012*
k-RealSeq is NP-complete for every $k \geq 3$.

Obviously, if a realization (V_1, V_2) of τ' in G'_F exists, then (U, V_1, V_2), where U contains all the vertices from the star subgraph of G_F, is a correct realization of τ in G_F. Conversely, in a realization (U, V_1, V_2) of τ in G_F, all the vertices of the new star subgraph have to be contained in U since otherwise $G_F - U$ would contain too many small connected components. Therefore, (V_1, V_2) is a realization of τ' in G'_F.

Hence, we get that τ is realizable in G_F iff τ' if realizable in G'_F. By transitivity, we get that τ is realizable in G_F iff F is satisfiable in a 1-in-3 way. ■

Clearly, this graph and sequence augmentation can be repeated as many times as wanted to prove that k-RealSeq is NP-complete for any fixed $k \geq 4$.

Another partition constraint

Let us now consider the following stronger definition...

Def. *Prescription - Realization under prescription*

A *k-prescription of G* is a sequence of k pairwise distinct vertices $(v_1, ..., v_k)$ of G. If $k \leq \|\tau\|$, we say that τ is *realizable in G under* $(v_1, ..., v_k)$ if there exists a realization $(V_1, ..., V_p)$ of τ in G such that for every $i \in \{1, ..., k\}$ we have $v_i \in V_i$.

... and the associated decision problem.

PRESCRIPTIBLE SEQUENCE - PRESCSEQ

Instance: A graph G, a sequence τ and a k-prescription P of G with $k \leq \|\tau\|$.
Question: Is τ realizable in G under P?

PrescSeq is complete in NP

Prescriptible Sequence - PrescSeq
Instance: A graph G, a sequence τ and a k-prescription P of G with $k \leq \|\tau\|$.
Question: Is τ realizable in G under P?

Surprisingly, this problem has the same complexity as RealSeq without any regard to what are the size and the elements of P.

PRESCSEQ is complete in NP

PRESCRIPTIBLE SEQUENCE - PRESCSEQ
Instance: A graph G, a sequence τ and a k-prescription P of G with $k \leq \|\tau\|$.
Question: Is τ realizable in G under P?

Surprisingly, this problem has the same complexity as REALSEQ without any regard to what are the size and the elements of P.

Thm. *B. - 2012*
PRESCSEQ is NP-complete.

Proof.
One can modify the checking algorithm for REALSEQ in such a way that it also makes sure that the vertices of the prescription belong to the associated parts of the input realization. This modification does not alter the complexity of the algorithm. Therefore, PRESCSEQ is in NP.

Thm. *B. - 2012*
PRESCSEQ is NP-complete.

We now show that PRESCSEQ is complete in NP by reduction from REALSEQ. Given a graph G and a sequence τ, we construct a graph G', a sequence τ' and a prescription P of G' such that

$$\tau \text{ is realizable in } G \\ \Leftrightarrow \\ \tau' \text{ is realizable in } G' \text{ under } P.$$

Thm. *B. - 2012*
PrescSeq is NP-complete.

We now show that PrescSeq is complete in NP by reduction from RealSeq. Given a graph G and a sequence τ, we construct a graph G', a sequence τ' and a prescription P of G' such that

$$\tau \text{ is realizable in } G$$
$$\Leftrightarrow$$
$$\tau' \text{ is realizable in } G' \text{ under } P.$$

Consider a vertex v of G, and link v to one extremity of a path on a vertices for some arbitrary integer $a \geq 1$. Let us denote by u the other endvertex of this path, and by G' the resulting graph.

Then observe that if $\tau = (n_1, ..., n_p)$, then $\tau' = (a, n_1, ..., n_p)$ is realizable in G' under $P = (u)$ iff τ is realizable in G since there is only one connected subgraph of G' with order a that contains u. ■

About our reduction from REALSEQ to PRESCSEQ

Some remarks about the latter reduction.

- Our graph, sequence and prescription augmentation can be performed as many times as wanted.
- The integer values added to the prescription can be chosen arbitrarily.
- One can perform this reduction from one of the k-REALSEQ problems instead of REALSEQ.

Thanks to these, we get that PRESCSEQ is NP-complete as soon as τ has at least two part sizes that are not associated with the prescription. Besides, this statement does not depend on the size of P or on the integer values in P.

Once again, we consider a definition...

Def. *AP graph*
A graph G is *arbitrarily partitionable* if every sequence that sums up to $|V(G)|$ is realizable in G.

... and the decision problem related to it.

AP GRAPH
Instance: A graph G.
Question: Is G an AP graph?

About AP Graph

AP Graph is known to be in P when restricted to some families of graphs like

- trees with exactly one node whose degree is at least 3 [BBP02, BF06],
- split graphs [BKW09],
- etc.

The general problem is not known to belong to either NP or co-NP. Moreover, it is still unknown whether it is NP-hard. Hence, the hardness of AP Graph does not seem to be catchable thanks to the usual complexity classes at first glance.

Qst. Is AP Graph NP-hard?

About AP GRAPH

AP GRAPH is known to be in P when restricted to some families of graphs like

- trees with exactly one node whose degree is at least 3 [BBP02, BF06],
- split graphs [BKW09],
- etc.

The general problem is not known to belong to either NP or co-NP. Moreover, it is still unknown whether it is NP-hard. Hence, the hardness of AP GRAPH does not seem to be catchable thanks to the usual complexity classes at first glance.

Qst. Is AP GRAPH NP-hard?

AP GRAPH can be located in the second level of the *polynomial hierarchy*: using an oracle dealing with REALSEQ, we can easily check that a graph G is not AP. Since we can check whether an instance of AP GRAPH is a no-instance in polynomial time thanks to an algorithm dealing with a problem in NP $\cup$ co-NP, AP GRAPH is in Π_2^p.

Is AP GRAPH a Π_2^p-complete problem?

We proved that REALSEQ is NP-complete thanks to the following reduction scheme.

$$\text{SAT} \leq_p \text{3SAT} \leq_p \text{1-IN-3 SAT} \leq_p \text{REALSEQ}$$

One possible way to show that AP GRAPH is Π_2^p-complete would be to show that

$$\forall\exists\text{SAT} \leq_p \forall\exists\text{3SAT} \leq_p \forall\exists\text{1-IN-3 SAT} \leq_p \text{AP GRAPH}$$

holds. This reduction chain holds until $\forall\exists$1-IN-3 SAT by modifying the $\forall\exists\text{SAT} \leq_p \forall\exists\text{3SAT}$ and $\forall\exists\text{3SAT} \leq_p \forall\exists\text{1-IN-3 SAT}$ reductions.

Is AP GRAPH a Π_2^p-complete problem?

We proved that REALSEQ is NP-complete thanks to the following reduction scheme.

$$\text{SAT} \leq_p \text{3SAT} \leq_p \text{1-IN-3 SAT} \leq_p \text{REALSEQ}$$

One possible way to show that AP GRAPH is Π_2^p-complete would be to show that

$$\forall\exists\text{SAT} \leq_p \forall\exists\text{3SAT} \leq_p \forall\exists\text{1-IN-3 SAT} \leq_p \text{AP GRAPH}$$

holds. This reduction chain holds until $\forall\exists$1-IN-3 SAT by modifying the $\forall\exists\text{SAT} \leq_p \forall\exists\text{3SAT}$ and $\forall\exists\text{3SAT} \leq_p \forall\exists\text{1-IN-3 SAT}$ reductions.

However, our reduction from 1-IN-3 SAT to REALSEQ does not seem to be generalizable to some reduction from $\forall\exists$1-IN-3 SAT to AP GRAPH. Recall that we "translated" all the constraints attached to a formula F by adding some strong substructures to the resulting graph G. Because of these substructures, G is far from being AP.

Qst. Is AP GRAPH complete in Π_2^p?

An example of Π_2^p-complete graph partition problem

We can however imagine some Π_2^p-complete problems based on our definitions. Recall that G and $\tau = (n_1, ..., n_p)$ are a graph and a sequence that sums up to $|V(G)|$.

Def. *Partition level - Partition hierarchy - Realization under a partition hierarchy*

Let $l \in \{1, ..., p\}$. A *n_l-partition-level L_l for τ and G* is a set of subsets of $V(G)$ inducing connected subgraphs of G with order n_l. A *$(n_1, ..., n_l)$-partition-hierarchy L for τ and G* is a collection $L = (L_1, ..., L_l)$ of n_1-, ..., n_l-partition-levels for τ and G such that no subset of L_i intersects a subset of L_j for every $i \neq j$. We say that τ is realizable in G under L if for every combination of subsets $(V_1, ..., V_l)$ from L where V_i is a vertex subset of L_i, there exists a realization $(V_1, ..., V_p)$ of τ in G.

In clear, the partition hierarchy forces us to consider some given parts as the first parts of a realization of τ in G. The following decision problem

Dynamic Realizable Sequence - DynRealSeq

Instance: A graph G, a sequence $\tau = (n_1, ..., n_{p'}, n_{p'+1}, ..., n_p)$ admissible for G with $p \geq p'$ elements, and a $(n_1, ..., n_{p'})$-partition-hierarchy L for τ and G.
Question: Is τ realizable in G under L?

asks whether every partial realization of τ in G deduced from the partition-levels of L can be extended to a realization of τ in G.

Thm. *B. - 2012*
DYNREALSEQ is Π_2^p-complete.

Proof.
One can point out a combination of subsets $(V_1, ..., V_{p'})$ of L that is not extendable to a realization of τ in G. A polynomial-time algorithm can then check that the sequence $(n_{p'+1}, ..., n_p)$ is not realizable in $G - \bigcup_{i=1}^{p'} V_i$ thanks to an oracle for REALSEQ.
Therefore, DYNREALSEQ is in Π_2^p.

We now show that DYNREALSEQ is complete in Π_2^p by reduction from $\forall\exists$1-IN-3 SAT.

$\forall\exists$1-IN-3 SAT
Instance: A 3CNF formula F over variables $X \cup Y$, where $X = \{x_1, ..., x_{n'}\}$, $Y = \{x_{n'+1}, ..., x_n\}$ and $n' \leq n$, and clauses $\{C_1, ..., C_m\}$.
Question: For every truth assignment of the variables of X, does there exist a truth assignment of the variables of Y such that F is satisfied in a 1-in-3 way?

Thm. *B. - 2012*
DYNREALSEQ is Π_2^p-complete.

Our reduction is based on the reduction we gave from 1-IN-3 SAT to REALSEQ. Recall that in the latter reduction, setting a literal of F true is simulated by putting a literal vertex of G_F into the first part V_1 of a realization of τ in G_F.

We want to keep that relationship somehow. Hence, for every truth assignment ϕ_1 to the literals deduced from X, we have to check whether there is a realization of τ in G_F such that the literal vertices associated with the true literals via ϕ_1 belong to V_1.

All these possible truth assignments are simulated "dynamically" thanks to a partition-hierarchy for τ and G_F. We create the instance of DYNREALSEQ as follows.

- G_F is obtained similarly as in the reduction from 1-IN-3 SAT to REALSEQ.
- $\tau = (1, ..., 1, n_1 + n - n', n_2 - n)$.
- For every $i \in \{1, ..., n'\}$, let $L_i = \{\{v_{x_i}\}, \{v_{\overline{x_i}}\}\}$ be a 1-partition-level for τ and G_F.
- $L = (L_1, ..., L_{n'})$.

Thm. *B. - 2012*
DYNREALSEQ is Π_2^p-complete.

Observe that in the realizations of τ in G_F under L, the union of the $n' + 1$ first parts performs a connected part with size $n_1 + n$ containing n' literal vertices associated with literals over X. Thus, the arguments we pointed out to prove the correctness of the reduction from 1-IN-3 SAT to REALSEQ are still applicable here.

With every truth assignment ϕ_1 to the variables in X is associated a combination of parts from the 1-partition-levels of L. In other words, from every such ϕ_1 can be deduced a partial realization of τ in G_F whose extendibility has to be checked: if it can be extended, then we can deduce a truth assignment ϕ_2 of the variables in Y such that F is satisfied in a 1-in-3 way under ϕ_1 and ϕ_2. The converse is also true. Therefore, the reduction is correct. ∎

Part 1: Partitioning a graph into a few connected subgraphs

Part 2: Partitioning a graph following vertex prescriptions

Part 3: Partitioning a graph into arbitrarily many connected subgraphs

Part 4: Conclusions and open questions

Conclusions and open questions

1. Partitioning a graph into connected subgraphs is NP-complete even if
   - a constant number of parts greater than 2 is requested,
   - the part sizes are similar or mutually different,
   - some vertex prescriptions must be respected.

Conclusions and open questions

1. Partitioning a graph into connected subgraphs is NP-complete even if

- a constant number of parts greater than 2 is requested,
- the part sizes are similar or mutually different,
- some vertex prescriptions must be respected.

2. Partitioning a graph into small connected subgraphs with order 1 or 2 is an easy problem - this is related to the maximum matching problem. Moreover, our reduction from 1-IN-3 SAT to REALSEQ can be modified so that the resulting sequence only has 2's and 3's, but the number of 3 is linear in the size of the original instance.

> Qst. Is there a constant threshold $t \geq 1$ such that finding a realization of $(3^\alpha, 2^\beta)$ in a graph is generally easy when $\alpha \leq t - 1$ and hard otherwise?

Conclusions and open questions

1. Partitioning a graph into connected subgraphs is NP-complete even if

- a constant number of parts greater than 2 is requested,
- the part sizes are similar or mutually different,
- some vertex prescriptions must be respected.

2. Partitioning a graph into small connected subgraphs with order 1 or 2 is an easy problem - this is related to the maximum matching problem. Moreover, our reduction from 1-IN-3 SAT to REALSEQ can be modified so that the resulting sequence only has 2's and 3's, but the number of 3 is linear in the size of the original instance.

> Qst. Is there a constant threshold $t \geq 1$ such that finding a realization of $(3^\alpha, 2^\beta)$ in a graph is generally easy when $\alpha \leq t - 1$ and hard otherwise?

3. Except when restricted to some families of graphs, we still do not know much about the complexity of partitioning a graph into arbitrarily many connected subgraphs.

> Qst. What is the exact complexity of AP GRAPH?

Thank you for your attention!

References

D. Barth, O. Baudon, and J. Puech.
Decomposable trees: a polynomial algorithm for tripodes.
Discret. Appl. Math., 119(3):205–216, July 2002.

J. Bensmail.
On the computational complexity of partitioning a graph into a few connected subgraphs.
Technical Report, 2012.

D. Barth and H. Fournier.
A degree bound on decomposable trees.
Discret. Math., 306(5):469–477, 2006.

H. Broesma, D. Kratsch, and G. Woeginger.
Fully decomposable split graphs.
In *Combinatorial algorithms*, volume 5874 of *Lecture Notes in Computer Science*, pages 105–112. 20th International Workshop, IWOCA 2009, Springer, Berlin, 2009.

M.E. Dyer and A.M. Frieze.
On the complexity of partitioning graphs into connected subgraphs.
Discret. Appl. Math., 10:139–153, 1985.