Robust scheduling under uncertainty: an application to operations management at MDRS
Experimental Plan

Michael Saint-Guillain[1,2]

[1] ICTEAM, Université catholique de Louvain, Belgique
[2] INSA-Lyon, LIRIS, UMR5205, F-69621, France
michael.saint@uclouvain.be

Abstract

This report presents the experimental plan of a study to be conducted at Mars Desert Research Station (MDRS). We consider the problem of scheduling a set of (experimental and logistic) operations in a constrained context such as the MDRS. When taking uncertainty into account when designing models, a robust solution can be obtained such that it minimizes the probability of a failure in the mission global objectives. Based on real operations at MDRS, we will investigate different mathematical formulations for both deterministic and robust stochastic modelings of the problem, the later taking uncertainty into account. In order to solve the deterministic problem, optimal and heuristic algorithms will be designed. Our principal goal is to show how the later can be modified in order to take stochasticity into account while maximizing the probability of mission completion. Finally, computational experimentations will compare both approaches and highlight good practices for the best of future real operation planning on Mars and similar environments.

Figure 1: The Mars Desert Research Station.

1 Introduction

The problem of scheduling a set of operations in a constrained context such as the *Mars Desert Research Station* (MDRS, Figure 1) is already a non-trivial problem, even in its classical deterministic version. It should be seen as a generalization of the well-known *job-shop scheduling problem*, which is not only NP-complete [4] but has also the reputation of being one of the most computationally demanding combinatorial optimization problems [1].

At the MDRS, computing an optimal schedule for all the operations becomes significantly less attractive as soon as some problem data, such as the manipulation time of some experiments, reveals to be different from its predicted exact value. In a constrained environment with shared resources and devices, such deviation can propagate to the remaining planned operations, eventually leading to a global infeasibility (e.g. a crew member cannot achieve all its planned experiments).

In this report, we present the experimental study to be conducted at MDRS. We will propose and compare several different mathematical formulations of the operational problem faced at MDRS. Namely, the time-constrained scheduling of the set of activities, or jobs, to be processed by the MDRS scientific staff sharing limited devices and working spaces during a fixed-length stay at the station. The scheduling problem variant we study being very specific to our MDRS case study, its modeling will probably significantly differ from the literature. Some of the models we will investigate are deterministic, but the unpredictable nature of the operational environment motivates the need of robust models, leveraging the stochastic knowledge to take decisions that maximize the probability of success. We will design both optimal and heuristic solving methods for the deterministic problem, and a extended heuristic approach for the stochastic variant of our problem. Using experimental, realistic benchmarks generated in light of real observations during two weeks as part of a scientific staff operating in the MDRS, we will finally compare the different approaches proposed.

This report is organized as follows. In section 2 we further discuss the combinatorial problem faced in our case study. Section 3 describes, steps by steps, the different planned milestones towards the realization of the study. Conclusions and perspectives are discussed in section 4.

2 The MDRS case study

The Mars Desert Research Station (MDRS), owned and operated by the Mars Society, is a full-scale analog facility in Utah that supports Earth-based research in pursuit of the technology, operations, and science required for human space exploration.[1]

The scientific staff will be constituted of 8 crew member, and the whole mission will lasts fifteen days. When arriving at MDRS, each crew member will conduct its own scientific study, involving different fields such as medicine, biology, geospatial engineering, building engineering, chemistry and particle physics.

Whereas each crew member has independent goals, the whole mission will be considered as a success only in the case where all the projects are completed. Interaction is likely to appear in the scientific staff, as some manipulations can be performed by more than one person. In addition, some of the projects contain optional (prioritized) parts.

In this application, we study the effects of job processing time uncertainty in the robustness (*i.e.* reliability) of operation scheduling in an environment such as MDRS. Thanks to information gathered from the staff members and real data collected during the monitoring of the mission, we will identify stochastic patterns about the job processing times. The stochastic patterns can then be exploited for generating realistic benchmarks of variable sizes.

[1]Source: http://mdrs.marssociety.org

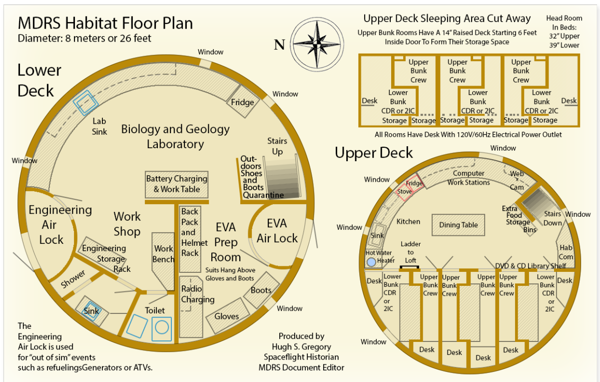


Figure 2: Plan of the Mars Desert Research Station habitat. It is divided in two floors, and its diameter is of 8 meters. *Source: http://mdrs.marssociety.org*

3 Experimental plan

This section summarizes the different intermediate steps that constitutes the case study.

3.1 Before landing

1. **Requirement analysis**: Maybe the most important part of the project consists in collecting and compiling an exhaustive list of requirements from the scientific staff involved in the mission. When arriving at MDRS, each crew member carrying its own project, from different scientific fields.

   In addition, a number of daily logistic operations will have to be shared amongst the crew members. Figure 2 shows a plan of the MDRS habitat. In a constrained environment with limited space, tools and devices, even simple activities such as cooking or cleaning may require a significant amount of time (previous inhabitants of the station estimate cooking time at about 2 man-hours).

   The very first objective will therefore be to collect the a priori data from the scientific staff, and compile a full dataset containing:

   (a) The list of available devices (*machines*) together with their respective constraints (*e.g.* time outside the pressurized environment is limited to 3 hours per day and per person — so are the use of related machines, such as the rovers);

   (b) The overall project of each scientific will be segmented in a set of jobs to be processed at MDRS, each of these coming with:

      i. type of machine required by the job;
      ii. required helping people, if needed;

iii. precedence constraints, if any;
iv. the estimated processing time of the job;
v. a level of confidence on the estimation;
vi. the job priority.

Some jobs can only to be processed once other specific jobs have been achieved (precedence constraints). Each job is given a priority by its owner, from 1 (optional) to 5 (mandatory, fail of the mission otherwise).

(c) A list of daily jobs shared between crew members with the corresponding machine requirements, processing time estimations and confidence levels.

The purpose of the a priori dataset will be to design the model and thus, the adequate solving algorithms.

2. **Deterministic model and algorithms**: The deterministic optimization problem associated to our case study can be formulated as the following integer program:

$$\begin{aligned} z^* &= \max_x \ f(x) \\ \text{s.t.} & \quad x \in X, \end{aligned}$$

where X represents the set of linear inequalities, or constraints, that define our complex job-shop scheduling problem at MDRS. In particular, this solution space is meanly based on the whole set of mandatory jobs, their constraints, and the processing time estimations which we take here (in the deterministic model) as granted. Objective function $f(x)$ is a weighted sum reflecting the amount of valuable optional task processed in solution x, in addition to the mandatory jobs.

We will use the Constraint Programming [7] paradigm to solve this deterministic problem. Constraint programming dissociates the modeling part of the problem from the systematic exploration of the solution tree:

$$SOLVE = MODEL + SEARCH$$

Whilst constraint programming (CP) allows a very descriptive and easy way to model a problem, modern CP solvers come with efficient search algorithms out of the box. The user simply provides a description of the problem (the model) in a language that the solver understands. The solver then triggers the adequate algorithms in order to search the solution space, and eventually come up with a provably optimal solution.

A heuristic solving method will also be designed. Provided the CP model of the problem, a simple heuristic solving method is the one easily obtained by considering the so-called *Large Neighborhood Search* (LNS, [8]) approach. LNS is based on the Local Search paradigm. From a feasible (non-optimal) initial solution x, LNS obtains a different, neighboring solution x' by first selecting a subset of the decision variables in x to be fixed, and re-optimizing on the remaining variables only. By iteratively repeating the operation, we explore the solution space by jumping efficiently from one solution to another whilst never degrading the objective function. Eventually, the process can be stopped whenever the solution looks near-optimal enough.

3. **A priori planning**: The last operation before flying to MDRS will obviously be to compute a (near-)optimal solution to be following by the scientific staff as soon as the mission starts in Utah. Provided that such a priori solution, the crew will be given the operational planning of the whole stay at MDRS, stating which task each crew member should perform (for either shared tasks or its own research project) during each of the fifteen days of the mission.

3.2 Operations at MDRS

- **Operation monitoring**: During the entire duration of the mission, we will monitor and record everything related to the uncertainty in the job processing times in order to improve the quality of the a priori estimations. Furthermore, the requirement analysis (described at phase 3.1-1) will be either validated or improved according to observation.
- **Daily re-scheduling**: As executive officer of the crew, my job will be to provide the staff everyday with an optimal planning. Therefore, it is very likely that the data and/or the model will have to be modified in a regular basis in order to take random events realizations into account. Each time the data or the model changes, a new solution (planning) will be computed directly at MDRS.

3.3 Post analysis

1. **Stochastic robust model and solution method**; The stochastic problem of finding robust solutions will can be formulated as a *chance-constrained program* (CCP, see [3]). Let $\boldsymbol{\xi}$ be the random vector describing the uncertainty in our problem data. We then have one random variable

$\boldsymbol{\xi}_i \in \boldsymbol{\xi}$ per stochastic job processing time, the distribution of which being approximated based on experience at MDRS (steps *3.1-1* and *3.2-(a)*). Let S be the set of distinct realizations of $\boldsymbol{\xi}$ having a positive probability. Namely, S is the set of all possible scenarios.

If $\alpha \in [0, 1]$ is the desired level of robustness that is, the required probability that the solution x remains feasible, the CCP can be formulated as:

$$
\begin{aligned}
z^* = \quad & \max_x \; f(x) \\
\text{s.t.} \quad & \sum_{s \in W(x)} \Pr(\boldsymbol{\xi} = s) \quad \geq \alpha \\
& W(x) = \{s : x \in X_s,\; s \in S\},
\end{aligned}
$$

where X_s defines the solution space of our problem under specific scenario $s \in S$. $W(x)$ is the set of scenarios in which a priori solution x remains feasible. In other words, the optimal solution x^* of cost z^* is the one that maximizes $f(x)$ whilst maintaining a probability α of remaining feasible.

We argue that the term

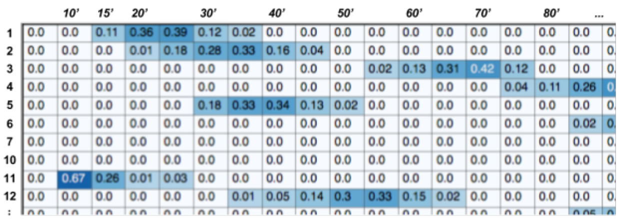
$$\sum_{s \in W(x)} \Pr(\boldsymbol{\xi} = s)$$

can be efficiently computed (in pseudo-polynomial time). An a priori solution x is actually a set of sequences of scheduled jobs, one sequence per crew member. The key idea is to think at this set of sequences as of a set of vehicle routes, where each job is a customer in the Vehicle Routing Problem with Time Windows (VRPTW). In particular, in our case the travel times to (or equivalently service times at) each customer correspond to the job processing times and are stochastic. Therefore, any of the following papers [5, 6, 2] that propose a solution method for the VRPTW with Stochastic Travel Times (or service times) describes, provided some minor adaptations, an pseudo-efficient way to compute $\sum_{s \in W(x)} \Pr(\boldsymbol{\xi} = s)$.

The solving method will be adapted from the LNS algorithm used for the deterministic problem, by simply using the computation of the feasibility likelihood shown above as criterion for selecting robust solutions during the local search process, as shown at line 5 of Algorithm 1. The algorithm assumes the existence of a solver $\mathcal{O}$ for the deterministic problem.

Algorithm 1: LNS based local search method to seek for robust a priori solutions

```
1  Let x be an initial a priori solution such that
     x ∈ X.
2  while some stopping criterion is not met do
3      Select a random subset V of problem decision
         variables.
4      Solve the reduced deterministic optimization
         problem obtained by fixing values of V
         according to current values in x:
                    x' ← O(V, x)
5      if Σ_{s∈W(x')} Pr(ξ = s) ≥ α and x' ≥ f(x*)
         then
6          set x* to x'
7  return the best a priori solution x* found
```



		10'	15'	20'		30'		40'		50'		60'		70'		80'	...	
1	0.0	0.0	0.11	0.36	0.39	0.12	0.02	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
2	0.0	0.0	0.0	0.01	0.18	0.28	0.33	0.16	0.04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.02	0.13	0.31	0.42	0.12	0.0	0.0	0.
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.04	0.11	0.26	0.
5	0.0	0.0	0.0	0.0	0.0	0.18	0.33	0.34	0.13	0.02	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.02	0.
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
11	0.0	0.67	0.26	0.01	0.03	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.
12	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.05	0.14	0.3	0.33	0.15	0.02	0.0	0.0	0.0	0.0	0.
:	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.05	0.

Figure 3: Stochasticity in the job processing times. Cell (i, j) gives the probability that job i actually requires a processing time of j minutes.

2. **Benchmark generation**: The data collected at step 3.1-1 will be exploited in order to create realistic benchmarks to compare the robustness (and hence the reliability) of solutions obtained using the different formulations.

   In particular, we will try to best represent the uncertainty in the job processing times, using estimations provided by the scientific staff. We will draw a map of the stochastic knowledge from these estimations together with the confidence levels indicated by the staff members, and the observations made during the mission. This is illustrated in Figure 3.

   Additional benchmarks of varying size (size of the scientific staff, duration of the mission, number of jobs and machines) will then be generated, based on the real benchmark representing our mission. Those additional benchmarks will be useful to study how value of robustness varies with the size of the mission.

3. **Experimental study**: Using the real and the generated benchmarks, we will compare the behavior of the solutions one obtains using either a deterministic or a stochastic robust formulation

when planning the operations, in the hope of emphasize the superiority of one formulation over the other.

4 Conclusions and perspectives

In this report, we exposed the experimental plan that will be followed during our study on *robust scheduling of scientific operations at MDRS*. The combinatorial problem as been informally described, and we discussed the solving methods that will be considered. The implementation of the experimental setting is discussed as well.

At early current time of writing, most of the technical aspects of the project are still to be defined, as these mainly depend on the requirement analysis on the scientific staff. Nonetheless the deterministic part of our study relies on well-studied technologies and paradigms such as Constraint Programming [7] and Large Neighborhood Search [8]. The stochastic models and algorithms will have to be defined, mainly depending on the key observations to be made during the mission at MDRS.

We are confident about the scientific value of the data collected during the mission and the potential contributions of models, algorithms and final empirical analysis. Eventually, our goal is to publish the study in an international scientific conference on AI. Collaborators and sponsors will be given an thankful acknowledgment section.

References

[1] David Applegate and William Cook. A computational study of the job-shop scheduling problem. *ORSA Journal on computing*, 3(2):149–156, 1991.

[2] Simen Braaten, Ola Gjønnes, Lars Magnus Hvattum, and Gregorio Tirado. Heuristics for the Robust Vehicle Routing Problem with Time Windows. *Expert Systems with Applications*, 2017.

[3] A Charnes and W. W Cooper. Chance-constrained programming. *Management science*, 6(1):73–79, 1959.

[4] Jan K Lenstra and A H G Rinnooy Kan. Computational complexity of discrete optimization problems. *Annals of Discrete Mathematics*, 4:121–140, 1979.

[5] Xiangyong Li, Peng Tian, and Stephen C.H. H Leung. Vehicle routing problems with time windows and stochastic travel and service times: models and algorithm. *International Journal of Production Economics*, 125(1):137–145, may 2010.

[6] Douglas Miranda and Samuel Vieira Conceição. The Vehicle Routing Problem with Hard Time Windows and Stochastic Travel and Service time. *Expert Systems with Applications*, 64:104–116, 2016.

[7] Francesca Rossi, Peter Van Beek, and Toby Walsh. *Handbook of constraint programming.* Elsevier, 2006.

[8] Paul Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *Principles and Practice of Constraint Programming—CP98*, pages 417–431. Springer, 1998.

About the author

Michael Saint-Guillain assumes the role of executive officer of the team *UCL to Mars 2018*. He started his PhD studies in 2013 at Université catholique de Louvain (Belgium), and in 2016 at Université de Lyon (France), in Computer Sciences. Specialized in operations research (a sub-domain of applied mathematics involved in planning and decision making), he is mostly active in dynamic and stochastic vehicle routing problems.