STOCHASTIC NOTIFICATION SCHEDULING WITH PUSH CONSTRAINTS

Yang Sun, College of Business Administration, California State University, Sacramento, 6000 J Street, Sacramento, CA 95819, 916-278-6001, suny@csus.edu
Yang Li, College of Business Administration, California State University, Sacramento, 6000 J Street, Sacramento, CA 95819, 916-278-6003, yang.li@csus.edu

ABSTRACT

Push notifications allow individuals to receive information in a timely manner; however, users can easily be overwhelmed that they may turn off push notifications. In this research we address the push notification scheduling problem in which the user is allowed to specify a constraint to limit the number of times a push notification service can ding or vibrate the mobile device in a day. In addition, the user can specify blackout time periods during which no push notification is allowed. The scheduling function can be implement at the server side or the client side and can be embedded in an app or the operating system. The objective is to minimize the total message (job) waiting time before a batch of messages is released/pushed to the user. Job release control is an important operational decision to make in manufacturing environments in order to regulate factory workload and avoid bottleneck starvation; however, job release problems in service systems are largely overlooked in the literature. How to allocate the restricted number of releases during the allowed time periods has a significant impact on the quality of service measured by the timeliness of the delivery of the jobs under certain user behavior and energy constraints.

Consider a message ready to be pushed as a job arrival. In this research we assume that job arrivals are random and follow a known stochastic pattern in the given time period T. Specifically, we assume that job arrivals follow a Poisson process, either stationary or nonstationary. In practice, the pattern can be obtained by analyzing historical arrival data. Jobs can be released for exactly m times during T. Arrived jobs wait at a "gate block". As soon as a release occurs, all jobs currently waiting are released for pushing and their waiting times at the "gate" are added into the total waiting time. Releases are not allowed during a set of pre-identified sub-periods that are considered blackout times. Job flow times after the release are considered sunk costs and are not counted in the performance measure. Future jobs will accumulate at the "gate block" again and wait for the next release. The last release always happens at the end of the period T. The problem is therefore to allocate or schedule the $(m - 1)$ releases during the period to minimize total waiting time.

Let r_i be the release time of the i^{th} release, $i = 1, \cdots, m$, where m is the total number of releases within the planning cycle, and $r_m = T$. The following optimality property under stationary Poisson arrivals can be mathematically proved.

1. To minimize the total expected waiting time of all jobs under stationary Poisson arrivals, the optimal i^{th} release time $r_i^* = \frac{i}{m}$.

Assume the job arrival is a nonstationary Poisson process with arrival rate $\lambda(t)$ at time t. The following optimality property under nonstationary Poisson arrivals can be mathematically proved.

2. To minimize the total expected waiting time of all jobs, the optimal release times $r_1^*, r_2^*, \cdots, r_{m-1}^*$ satisfy the following equations: $\int_0^{r_1} \lambda(t)dt = (r_2 - r_1)\lambda(r_1)$, $\int_{r_1}^{r_2} \lambda(t)dt = (r_3 - r_2)\lambda(r_2)$, $\cdots$, $\int_{r_{m-2}}^{r_{m-1}} \lambda(t)dt = (T - r_{m-1})\lambda(r_{m-1})$.

3. To garauntee the solutions solved from the above first order conditions minimize the total waiting time, $r_1^*, r_2^*, \cdots, r_{m-1}^*$ need to further satisfy the following second order conditions: $\lambda'(r_1) \leq \frac{2\lambda(r_1)}{r_2 - r_1}$, $\lambda'(r_2) \leq \frac{2\lambda(r_2)}{r_3 - r_2}, \cdots, \lambda'(r_{m-1}) \leq \frac{2\lambda(r_{m-1})}{T - r_{m-1}}$. These conditions ensure the joint convexty of the objective function in $r_1, \cdots, r_{m-1}$.

Note that as long as the second order conditions above hold, the results can be easily extended to a job release problem with linear release constraints. For example, if a blackout interval constraint (no release is allowed during the backout intervals) is imposed on the problem, the optimization problem can be solved by two steps. The first step ignores the blackout interval constraint and obtains the uncontrained solutions by the first order conditions. If any r_i solved in the first step falls within a blackout interval (i.e., r_i is infeasible), the optimal release time should be chosen at either the lower-bound or the upper-bound of the blackout interval (whichever yields lower total waiting time) due to the joint-convexity property of the objective function and this selection can easily be implemented algorithmatically.

4. Special case: non-stationary Poisson arrvial with m = 2. When there are only two releases, the optimal first release time r^* satisfies the first order condition $\int_o^r \lambda(t)dt = (T - r)\lambda(r)$. The second order condition required for the convexity of the objective function is $\lambda'(r) \leq \frac{2\lambda(r)}{T - r}$.

The second order condition above also implies that the optimal first release time is likely to be postponed when the job arrival rate exhibits an increasing trend ($\lambda'(t) > 0$), and is likely to be chosen when the job arrival rate exhibits a decreasing trend ($\lambda'(t) < 0$).

Optimal solutions can be obtained using the above properties for stylized cases in which the function $\lambda(t)$ has a relatively simple form. Such solutions provide insights for developing scheduling heuristics for general cases. A number of decision heuristics are developed for general cases where the nonstationary Poisson arrival rate does not follow a stylized pattern. Discrete event simulation models are used to compare various heuristic policies under the push constraints. Simulation experiments show that heuristic-based policies that release jobs based on queue status consistently provide low total waiting time.

Keywords: stochastic job release scheduling, heuristics, discrete event simulation