From Reactive to Predictive Regulation in Metros

K. Kecir[1,2], L. Hélouët[1], P. Dersin[2], B. Adeline[2], A. D'Ariano[3]

[1]ALSTOM, [2]INRIA Rennes, FRANCE [3]UNIV. ROMA 3, ITALY

{karim.kecir,loic.helouet}@inria.fr {pierre.dersin,bruno.adeline}@alstom.com a.dariano@ing.uniroma3.it

Urban networks must ensure passengers satisfaction and meet QoS criteria (punctuality, regularity of service, energy consumption...) fixed by operators or local authorities. For these systems, forecasts are designed in the form of timetables describing departure and arrival dates of trains at stations, or with promises of regular pace at some stations. However, due to unpredictable delays (caused by weather conditions, users misbehavior,...), these plannings are never met. To recover from delays, metro networks are equipped with corrective mechanisms called *regulation algorithms*. Currently, regulation is mainly *event-based* and *reactive*: upon arrival of a train, the difference between the forecast arrival date and the observed arrival date calls for corrections to the forecast: next departures can be delayed, dwell times can be shortened, commercial speeds can be increased and, in some cases, trains ordering can even be modified. These algorithms apply rules of the form `"if train` x `is late by more than` y `seconds, shorten dwell time by` v `seconds"`. Of course, algorithms can be more involved and consider more parameters than a single train delay; yet, they remain quite local decisions. In this setting, regulation advices are application of rather logical rules aiming at recovering delays, but whose optimality is not certain. Experience shows nevertheless that these systems are sufficient in practice to recover from small delays. However, most cities face traffic increase, but yet ask for high QoS and energy optimization. There is hence a clear need for algorithms with optimal performance, and for tools to demonstrate this optimality.

We propose a framework to model networks that integrate optimization schemes in their regulation, and tools to evaluate the performance of these optimized regulation algorithms. The main idea here is to consider more global decisions: instead of reacting to a local delay, we advocate the fact that solutions returned by regulation algorithms have to consider elements from the whole network and its planned forecast. We also advocate the fact that decisions that are proposed by regulation algorithms have to be optimized: instead of computing a rescheduling that simply postpones forecasts depending on a measured delay and on fixed thresholds, regulation can reconsider ordering of trains (and even insert or remove trains from a network) and search solutions that optimize some criteria (mean delay, energy consumption,...) a priori, at least for a bounded time window. This setting is *predictive*, as one reconsiders for a bounded time windows the whole forecast ordering and timing. It tries to be optimal, by returning the best solution for a given time window. Note here that one cannot expect a regulation algorithm to return the best possible solution as, first, optimization algorithms use heuristics to solve problems in decent time; and, second, even if an optimal solution is proposed for a given setting, disturbances can still occur thus invalidating a planning that has been recomputed, and which will hence not be optimal in this modified setting.

We are developing the SIMSTORS tool to evaluate performance of regulation algorithms. This tool is decomposed as follows: the first part is a simulator that represents a network, animates trains and introduces random perturbations. The second part is a schedule representing the predicted timetable, and the last part is a regulation algorithm. The simulator part models trains moves, accepts inputs from the regulation part, and plans departures as expected from the schedule. In turns, it provides information to the regulation algorithm on arrival and departure dates of trains. The schedule is part of the input data of the system and evolves during the simulation. The regulation part is an *actual algorithm* running in a real system. It receives occurrence dates for departures and arrivals, and reschedules the timetable according to rules. Simulation campaigns (à la Monte-Carlo) with this model allow to derive statistics, and measure efficiency of regulation algorithms. We propose to extend this simulation scheme to evaluate integration of optimization techniques in regulation. The main idea is to use the output of the optimizer AGLIBRARY to reschedule timetables. There are constraints for this new regulation scheme. First, the optimization has to be fast; this means that optimization has to be performed for a bounded time window. This is not a disadvantage as optimal solutions for large time windows have high chances to be reconsidered. This solution also has to provide fast reactive decisions for events that occur during computation time of the optimizer (optimization applies only to the next events). This means that our regulation algorithm has to be an hybrid solution applying fast reactive solutions for early events, and quasi optimal solutions to remaining events occurring after the delay needed to compute optimized solutions.