

Robust Operations Management on Mars

Michael Saint-Guillain

Université catholique de Louvain (UCLouvain, Belgium)
Institut National des Sciences Appliquées de Lyon (INSA-Lyon, France)
michael.saint@uclouvain.be

Abstract

We compare both deterministic and robust stochastic approaches to the problem of scheduling a set of scientific tasks under processing time uncertainty. While dealing with strict time windows and minimum transition time constraints, we provide closed-form expressions to compute the exact probability that a solution has to remain feasible. Experiments, taking uncertainty on the stochastic knowledge itself into account, are conducted on real instances involving the constraints faced and objectives pursued during a recent two-week Mars analog mission in the desert of Utah, USA. The results reveal that, even when using very bad approximations of probability distributions, solutions computed from the stochastic models we introduce, significantly outperform the ones obtained from a classical deterministic formulation, while preserving most of the solution's quality.

Unlike most classical scheduling problems, operations in a space mission must be planned days ahead. Complex decision chains and communication delays prevent schedules from being arbitrarily modified, hence online reoptimization approaches are usually not appropriate. The problem of scheduling a set of operations in a constrained context such as the *Mars Desert Research Station* (MDRS, Fig. 1) is not trivial, even in its classical deterministic version. It should be seen as a generalization of the well-known NP-complete *job-shop scheduling problem* (Lenstra and Kan 1979), which has the reputation of being one of the most computationally demanding (Applegate and Cook 1991). (Hall and Magazine 1994) insist on the importance of mission planning, citing the Voyager 2 space probe for which the development of the a priori schedule involving around 175 experiments requiring 30 people during six months. Nowadays, hardware and techniques have evolved and it is likely that a couple of super-equipped (*i.e.* with a brand new laptop) human brains may suffice in that specific case. Yet, the problems and requirements have evolved too and space missions have to deal with teams of astronauts.

At the MDRS, computing an optimal schedule becomes significantly less attractive as problem data, such as the manipulation time of experiments, are different from their predicted values. In a constrained environment with shared resources and devices, such deviations can propagate to the



Figure 1: The Mars Desert Research Station (MDRS) in the Utah desert, U.S, is a Mars analog planetary habitat.

remaining operations, eventually leading to global infeasibility. The purpose of this paper is to investigate, based on the real case study of a Mars analog mission, the impact of stochastic robust modeling against a classical deterministic approach on the reliability of a priori mission planning.

Illustrating problem. Consider four jobs $\{A, B, C, D\}$ to be scheduled on one single machine. Each job has one or several time windows, precedence constraints and minimum transition time constraints, the latter stipulating a minimum delay between the completion time of a job and the beginning of another. We assume jobs to be atomic, the processing of a job cannot be split on several time windows. The horizon comprises two working days of five hours each (9am to 2pm). Fig. 2 provides the remaining details of the problem, which we will refer to as the ABCD problem.

A common goal for a deterministic scheduling problem is to minimize the completion time of the entire project, which is achieved here by the sequence solution $s = \langle A, B, C, D \rangle$. It is in fact the only optimal solution: A starts at 9am, then job B at 10, job C from 12 to 1pm, job D from 1 to 2pm. Now suppose that A's processing time is uncertain. If A reveals to require slightly more than one hour, then job B will not be completed within its time window, and is reported to the second day. Solution s is no longer feasible, as job C is not completed during the first day. Assuming the mean processing time of A to be of 1 hour, it has a significant probability to exceed it, whatever the distribution is (0.5 if it is symmetric). No matter how confident we are about A's processing time and the degree of its uncertainty (*e.g.* standard deviation),

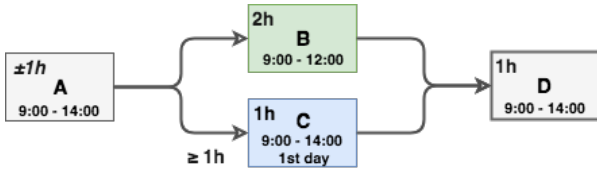


Figure 2: The ABCD problem. Each job has a processing time of 1 or 2 hours and a time window spanning either the entire work day (9am to 2pm) or part of it (9am to 12am). Job C must wait at least one hour after completion of A to start and must be completed during the first day.

tion), a non null probability for A's processing time to reveal not to be exactly 1 hour leads a high probability of failure ($\approx 50\%$ in most of the cases). A reliable scheduler should rather suggest the more robust solution $s = \langle A, C, B, D \rangle$.

Application and contributions. How to deal with processing time uncertainty when facing a larger, complex, scheduling problem which possibly involves multiple operators, unknown probability distributions and exotic constraints? In this paper, we show how even such simple insights can actually be applied to real project development and planning. As a proof of concept, we demonstrate the contribution of robust modeling on the preparation and implementation of the *UCL to Mars 2018* project that took place at the Mars Research Desert Station (Utah), March'18.

Project management realizes about 30% of the world gross product (Turner et al. 2010). Besides the space analog mission, our approach naturally applies to a larger set of project scheduling contexts whereas most of existing studies have solely been done in machine scheduling environments (Herroelen and Leus 2005).

We formulate the core problem as a *robust single-machine scheduling problem with random processing times*. Tasks are subject to precedence constraints, strict time windows and minimum transition times. We show how the robustness of a solution, in terms of its probability to remain feasible to operational time constraints, can be exactly computed in pseudo-polynomial time. We also identify theoretical limitations to computational tractability and propose alternatives. Finally, we adapt the core problem to the goals pursued and constraints faced during the *UCL to Mars 2018* mission, and empirically measure the average gain of using our generalized stochastic formulation instead of a deterministic one.

Core problem

For now we consider a *robust single-machine scheduling problem with random processing times* (R-SMS-T), involving job precedence, time windows and minimum transition times constraints. Approximation functions will thereafter be used to generalize the core theoretical results to the specific problem we face in our MDRS case-study, a *robust job-shop scheduling problem* involving additional specific constraints. A recent review on stochastic scheduling is provided by (Chaari et al. 2014).

Input. A discrete time horizon $H = \{0, \dots, h\}$ on which a set $J = \{j_1, \dots, j_n\}$ of jobs must be scheduled on an unique machine (or operator). The machine processes one job at a time and each job must be processed exactly once. Each job j comes with a probability p_j^d that j requires a processing time $d \in H$, with $\sum_{d \in H} p_j^d = 1$. Each job j is associated a set $TW_j \subseteq H$ of valid time intervals to process j , called time windows. Precedence constraints state that a (possibly empty) set $J_j^< \subset J$ of jobs must be completed before starting job j . A *minimum transition time* states a minimum delay $w_{j',j}^{\min}$ to be observed between completion of a job $j' \in J_j^<$ and the beginning of j .

Solution and formulation. A solution to the R-SMS-T is an ordered sequence of jobs: $s = \langle j_i, \dots, j_{i'} \rangle$. Every job $j \in J$ appears exactly once in the sequence. We use a two-index flow formulation in order to describe s : binary decision variables x_{ij} denote whether or not the job $j \in J$ is scheduled immediately after $i \in J$. Additional variables x_{0j} (resp. x_{j0}), for $j \in J$ refer to the first (resp. last) job j of the sequence. We formulate our R-SMS-T as:

$$\max_s r(s) \quad (1)$$

$$\text{s.t.} \quad \sum_{i \in J_0 \setminus \{j\}} x_{ij} = \sum_{i \in J_0 \setminus \{j\}} x_{ji} = 1 \quad j \in J \quad (2)$$

$$u_j - u_i + nx_{ji} \leq n - 1 \quad j, i \in J \quad (3)$$

$$u_i \leq u_j - 1 \quad j \in J, i \in J_j^< \quad (4)$$

$$x_{ji} \in \{0, 1\} \quad j, i \in J_0 \quad (5)$$

where we note $J_0 = J \cup 0$ for conciseness. The robustness measure $r(s)$, detailed in the next section, gives the probability of s to remain feasible. Flow conservation constraints (2) state that a job is preceded (and followed) by exactly one job i . In fact, inequalities (2)-(3) define the solution space of a *directed traveling salesman problem* (TSP), when using a so-called *MTZ-formulation* (Miller, Tucker, and Zemlin 1960) in order to explicitly formulate as u_j (3) the position of job j in the sequence. We then express precedence constraints quite naturally in (4). Since the time dimension is stochastic, time windows and minimum transition time constraints cannot be part of the description of a solution to the R-SMS-T. Instead, they contribute to objective function $r(s)$ as described in the next section. Constraints (2)-(5) are then sufficient to define the solution space of our robust single-machine scheduling problem with random processing times.

Recourse assumptions. In order to be efficiently computed, $r(s)$ requires assumptions (so-called *recourse strategy*) on how s is adapted to the realizations of the random processing times:

1. The machine (operator) executes its jobs according to the ordered sequence defined by s ;
2. When starting a job j , one does not know its processing time until it is actually completed.
3. A job that is not completed by the end of its current time window must be re-processed from scratch at the beginning of its next time window, if any (otherwise 4).

4. In case the machine fails at processing a job due to unfortunate processing times, the sequence is interrupted.

These assumptions directly come from the definition of our problem. In particular, by fixing the sequences of s assumption 1 explicitly forbids reoptimization. A relaxation would lead to a dynamic and stochastic problem, and would no longer permit the exact evaluation of $r(s)$ in pseudo-polynomial time (unless P=NP). Furthermore, operational contexts such as space missions do not allow the modification of the schedule in the middle of a work day. Our recourse strategy still provides a good indicator for situations that suggest reoptimization at the end of each work day. In fact, under assumptions 1-4 the probability to remain feasible as computed by $r(s)$ is a lower bound to the probability of perfect reoptimization (optimally reoptimizing each time a random event realizes) to remain feasible.

Robustness of a solution

We define the robustness $r(s)$ of a given solution s as its probability to remain feasible. This can be trivially expressed in terms of the set of possible scenario realizations:

$$r(s) = \sum_{\xi \in E} \Pr\{\xi\} f(s, \xi) \quad (6)$$

where E is the set of all probable scenarios and $f(s, \xi)$ is the indicator function returning 1 if and only if solution s remains feasible under scenario ξ , that is, the case described by assumption 4. is not encountered. Since the size of E grows exponentially with respect to the number of jobs, the computation of (6) rapidly becomes intractable. The most natural approximation for such an enumerative function is called *Sample Average Approximation* (Ahmed and Shapiro 2002), which relies on Monte Carlo sampling to evaluate only a subset of E . Its accuracy however depends on the number of samples taken into consideration.

Instead, one can reason on the jobs themselves (and their associated random variables) in order to derive a tractable formula to compute $r(s)$ exactly. Let us consider that a job j is *correctly processed* in a scenario ξ if it is completed within one of its valid time windows and fulfills all minimum transition time constraints with other jobs, if any. It follows that the probability that a solution s remains feasible is the probability that every job succeeds in that sense,

$$\begin{aligned} r(s) &= \Pr\left\{\bigwedge_{j \in J} \text{job } j \text{ correctly processed in } s\right\} \\ &= \Pr\{j_{\text{last}} \text{ correctly processed in } s\} \end{aligned} \quad (7)$$

which, by following assumption 4., is also the probability (7) that we succeed at proceeding the last job $j_{\text{last}} \in J$ of the sequence $s = \langle j, \dots, j_{\text{last}} \rangle$.

In our ABCD example, $r(s)$ is the probability that job D eventually gets completed. Our time horizon is composed by two consecutive five-hour work days, modeled as a discrete set $H = \{0, \dots, 599\}$ which contains 600 time units of one minute each. The first 300 time units belong to the first work day, and so on. Since job B can only be processed during the first three hours of each day, we have $TW_B =$

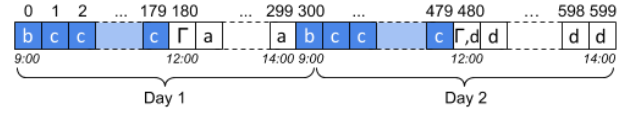


Figure 3: Time horizon of job B. Blue cells represent $TW_B = \{0, \dots, 179, 300, \dots, 479\}$. When $t \leq 179$ then $TW_B^-(t) = \emptyset$. For $180 \leq t \leq 479$, we have $TW_B^-(t) = \{0, \dots, 179\}$ and, for $t \geq 480$, $TW_B^- = \{300, \dots, 479\}$.

$\{0, \dots, 179, 300, \dots, 479\}$. Similarly, $TW_C = \{0, \dots, 299\}$ as job C must be completed the first day. We also denote by $TW_j^-(t) \subseteq TW_j$ the set of time units which only belong to the time window that directly precedes time t and does not contain it. Fig. 3 illustrates H according to job B. We note $j^- \in J$ the job that directly precedes j in solution s .

Let $P_j^{\text{end}}(t)$ the probability that job $j \in J$ is completed at time t exactly, and $P_j^{\text{start}}(t)$ the probability that j is started at time t precisely. Note that, consequently to assumptions 2. and 3., starting a job does not systematically involve its completion. Job j_{last} is therefore correctly processed if and only if it is completed during one of its time windows:

$$r(s) = \sum_{t \in TW_{j_{\text{last}}}} P_{j_{\text{last}}}^{\text{end}}(t) \quad (8)$$

Completion times. For every job $j \in J$, the probability $P_j^{\text{end}}(t)$ that j finishes exactly at discrete time unit $t \in H$ is constituted of all the possible processing times that could have led to complete the job at that specific time t :

$$P_j^{\text{end}}(t) = \begin{cases} \sum_{d=0}^t p_j^d \cdot P_j^{\text{start}}(t-d) & \text{if } t \in TW_j \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

Starting times. We use starting times to handle time windows. Suppose we know the probability $P_j^{\text{ready}}(t')$ that the machine becomes ready for job j at time $t' \leq t$, in the sense that it is ready as soon as it completes the previous job j^- , and that any minimum transition time constraint between a job j' and j is fulfilled. Then, $P_j^{\text{start}}(t)$ is the probability that, according to P_j^{ready} and j 's time windows, one actually starts to process job j at a current time t . We decline the computation of $P_j^{\text{start}}(t)$ in three different cases, depending on t :

- $t \in TW_j \wedge t-1 \notin TW_j$: t is the first time unit of the current time window. For job B, this corresponds to b cells in Fig. 3. There are two possible reasons for starting a job at such particular moments:
 - Previous job, if any, just completed or did earlier. j must wait for current time window to begin (first summation term below). In the case of job B, at $t = 300$ the first summation ranges from $\Gamma_B(300) = 180$ to 300.
 - j had to be reprocessed (second summation term below), as last attempt (during previous time window TW_j^-) revealed to require too much time to complete and had to be interrupted due to assumption 4. In the case of job B, at $t = 300$ the second summation ranges in $\{(t', d) : 0 \leq t' \leq 179, t' + d > 180\}$.

Putting the two cases together then leads to:

$$P_j^{\text{start}}(t) = \sum_{t'=\Gamma_j(t)}^t P_j^{\text{ready}}(t') + \sum_{\substack{t' \in TW_j^-(t) \\ d: t'+d-1 \notin TW_j}} P_j^{\text{ready}}(t') p_j^d \quad (10)$$

where $\Gamma_j(t)$ is the first time unit that directly follows the previous time window (from time t) that is, the first moment from which we could wait for the opening of the current time window. For job B, this corresponds to the Γ cells in Fig. 3: $\Gamma_B(0) = 0$ and $\Gamma_B(300) = 180$.

- $t \in TW_j \wedge t-1 \in TW_j$: t lies on a legal time unit, but not the first of the current time window. The only reason for starting the job at that moment is that the machine just becomes available (ready) at current time t :

$$P_j^{\text{start}}(t) = P_j^{\text{ready}}(t) \quad (11)$$

For job B, this corresponds to c cells in Fig.3.

- $t \notin TW_j$: t is not a legal time unit for processing j ,

$$P_j^{\text{start}}(t) = 0. \quad (12)$$

For job B, this corresponds to cells a and d in Fig.3. Note that the schedule definitely fails at processing j if there is no future legal time unit. This happens in Fig.3 if the machine becomes available at d cells.

Availability times. We define $P_j^{\text{ready}}(t)$ as the probability that the machine becomes available for job j at a time t . It is the probability that job j could be started at time t , regarding both the completion time of j^- and minimum transition time constraints only (i.e. regardless time windows of j).

First job of a sequence. If job j is the first of the sequence s then the moment at which the machine becomes available is obviously time unit zero:

$$P_j^{\text{ready}}(t) = 1 \text{ if } t = 0, \quad 0 \text{ otherwise.} \quad (13)$$

Subsequent jobs. We now consider a job j which is not the first of its sequence. If there is no minimum transition time constraint associated from any job j' to j , then the machine is ready for job j as soon as previous job j^- is completed:

$$P_j^{\text{ready}}(t) = P_{j^-}^{\text{end}}(t) \quad (14)$$

If j has exactly one minimum transfer time constraint with an unique job $j' \in J_j^<$, probability $P_j^{\text{ready}}(t)$ becomes

$$P_j^{\text{ready}}(t) \equiv \Pr\{t = \max(\text{end}(j^-), \text{end}(j') + w_{j',j}^{\min})\} \quad (15)$$

where $\text{end}(j)$ is the time at which job j reveals to be completed. Namely, either job j' has completed for long enough (at a time $t' \leq t - w_{j',j}^{\min}$) to not worry about minimum transition time $w_{j',j}^{\min}$ and the operator waits for the completion of j^- in order to start j , or the previous job j^- is completed yet but j must be delayed until current time t coincides with the completion time of j' plus minimum transition time $w_{j',j}^{\min}$. If the sequence is $s = \langle A, B, C, D \rangle$ in our example, $P_C^{\text{ready}}(t)$

is the probability for t to be exactly equal to the maximum of 1) the completion time of previous job B and 2) that of A plus one hour. Mathematically,

$$P_j^{\text{ready}}(t) = \Pr\{\text{end}(j^-) = t \wedge \text{end}(j') \leq t - w_{j',j}^{\min}\} + \Pr\{\text{end}(j^-) < t \wedge \text{end}(j') = t - w_{j',j}^{\min}\} \quad (16)$$

Indeed, in absence of time windows for j either we start j as soon as j^- finishes (first term of (16)), or we wait after the completion j^- until we reach appropriate time $\text{end}(j') + w_{j',j}^{\min}$ (second term of (16)). Since the completion time of j^- clearly depends on that of j' , it finally leads to:

$$P_j^{\text{ready}}(t) = \sum_{t' \leq t - w_{j',j}^{\min}} P_{j'}^{\text{end}}(t') \cdot P_{j^-|j'}^{\text{end}}(t, t') + \sum_{t' < t} P_{j'}^{\text{end}}(t - w_{j',j}^{\min}) \cdot P_{j^-|j'}^{\text{end}}(t', t - w_{j',j}^{\min}) \quad (17)$$

where $P_{j^-|j'}^{\text{end}}(t, t')$, the probability previous job j^- completes at time t conditionally that job j' completes at time $t' \leq t$, can be computed recursively by following (9) where $P_{j'}^{\text{end}}(t') = 1$ and $P_{j'}^{\text{end}}(t'') = 0$, for $t'' \neq t'$. In particular, in case $j' = j^-$ then (17) reduces to $P_{j^-}^{\text{end}}(t - w_{j',j}^{\min})$, since $P_{j^-|j^-}^{\text{end}}(t, t') = 1$ if $t' = t$, zero otherwise. Note that supplementary minimum transition time constraints can be associated to j by adding terms to the max operator in (15). *Intractability issue.* Having two jobs j', j'' on which j depends for minimum transition time, one has to take conditional probabilities $P_{j^-|j',j''}^{\text{end}}(t, t', t'')$ into account for every combination of $t', t'' \in H$. In the case the last job j_n of a solution $s = \langle j_1, \dots, j_n \rangle$ is constrained by minimum transition times involving previous jobs j_1, \dots, j_{n-1} , computing all the conditional probabilities is equivalent to enumerating all the $\mathcal{O}(h^n)$ possible scenarios. In order to keep the computation tractable, in what follows we assume that there is at most one such constraint per job.

Computational complexity. Provided a solution to n jobs, the complexity of computing (7) is equivalent to the one of filling up three matrices P^{ready} , P^{start} and P^{end} , each of size nh , containing respectively all the $P_j^{\text{ready}}(t)$, $P_j^{\text{start}}(t)$ and $P_j^{\text{end}}(t)$ probabilities. The computational effort required to compute each cell significantly varies depending on the presence of minimum transition time constraints.

No minimum transition times. Once the probabilities in cells $(j, 1 \dots t)$ of P^{end} are known, the cell $P_{(j,t)}^{\text{ready}}$ can be computed in $\mathcal{O}(1)$ using (14). Then, using probabilities in cells $(j, 1 \dots t)$ of P^{ready} , cell $P_{(j,t)}^{\text{start}}$ can be computed in $\mathcal{O}(h)$ according to equation (10)-(12). In fact, the double summation in the second term of (10) is amortized in $\mathcal{O}(h)$. Finally, probabilities $(j, 1 \dots t)$ of P^{start} allows to compute cell $P_{(j,t)}^{\text{end}}$ in $\mathcal{O}(h)$ according to equation (9). A solution consisting in n jobs and a time horizon of length h leads to a worst case complexity of $\mathcal{O}(nh^2)$.

Minimum transition times. In the case where each job can have a most one minimum transition time with another job,

we consider the worst case involving a sequence $\langle j_1, \dots, j_n \rangle$ in which all j_2, \dots, j_n jobs have a minimum transition time constraint with j_1 . It then requires to compute $\mathcal{O}(n^2)$ conditional probability matrices $P_{j|j'}^{\text{pend}}(t, t')$ each of size h^2 , each cell still computable in $\mathcal{O}(h)$. That enables the computation of P^{ready} values in $\mathcal{O}(h)$ according to equation (17). The overall complexity is now of $\mathcal{O}(nh^2 + n^2h^3)$. However, if we allow a job to have minimum transition time constraints with q other jobs, then the $P_{j|j'_1 \dots j'_q}^{\text{pend}}$ matrix will be of exponential size h^q .

Stochastic transition times. There are some contexts in which transition times could also be considered stochastic. In fact, for a fixed planning, transitions may be equivalently seen as jobs. Stochastic transition times may be thus trivially handled, by simply replacing in a planning each stochastic transition time by a job with same probability distribution. Hence we equivalently end up with sequences of jobs, but no transition times.

The UCL to Mars 2018 case study

During our stay at the Mars Desert Research Station (MDRS, Fig. 1), our crew conducted 10 experiments (see ucltomars.org/#!/crews/190/projects) regrouped in 7 research projects. Each researcher from the team was associated to a project, composed of a set of jobs. A researcher acts as an operator (machine), subject to constraints such as those covered by the R-SMS-T, plus a few additional ones. The goal of the crew's executive officer was therefore to model, and solve, the associated scheduling problem globally.

In-place operations

The a priori horizon covered the thirteen 9-hour work days of the entire mission. At the end of each day, we solved an updated schedule on the remaining horizon, depending on the scientific outcomes of the current day. For practical reasons, each researcher was only able either to perform his/her own jobs, or to assist other researchers.

Deterministic model. As we were provided 7 operators instead of a single one, the problem we faced at MDRS had been modeled in-place as a deterministic *job-shop scheduling problem* (JSP), assuming processing times to be perfectly known. The modeling of all crew members' research projects merged within a global problem was inevitable, as researchers at MDRS depend on limited and shared resources. Some projects required *extra vehicular activities* (EVAs), which for security reasons require between three and five participants. EVAs usually take half a day (four hours) in total, should be planned and approved days ahead and happen at most once a day. In such context, all crew members have their schedule linked to each others even concerning research projects that do not involve EVAs.

The time horizon consists of 10 minutes time units, each operational day counting 24×6 time units. In fact, because of transition time constraints the 15 non-working hours (out of 24) must be considered as well. An example of project

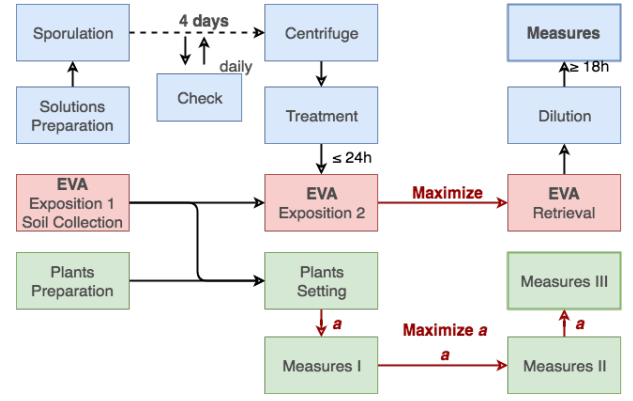


Figure 4: Modeling of a research project combining botanics (green) and biology (blue), conducted by crew member Frédéric Peyrusson (UCLouvain, Belgium) .

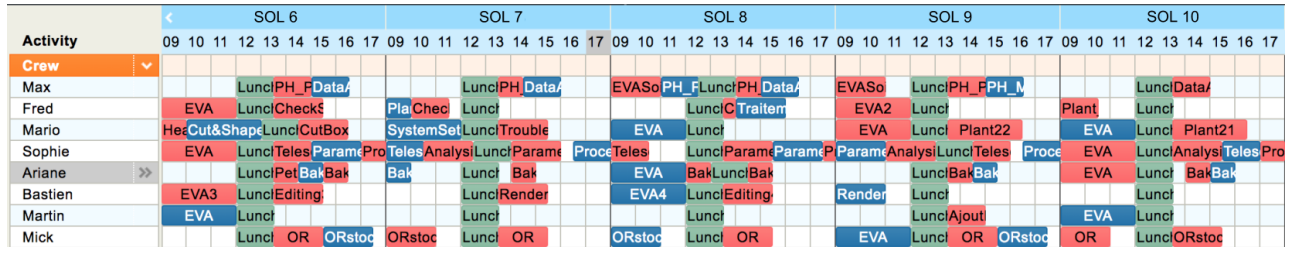
model is depicted in Fig. 4. The model requires three distinct EVAs and combines two different projects lead by the same researcher. Because a minimum number of people is required to validate an EVA, it is however likely that the researcher attends additional EVAs. In fact, the model depicted is actually connected to the projects lead by the seven other researchers thorough these EVA jobs.

In addition to precedence and minimum transition time constraints, we also notice a 24h *maximum transition time* between completion *Treatment* and that of *Exposition2*. Maximum transition time constraints are easily handled in the deterministic problem. The stochastic formulation however requires to be adapted, as explained hereafter.

Finally, beyond feasibility the model maximizes a quality measure, depending on several preferences predefined by the crew members, such as maximizing the delay between second and third EVA in Fig. 4. We refer to this solution quality function as f^{mdrs} hereafter.

Solutions. Optimizing the deterministic problem was achieved in-place using a basic local search (LS) approach, exploiting well-known sequence neighborhood operators and a simulated annealing meta-heuristic. Our LS algorithm is directly adapted from the one for stochastic VRPs of Saint-Guillain *et al.* (2017) , while replacing vertices, service times and vehicles by jobs, processing times and machines, respectively. On Mars, a solar day (commonly called a "sol") lasts approximately 24h39m. We refer to the day preceding the first mission sol as SOL0. The n th day of the mission is called SOL n . A new problem is reformulated at the end of each day, based on the past operations, and sometimes by integrating new specific constraints or preferences. Fig. 5 shows an example of a schedule as recomputed during the mission, from SOL6. Taking into account the initial complexity of the model, this motivated the choice for a heuristic algorithm over an exact approach.

The problem counts from 237 jobs at SOL0, to 149 jobs at SOL7. Time horizon, at SOL0, is composed of 13 days of 144 time units, each of 10 minutes.



A posteriori analysis: stochastic robust approach

This section investigates the impact of taking uncertainty into account at planning phase of a Mars analog mission, provided the robustness measure we propose.

Solution method and solution evaluation. In order to study the impact of our robust formulation, independently of the technology used, we use the same solution algorithm as used for the deterministic problem.

The objective function computed by the solver at each solution s is replaced by the heuristic:

$$f(s) = f^{\text{mdrs}}(s) \times r^{\text{mdrs}}(s)$$

where $f^{\text{mdrs}}(s)$ is the quality of s as computed in the deterministic context. If the solution is not *deterministic-feasible*, that is if s does not fulfill all the constraints of the problem as described in section (including constraints which are specific to the MDRS, such as minimum and maximum people attendance during EVAs), then $f^{\text{mdrs}}(s) = 0$. In other words, we optimize based on the deterministic solution quality, processing times hence being assumed to be fixed to their expected values, multiplied by the robustness measure of the solution against processing time variability.

MDRS specific constraints and issues. Note that following (19) our definition of $r(s)$ does not involve the *multiple parallel machines* context nor the *maximum transition time* constraints, whereas they were both present on projects we conducted at MDRS.

In case there is no minimum transition time constraint between jobs of a set of machines M , as it case the case at MDRS, our definition of $r(s)$ can be easily generalized:

$$r(s) = \prod_{m \in M} \left(\sum_{t \in TW_j} P_{j_{\text{last}}}^{\text{pend}}(t) \right) \quad (18)$$

where j_{last}^m is the last job assigned by machine $m \in M$, since the execution of the jobs (and hence the $P_{j_{\text{last}}^m}^{\text{end}}(t)$ probabilities) are consequently independent between the different machines. This is true even for EVAs, despite their minimum and maximum people requirement. For practical reasons, an EVA could only take place during mornings, from 9am to 12am. Together with the lunch activities, EVAs are the only jobs having a fixed deterministic duration (3h). The probability of respecting the scheduled people attendance to an EVA is then equivalent to the probability for all the assigned people to respect this 9am-12am time window.

Similarly to the minimum transition time constraints, maximum transitions are computationally very hard to take into account while computing an exact $r(s)$. Instead, here we use the following approximation:

$$r^{\text{mdrs}}(s) \approx \prod_{m \in M} \left(\sum_{t \in TW_j} P_{j_{\text{last}}}^{\text{end}}(t) \right) \\ \times \left[1 - \sum_{j \in J} \sum_{j' \in J_j^<} \left(\sum_{t' \in H} \sum_{t \geq t' + w_{j',j}^{\text{max}}} P_{j'}^{\text{end}}(t') P_j^{\text{end}}(t) \right) \right] \quad (19)$$

by supposing independence between completion times. Here $w_{j',j}^{\max}$ is the maximum allowed transition time between completion of j' and that of j , if any, otherwise $w_{j',j}^{\max} = h$. The additional second term is one minus (an upper bound on) the approximated probability that a maximum transition time constraints is violated.

Horizon Partitioning Approximation (HPA). Another issue with the MDRS case study is the computational effort required to compute $r(s)$, which critically depends on the length of the horizon. In practice, computing $r^{\text{mdrs}}(s)$ on the entire horizon takes at least several minutes, even at SOL7. Instead, we introduce the $r_{\Delta}^{\text{mdrs}}(s)$ measure, which we denote by *Horizon Partitioning Approximation*. HPA consists in the robustness of s when the horizon is partitioned in independent parts of Δ days each. For instance, $r_2^{\text{mdrs}}(s)$ is computed by first applying $r^{\text{mdrs}}(s)$ on days $\{1, 2\}$ only, using a two-days horizon and considering only the jobs a priori planned at days $\{1, 2\}$ by s . The same computation is then performed on days $\{3, 4\}$, by considering the days $\{1, 2\}$ to be fixed, and so on until we cover the entire horizon. A pass at days $\{d, \dots, d + \Delta - 1\}$ computes the probability that, if everything happens as planned by s up to start of day d , all the jobs planned for days $\{d, \dots, d + \Delta - 1\}$ get actually completed during those days. A job at day d that has a transition time constraint with a job j' from a day $d' < d$ simply sees its time window modified accordingly, since completion time of j' is supposed to be fixed. Finally, we multiply the probabilities obtained at each pass to get $r_{\Delta}^{\text{mdrs}}(s)$. Using $r_1^{\text{mdrs}}(s)$ then provides the probability that every job gets completed the day it is planned, which is a pessimistic approximation of the robustness of s , since in general delaying a some job during a few days does not necessarily break feasibility. The later can be viewed as optimizing a *stability* criterion (Goren and Sabuncuoglu 2008), in the sense that using $\Delta = 1$ we approximate the expected deviation from daily objectives.

$f =$	1. Exact mean values			2. $\hat{\mu} \pm 10\%$			3. $\hat{\mu} \pm 30\%$			4. Underestimates			5. Overestimates			δ
	f^{mdrs}	r_1^{mdrs}	$f^{\text{mdrs}} \times r_2^{\text{mdrs}}$	f^{mdrs}	r_1^{mdrs}	$f^{\text{mdrs}} \times r_2^{\text{mdrs}}$	f^{mdrs}	r_1^{mdrs}	$f^{\text{mdrs}} \times r_2^{\text{mdrs}}$	f^{mdrs}	r_1^{mdrs}	$f^{\text{mdrs}} \times r_2^{\text{mdrs}}$	f^{mdrs}	r_1^{mdrs}	$f^{\text{mdrs}} \times r_2^{\text{mdrs}}$	
SOL0	0.7	84.0	79.5	0.7	74.9	67.0	0.1	63.7	54.5	0.1	5.5	24.0	31.8	92.9	94.4	7
SOL1	6.1	90.2	96.8	0.7	75.5	87.0	0.1	55.6	66.7	0.1	8.0	15.3	20.3	93.3	97.3	4
SOL2	1.4	97.5	96.3	4.0	93.6	90.3	0.1	45.7	40.3	0.1	57.7	44.5	15.5	99.1	97.5	3
SOL4	28.3	90.0	80.9	25.4	89.6	79.5	21.1	66.1	40.3	0.3	59.9	36.2	71.4	86.5	75.3	8
SOL5	2.9	89.6	88.7	0.5	91.7	91.0	2.9	99.6	99.5	0.1	0.6	1.9	41.4	99.9	99.9	3
SOL7	4.8	91.6	89.3	0.1	64.3	60.1	9.7	94.3	90.9	0.1	5.4	5.3	22.5	99.2	99.0	17
Avg.	7.4	90.5	88.6	4.9	81.6	79.6	5.6	70.8	65.4	0.1	22.8	19.5	33.8	95.1	93.9	7

Table 1: Average percentage of times the solutions remain feasible despite processing time uncertainty, depending on the quality of available a priori stochastic knowledge. Column δ gives the average loss percentage of deterministic quality f^{mdrs} .

Experimental plan. Our empirical study is based on real data from the *UCL to Mars 2018* analog mission. Our benchmark is composed of the sequence of updated problems we faced at SOL1, SOL2, SOL4, SOL5, SOL7, in addition to the initial problem modeled at SOL0. Unfortunately, models faced at sols 3, 6, 8-12 were lost due to a technical issue.

However, we ignore the exact probabilities that describe the processing times of our jobs. Good predictions require a significant amount of observations, whereas we are only provided the scenario that realized during the mission. Collecting a sufficiently large set of observations is often impossible in practice and it is often both necessary and realistic to consider the real distributions as unknown (or hidden). Let X_j be the real probability distribution of j 's processing time, and \hat{X}_j the predicted one used by $r(s)$. As X_j is unknown, we approximate it using a normal distribution: $\hat{X}_j \sim N(\hat{\mu}_j, \hat{\sigma})$. We consider five different experimental contexts, depending on the quality of the approximations:

1. We know exactly the mean value of each distribution:
 $E[X_j] = \hat{\mu}_j, \forall j \in J$;
2. The approximations are fairly good:
 $E[X_j] \sim Un(\hat{\mu}_j \pm 10\%), \forall j \in J$;
3. The approximations are of poor quality:
 $E[X_j] \sim Un(\hat{\mu}_j \pm 30\%), \forall j \in J$
4. The approximations are globally underestimating:
 $E[X_j] \sim Un([\hat{\mu}_j - 10\%, \hat{\mu}_j + 30\%]), \forall j \in J$
5. The approximations are globally overestimating:
 $E[X_j] \sim Un([\hat{\mu}_j - 30\%, \hat{\mu}_j + 10\%]), \forall j \in J$

whereas in all five cases each hidden discrete distributions X_j is randomly generated according to $E[X_j]$, by using a normal distribution with 50 rolls only which results in a highly imperfect normal distribution. Note that 4. stands for a pessimistic context in which processing times will often reveal to be longer than initially estimated. On the contrary, context 5. assumes the estimations to globally overestimate the real distributions. Such an optimistic assumption is however of particular interest in project management, where overestimating is a widespread practice to mitigate the risks.

A solution s is optimized by using either the deterministic objective function, $f = f^{\text{mdrs}}$, or the proposed stochastic measure, $f = f^{\text{mdrs}} \times r_{\Delta}^{\text{mdrs}}$, exploiting the provided \hat{X}_j

approximate distributions. In order to assess the quality of s , we measure its true robustness by simulating its execution on a sufficiently large number of scenarios (10^5), hence using *Sample Average Approximation* (Ahmed and Shapiro 2002). For each experimental context, from 1. to 5., the 10^5 scenarios are randomly sampled from the X_j hidden real distributions. We then measure the average proportion of the scenarios in which s remains feasible. We refer to this robustness measure as $SAA_{10^5}(s)$.

Results. For each instance $SOLn$, we compute a set of 10 solutions by first using the deterministic objective function $f = f^{\text{mdrs}}$, then by using $f = f^{\text{mdrs}} \times r_1^{\text{mdrs}}$ and finally by using $f = f^{\text{mdrs}} \times r_2^{\text{mdrs}}$, with 30 minutes of computation time. Table 1 shows the average results obtained by these sets of solutions as *the percentage of simulations in which the schedule remains feasible*, depending on the instance and the experimental context 1. to 5. We obtain these results by computing $SAA_{10^5}(s)$. For each instance, the δ column gives the solutions average relative difference in their deterministic attractiveness, as computed by f^{mdrs} .

Clearly, the solutions obtained using the two stochastic robust approaches strongly outperform those of the deterministic model. Obviously, stochastic models perform better under optimistic assumptions, namely exact (1.) and overestimating (5.) distributions. The largest gaps in the average robustness of both deterministic and stochastic models appear under context 1., since at that point the stochastic HPA functions r_1^{mdrs} and r_2^{mdrs} are computing values being almost equivalent to the real ones. The most preferred situation is naturally context 5. in which the mean processing times are globally overestimated, since whatever the solution is, it is likely to be more robust than under other contexts. However, it is interesting to note that even under such fortunate conditions, the deterministic model produces solutions that success $\pm 34\%$ of the time, on average, against $\pm 95\%$ for those obtained using the proposed stochastic model. Furthermore, this improved robustness comes at the price of deteriorating by only 7% of the solution's deterministic quality (δ column) on average. Under contexts 2. and 3., average robustness is increased from the deterministic approach by more or less 75% and 65% respectively, when using r_1^{mdrs} . Finally, using $\Delta = 1$ compared to $\Delta = 2$ in r_{Δ}^{mdrs} reveals to be more advantageous here, although this is strongly problem dependent.

$f =$	1.	2.	3.	4.	5.	δ
$f^{\text{mdrs}} \times r_1^{\text{mdrs}}$	90.5	81.6	70.8	22.8	95.2	7
$f^{\text{mdrs}} \times$	SAA _{10²}	89.4	78.6	47.3	8.3	96.6
	SAA _{5,10²}	90.6	79.8	49.7	16.7	97.6
	SAA _{10³}	89.7	78.3	56.2	19.4	97.1
	SAA _{5,10³}	88.8	79.3	62.3	21.6	98.2
	SAA _{10⁴}	87.0	76.1	55.7	19.3	96.8

Table 2: Average results obtained by using the SAA-based method, depending on the experimental context (from 1. to 5.), compared to average results obtained by $f \times r_1^{\text{mdrs}}$. Different scenario pool sizes are considered, from 10^2 to 10^4 .

$f =$	1.	2.	3.	4.	5.	δ
$f^{\text{mdrs}} \times r_1^{\text{mdrs}}$	89.9	81.0	68.1	19.6	95.1	8
$f^{\text{mdrs}} \times$	SAA _{5,10²}	86.9	74.7	46.1	12.4	96.5
	SAA _{10³}	85.7	74.5	51.8	12.1	96.2
	SAA _{5,10³}	69.5	56.5	39.4	8.6	86.3

Table 3: Average results obtained by using the SAA-based method, compared to average results obtained by $f \times r_1^{\text{mdrs}}$, computation time reduced to 10 minutes.

In fact, in our MDRS case study involving time horizons that span from 6 to 13 days, Δ provides a parameterizable trade-off between accuracy and computational effort.

Comparison with SAA. Results reported in Table 1 clearly motivate the use of a robust formulation. However, the design and implementation of r^{mdrs} and r_{Δ}^{mdrs} can only be justified if it allows better average results compared to the Sample Average Approximation (SAA) method.

Table 2 reports the average results obtained by repeating all the experiments, while using $f \times \text{SAA}_N$ as LS objective function, with the number N of sampled scenario varying from 10^2 up to 10^4 . Both accuracy and computational efficiency of SAA depends on N , which is in fact problem dependent. In our case, $N = 5 \cdot 10^3$ seems to constitute the best compromise on average, when computational time is limited to 30 minutes. Despite the interesting results obtained by SAA_{5,10³}, it is however outperformed by r_1^{mdrs} in contexts 2., 3. and 4. In terms of robustness, these contexts are of the utmost importance for anyone concerned by the issues and limitations of processing time estimations.

The superiority of a closed-form function, such as r_1^{mdrs} , over SAA is closely related to the available computation time. It is likely that, provided a couple of hours rather than of 30 minutes, SAA would eventually outperform r_1^{mdrs} . In contrary, reducing computation time to 10 minutes tends to significantly reinforce the superiority of r_1^{mdrs} , as show in Table 3. We note that, as we reduce computation time, SAA obtains better results by reducing the number of samples, hence improving diversification in the LS process.

Accuracy of r^{mdrs} . Moving from the theoretical core problem to the real one faced at MDRS naturally introduces limitations, leading to the proposed simplifying assumptions. In Fig. 6, we show how this impacts the accu-

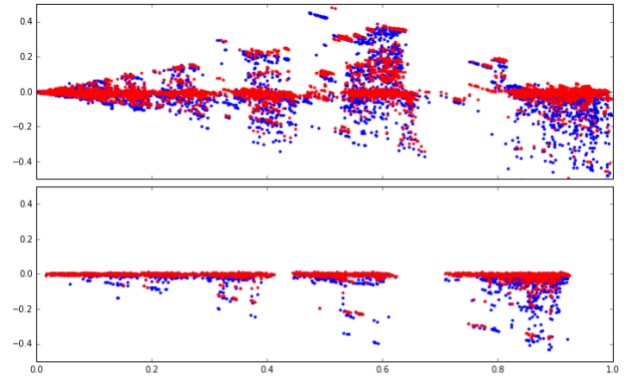


Figure 6: Average deviation of r_1^{mdrs} (blue) and r_2^{mdrs} (red) robustness measures (y -axis), depending on the solutions' true robustness (x -axis). Top: SOL0. Bottom: SOL7.

racy of the robustness functions r_1^{mdrs} and r_2^{mdrs} , on both instances SOL0 and SOL7. For every incumbent solution s encountered by the LS algorithm during the previous experiments (± 8700 solutions for SOL0), we recomputed the true robustness of s under experimental context 1., thus by using SAA_{10⁵}. Fig. 6 shows how the value $r_{\Delta}^{\text{mdrs}}(s) - \text{SAA}_{10^5}(s)$ evolves with respect to SAA_{10⁵}(s).

We naturally observe that the accuracy of the robustness measure, as computed by r_1^{mdrs} and r_2^{mdrs} , first depends on the size of the problem: there is a clear difference between SOL0 (237 jobs) and SOL7 (149 jobs). We note that r_2^{mdrs} seems globally more accurate than r_1^{mdrs} . Having a closer look reveals that under SOL0 (*resp.* SOL7), around 82% (*resp.* 99%) of values computed by r_2^{mdrs} are comprised in $\text{SAA}_{10^5}(s) \pm 0.01$, whereas only 75% (*resp.* 99%) for r_1^{mdrs} .

The average accuracy of r_{Δ}^{mdrs} globally decreases as the true robustness of the solutions increases. This suggests a hybrid approach, mixing r_{Δ}^{mdrs} in the early stage of the LS process, whereas SAA (e.g. SAA₅₀₀₀) as soon as the robustness of the incumbent solution exceeds some predefined threshold. Finally, we observe that under SOL7, r_{Δ}^{mdrs} is globally not overestimating the true robustness. As a matter of fact, while examining the other SOL n instances, the proportion of over-estimations tends to decrease with the problem size (*i.e.* the number of jobs).

Length of the horizon and deterministic quality. It is interesting that, in the experimental results, no relation appears between the number of operational days and the deterministic quality of a solution. Our first intuition tells us that the shorter is the horizon, the better the deterministic quality, because the more likely we respect the a priori schedule. We think that what actually happens is that the length of the horizon, in each instance, is compensated by the urgency of the tasks. In fact, when 10 days remain, one can afford postponing tasks because of a poor decisions. When there are only 2 days left, things start to be too urgent and postponing may not remain an option anymore. In the end, both compensate so that the average quality of a deterministic planning may after all not depend on the length of the horizon.

Conclusions and research directions

In the context of a recent Mars analog mission, we propose robust models for daily decisions in operations scheduling. Simulations show that our method, by taking the processing time uncertainty into account when designing a schedule, produces solutions significantly more reliable than those obtained using a classical deterministic model, even when the available stochastic knowledge is of very poor quality.

Our experiments take into account the uncertainty on the available probability distributions themselves. We study how the results are impacted by their accuracy, hence exploring different experimental contexts. In particular, even when using a widespread risk mitigation practice that consists in overestimating (at planning phase) all the average processing times, the average probability of a mission success can be multiplied by *three* when using our robust approach. The solutions' robustness comes at a relatively low price, their quality being impacted by 7% on average, whereas the probability to stay feasible is significantly increased.

We explore two fundamentally different approaches for evaluating the robustness of a solution: the proposed closed-form formulas and the well known Sample Average Approximation method. Depending on the problem and available computation time, results suggest that the strengths of both techniques could be combined into a hybrid algorithm.

Our current understanding of the problem could be improved by conducting further experiments, on a broader set of operational contexts. In fact, exploiting available data from different projects is likely to require new specific, exotic constraints to be considered at planning and optimization phases, leading to a more comprehensive model.

Existing techniques for robust (*a.k.a.* proactive) scheduling are mostly *redundancy-based*, or make use of *temporal protection* (Herroelen and Leus 2005). Based on random variables, our method considers the original set of tasks without duplicating nor modifying the data, and searches for the expected best sequencing of the tasks. This makes our approach compatible with the two previously cited ones, and experiments should be conducted while mixing for example with redundancy. Further research should also be considered for alternative computational models such as Bayesian networks (Darwiche 2009), which naturally apply to our random objective function, and for alternative representations of the uncertainty, e.g. by using fuzzy numbers instead of random variables (Huang and Teghem 2012).

Application to other domains. Whereas the paper considers a Mars analog mission, the proposed approach can be of interest in many other domains. Consider for instance the case of the biotechnology industrial domain. In biotech companies, scheduling the manufacturing projects (*e.g.* production of vaccines, drugs, *etc.*) is a problem for which our approach is potentially particularly valuable. The main reasons are that: 1) their tasks must be performed by humans, hence having highly variable processing times, and 2) they must cope with really strict operational constraints (the so-called GMPs). In fact, we claim that our approach applies to any operational context for which planning or scheduling

involves time uncertainty and hard operational constraints.

Acknowledgements. This work could not have been conducted without the involvement of the entire *UCL to Mars 2018* crew: Bastien Baix, Frédéric Peyrusson, Martin Roumain, Ariane Sablon, Mario Sundic, Sophie Wuyckens and Maximilien Richald. We also thank the Mars Society, in particular Shannon Rupert and the Mission Support personnel, for creating the conditions that make rotations at the MDRS realistic Mars analog sojourns.

References

- Ahmed, S., and Shapiro, A. 2002. The sample average approximation method for stochastic programs with integer recourse. *E-print available at <http://www.optimization-online.org>*.
- Applegate, D., and Cook, W. 1991. A computational study of the job-shop scheduling problem. *ORSA Journal on computing* 3(2):149–156.
- Chari, T.; Chaabane, S.; Aissani, N.; and Trentesaux, D. 2014. Scheduling under uncertainty: Survey and research directions. *2014 International Conference on Advanced Logistics and Transport (ICALT)* 229–234.
- Darwiche, A. 2009. *Modeling and reasoning with Bayesian networks*. Cambridge university press.
- Goren, S., and Sabuncuoglu, I. 2008. Robustness and stability measures for scheduling: Single-machine environment. *IIE Transactions (Institute of Industrial Engineers)* 40(1):66–83.
- Hall, N. G., and Magazine, M. J. 1994. Maximizing the value of a space mission. *European Journal of Operational Research* 78(2):224–241.
- Herroelen, W., and Leus, R. 2005. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research* 165(2):289–306.
- Huang, S.-Y., and Teghem, J. 2012. *Stochastic versus fuzzy approaches to multiobjective mathematical programming under uncertainty*, volume 6. Springer Science & Business Media.
- Lenstra, J. K., and Kan, A. H. G. R. 1979. Computational complexity of discrete optimization problems. *Annals of Discrete Mathematics* 4:121–140.
- Miller, C. E.; Tucker, A. W.; and Zemlin, R. A. 1960. Integer Programming Formulation of Traveling Salesman Problems. *Journal of the ACM* 7(4):326–329.
- Saint-Guillain, M.; Solnon, C.; and Deville, Y. 2017. The Static and Stochastic VRP with Time Windows and both random Customers and Reveal Times. In *Evolutionary Algorithms and Meta-heuristics in Stochastic and Dynamic Environments (EvoSTOC)*, 110–127. Springer International Publishing.
- Turner, R. J.; Huemann, M.; Anbari, F. T.; and Bredillet, C. N. 2010. *Perspectives on projects*. Routledge.