**Fatigue Models & Optimizers**

While a rule is conceptually easy to understand, the impact of introducing it to a complex system like airline crew scheduling is hard to predict. The impact of a far more complex fatigue model is, quite paradoxically, easier to predict.

The integrated system of automatic schedule creation and fatigue models has several advantages over using the software separately. It allows the complete process to be automated and also allows for post-hoc manual fatigue analysis, without ever leaving the scheduling software. There are also significant schedule efficiency gains to be had by letting the optimizer dynamically handle the rebuilding of fatigue patterns, instead of doing it later by hand.

With a tightly integrated system, there are different ways of integrating the fatigue model into the schedule creation process. The simplest approach, to just move the “too fatiguing to fly” threshold from the post-planning step into the optimizer, will un-lock some productivity that is lost in the manual planning step, but will still rely on a threshold below which no flight should be operated. This threshold is inherently problematic, and one of the primary reasons for moving to a distribution-centric view of fatigue risk.

Even in cases where a rule-based fatigue risk management system was shown to do a nice job in limiting fatigue, there are other potential gains to be had from replacing parts of the system with a fatigue model. Examples of these benefits include finding productivity that is inhibited by the fatigue rules but not particularly fatiguing, or increasing crew satisfaction with work schedules. Using a bio-mathematical model with an incentive-based distribution-centric approach, allows the optimizer to find good solutions that were previously hidden by poorly designed rules.

**Aligning Crew Scheduling Rules with Human Physiology**

When using a crew scheduling optimizer it is actually not that difficult to align a rule set to human physiology using a bio-mathematical model:

1. **Prepare.** Take your normal planning solution and use as baseline. Quantify the overall "goodness" (G) of that solution. Quantify the overall fatigue risk (R) of the baseline solution.
2. **Appraise.** Quantify the maximum achievable potential by creating a scheduling solution with all rules, solely there for protecting from fatigue, shut off. Instead, limit fatigue risk using the model. Quantify again G and R.
3. **Relax.** Form a hypothesis of what rule relaxations in the baseline would be the most effective for improving G. Re-run the planning solution using those relaxations, and call the solution A. Measure G and R again. You should now see that G has improved significantly but R is somewhat worse.
4. **Constraint.** Use the fatigue model to find and study the cases (scheduling patterns) that you now allow in A and that the fatigue model considers very poor. Reformulate relaxations or add new rules to constrain those patterns. Make a new run with this rule set, calling the result solution B. Measure G and R again. Hopefully R is now improved, but G has most likely taken a small hit. Make sure you improve R to beat what you had in your baseline solution.
5. **Evaluate.** Unless happy with the improvement reached in G and R, go back to step 3 using solution B as the new baseline. Typically some 60-80% of the potential found in step 2 should be possible to reach by these rule reformulations.
6. **Stress test.** Provided you have a fatigue model, you can also use real-time with the optimizer; introduce an incentive for the optimizer to reward fatiguing flights. Make new worst case runs for both the new rule set and the original rule set, now pushing the optimizer to plan as poorly as possible (staying within the rules). Measure R and make sure the new rules do not allow for more fatigue risk to "slip through" than the baseline rules did. If they do, go back to step 3.
7. **Validate.** Make sure you also do the above steps (1-6) over several or larger data sets (flight schedules) so that you verify that the rules are not in any case (e.g. during another season) resulting in making G or R worse than the baseline(s).
8. **Apply.** R can be further improved by gently pushing the optimizer to a corner of the solution space with lower fatigue risk. Typically quite a lot can be achieved by this, despite fixed constraints, with almost no negative impact on G at all.