Notes on Online Combinatorics

Rod Downey
Victoria University
Wellington, New Zealand

ACC, December 2025

Online

- Your job is to put an unknown number of objects $\{a_0, a_1, \ldots\}$ into bins $\{b_j \mid j \in \omega\}$ of a certain maximum capacity and they are given one at a time. You must put a_i into some b_j at stage i before I give you a_{i+1} and try to minimize the number of bins used as you go along. '
- I give you an unknown graph graph G one point at a time, giving the induced subgraph $\{a_0, a_1, \ldots, a_i\}$ at step i and you must decide a colour before given a_{i+1}. Minimize the number of colours.
- You are in a vast graph and need to build, for example, an object incrementally, but don't have time to see more than a local neighbourhood.
- You are a triage nurse and patients arrive and you must order them in some priority ordering to be seen dynamically.
- You are answering processing questions about an infinite group on your laptop, but have no idea about the complexity, except hope.

Online

- In all of the above you are in an online situation, though some are different than others.
- There are hundreds of algorithms for such problems both in the finite case, and in the case where e.g. a scheduler needs to works "forever".
- There are books with taxonomies of such algorithms.
- As data gets bigger there is no doubt in my mind that algorithms concerned with incremental changes in data and their relationship with updating solutions will be ever more important.
- My goal is to give a theoretical basis for the above.
- What is a good setting? What kinds of questions can be answered for such online structures, for example, in terms of logical description, and for what kinds of parameterizations? Etc.
- One such basis has been talked about by Melnikov and Ng, and others.

Two approaches

- There seem two basic criteria needed for "online-ness".
- And there are two non-independent approaches based on them.
- The first is based on the punctuality of the online algorithm. We must do something immediately before the next item arrives. (Or leaves, this could be a Δ_2^0 process.) This has a growing and rich theory, and intersects with "temporal graphs".
- Temporal graph is one on vertices $v_1, \ldots, v_n$ for which edges $v_i v_j$ appear and disappear with time.
- The second is based on the uniformity of the operators against a hostile universe. This has almost no theory, yet!
- In this talk we'll look at the second approach, and mainly be concerned with the setting of graphs.

The Operator Approach

- Consider online colouring of a graph with the simple monotone model.
- The online algorithm A acts on G_{s+1} to (irrevocably) colour $v = s + 1$.
- You could include delay where it sees $f(s+1)$ many new points, where f would be primitive recursive, before making its decision, but we'll stick with the simple version.
- The crucial insight is that A must act *uniformly* on any sequence $G_0, \ldots, G_{s+1}, \ldots$. The offline algorithm can be considered as a sequence of algorithms $\hat{A}_s$ acting on G_s for each s.

- We think of the possible inputs as being represented by nodes in a tree of possibilities, with nodes of length n representing greaphs with n vertices.
- A node of length $n+1$ will represent a graph extending the one of length n which it is a child of.
- The key observation: whilst there are only a primitive recursive number of graphs of size s, there is no reason that the limit graph the opponent builds is even remotely primitive recursive.
- There are $2^{\aleph_0}$ many possible graphs.
- We are thinking of the algorithm acting on objects represented as paths in a computable tree. An operator.

- We could argue that any countable structure could be considered, where A_n is some kind of n-bounded fragment of the open diagram.
- But, really, in practice online structures are given in "layers". For instance, the n-th step in colouring is to consider an induced subgraph on n vertices, not, for instance, taking some enumeration of the vertices and edges and giving only part of the picture.
- Moreover functions make everything problematical. (How many iterations should we allow?)
- We want a theory which reflects practice.

Definitions

Definition

A class $\mathcal{C}$ of relational structures is called inductive if $A \in \mathcal{C}$ implies A has a filtration or online presentation $A = \cup_s A_s$ where each A_s is finite and has universe $\{1, \ldots, s\}$ and for all $s' > s$ the substructure induced by $\{1, \ldots, s\}$ in $A_{s'}$ is A_s.

Definition

A *representation* of a class $\mathcal{C}$ of structures is a surjective function $F : \omega^{<\omega} \to \mathcal{C}^{<\omega}$, which acts computably in the sense that $F(\sigma) = C_n$ for $|\sigma| = n$ and $|C_n| = n$, and if $\sigma \preceq \tau$ then $F(\sigma)$ is an induced substructure of $F(\tau)$. (Later this might be partial, and objects might have several names.)

Online

Definition

A on-line problem is a triple (I, S, s) where I is the space inputs viewed as strings in a finite or infinite computable alphabet, S is the space of outputs (solutions) viewed as strings in (perhaps, some other) alphabet, and $s : I \to S^{<\omega}$ is a function which maps I to the set of admissible solutions of σ of S.

Intuitively, to solve a problem (I, S, s) we need to find an "online" computable function f which, on input i, chooses an admissible solution from the finite set $s(i)$.

Definition

A solution to an online problem (I, S, s) is a function $f : I \to S$ with the properties:

(O1) f is computable without delay;

(O2) $f(\sigma) \in s(\sigma)$ for every $\sigma \in I$;

(O3) $f(\sigma)$ uses only σ in its computation.

Remark:

- Suppose that the representation is e.g. 2^ω, the space of infinite binary sequences, The algorithm is uniform on all paths.

Finite=Infinite

Proposition

- *Suppose that A acts in an online fashion uniformly on all finite strings. Then A acts uniformly online on all computable paths through the representing space.*
- *Suppose that the algorithm A is total and acts uniformly online on all computable paths. Then A acts uniformly on all paths.*

Proof.

(i) If A fails on some computable path α it must fail on some finite initial segment. (ii) Computable paths are dense. □

More precisely

- Suppose f is a solution to an online problem (I, S, s).
- The space of inputs carries a natural totally disconnected topology, and the completion of I forms the space of "paths" or infinite words in the language of I.
- The solution f induces a solution for the completion of the initial problem (I, S, s), in the sense that f can be uniquely extended to a functional $\overline{f} : [I] \to [S]$ between completions.
- Then $\overline{f}$ is a primitive recursive ibT operator (which means that its oracle use is bounded by the identity) with the property that, for every $n, f(p \upharpoonright n) \in s(p \upharpoonright n)$.
- In this case we say that $\overline{f}$ is a solution to the completion of (I, S, s).

Graphs; incremental computation

- There is a notion of incremental computation due to Milterson et. al. and we can show that this aligns to an online version of "Weihrauch reduction."
- We can also have ratio preserving Weihrauch reductions.
- The performance ratio of a minimization problem (e.g. coloring here) is
$$\frac{|\{f_\chi(G \upharpoonright n)\}|}{|\{\chi(G \upharpoonright n\}|}.$$
- E.g. Famously First Fit Bin Packs with Performance ratio 2. (currently 1.7)

Shape counts

- It is pretty hopeless to find properties with small performance ratios on general graphs, but we know that bounding inputs with topological parameters seems to sometimes give good online performance.
- We look at some below.

- One example as above is colouring. E.g. an interval graph of width k can be online coloured by $3k+1$ many colours reduces to chain covering of interval orderings. (Kierstead and Trotter)
- G is an interval graph of width k, if it can be represented as intervals I_x for each $x \in V$ with $I_x \cap I_y \neq \emptyset$ iff $xy \in E$, and the cutwidth is $\leq k+1$. A subgraph is said to have bounded pathwidth
- Another formulation is that G has a path decomposition with maximal bag size $k+1$. That is there is a path $w_1, \ldots, w_n$ such that each w_i has a bag B_i, with $|B_i| \leq k+1$ and such that
    - $v \in V(G)$ implies v is in some bag B_i.
    - $vw \in E(G)$ implies both v and w appear together in some B_j.
    - $v \in B_i \cap B_j$ implies $v \in B_k$ for $i \leq k \leq j$.
- If the decomposition above is a tree rather than a path, then we have tree decomposition, rather than a path decomposition.

Theorem (Askes and Downey-Building on Kierstead and Trotter, and Downey and Fellows)

]

1. *Graphs of pathwidth k can be online coloured with $3k+1$ many colours.*
2. *Hence there is a linear time to construct such colourings. (More later)*

- This is proven by induction on the width k, and for each k we recursively construct an algorithm A_k. If $k=1$ then G is a caterpillar graph and we can use greedy minimization which will use at most 3 colours.
- So suppose $k>1$, and let G_n have vertices $\{v_1, \ldots, v_n\}$. The computable algorithm A_k will have computed a partition of G, which we denote by $\{D_y \mid y<k\}$. Consider v_{n+1}. If the pathwidth of $G_{n+1} = G_n \cup \{v_{n+1}\}$ is $<k$, colour v_{n+1} by A_{k-1}, and put into one of the cells D_y, for $y<k-1$ recursively. (In the case of pathwidth 1, this will all go into D_0.) We will be colouring using using the set of colours $\{1, \ldots, 3k-2\}$.

- If the pathwidth of G_{n+1} is k, consider H_{n+1}, the induced subgraph of G_{n+1} generated by $G_{n+1} \setminus D_k$. If the pathwidth of H_{n+1} is $< k$, then again colour v_{n+1} by A_{k-1}, and put into one of the cells D_y, for $y < k - 1$, recursively, and colour using the set of colours $\{1, \ldots, 3k - 2\}$.
- If the pathwidth of H_{n+1} is k, then we put v_{n+1} into D_{k-1}. In this case, that is in D_{k-1}, we will use first fit using colours $3k - 2 < j \leq 3k + 1$.
- The remainder of the proof is an induction on the construction to show that this works, roughly showing that is the process does not work then it would have failed earlier and the pathwidth of G_n would be too big.
- This is done by taking a path decomposition of width k of G and seeing how this interacts with the algorithm.

Complexity

- We remark that the proof of the theorem above gives an algorithm which is linear time (as k-PATHWIDTH is linear time FPT), but is inefficient as the constants for the pathwidth algorithm (Bodlaender's Algorithm for treewidth.) are of the order of 2^{35k^3} which is pretty horrible.
- This algorithm is invoked at each step. We don't know the best complexity for the following (online) promise problem, which is highlighted by such considerations.
  Input: An online graph G, and a vertex v and a graph H with vertices $V(G) \cup \{v\}$ G a subgraph of H.
  Promise: G has pathwidth k.
  Parameter: An integer k.
  Question: Does H have pathwidth k?

Efficient algorithm

- Amazingly, first fit gives a good approximate algorithm.

Theorem (Dujmovic, Joret, Wood, SIAM Discrete Math, 2012)

First Fit colours graphs of pathwidth k with at most $8(k+1)$ many colours.

- The proof uses static analysis of chain decompositions of posets.

- For online applications, ratio preserving Weihrauch Reductions.

Definition

Let f, g be functions on 2^ω. Then f is called *ratio preserving online reducible to g*, $f \leq_O^r g$, if there are (type II) online computable functions A and B with and a constant d, such that for all n,

$$f(\alpha \restriction n) = A(\alpha \restriction n, g(B(\alpha \restriction n)),$$

and the ratio of $c(f(\alpha \restriction n))$ to $c(f_{\mathrm{off}}(\alpha \restriction n))$ is at most d times the ration of $c(g(B(\alpha \restriction n)))$ to $c(g_{\mathrm{off}}(B(\alpha \restriction n)))$.

Fact

If $f \leq_O^r g$ *then , for some* $d > 0$, $\frac{c(f \restriction n)}{c(f_{off} \restriction n)} \leq d \; \frac{c(g \restriction n)}{c(g_{off} \restriction n)}$.

- A classical reduction is a ratio preserving Weihrauch reduction from colouring interval graphs to chain cover for interval orderings.

- Lots other applications of this setting.
- E.g. EX-learning, Distributed computing, Büchi automata, etc.
- The idea is to somehow tie these together.
- Here are two examples, one from Tree Decompositions, and one from proof theory:

- Graphs of bounded treewidth are usually solved by tree automata.
- But if we present a graph by a root to leaf online representation, we call a promise, then the apparatus of Courcelle's Theorem on MS_2 theory of graphs of bounded treewidth applies.

Theorem (D and Long Qian)

Given a formula $\varphi(X)$ which is first order on graphs and X only occurs positively or negatively in $\varphi(X)$, then the online problem corresponding to $\varphi(X)$ has an online algorithm (the greedy algorithm) which has constant competitive ratio for graphs of bounded degree.

- Principal tool: Gaifman's Locality Lemma, and greed.
- Associated with any structure is a Gaifman graph where more or less edges correspond to relations, in the sense that x and y are joined if they occur in some tuple of the structure A together. The Locality Theorem says every first order formula is equivalent to a boolean combination of "basic local sentences" of a small radius.
- Obviously there are lots of interesting questions open here. For example, is there any analog of this result for some set of φ where there is no promise. Maybe in bounded pathwidth?

Speculation

- You can topologise, for example the space of online presentations of graphs (or graphs of bounded pathwidth) as a totally disconnected space with distance 2^{-n} at level n.
- Then considerations like the above says that we have an online algorithm acting of the paths providing a $3k+1$ approximation algorithm, and there is no continuous $k+1$ colouring.
- Perhaps the correct logic here is continuous model theory, rather than first order logic.

New directions

- Many counterexamples come from not knowing Skolem functions in an online situation.
- Imagine you are in a maze and navigating. Unless you were in Harry Potter, you are not going to get new egresses from the current location appearing. In many situations in e.g. online graphs you would expect at least to know your immediate surroundings, without a global picture.
- This idea leads to a new class of online structures with algorithmic parameterizations:

Definition

1. A locally strongly online graph is an given by a filtration (G_s, N_s) where $N_s = N(G_s)$, the neighbours of G_s in $\lim_s G_s = G$.
2. A strongly online graph G is a filtration (G_s, H_s) where $H_{s+1} = N(G_s \cup H_s \cup \{v_{s+1}\})$.

- This could be re-prhrased as adding certain function symbols to the langauge.
- Naturally realized in online algorithms on algebraic structures of unknown size.
- You can think of G_s as the blue part of the graph and $H_s - G_s$ (or $N_s - G_s$) as the red part of the graph. The online algorithm is running on the blue part of the graph only.
- In, for example, navigation we can see a space around us but only process the algorithm when we traverse it. We could also specify N_k instead of N for the k-neighbourhood of G_s.
- This accords with old work on "highly recursive" graphs. (Kierstead, Bean, Schmerl, etc)

Theorem (Askes)

If G is a strongly online graph, then G can be strongly online coloured in $2\chi(G)$ many colours.

We compare this with

Theorem (Schmerl 1980)

If G is a highly computable, k-colourable graph then, G is computably $(2k-1)$-colourable.

Theorem (Askes)

1. *For all k there is a strongly online k-colourable graph that cannot be strongly $(2k-1)$-online coloured.*
2. *There is a locally strongly online tree T which cannot be finitely coloured online.*

Pathwidth

Recall that G has pathwidth d if every vertex x can be represented as an interval I_x and such that if if $xy \in E$, then $I_x \cap I_y \neq \emptyset$. The width is the max cutqwidth of the decomposition, and this must be $\leq d+1$. The cuts determine a set of bags of size $\leq k+1$ in an order.

Theorem (Askes)

1. *Every strongly online graph, G, with strongly online pathwidth k can be strongly online coloured in $2k+1$ colours.*
2. *There is a strongly online graph with pathwidth k that cannot be strongly online coloured in $2k$ colours.*

Can online path decompositions be built?

Theorem (Askes)

For all $n > 0$, there is a strongly online graph of pathwidth 2 but whose strongly online pathwidth is $\geq n$

A sample proof

We will diagonalize against all possible φ_e, online algorithms attempting to show that the pathwidth is $\leq n$. We present $n(2n+1)$ many vertices as the centres of paths which are ever growing. The blue vertices are the presented ones which are the centre. We wait till φ_e declares bags (intervals I_x) for vertices in these paths. As G has $n(2n+1)$ many vertices into bags of size $\leq n$, there must be $2n+1$ bags containing a vertex not in the other $2n+1$ many bags.
We now join up the ends of the lines (with vertices) to force the bags to grow.
The next pictures might help.

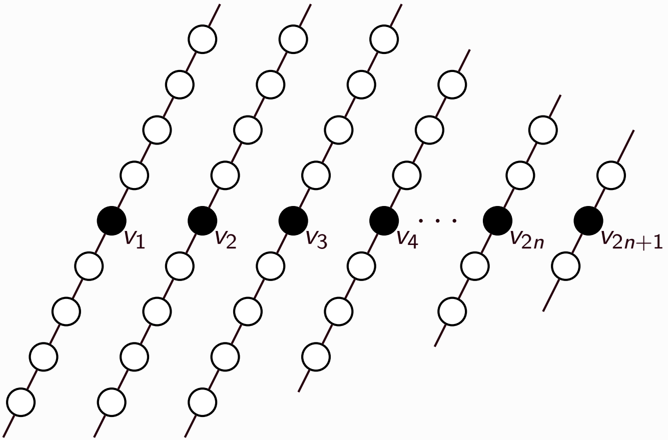

Figure: The initial $n + 1$ paths and the presented vertices (in black)

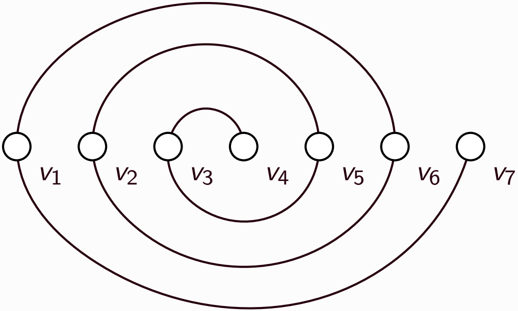

Figure: The connections between vertices (paths) for $n = 3$

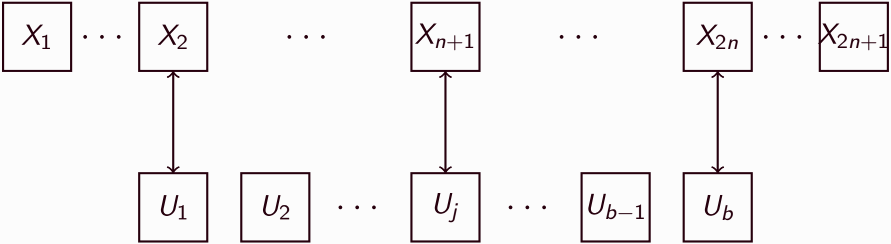

Figure: The bags in $\mathcal{Q}$ and the bags corrosponing to the path v_i—v_{2n+2-i}.

Thus X_{n+1} appears between U_1 and U_b. Hence there must be some U_j such that $U_j = X_{n+1}$. Therfore X_{n+1} contains some u_j. Then by letting i vary we can see that X_{n+1} must contain $n+1$ vertices (one from each path along with the vertex v_{n+1}). Therefore G_e satisfies R_e.

A model theoretical framework

- Speculation: We can have a constant ratio approximation scheme for first order properties if the Gaifman rank of the formula is below the online neighbourhood distance.
- Current model theory needs enriching to fit into this framework. For example, structures which seem to be reasonably well behaved online (e.g. good approximation ratios) seem to have a good shape according to some pseudo-metric on them. How to include this in model theory, whose inspiration was more about fields and the like.
- Further finitization of computability theory.

References

- Foundations of Online Structure Theory, BSL, 2019.
- Foundations of Online Structure Theory, II, https://arxiv.org/abs/2007.07401 final version in Logical Methods in Computer Science, 2021.
- M. Askes "Adversarial and Online Adversarial Games" MSc Thesis, Victoria University, in preparation.
- Lots of papers on Melnikov's and Ng's home pages.
- M. Askes and R. Downey, Online, Computable and Punctual Structure Theory, Logic Journal of the IGPL, 2023.

Thank You