The parameterized complexity of fixing number and vertex individualization in graphs

V. Arvind *Frank Fuhlbrück* Johannes Köbler
Sebastian Kuhnert Gaurav Rattan

MFCS 2016, Kraków

Outline

1. Graph Classes and Problems related to Individualization
   - Color Refinement and Individualization
   - Graph classes based on the number of individualized vertices

2. Parameterized complexity
   - The class MINI[1]
   - Overview of results
   - A W[P]-hardness proof

Outline

1. Graph Classes and Problems related to Individualization
   - Color Refinement and Individualization
   - Graph classes based on the number of individualized vertices
2. Parameterized complexity
   - The class MINI[1]
   - Overview of results
   - A W[P]-hardness proof

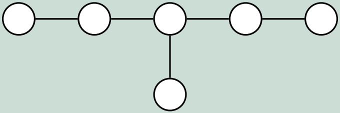
Color Refinement Procedure

(aka. 1-dim. Weisfeiler-Lehman algorithm)

Input: colored graph $X = (V, E, c)$:

1. assign a color $c'(v)$ to every vertex v such that $c'(v) = c'(u)$ iff $c(v) = c(u)$ and for each color i, u and v have the same number of i-colored neighbors.
2. If c' induces a finer color class partition than c, set $c := c'$ and goto 1
3. otherwise output the *stable coloring*.

Example



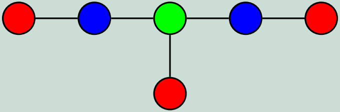
Color Refinement Procedure

(aka. 1-dim. Weisfeiler-Lehman algorithm)

Input: colored graph $X = (V, E, c)$:

1. assign a color $c'(v)$ to every vertex v such that $c'(v) = c'(u)$ iff $c(v) = c(u)$ and for each color i, u and v have the same number of i-colored neighbors.
2. If c' induces a finer color class partition than c, set $c := c'$ and goto 1
3. otherwise output the *stable coloring*.

Example



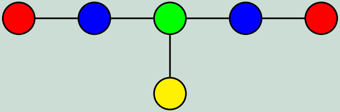
Color Refinement Procedure

(aka. 1-dim. Weisfeiler-Lehman algorithm)

Input: colored graph $X = (V, E, c)$:

1. assign a color $c'(v)$ to every vertex v such that $c'(v) = c'(u)$ iff $c(v) = c(u)$ and for each color i, u and v have the same number of i-colored neighbors.
2. If c' induces a finer color class partition than c, set $c := c'$ and goto 1
3. otherwise output the *stable coloring*.

Example



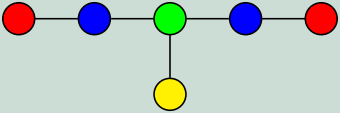
Individualization

Individualization-Refinement

1. Refinement: do color refinement
2. Individualization: If not all color classes are singletons, *individualize* a vertex in a color class of size ≥ 2 (i.e., assign an unused color to it). Goto 1.

- Used by practical isomorphism algorithms, e.g. nauty and Traces [McKay, Piperno 14].
- Here an entire tree of colorings is constructed by choosing different vertices in each step 2.

Example



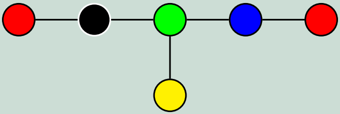
Individualization

Individualization-Refinement

1. Refinement: do color refinement
2. Individualization: If not all color classes are singletons, *individualize* a vertex in a color class of size ≥ 2 (i.e., assign an unused color to it). Goto 1.

- Used by practical isomorphism algorithms, e.g. nauty and Traces [McKay, Piperno 14].
- Here an entire tree of colorings is constructed by choosing different vertices in each step 2.

Example



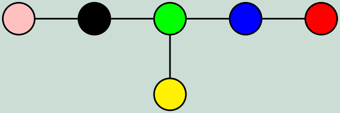
Individualization

Individualization-Refinement

1. Refinement: do color refinement
2. Individualization: If not all color classes are singletons, *individualize* a vertex in a color class of size ≥ 2 (i.e., assign an unused color to it). Goto 1.

- Used by practical isomorphism algorithms, e.g. nauty and Traces [McKay, Piperno 14].
- Here an entire tree of colorings is constructed by choosing different vertices in each step 2.

Example



Graph classes and problems

Definition (for a colored graph X)

- $X \in$ Rigid if the identity is the only automorphism of X.
- $X \in$ Refinable if two vertices u and v of X receive the same stable color only if there is an automorphism of X that maps u to v.
- $X \in$ Discrete if running color refinement on X results in singleton color classes, Discrete = Rigid $\cap$ Refinable.
- $X \in$ Discrete$[\ell]$ if $X \in$ Discrete and a stable coloring occurs after $\leq \ell$ rounds

Graph classes and problems

Definition (for a colored graph X)

- $X \in$ Rigid if the identity is the only automorphism of X.
- $X \in$ Refinable if two vertices u and v of X receive the same stable color only if there is an automorphism of X that maps u to v.
- $X \in$ Discrete if running color refinement on X results in singleton color classes, Discrete = Rigid $\cap$ Refinable.

Problem k-$\mathcal{C}$ for a class $\mathcal{C}$ of colored graphs

Input: A colored graph $X = (V, E, c)$ and an integer k

Parameter: k

Question: Are there k vertices of X so that individualizing them results in a graph in $\mathcal{C}$?

Graph classes and problems

Definition (for a colored graph X)

- $X \in$ Rigid if the identity is the only automorphism of X.
- $X \in$ Refinable if two vertices u and v of X receive the same stable color only if there is an automorphism of X that maps u to v.
- $X \in$ Discrete if running color refinement on X results in singleton color classes, Discrete = Rigid $\cap$ Refinable.

Problem $(n-k)$-$\mathcal{C}$ for a class $\mathcal{C}$ of colored graphs

Input: A colored graph $X = (V, E, c)$ and an integer k

Parameter: k

Question: Are there $|V| - k$ vertices of X so that individualizing them results in a graph in $\mathcal{C}$?

Graph classes and problems

Definition (for a colored graph X)

- $X \in$ Rigid if the identity is the only automorphism of X.
- $X \in$ Refinable if two vertices u and v of X receive the same stable color only if there is an automorphism of X that maps u to v.
- $X \in$ Discrete if running color refinement on X results in singleton color classes, Discrete = Rigid $\cap$ Refinable.

Problem k-$\mathcal{C}$ for a class $\mathcal{C}$ of colored graphs

Input: A colored graph $X = (V, E, c)$ and an integer k

Parameter: k

Question: Are there $|V| - k$ vertices of X so that individualizing them results in a graph in $\mathcal{C}$?

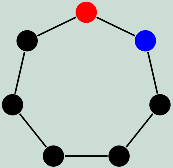
Definition (t-rigid etc.)

We say a colored graph X is *t-rigid* if $(X, t) \in k$-Rigid. t-discrete and t-refinable are defined analogously.

Examples

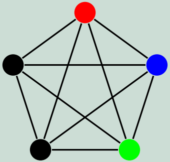
Example

For every n the circle C_n is 2-rigid, but not 1-rigid.



Example

For every n the complete graph K_n is not $(n-2)$-rigid (every graph on n vertices is $(n-1)$-rigid).



Outline

1 Graph Classes and Problems related to Individualization
Color Refinement and Individualization
Graph classes based on the number of individualized vertices

2 Parameterized complexity
The class MINI[1]
Overview of results
A W[P]-hardness proof

Parameterized complexity

Problem Mini-3SAT [Cai, Juedes 03; Downey et al. 03]

Input: A formula F in 3-CNF of size bounded by $k \log n$ and the number n in unary

Parameter: k

Question: Is F satisfiable?

Definition

The class MINI[1] contains all parameterized problems that are FPT many-one reducible to Mini-3SAT.

Parameterized complexity

Problem Mini-3SAT [Cai, Juedes 03; Downey et al. 03]

Input: A formula F in 3-CNF of size bounded by $k \log n$ and the number n in unary

Parameter: k

Question: Is F satisfiable?

Definition

The class MINI[1] contains all parameterized problems that are FPT many-one reducible to Mini-3SAT.

Lemma [Cai, Juedes 03; Downey et al. 03]

If MINI[1] = FPT then there is a $2^{o(n)}$ time algorithm for 3SAT.

Results

Problem(s)	Parameterized Complexity
k-RIGID	MINI[1]-hard
$(n-k)$-RIGID	FPT
k-$\mathcal{C}$	W[P]-hard for DISCRETE $\subseteq \mathcal{C} \subseteq$ REFINABLE
k-$\mathcal{C}$	optimization problem in logspace for graphs with color class size ≤ 3 and $\mathcal{C} \in \{$DISCRETE, REFINABLE, RIGID$\}$
$(n-k)$-DISCRETE	FPT
k-DISCRETE[ℓ]	W[2]-hard for every $\ell \geq 1$

All hardness results are w.r.t. FPT many-one reductions.

Proof

Theorem

For $\mathcal{C}$ with DISCRETE $\subseteq \mathcal{C} \subseteq$ REFINABLE, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-DISCRETE (Outline).

- Reduction from WEIGHTED MONOTONE CIRCUIT SATISFIABILITY.

Proof

Theorem

For $\mathcal{C}$ with DISCRETE $\subseteq \mathcal{C} \subseteq$ REFINABLE, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-DISCRETE (Outline).

WEIGHTED MONOTONE CIRCUIT SATISFIABILITY.

Input: A monotone boolean circuit C on n inputs and an integer k

Parameter: k

Question: Is there an input string $x \in \{0,1\}^n$ of Hamming weight k so that $C(x) = 1$?

Proof

Theorem

For $\mathcal{C}$ with DISCRETE $\subseteq \mathcal{C} \subseteq$ REFINABLE, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-DISCRETE (Outline).

- Reduction from WEIGHTED MONOTONE CIRCUIT SATISFIABILITY.
- Input: monotone circuit C, integer t, Output: graph X_C, t

Proof

Theorem

For $\mathcal{C}$ with Discrete $\subseteq \mathcal{C} \subseteq$ Refinable, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-Discrete (Outline).

- Reduction from Weighted Monotone Circuit Satisfiability.
- Input: monotone circuit C, integer t, Output: graph X_C, t
- There is some x of weight t s.t. $C(x) = 1$ iff X_C is t-discrete.

Proof

Theorem

For $\mathcal{C}$ with DISCRETE $\subseteq \mathcal{C} \subseteq$ REFINABLE, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-DISCRETE (Outline).

- Reduction from WEIGHTED MONOTONE CIRCUIT SATISFIABILITY.
- Input: monotone circuit C, integer t, Output: graph X_C, t
- There is some x of weight t s.t. $C(x) = 1$ iff X_C is t-discrete.
- Idea: represent each gate g_i by a pair $P_i = \{v_i, v_i'\}$ (also a color class) and connect the pairs by gadgets corresponding to $\wedge$ or $\vee$

Proof

Theorem

For $\mathcal{C}$ with Discrete $\subseteq \mathcal{C} \subseteq$ Refinable, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-Discrete (Outline).

- Reduction from Weighted Monotone Circuit Satisfiability.
- Input: monotone circuit C, integer t, Output: graph X_C, t
- There is some x of weight t s.t. $C(x) = 1$ iff X_C is t-discrete.
- Idea: represent each gate g_i by a pair $P_i = \{v_i, v_i'\}$ (also a color class) and connect the pairs by gadgets corresponding to $\wedge$ or $\vee$
- Setting an input gate $g_i = 1$ corresponds to individualizing v_i.

Proof

Theorem

For $\mathcal{C}$ with Discrete $\subseteq \mathcal{C} \subseteq$ Refinable, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-Discrete (Outline).

- Reduction from Weighted Monotone Circuit Satisfiability.
- Input: monotone circuit C, integer t, Output: graph X_C, t
- There is some x of weight t s.t. $C(x) = 1$ iff X_C is t-discrete.
- Idea: represent each gate g_i by a pair $P_i = \{v_i, v_i'\}$ (also a color class) and connect the pairs by gadgets corresponding to $\wedge$ or $\vee$
- Setting an input gate $g_i = 1$ corresponds to individualizing v_i.
- Output pair P_ℓ gets distinct colors iff $g_\ell = 1$

Proof

Theorem

For $\mathcal{C}$ with DISCRETE $\subseteq \mathcal{C} \subseteq$ REFINABLE, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-DISCRETE (Outline).

- Reduction from WEIGHTED MONOTONE CIRCUIT SATISFIABILITY.
- Input: monotone circuit C, integer t, Output: graph X_C, t
- There is some x of weight t s.t. $C(x) = 1$ iff X_C is t-discrete.
- Idea: represent each gate g_i by a pair $P_i = \{v_i, v_i'\}$ (also a color class) and connect the pairs by gadgets corresponding to $\wedge$ or $\vee$
- Setting an input gate $g_i = 1$ corresponds to individualizing v_i.
- Output pair P_ℓ gets distinct colors iff $g_\ell = 1$
- X_C consists of multiple copies of this graph connected to some global input pairs to assure that individualization is only sensible in the input pairs

Proof

Theorem

For $\mathcal{C}$ with Discrete $\subseteq \mathcal{C} \subseteq$ Refinable, k-$\mathcal{C}$ is W[P]-hard,
even for graphs of color class size at most 4.

Proof for k-Discrete (Gadgets).

- For $g_h = g_i \wedge g_j$ we use $\mathrm{CFI}(P_i, P_j, P_h)$[Cai, Fürer, Immerman 92]:
  Works as a parity gate for automorphisms, here it works like $\wedge$.

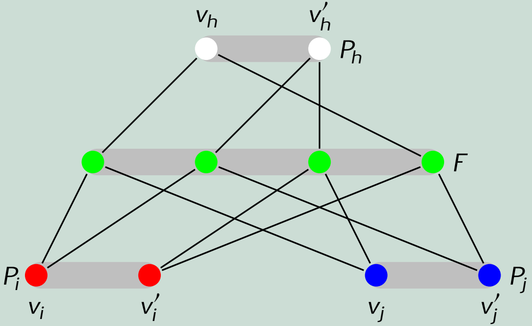
Proof

Theorem

For $\mathcal{C}$ with DISCRETE $\subseteq \mathcal{C} \subseteq$ REFINABLE, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-DISCRETE (Gadgets).

- For $g_h = g_i \wedge g_j$ we use $\mathrm{CFI}(P_i, P_j, P_h)$[Cai, Fürer, Immerman 92]:



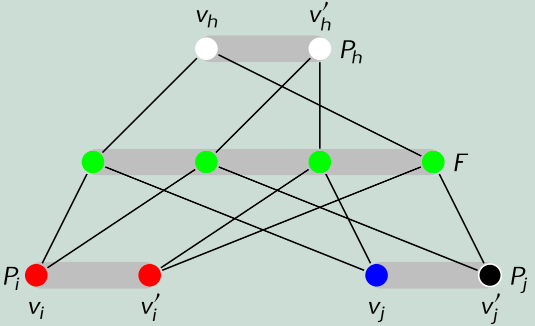
Proof

Theorem

For $\mathcal{C}$ with Discrete $\subseteq \mathcal{C} \subseteq$ Refinable, k-$\mathcal{C}$ is W[P]-hard,
even for graphs of color class size at most 4.

Proof for k-Discrete (Gadgets).

- For $g_h = g_i \wedge g_j$ we use $\mathrm{CFI}(P_i, P_j, P_h)$[Cai, Fürer, Immerman 92]:



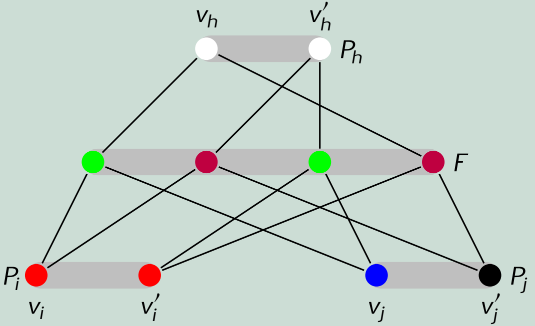
Proof

Theorem

For $\mathcal{C}$ with DISCRETE $\subseteq \mathcal{C} \subseteq$ REFINABLE, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-DISCRETE (Gadgets).

- For $g_h = g_i \wedge g_j$ we use $\mathrm{CFI}(P_i, P_j, P_h)$[Cai, Fürer, Immerman 92]:



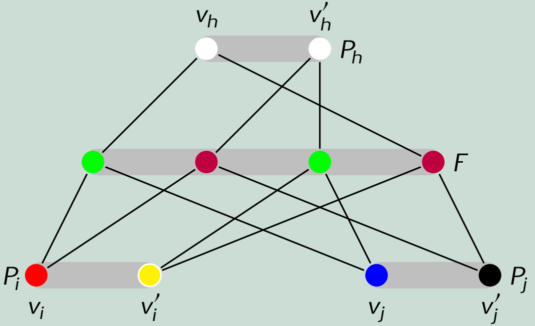
Proof

Theorem

For $\mathcal{C}$ with DISCRETE $\subseteq \mathcal{C} \subseteq$ REFINABLE, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-DISCRETE (Gadgets).

- For $g_h = g_i \wedge g_j$ we use $\mathrm{CFI}(P_i, P_j, P_h)$[Cai, Fürer, Immerman 92]:



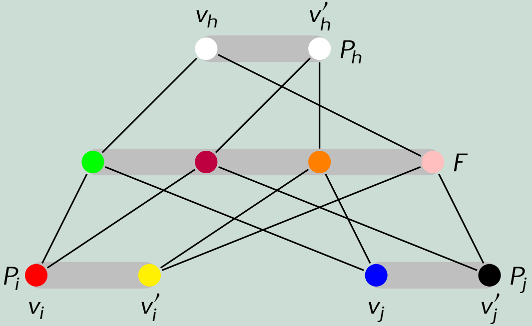
Proof

Theorem

For $\mathcal{C}$ with Discrete $\subseteq \mathcal{C} \subseteq$ Refinable, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-Discrete (Gadgets).

- For $g_h = g_i \wedge g_j$ we use $\mathrm{CFI}(P_i, P_j, P_h)$[Cai, Fürer, Immerman 92]:



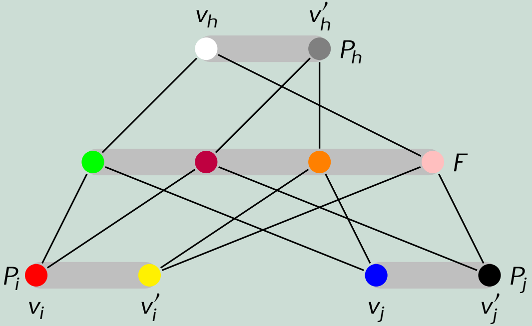
Proof

Theorem

For $\mathcal{C}$ with DISCRETE $\subseteq \mathcal{C} \subseteq$ REFINABLE, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-DISCRETE (Gadgets).

- For $g_h = g_i \wedge g_j$ we use $\mathrm{CFI}(P_i, P_j, P_h)$[Cai, Fürer, Immerman 92]:



Proof

Theorem

For $\mathcal{C}$ with DISCRETE $\subseteq \mathcal{C} \subseteq$ REFINABLE, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-DISCRETE (Gadgets).

- For $g_h = g_i \vee g_j$ we use $\mathrm{IMP}(P_i, P_h)$ and $\mathrm{IMP}(P_j, P_h)$,

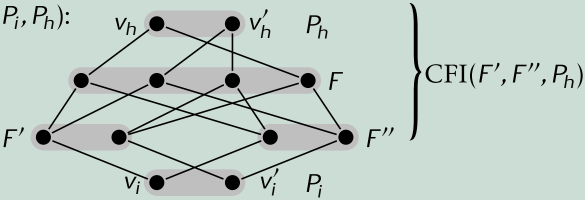
Proof

Theorem

For $\mathcal{C}$ with Discrete $\subseteq \mathcal{C} \subseteq$ Refinable, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-Discrete (Gadgets).

- For $g_h = g_i \vee g_j$ we use $\mathrm{IMP}(P_i, P_h)$ and $\mathrm{IMP}(P_j, P_h)$,
- where $\mathrm{IMP}(P_i, P_h)$:



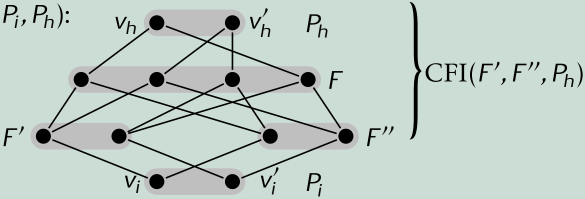
Proof

Theorem

For $\mathcal{C}$ with DISCRETE $\subseteq \mathcal{C} \subseteq$ REFINABLE, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-DISCRETE (Gadgets).

- For $g_h = g_i \vee g_j$ we use $\mathrm{IMP}(P_i, P_h)$ and $\mathrm{IMP}(P_j, P_h)$,
- where $\mathrm{IMP}(P_i, P_h)$:



- We do not connect P_i and P_h directly because F stops backflow.

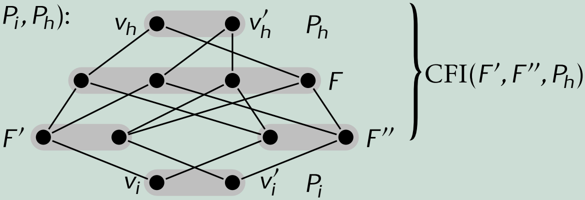
Proof

Theorem

For $\mathcal{C}$ with DISCRETE $\subseteq \mathcal{C} \subseteq$ REFINABLE, k-$\mathcal{C}$ is W[P]-hard, even for graphs of color class size at most 4.

Proof for k-DISCRETE (Gadgets).

- For $g_h = g_i \vee g_j$ we use $\mathrm{IMP}(P_i, P_h)$ and $\mathrm{IMP}(P_j, P_h)$,
- where $\mathrm{IMP}(P_i, P_h)$:



- We do not connect P_i and P_h directly because F stops backflow.
- Output pair P_ℓ is connected to all input pairs via IMP. Thus, if P_ℓ gets distinguished, all pairs will be distinguished.

□

Open problems

Open Problems

- Upper bounds (or better lower bound) for all hard problems except k-Discrete (which is W[P]-complete).
- Restriction to graph classes (or parameterizations by other parameters like degree, treewidth etc.)

Thank You!

Literature

Cai, Jin-Yi, Martin Fürer, Neil Immerman (1992). 'An optimal lower bound on the number of variables for graph identification'. In: *Combin.* 12.4, pp. 389–410.

Cai, Liming, David Juedes (2003). 'On the existence of subexponential parameterized algorithms'. In: *J. Comput. Syst. Sci.* 67.4, pp. 789–807.

Downey, Rodney G., Vladimir Estivill-Castro, Michael R. Fellows, Elena Prieto, Frances A. Rosamund (2003). 'Cutting up is hard to do: The parameterised complexity of k-cut and related problems'. In: *Electronic Notes in Theoretical Computer Science* 78, pp. 209–222.

McKay, Brendan D., Adolfo Piperno (2014). 'Practical graph isomorphism, II'. In: *J. Symb. Comput.* 60, pp. 94–112.

Relations among the problems

k-Discrete $\subseteq$ k-Rigid

If two vertices can be mapped onto each other by an automorphism, then they will receive the same color after color refinement, i.e., partition into orbits is at least as fine as partition into color classes.

k-Discrete $\subseteq$ k-Refinable

By the same argument the orbit partition cannot be finer than a set of singletons.

k-Refinable $\not\subseteq$ k-Rigid

Any transitive graph on at least 2 vertices is refinable but not rigid.

k-Rigid $\not\subseteq$ k-Refinable

Take $t+1$ copies of the Frucht graph (rigid cubic graph) as X. Then $(X, t) \in k\text{-Rigid} \setminus k\text{-Refinable}$ (also holds for the connected graph $\overline{X}$).