On Probabilistic Activity Drag: The structure behind the distribution[1]

LETTER TO THE EDITOR

Ref: Lyaschenko, A. (2026). Probabilistic Activity Drag; *PM World Journal*, Vol. XV, Issue VII, July. Available online at https://pmworldjournal.com/wp-content/uploads/2026/07/pmwj166-Jul2026-Lyaschenko-Probabilistic-Activity-Drag.pdf

Dear Editor,

Alex Lyaschenko and I are connected on LinkedIn, and I have followed his writing on schedule optimisation for some time, so I read his featured paper on Probabilistic Activity Drag as soon as it appeared.

It is a careful and useful piece of work, and it closes a gap that has sat under the Drag family since Devaux introduced the metric in 1999. Drag has always been computed on a schedule whose durations are treated as known. Every planner knows they are not. By recomputing Drag inside each Monte Carlo iteration and aggregating the results, the paper replaces a single promised number with a range and a probability of achieving it. That is the right move, and it is made without losing the intuition that made Drag worth having.

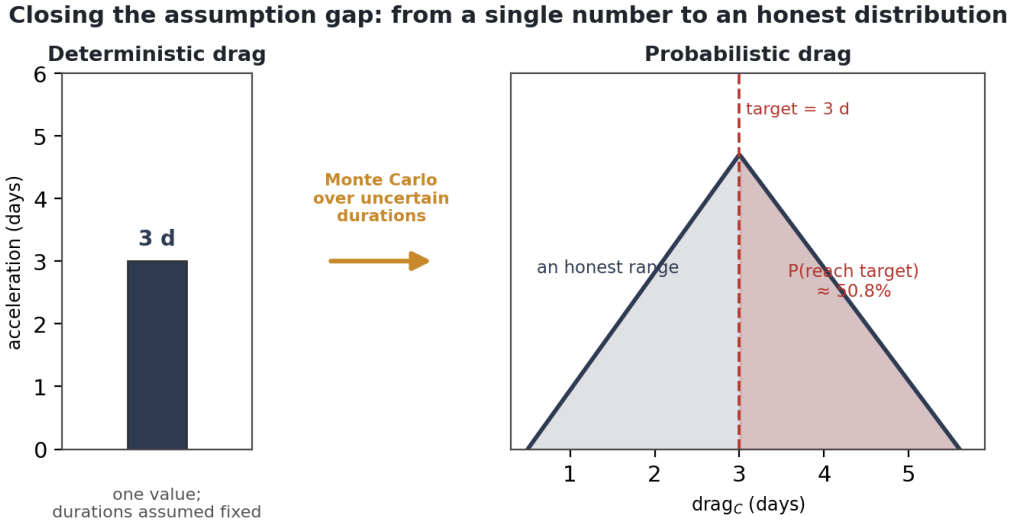

Figure 1. *The gap the paper closes. Deterministic Drag reports one number and assumes the durations feeding it are known. Recomputing Drag inside each simulated iteration returns a range and the probability of reaching the target. On the paper's own example the probability of achieving the three-day target on Activity C is 50.8%.*

[1] How to cite this work: Dhand, Nikhil (2026). On Probabilistic Activity Drag: The structure behind the distribution, Letter to the Editor, *PM World Journal*, Vol. XV, Issue IX, September.

Several judgements in the paper deserve particular credit. Insisting that the schedule be calculated with real constraints in place, including resource availability, is not a small remark: it is the difference between a Drag figure a planner can act on and one that dissolves the moment the resource histogram is honoured. The paper is candid about what the metric does not do. It states plainly that Drag is not commutative, that reducing an activity to zero is usually not the right target and that crashed duration is the better one, and that acceleration cost sits outside the model although it often decides the answer. It also warns that a DCPM-based model can mislead precisely because it ignores resource limits.

Papers that advance a metric rarely list its boundaries this openly.
It is worth saying why this matters now rather than at any point in the previous twenty years. Drag is the rare schedule metric that answers the question management does ask, which is not where the risk sits but where money spent buys time (Devaux, 1999; Duncan and Devaux, 2009). Its recognition in the eighth edition of the PMBOK Guide (Project Management Institute, 2025) puts it in front of a readership that will now try to use it, and the first thing that readership will find is that the number moves. A planner who commits acceleration cost against a deterministic Drag of five days and recovers three has not been unlucky; they were handed a point estimate of a quantity that was never a point. Putting a distribution around it before that readership arrives is timely, and the paper deserves credit for getting there first.

The worked example is well chosen, and it is worth restating in scheduling terms because everything below depends on it. Two Finish-to-Start chains run in parallel: A, ten days on resource R1, into B, four days on R2; and C, thirteen days on R3, into D, six days on R1. Only one unit of each resource exists. The project runs nineteen days and the critical path is C into D. Under duration-driven CPM, A and B carry five days of float and C and D carry five days of Drag each, as both Picture 1.1 and the table in Picture 1.2 show. The sentence beneath Picture 1.2 names A and B as the activities with positive Drag, which I take to be a slip for C and D. Nothing in the logic network connects A to D. They are joined only by both requiring the single unit of R1, and that is a resource contention resolved when the schedule is levelled, not a dependency a planner ever drew.

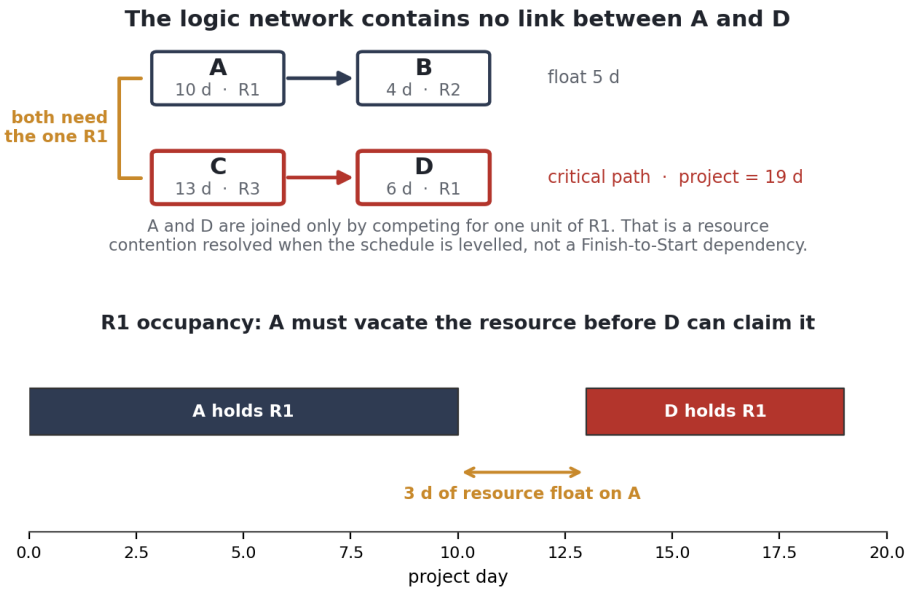

Figure 2. *The example in scheduling terms. The logic network contains only A into B and C into D. A and D are connected by nothing except competition for the one unit of R1, shown below as resource occupancy: A holds R1 until day 10, D needs it from day 13, which leaves three days of resource float on A.*

What the resource contention does to the numbers is the paper's sharpest observation. Once R1 is honoured, A can slip only three days rather than five before the project moves, because A must release R1 before D can claim it, and C's Drag falls from five days to three for the same reason: pulling C earlier only helps until D runs into the resource still held by A. D's Drag stays at five.

The author calls this a hidden resource dependency, and the phrase is exact. It is hidden because it appears in no dependency the planner entered.

One further passage deserves more attention than it will get, because it is the one that makes the method usable by readers who have no access to a system that computes any of this. The paper points out that a team can build optimistic, expected and pessimistic versions of the schedule, calculate Drag on each, and read the range off those three runs; and that where a tool will not compute Drag at all, setting the analysed activity to zero duration and re-running gives the same answer directly. That is a practical instruction a planner can follow on a Monday morning with the software already on the desk. An author advancing a metric has every incentive to make it sound as though the metric requires specialist machinery. This one does the opposite, and the field is better served by it.

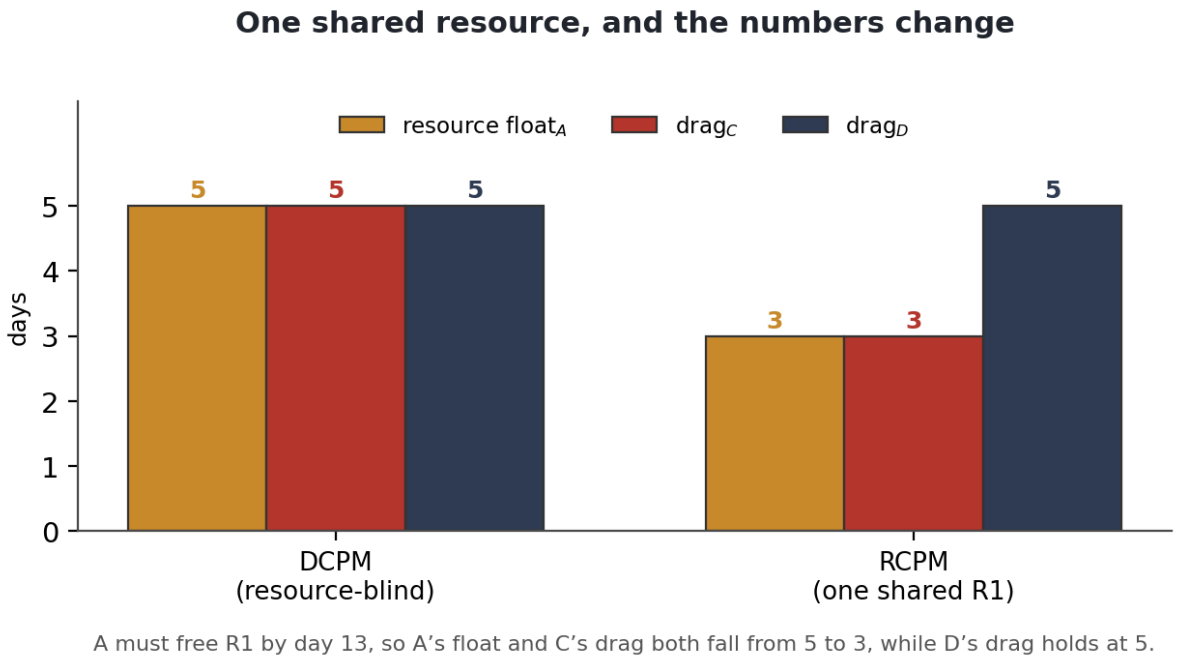

Figure 3. *The effect of honouring one shared resource. Moving from resource-blind DCPM to RCPM, A's float and C's Drag both fall from five days to three, while D's Drag holds at five. The paper's note that the two Drags do not sum to an eight-day acceleration follows directly from this.*

The simulation results then make the case concrete. Activity C returns a Drag between roughly half a day and 5.6 days with a 50.8% chance of reaching three days, and Activity D reaches its deterministic five days only 15.1% of the time although a four-day acceleration is close to even money. A planner told that the five-day acceleration they were counting on will arrive roughly one time in seven has learned something a deterministic register cannot tell them. Spider Project, which the paper notes is at present the only system implementing the method, evidently does this work with care.

One observation: the randomness is inherited, not intrinsic

There is a single place where the reported object seems to me to fall short of the structure that produced it, and it is worth stating precisely because the example makes it checkable. Within any one iteration nothing is random. The durations are drawn, and then Drag is computed from them by ordinary network arithmetic, deterministically.

On this network that arithmetic collapses to an identity. Activity D cannot start until both its predecessor C has finished and A has released R1, so its start is the later of the two, and the days that reducing C can remove are the days between C's finish and A's. In other words drag_C equals C's duration minus A's finish whenever A's release of R1 binds before the parallel path through A and B, which holds across about 96 per cent of the sampled range.

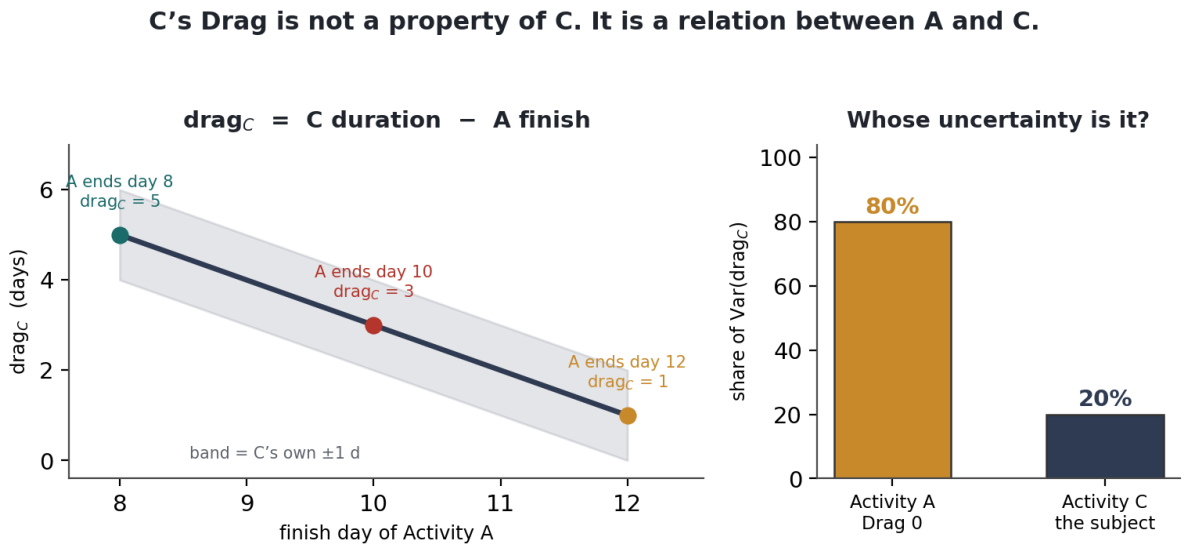

Figure 4. *Drag is a relation, not an attribute. On this network drag_C equals C's duration minus A's finish, so the quantity reported against C is a statement about A and C together. Four fifths of its spread is contributed by A, an activity that is not critical and whose own Drag is zero.*

Two consequences follow. The first is that the reported histogram for C is largely a picture of Activity A, which carries plus or minus two days against one day for C and so contributes about eighty per cent of the variance in C's Drag while holding a Drag of zero itself. The second is that the 50.8% has a reason: both durations are symmetric about their expected values, so their difference is symmetric about three days and the probability of clearing a three-day target sits very close to one half. The simulation is recovering a property of the network rather than an accident of sampling, and the gap between 50.8% and 50% is of the order of sampling error at ten thousand iterations.

What the simulation returns is co-movement, and co-movement has no direction

This leads to what I think is the substantive limitation, and it is a limitation of the shape of the output rather than of the method. Drag is defined by removal: take the activity out, recompute the network, record what the project saved. Run that inside a Monte Carlo and each iteration produces a set of Drag values that were computed on the same network from the same draws, so they necessarily move together. Aggregate ten thousand iterations and the co-movement is strong. On the paper's own inputs the Drags of C and D correlate at about 0.81.

It is worth being exact about where the direction goes, because it is not lost in the arithmetic. It is lost in the handover. Inside the network Drag is a derived quantity: it exists only as a consequence of the links, and it inherits their orientation, so drag_C is a statement about A and C in that order and about nothing else. The moment Drag is taken up as a simulation parameter, it stops being derived and becomes an attribute, a number attached to an activity row alongside its duration and its resource. The row

records the value. It does not record the parent that produced it. Ten thousand iterations later the output is a column of Drag samples per activity, and the only relationship those columns can still express is that they rise and fall together. Direction has not been calculated away; it was simply not carried across, and correlation is what is left when it is not.

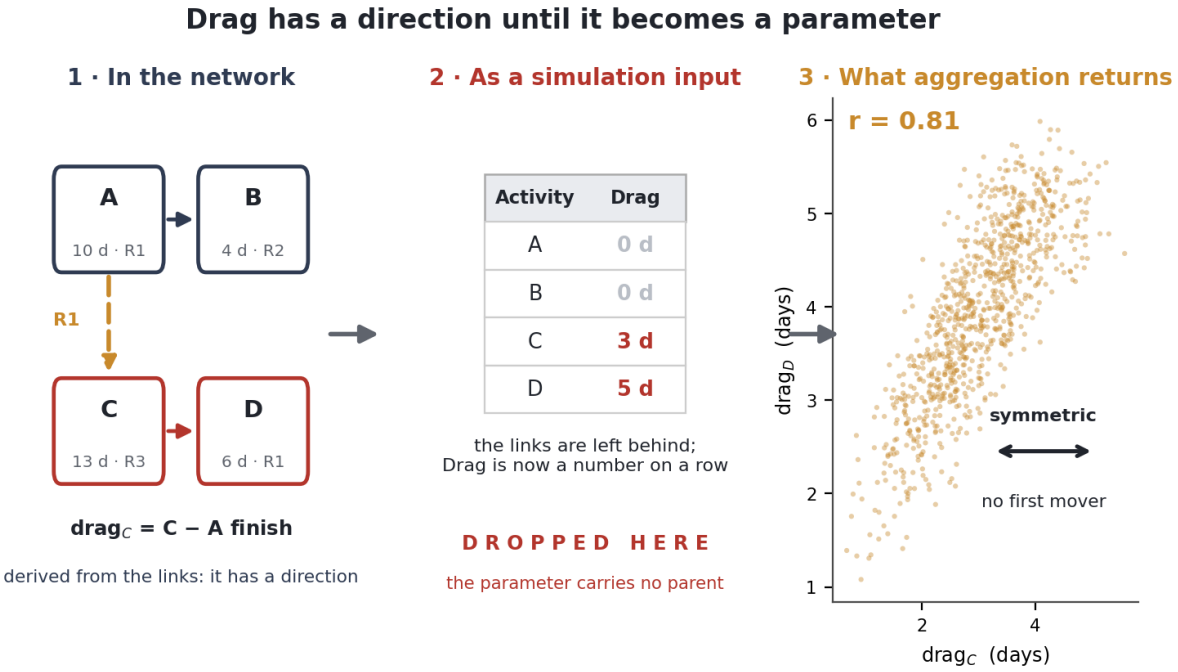

Figure 5. *Where the direction is lost. Inside the network Drag is a derived quantity and carries the direction of the links that produced it. Lifted out as a simulation input it becomes a number on an activity row, and the parent that produced it is not carried with it. What aggregation returns is co-movement between columns of samples, symmetric and silent on which activity moves first.*

This matters because the two readings recommend different things. Correlation is symmetric, so it reports that C and D move together and stays silent on which moves first, which is the only question an acceleration decision asks. If the 0.81 is taken as the relationship, then C and D are the pair to manage and every co-moving activity has an equal claim on the budget. But hold Activity A's finish fixed and that correlation falls to about 0.52.

Read as shared variation, the two Drags hold about sixty-five per cent in common before A is accounted for and about twenty-seven per cent after it is, so the majority of what looks like a relationship between C and D is a third activity acting on both. And the strongest pairwise association among the four durations and the two Drags is not between the published Drags at all. It is between C's Drag and Activity A's finish, at about minus 0.90. The paper anticipates part of this when it observes that an activity with zero Drag may still hold an optimisation opportunity once uncertainty is considered.

The point here is slightly different: A is not a hidden opportunity in its own right, it is the ceiling on someone else's.

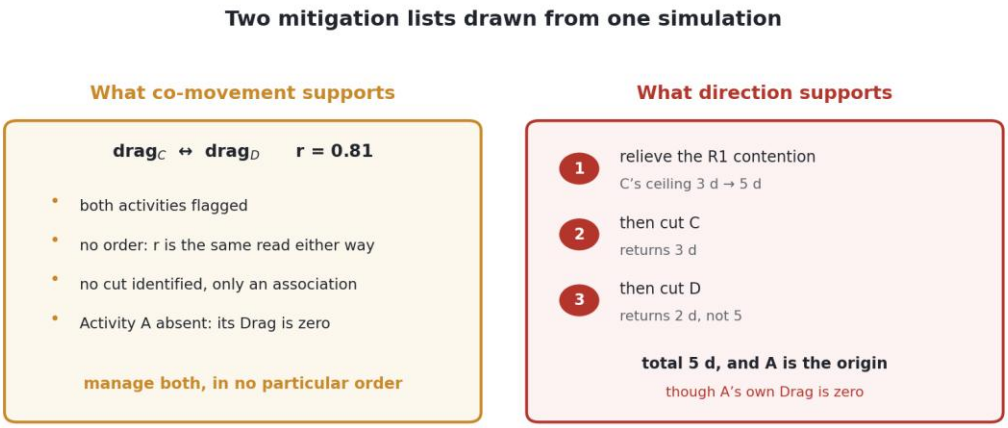

Figure 6. *Two lists from the same run. Co-movement flags both C and D, gives no order because the correlation reads the same in either direction, and never mentions A. The directed reading names a constraint to relieve, an order to cut in, and the correct second figure of two days rather than five, and it identifies A as the origin although A's own Drag is zero.*

Drag is a property of a chain, not of an activity

Once the direction is restored, a second point follows, and it is the one that matters most for how the metric is used. Drag is reported activity by activity, so it is read as though each activity owned a quantity. On a constrained network it does not. Accelerating anything re-prices everything downstream of it, and the register has to be recomputed after every cut. Following the paper's own example through two accelerations shows how quickly this bites. Cut C by its three days and the project falls to sixteen, but D is now worth two days rather than five. Cut D by that two and the project falls to fourteen, and at that point every activity in the network, all four of them, has a Drag of zero. The register reports no remaining opportunity.

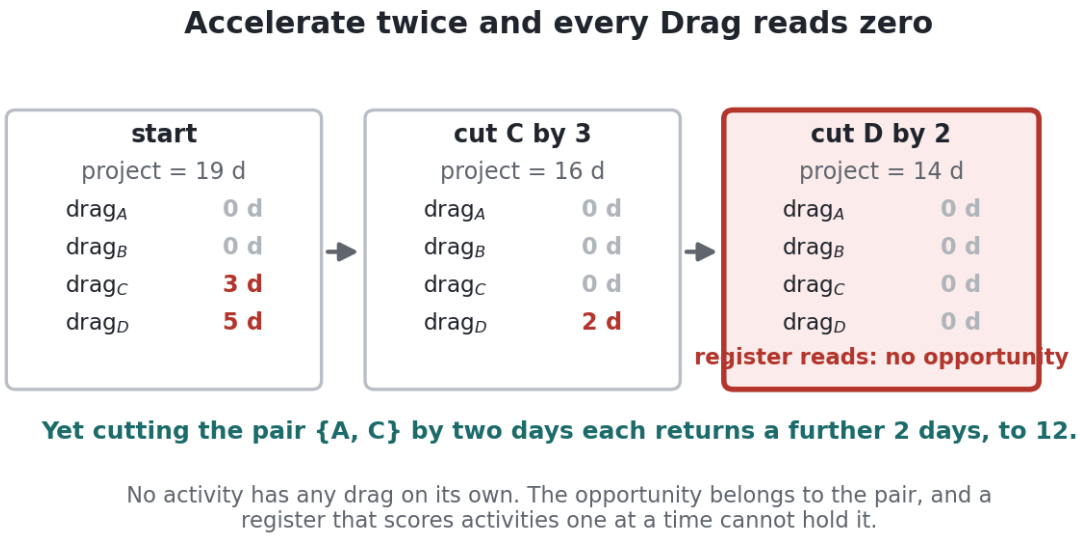

Figure 7. *The register goes blind. After two accelerations that the metric itself recommends, every activity has a Drag of zero and the register reports no further opportunity. Cutting the pair A and C together by two days each nevertheless returns a further two days. The opportunity belongs to a set of activities, and a metric that scores them one at a time cannot represent it.*

The project can still be accelerated. Two parallel paths now bind at fourteen days, so shortening either alone achieves nothing and shortening both achieves something: taking two days from A and two from C brings the project to twelve. Each activity has a Drag of zero and the pair has a Drag of two. Several pairs will do it, but not every pair, and the exception is the instructive one. Taking two days from B and two from C leaves the project at fourteen, because shortening C cannot start D any earlier while A is still holding R1. Which pairs work is a resource question, and no column of per-activity Drags contains the answer.

This is not a defect in Devaux's definition, which is exact for what it defines. It is a limit of what any per-activity number can carry, and it appears in the paper's four-activity example after two moves. On a programme of any size, where an activity of small individual Drag may sit at the head of a chain whose members carry a great deal of it, the effect is not an edge case. Ranking by Drag returns the activities where the acceleration is booked. It does not return the constraint that generates it.

What the method already computes is enough to fix this, and no new simulation machinery is needed. Each iteration builds the whole network, so alongside each Drag the run can store the quantity that capped it, which here is A's release of R1, and the Drag of each remaining candidate recomputed after each cut. What comes out is not a family of histograms but a directed structure with conditionals on its edges: the probability that an activity binds given that its parent has overrun, against the probability given that it has not. Where several constraints converge on one activity, a bounded combination rule such as noisy-OR (Fenton and Neil, 2018) lets them accumulate without any one of them being asked to carry the whole relationship. The apparatus is

long established (Pearl, 1988); what a levelled schedule adds is that its edges are already known, so nothing need be inferred from co-movement after the event. The decision object then becomes the edge rather than the node: which constraint, if it were relieved, returns the most. Freeing the R1 contention in this example returns C's ceiling from three days to five, an intervention that no ranking of activity Drags can suggest, because the activity responsible has a Drag of zero.

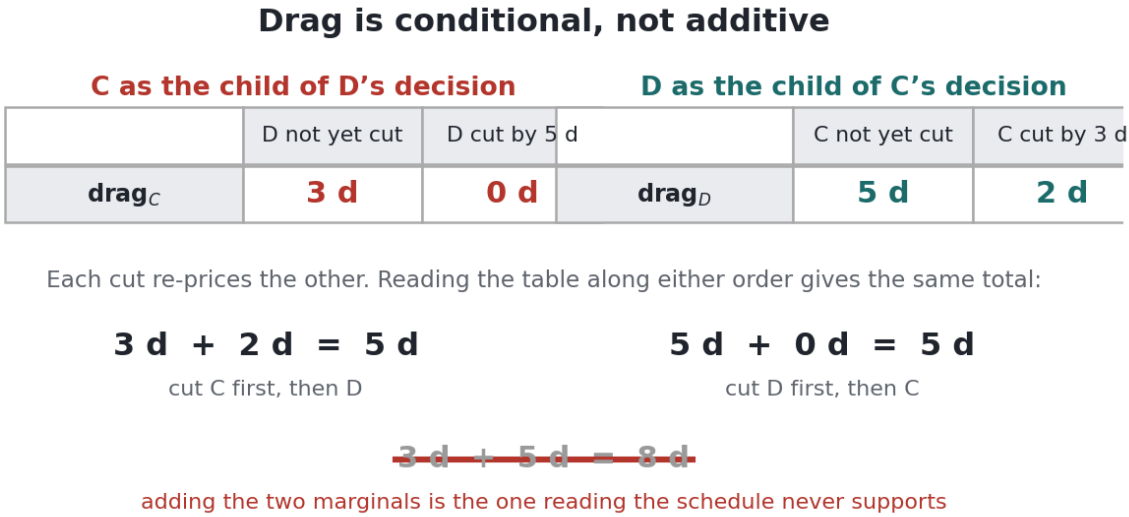

Figure 8. *Drag as a conditional table rather than a ranked list. Each activity is the child of the other's decision. Cutting C first buys three days and leaves D worth two; cutting D first buys five and leaves C worth nothing. Both orders total five days, while adding the two published Drags gives eight, a figure the schedule never supports.*

The cherry, if such a model is to live alongside a delivery, is what happens once the project starts reporting. Activity A finishes on some actual day, and at that moment most of the uncertainty in C's Drag is not updated but resolved, because it was A's uncertainty throughout. Rather than re-running the simulation from the original three-point estimates as though nothing had been learned, the model can be conditioned on the observation. A posterior-sampling scheme holds the topology fixed, so the *shape* of the structure is preserved and only the parameters move. Successive estimates then converge instead of contradicting one another from one monthly re-run to the next. The point is a modest one. We do not need to fight uncertainty by sampling it ever more finely. We need to accept it and record it, so that a schedule learns as it is delivered.

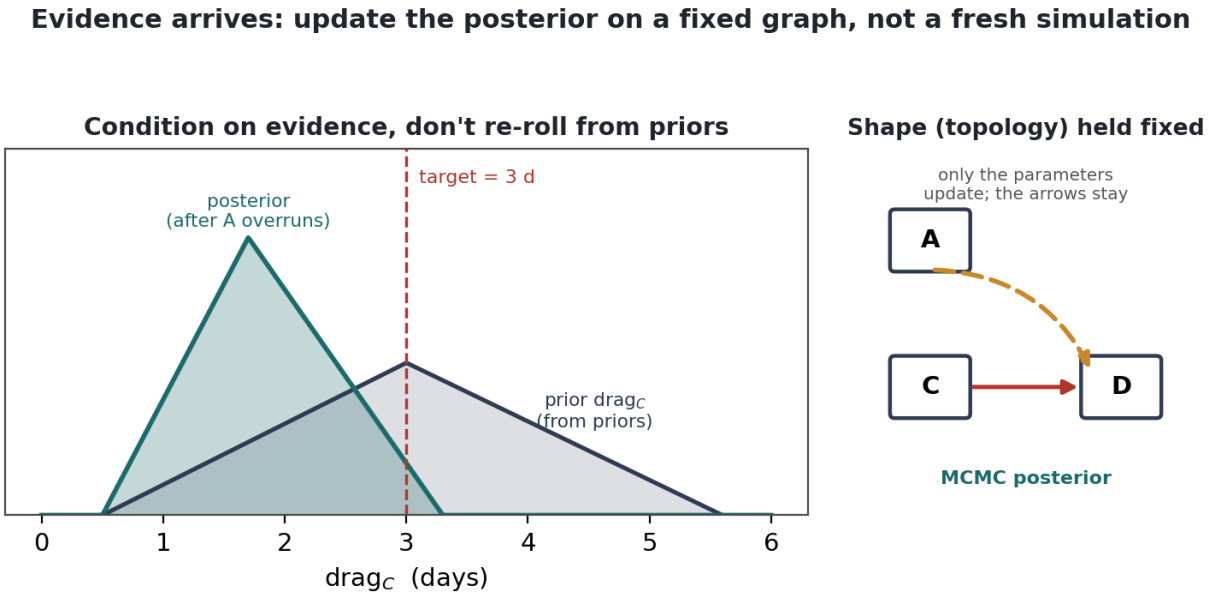

Figure 9. *Updating rather than re-rolling. As real durations arrive, condition the model on them while holding the network topology fixed, so only the parameters move. On this example, observing A resolves most of the uncertainty in C's Drag rather than merely revising it.*

In closing

It should be said plainly that everything above is available only because the paper supplies it. The identity, the variance split, the conditional table and the pair effect are all read off Scenario 1, which the author set out completely enough for a reader to rebuild it and argue with it. That is not the norm. A great many schedule-risk papers report outputs from a model nobody outside the author's office can reconstruct, and no reader can argue with a histogram. Here the network, the resources, the three-point ranges and the iteration count are all on the page, so the work can be tested rather than merely believed. What is above is a consequence of that openness, not a criticism of it.

None of this argues against Probabilistic Drag. It argues that the paper has built the apparatus that makes the conditional reading possible and is one short step from it. Recomputing Drag inside a resource-levelled simulation is the hard part, and the paper does it and reports it honestly. The example was chosen well enough that its hidden R1 contention, offered as an aside, turns out to govern the headline number, and that is a sign of a well-made example rather than a flaw in it. A metric that began by replacing a deterministic promise with an honest distribution is close to replacing that distribution with the structure underneath it, which is what a planner deciding where to spend acceleration money needs.

I congratulate the author on a valuable and well-judged contribution, and I hope these notes are read in the constructive spirit in which they are offered.

If there is one sentence to take from all of the above, it is that this paper has already done the difficult half. Recomputing Drag inside a resource-levelled Monte Carlo, iteration by iteration, is the part that requires a real delivery model and a system that will carry it; reading the output as a structure rather than as a set of histograms is comparatively cheap once the first part exists. Everything I have suggested is a change of what gets recorded and reported from a run the author is already performing. That is the mark of a paper worth building on rather than one worth answering, and I would rather be the first of many readers to push at it than one of the many who did not notice it.

A note for readers who rebuild the example. The figures and numbers above use the three-point durations in Picture 1.4, where Activity D is given as four, five and six days against a remaining duration of six. The prose sets uncertainty at plus or minus one day for activities other than A, which would put D at five, six and seven, and the two choices diverge sharply: only the values in Picture 1.4 come close to the 15.1% the paper reports for D, returning about thirteen per cent against the prose reading's fifty. My reconstruction samples the four durations independently from triangular distributions over those ranges at four hundred thousand iterations, without calendars, which is also why the paper's half-day minimum for C appears as zero here; the Activity Calendar named among its parameters sits outside my model.

Yours faithfully,

Nikhil Dhand, PMP

Creator of Probabilistic Chain Analysis and the RiskPulseV15 engine

Delhi, India

References:

Devaux, S. A. (1999). *Total Project Control: A Manager's Guide to Integrated Project Planning, Measuring, and Tracking*. John Wiley & Sons.

Duncan, W. and Devaux, S. (2009). Scheduling is a Drag.

Fenton, N. and Neil, M. (2018). *Risk Assessment and Decision Analysis with Bayesian Networks*, 2nd ed. CRC Press.

Lyaschenko, A. (2026). Probabilistic Activity Drag. *PM World Journal*, Vol. XV, Issue VII, July. https://pmworldlibrary.net/wp-content/uploads/2026/07/pmwj166-Jul2026-Lyaschenko-Probabilistic-Activity-Drag.pdf

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann.