The dynamic bowser routing problem

Citation for published version:
Rossi, R, Tomasella, M, Martin-Barragan, B, Embley, T, Walsh, C & Langston, M 2019, 'The dynamic bowser routing problem', *European Journal of Operational Research*, vol. 275, no. 1, pp. 108-126. https://doi.org/10.1016/j.ejor.2018.11.026

Digital Object Identifier (DOI):
10.1016/j.ejor.2018.11.026

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
European Journal of Operational Research

The Dynamic Bowser Routing Problem

Roberto Rossi*[1], Maurizio Tomasella[1], Belen Martin-Barragan[1], Tim Embley[2],
Christopher Walsh[3], and Matthew Langston[4]

[1]Business School, University of Edinburgh, Edinburgh, UK
[2]Costain Group Plc., Maidenhead, UK
[3]Cenex Advanced Technology Innovation Centre, Loughborough, UK
[4]J C Bamford (JCB) Excavators Ltd., Rocester, UK

Abstract

We investigate opportunities offered by telematics and analytics to enable better informed, and more integrated, collaborative management decisions across construction sites. We focus on efficient refuelling of assets across construction sites. More specifically, we develop decision support models that, by leveraging data supplied by different assets, schedule refuelling operations by minimising the distance travelled by the refuelling truck — the so-called "bowser" — as well as fuel shortages. Motivated by a practical case study elicited in the context of a project we recently conducted at Crossrail, we introduce the Dynamic Bowser Routing Problem. In this problem the decision maker aims to dynamically refuel, by dispatching a bowser truck, a set of assets which consume fuel and whose location changes over time; the goal is to ensure that assets do not run out of fuel and that the bowser covers the minimum possible distance. We investigate deterministic and stochastic variants of this problem and introduce effective and scalable mathematical programming models to tackle these cases. We demonstrate the effectiveness of our approaches in the context of an extensive computational study designed around data collected on site.

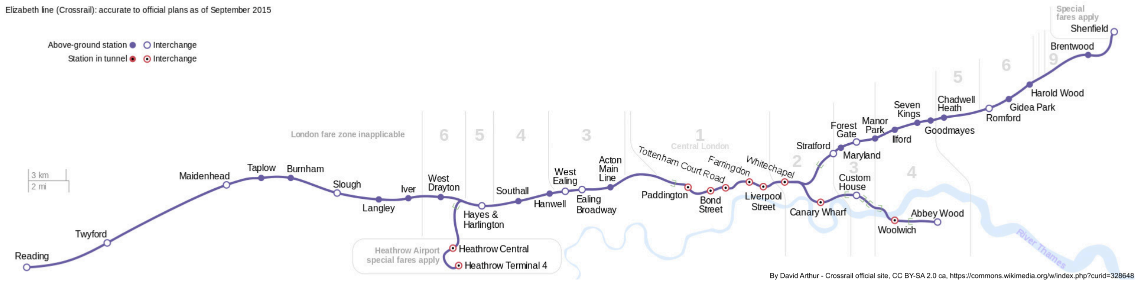
Keywords: Routing; Dynamic Bowser Routing Problem; Stochastic Bowser Routing Problem; Mixed-Integer Linear Programming; Construction.

1 Introduction

The United Kingdom (UK) National Infrastructure Plan comprises a pipeline of public investment in infrastructure worth over £100 billion between 2016 and 2020 [35] and has clear aspirations for low-carbon solutions [34, 36]. Unfortunately, the fragmented nature of construction logistics represents a challenge to these aspirations.

In the context of a public-private partnership funded by Innovate UK, our team investigated opportunities offered by telematics and analytics to enable better informed, and more integrated, collaborative management decisions in construction sites. We considered

*Corresponding author. Address: 29 Buccleuch place, EH89JS, Edinburgh, UK. Email: roberto.rossi@ed.ac.uk



By David Arthur - Crossrail official site, CC BY-SA 2.0 ca, https://commons.wikimedia.org/w/index.php?curid=328648

Figure 1: The Crossrail project

a selection of Crossrail costruction sites, a strategic UK infrastructure project for which Costain is one of the main contractors. Crossrail will deliver a new 118km high frequency, high capacity railway for London and the South East (Fig. 1). Our team mapped current construction processes and elicited barriers to the fully integrated, low-carbon construction supply chain. Building upon this analysis, we developed a set of solutions and related enabling business models; in this work we describe one of these solutions.

We focus on efficient refuelling of assets across construction sites. More specifically, we develop decision support models that, by leveraging data supplied by different assets, schedule asset refuelling operations by minimising the distance travelled by the bowser truck as well as fuel shortages. Surprisingly, there are only few comparable studies in the literature, none of which fully addresses this challenge.

Our work can be broadly positioned within the Inventory Routing literature. The field of Inventory Routing (IR), whose origins can be traced back to [9], encompasses problems which combine vehicle routing and inventory management decisions.

The origins of the “Truck Dispatching Problem,” a generalization of the Traveling Salesman Problem, date back to the seminal work by [24]. Since the early days, a sizeable literature developed on the so-called Vehicle Routing Problem (VRP), whose aim is to dispatch a fleet of vehicles on a given network to serve a set of customers while meeting a number of constraints. A comprehensive discussion on the VRP can be found in [22, 31].

Pioneering works in inventory control were carried out by [33], who introduced the concept of “economic order quantity,” and [61], who discussed the first lot-sizing algorithm for a finite-horizon inventory system subject to dynamic demand. Lot-sizing models are surveyed in [26, 53]; recent developments in stochastic lot sizing are surveyed in [59].

In IR optimisation is delegated to a central entity that jointly optimises all decisions. Recent surveys in the area include [15, 4, 14]. The first exact approach to the Inventory Routing Problem (IRP) was proposed by [5], who considered the single-vehicle case. Computationally efficient approaches to this problem were discussed in [6, 20]. Approaches for the multi-vehicle case include [21, 19, 2]. Stochastic IRP problems, in which customer demand is modeled as a random variable, include the seminal work by [27], and a number of more recent contributions [30, 42, 1, 43, 37, 38, 62, 58].

The Dynamic Bowser Routing Problem (DBRP), discussed in this work, presents a number of similarities and differences with the classical IRP and its existing variants presented in the literature. Like the classical IRP, the DBRP features a warehouse (the fuel cistern), a truck shipping an “item” (the fuel), and a number of recipients (the assets) for such item. Like retailers in the IRP, assets hold inventory (the fuel) that is depleted over time. The key difference from existing works in the IRP literature is the fact that assets are free to move in the network that represents our construction site layout; this means that an asset location may change over time and that an asset will move across the network over time together with its inventory. Moreover, the bowser can only move from one node of the network to an adjacent one at any given time period; while in classical IRP one needs to determine the routes visiting the customers served at that time period.

Works on Probabilistic Traveling Salesman Problem [40] and Stochastic Vehicle Routing Problem [29] have considered situations in which a customer may be present at a given node of the network with a prescribed probability, but we are not aware of works investigating the case in which the very same customer, which holds inventory that may need to be replenished, moves from one node to another over time in a deterministic or stochastic fashion.

A number of other works in the literature share similarities with our study, although in different application domains. We hereby provide a survey of what we believe are the most relevant application domains; for each domain, we try to stress the key differences from our problem.

In [45, 29] the authors investigated arc routing problems motivated by winter gritting applications where the “timing” of each intervention is crucial and service cost increases if intervention is carried out outside the prescribed time window. Our model also features similar “timing” constraint related to asset fuel stockout, but in road gritting, as in classical IRP, assets that are replenished (i.e. gritted roads) do not move from one period to the next.

A stream of literature that gained substantial momentum in recent times is that on electric vehicle refueling [44, 47, 7]. In electric vehicle refueling assets do move over time from one node of the road network to the next together with their battery, which can be seen as energy inventory. However, refueling stations have a fixed location and the problem typically addresses a strategic decision whose aim is to determine where to position refueling stations.

Rebalancing in bike sharing [13, 25] is another problem that attracted considerable interest in recent time. In this problem, the aim is to utilise one or more trucks in order to regularly rebalance the number of bikes located at self-service stations. The “pick-up and delivery” nature of this problem clearly differentiates it from our problem.

Moving our attention to robotics, recently [50, 51, 52] investigated stochastic collection and replenishment of agents motivated by use cases in mining and agricultural settings in which a replenishment agent transports a resource between a centralised replenishment point to agents using the resource in the field. They employ Gaussian approximations to quickly calculate the risk-weighted cost of a schedule; a branch and bound search then exploits these predictions to minimise the downtime of the agents. Previous works in this area mainly focused on scheduling the actions of the dedicated replenishment agent from a short-term and deterministic angle. The discussion in [52, p. 59] presents similarities to our setup; this emphasises the practical relevance of our study. However, our modeling and solution framework has the advantage of relying solely on MILP modeling and not on ad-hoc algorithms to predict future asset resource levels. Moreover, the discussion in [52] assumes that uncertain parameters and variables are normally distributed, while our approach based on piecewise linearisation of loss functions does not require this assumption and can accommodate any distribution.

Finally, our problem can be seen as a simplified version of the Aerial Fleet Refueling Problem (AFRP) [8] in which the aim is to schedule deployment of tankers and receiver aircraft, located at diverse geographical locations, in support of immediate and anticipated military operations. In contrast to the DBRP, the AFRP features a complex multicriteria and hierarchical objective function that does not simply minimize the distance covered by the tanker aircraft. Furthermore, this problem features a substantial number of additional constraints related to safety of crew and aircrafts. A number of more recent works investigate the Aerial Refueling Scheduling Problem (ARSP) [41], whose aim is to determine the refueling completion times for fighter aircrafts on multiple tankers in order to minimize the total weighted tardiness. The ARSP however does not consider the topology of the theatre of operations.

Having surveyed related works, we next outline our contributions:

- motivated by a practical case study elicited in the context of our project, we introduce the DBRP;
- we develop mathematical programming models to tackle deterministic as well as stochastic variants of the problem and we augment these models with valid inequalities that enhance computational performances;
- we introduce a comprehensive test bed built upon real world scenarios and data observed in the context of our experience at Crossrail sites and we carry out a thorough computational study based on it; our mathematical programming models scale well and can tackle instances of realistic size in reasonable time;
- for the stochastic variant of the problem, we contrast results obtained via our mathematical programming heuristic against the optimal policy obtained via stochastic dynamic programming; our analysis shows that our approximation is effective.

The rest of this work is organised as follows. In Section 2 we survey telemetry systems and related datasets that have been used in our study. In Section 3 we introduce the DBRP and an associated mathematical programming model. In Section 4 we discuss the Stochastic Bowser Routing Problem (SBRP), which generalises the DBRP to deal with stochastic factors, such as asset fuel consumption and location on site; we discuss a mathematical programming heuristic for the case in which asset fuel consumption is stochastic. In Section 5 we present our computational study investigating effectiveness and scalability of our models. Finally, in Section 6 we draw conclusions.

2 Telemetry at Crossrail construction sites

Modern construction machines, such as excavators, bowser trucks, but also power generators and pumps, feature a plethora of sensors including but not limited to GPS location, fuel level, and engine status. *Telemetry* is the transmission of measurements collected by these sensors from the equipment to the point of storage/consumption of the data [28, 48]. By building upon telemetry, *telematics* brings together sensor technologies, telecommunication, and computer science to monitor and control remote objects [49]. There is a close connection between telematics and analytics [57]. Telematics systems rely on visualisation technologies (descriptive analytics), on predictive algorithms (predictive analytics), and on optimisation models (prescriptive analytics) to support and automate decision making.

The introduction of sector-wide standards such as the Association of Equipment Manufacturers Professionals (AEMP) Telematics Standard [3] has made it possible to collect and integrate data from a wide range of sources located across a single or multiple construction sites. By relying on this standard, live as well as historical data for each piece of equipment can be obtained from Application Programming Interfaces (APIs) that rely on standard HTTP protocols. This opens up a wide range of opportunities to deploy analytics across the construction site. A sample of telemetry data in XML format that can be obtained from assets by leveraging the AEMP standard v1.2 is shown in our Supplementary Material. Sampling rate varies from one manufacturer to another, for instance the JCB LiveLink$^{\mathrm{TM}}$ system remotely samples data from assets every five minutes, while Komatsu Komtrax$^{\mathrm{TM}}$ features a coarser sampling rate of thirty minutes.

In this work we exploit two indicators that can be currently monitored via existing telemetry systems: *asset (GPS) location* and *fuel consumption*: Fig. 2 illustrates location of a JCB 540-170 telehandler at the Connaught bridge Crossrail site in London between the 15th and the 19th of March 2016; Fig. 3 illustrates cumulative fuel consumption of three JCB 540-170 telehandlers deployed on various Crossrail sites between February and March 2016.

Telemetry data can be used to track past fuel consumption and predict future consumption via predictive analytics techniques. The analysis of past performance generally proceeds from the visualisation of relevant indicators, for instance in the form of heatmaps (Fig. 4). Standard methods such as linear and non-linear regression, possibly taking into account asset location or other relevant predictor variables, can be employed to predict future fuel consumption. The thorough investigation of suitable predictive analytics strategies to determine future asset fuel consumption or location is beyond the scope of this work. However, in the economy of our investigation it is relevant to determine prototype consumption profiles for a range of assets deployed on construction sites, which we can then use in our computational study.

By looking at the heatmap in Fig. 4 it is clear that asset utilisation follows specific patterns; for instance, this asset operates from early morning to mid afternoon and does not generally operate over weekends — note that different assets feature different profiles, for instance some assets operate 24/7. In this specific instance, if we consider a time window during which the asset is in use — e.g. the time window from Mon, 6th of June

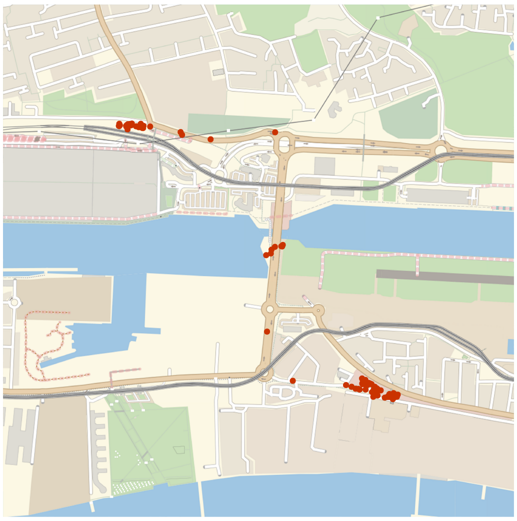


Figure 2: GPS location of a JCB 540-170 telehandler at the Connaught bridge Crossrail site in London between the 15th and the 19th of March 2016

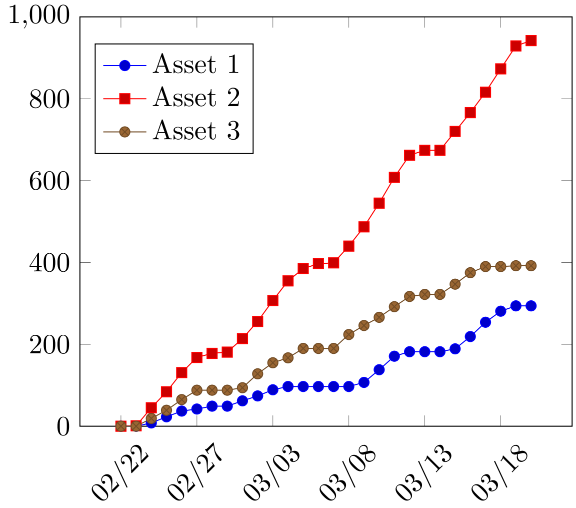


Figure 3: Fuel consumption (in liters) of three JCB 540-170 telehandlers deployed on Crossrail sites between February and March 2016

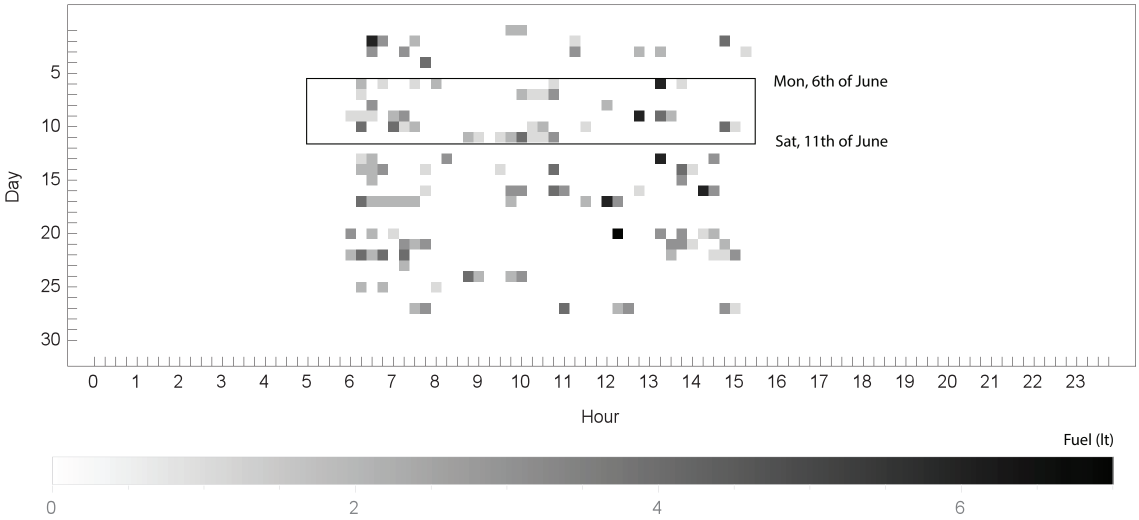


Figure 4: Heatmap representing fuel consumption (in litres) for a Crossrail JCB 540-170 telehandler between the 1st of June and the 30th of June 2016; note that time is divided into discrete time periods that last 15 minutes each.

to Sat, 11th of June highlighted in Fig. 4 — a compound Poisson distribution with parameter $\lambda = 0.502645$ and jump size distribution Poisson(0.602257) appears to provide a good fit for the consumption pattern observed during independent time buckets of 15 minutes. Our empirical study revealed that a compound Poisson distribution with Poisson jump size distribution generally provides a good fit for asset consumption data we collected in June 2016; by exploiting a month worth of fuel consumption data collected from different types of assets, including a variety of telehandlers and excavators, we excluded periods of inactivity (i.e. nights and weekend) and we fitted distribution parameters using a maximum likelihood approach; the resulting distributions are shown in Table 1, the respective p-values are also reported in the table. Further details on the analysis carried out, including heatmaps for assets analysed, are provided in our Supplementary Material.

In addition to analyzing asset fuel consumption distributions, we also analysed the behaviour of a 7.5 tonne bowser truck fitting a 950 litres fuel tank that refuels assets across a number of Crossrail construction sites. We installed a GPS logger on the truck and tracked its movements over a working week from Mon, 20th of June to Fri, 24 of June. The daily distance covered by the bowser truck and the number of journeys are reported in Fig. 5; Fig. 6 shows distance covered at different hours. A typical day for the bowser truck comprises two long (approx 40km) journeys across construction sites, and about 20 short journeys carried out within specific construction sites. An example of a long journey between the Connaught bridge site and the Plumstead site site of Crossrail is shown in Fig. 7; examples of short on-site journeys at Plumstead site are given in Fig. 8.

In the rest of this work we investigate optimisation models that rely upon telemetry data to schedule asset refuelling operations across construction sites.

3 The Dynamic Bowser Routing Problem

Motivated by the discussion in the previous sections, we introduce the DBRP. A comprehensive table of symbols used in the rest of this work is introduced in Appendix I. We consider a construction site with A assets (e.g. generators, telehandlers, excavators) all powered by a single type of fuel (e.g. diesel). The construction site map is given. This map takes the form of a directed graph $\langle V, E\rangle$, where V is the set of nodes, E is the set

	Compound Poisson		
Asset model	λ	jump size distribution	p-value
JCB 540-170	0.503	Poisson(0.602)	0.920
JCB 540-170	0.774	Poisson(0.684)	0.450
JCB 540-170	0.373	Poisson(1.005)	0.933
JCB JS130	1.039	Poisson(1.011)	0.461
JCB JS130	0.926	Poisson(0.394)	0.117
JCB 86C-1	0.477	Poisson(0.961)	0.779
JCB 531-70	0.283	Poisson(0.052)	0.516

Table 1: Fitted distribution for a selection of JCB assets deployed on Crossrail sites in June 2016; the distribution represent the fuel consumption over a 15 minutes time bucket.

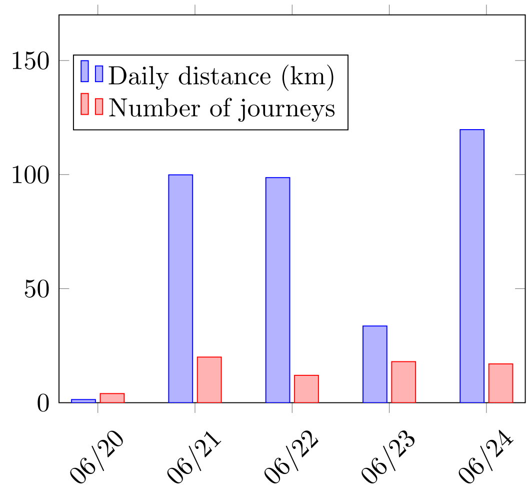


Figure 5: Daily distance covered by the bowser truck and number of journeys completed on a specific date

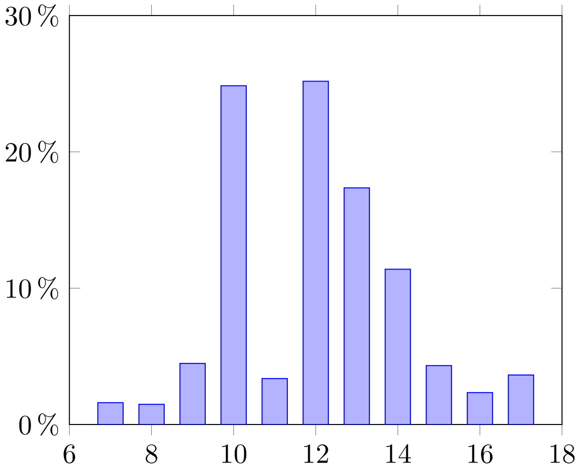


Figure 6: Percentage distance covered by the bowser at a given hour of the day

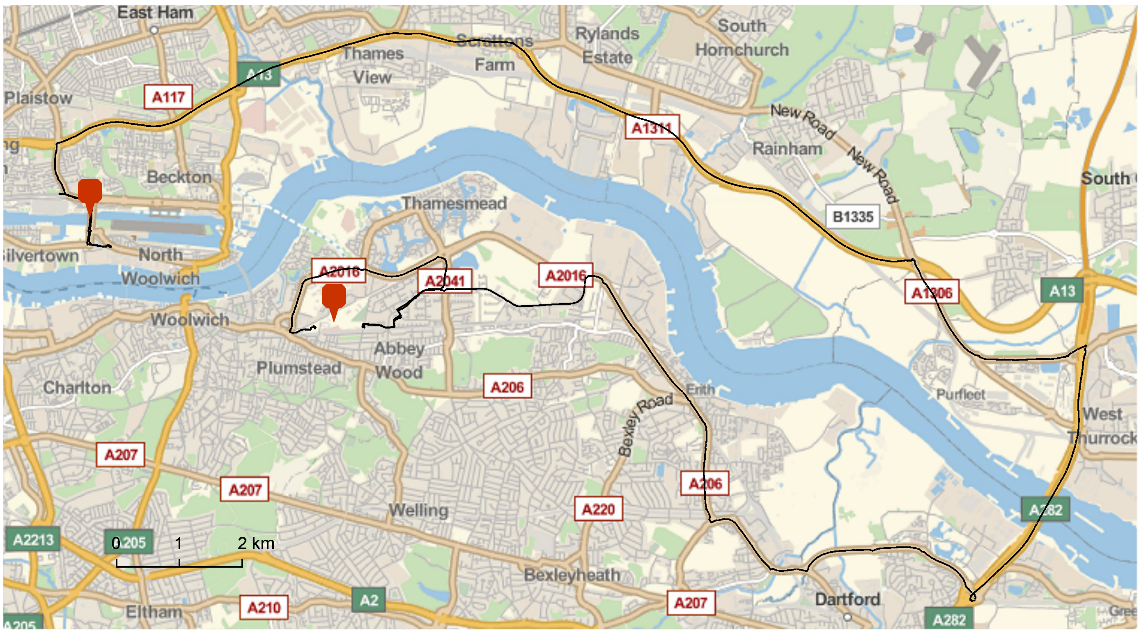


Figure 7: A bowser journey from the Connaught bridge site in North Woolwich to the Plumstead site of Crossrail on Tue, 21th of June; the bowser covered 47.6km in 1h and 31 minutes.

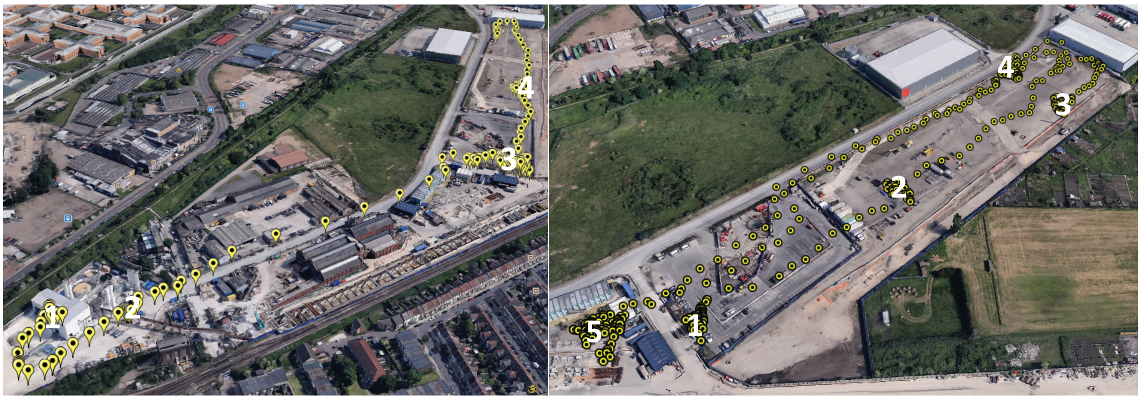


Figure 8: Two short on-site journeys at Plumstead site; the first journey (on the left hand side) covered 1.3km in 44 minutes and comprises four stops of less than five minutes each; the second journey (on the right hand side) covered 1.7km in 37 minutes and comprises five stops of less than five minutes each.

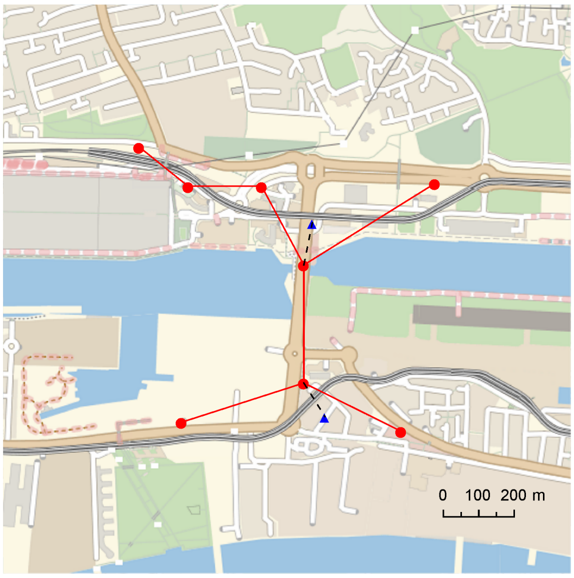


Figure 9: Sample site network for the Connaught bridge Crossrail site in London; triangles represent assets.

of arcs, and $N = |V|$. We assume this graph to be connected, but not necessarily fully connected. Nodes in V represent relevant locations across the construction site; an arc (v_1, v_2) in E connects two locations $v_1 \in V$ and $v_2 \in V$ if it is possible to travel from v_1 to v_2 in one time period. Essentially, this graph can be seen as a network that represents adjacent accessible locations (i.e. nodes), and their respective direct travel distances (i.e. arc weights), across the construction site. This form of map representation is common in GPS navigation systems and can be easily obtained via GPS traces.[1] It should also be noted that this setup is not limited to a single construction site and the graph may as well represent multiple interconnected sites.

We consider a discrete planning horizon that comprises T periods. At each point in time, an asset $a = 1, \ldots, A$ can be found in one and only one node $v \in V$; in practice, on

[1] see e.g. `https://www.openstreetmap.org/`

the basis of its GPS location an asset will be associated to the nearest node in the site network (Fig. 9). We assume that asset location $l_t^a \in V$ at each time period $t \in T$ is known with certainty. Each asset a features a fuel tank with capacity c_a and initial tank level s_a; fuel consumption $f_t^a \geq 0$ for an asset at a given time period $t \in T$ is given and known with certainty. We will relax the assumptions of certainty in Section 4.

Each construction site features a fixed site cistern where an infinite amount of fuel is assumed to be available; in a multi-site setting we assume, without loss of generality, a single cistern is available for all sites considered. There is a single bowser truck that can be used to refuel assets. The bowser features a tank with capacity c_b and initial level s_b, which is used to store fuel for refuelling assets. To refill its tank, the bowser must return to the site cistern. We do not model explicitly bowser fuel consumption, since it is unlikely the bowser will run out of fuel in between two visits to the site cistern: the daily distance covered by the bowser (Fig. 5) comfortably remains within the range a small truck can cover.

We model movements so that the bowser can only move from a node to an adjacent one within a single time period. We assume that refuelling of an asset takes a negligible time in relation to the size of time periods, and that refuelling can be performed if, at a given time period, both the bowser and the asset are located at the same node.

Our time modeling strategy resembles a "Large Bucket" strategy, as found in lot-sizing [12]. The rationale behind our choice of modeling macro (i.e. 10 minutes to 30 minutes) rather than micro (i.e. real time) periods is related to the fact that, as discussed in Section 2, asset refuelling operations generally require less than five minutes per asset and therefore fit within a period. Furthermore, the sampling rate of existing telemetry systems is quite low: between 5 minutes (JCB) and 30 minutes (Komatsu) between two readings.

Even in a deterministic setting, it is unrealistic to require that no asset stocks out, as the problem may admit no solution; we therefore choose to allow asset fuel stock outs. If an asset stocks out of fuel, we enforce a penalty cost p per litre of fuel short at the end of a given period. We assume operations do not stop as a consequence of a fuel shortage; a premium price will be paid to ensure work continuity, e.g. cost of expediting fuel, averting operation delays. Our asset therefore operate in a "lost sales" setting, i.e. when an asset is out of fuel, lost work and associated fuel consumption are not backlogged from one period to the next.

To summarize, the order of events within a given period t in the planning horizon is as follows. At the beginning of a period a bowser replenishment takes place if the bowser is at the cistern node and needs to be replenished; this replenishment is instantaneous. Immediately after, all assets that happen to be at the same node as the bowser are replenished according to the given refuelling plan. Assets then start operating and consume fuel according to the given consumption f_t^a. If an asset runs out of fuel, a premium price of p per litre of fuel short is paid to ensure business continuity. At the end of the period all assets, including the bowser, move instantaneously to their next location.

3.1 A Mixed-Integer Linear Programming model

We introduce a mixed-integer linear programming (MILP) formulation for the DBRP.

Our formulation features parameters $d_{i,j}$, denoting the distance between node i and node j in the site network; and binary parameter $\delta_{i,j}$ that is set to one iff it is possible to travel from node i to node j in one time period. There are five sets of decision variables

- V_t^i, a binary variable set to one iff, at time t, the bowser is at node i;
- $T_t^{i,j}$ a binary variable set to one iff the bowser transits from node i to node j at the end of period t, we only introduce variables $T_t^{i,j}$ for i, j such that $\delta_{i,j} = 1$;
- Q_t^a, the nonnegative quantity of fuel delivered to asset a at time t;
- S_t^a, the nonnegative fuel shortage for asset a at time t;
- B_t, the nonnegative quantity of fuel transferred from the cistern to the bowser at time t.

The MILP model is presented in Fig. 10. The objective function (1) minimises the total distance covered by the bowser plus the penalty cost associated with fuel shortages; note that both these values can be expressed in the same unit of measure, e.g. monetary units. Constraints (2) ensure that fuel cannot be transferred from the cistern to the bowser unless the bowser is at node 1; moreover, it makes sure that the nonnegative quantity of fuel B_t transferred from the cistern to the bowser at time t does not exceed the bowser capacity c_b. Constraints (3) capture the fact that the initial bowser tank level (s_b) plus all bowser replenishments (B_k) up to period t minus all fuel (Q_k^a) delivered to assets in periods $1, \ldots, t-1$ should not exceed bowser capacity c_b. Conversely, constraints (4) represent the so-called "inventory conservation constraints," which ensure that the initial bowser tank level (s_b) plus all bowser replenishments (B_k) up to period t minus all fuel (Q_k^a) delivered to assets in periods $1, \ldots, t$ should remain nonnegative. Constraints (5) capture the fact that, at each point in time, the bowser must be visiting one and only one network node. Constraints (6) establish that if the bowser is at node i at time t, it must transit to some node at the end of period t. Constraints (7, 8, 9) are channeling constraints linking variables $T_t^{i,j}$ (bowser transitions) and variables V_t^i (bowser visiting a node). Asset refuelling and inventory conservation constraints are expressed via constraints (10) and (11). More specifically, constraints (10) ensure that the initial fuel level of an asset (s_a) plus all fuel deliveries received from the bowser (Q_k^a) up to period t plus all previous fuel shortages (S_k^a) up to period $t-1$ (recall we operate in a "lost sales" setting) minus fuel consumption up to period t should be greater or equal to minus the shortage of fuel at the end of time t ($-S_t^a$). Conversely, constraints (11) ensure that the initial fuel level of an asset (s_a) plus all fuel deliveries received from the bowser (Q_k^a) up to period t plus all previous fuel shortages (S_k^a) up to period $t-1$ minus fuel consumption up to period $t-1$ should not exceed the asset tank capacity (c_a). The fact that an asset can be refuelled only if both machine and bowser are at the same node is captured by constraints (12), in which $l_{t,i}^a$ is a binary parameter that is set to one iff asset a is at node i during time period $t \in T$; i.e iff $l_t^a = i$. Finally, constraints (13) capture decision variable domains.

$$\min \quad \sum_{t=2}^{T} \sum_{i,j|\delta_{i,j}=1} T_{t-1}^{i,j} d_{i,j} + p \sum_{t=1}^{T} \sum_{a=1}^{A} S_t^a \tag{1}$$

Subject to

$$B_t \leq V_t^1 c_b \qquad t = 1, \ldots, T \tag{2}$$

$$s_b + \sum_{k=1}^{t} B_k - \sum_{k=1}^{t-1} \sum_{a=1}^{A} Q_k^a \leq c_b \qquad t = 1, \ldots, T \tag{3}$$

$$s_b + \sum_{k=1}^{t} B_k - \sum_{k=1}^{t} \sum_{a=1}^{A} Q_k^a \geq 0 \qquad t = 1, \ldots, T \tag{4}$$

$$\sum_{i=1}^{N} V_t^i = 1 \qquad t = 1, \ldots, T \tag{5}$$

$$\sum_{j|\delta_{i,j}=1} T_{t-1}^{i,j} = V_{t-1}^i \qquad t = 2, \ldots, T;\ i = 1, \ldots, N \tag{6}$$

$$T_{t-1}^{i,j} \geq V_{t-1}^i + V_t^j - 1 \qquad t = 2, \ldots, T;\ i,j|\delta_{i,j} = 1 \tag{7}$$

$$T_{t-1}^{i,j} \leq V_{t-1}^i \qquad t = 2, \ldots, T;\ i,j|\delta_{i,j} = 1 \tag{8}$$

$$T_{t-1}^{i,j} \leq V_t^j \qquad t = 2, \ldots, T;\ i,j|\delta_{i,j} = 1 \tag{9}$$

$$s_a + \sum_{k=1}^{t} Q_k^a + \sum_{k=1}^{t-1} S_k^a - \sum_{k=1}^{t} f_k^a \geq -S_t^a \qquad t = 1, \ldots, T;\ a = 1, \ldots, A \tag{10}$$

$$s_a + \sum_{k=1}^{t} Q_k^a + \sum_{k=1}^{t-1} S_k^a - \sum_{k=1}^{t-1} f_k^a \leq c_a \qquad t = 1, \ldots, T;\ a = 1, \ldots, A \tag{11}$$

$$Q_t^a \leq c_a \sum_{i=1}^{N} V_t^i l_{t,i}^a \qquad t = 1, \ldots, T;\ a = 1, \ldots, A \tag{12}$$

$$\begin{aligned} &T_t^{i,j},\ V_t^i \in \{0,1\} && t = 1, \ldots, T;\ i,j|\delta_{i,j} = 1 \\ &Q_t^a,\ B_t \geq 0 && t = 1, \ldots, T;\ a = 1, \ldots, A \\ &0 \leq S_t^a \leq f_t^a && t = 1, \ldots, T;\ a = 1, \ldots, A \end{aligned} \tag{13}$$

Figure 10: An MILP model for the Dynamic Bowser Routing Problem

3.2 Valid Inequalities

We next introduce three sets of valid inequalities [23] inspired by the discussion in [5].

The *first set of inequalities* captures the idea that the amount of fuel that can be delivered to assets is constrained by the number of visits the bowser pays to each asset as well as by the asset and bowser capacities.

Lemma 1. *For $t = 1, \dots, T$ and $a = 1, \dots, A$*

$$\sum_{k=1}^{t} \sum_{i=1}^{N} V_k^i l_{k,i}^a \geq \left(\sum_{k=1}^{t} f_k^a - s_a - \sum_{k=1}^{t} S_k^a \right) / \min(c_a, c_b) \tag{14}$$

Proof. $Q = \sum_{k=1}^{t} f_k^a - s_a - \sum_{k=1}^{t} S_k^a$ is the net amount of fuel that has been delivered to the asset by period t; the minimum value between asset a tank capacity c_a and the bowser capacity c_b can be used to determine the minimum number of visits required to deliver Q, the total number of visits paid must then be greater or equal to this value. □

The *second set of inequalities* makes sure that if an asset a has not been visited by the bowser in the time span $i, \dots, j$, then the sum of asset a refuelling quantities Q_k^a in this time span must be set to zero.

Lemma 2. *For $i, j = 1, \dots, T$, $i < j$, and $a = 1, \dots, A$*

$$M \sum_{n=1}^{N} \sum_{k=i}^{j} V_k^n l_{k,n}^a \geq \sum_{k=i}^{j} Q_k^a \tag{15}$$

where $M = \sum_{k=1}^{T} f_k^a$.

Proof. These constraints immediately follow from the description above; we let $M = \sum_{k=1}^{T} f_k^a$, since the total amount of fuel delivered to an asset should not exceed its total consumption up to time T. □

The *third set of inequalities* captures the fact that, if the bowser does not pay a visit to asset a in the time span $j, \dots, t$, then the net fuel level at period t is solely determined by deliveries $Q_1^a, \dots, Q_{j-1}^a$.

Lemma 3. *For $a = 1, \dots, A$, $t = 1, \dots, T$ and $j = 1, \dots, t$*

$$s_a + \sum_{k=1}^{j-1} Q_k^a + \sum_{k=1}^{t} S_k^a - \sum_{k=1}^{t} f_k^a \geq -M \sum_{n=1}^{N} \sum_{k=j}^{t} V_k^n l_{k,n}^a \tag{16}$$

where $M = \sum_{k=1}^{t} f_k^a$.

Proof. These constraints immediately follow from the description above; we let $M = \sum_{k=1}^{t} f_k^a$, since a right hand side value of $-\sum_{k=1}^{t} f_k^a$ deactivates the constraint. □

3.3 Numerical example

We now introduce a running example to support our discussion. We consider a planning horizon of $T = 10$ periods and a single site network with $N = 10$ nodes. The distance matrix is shown in Table 2. There are $A = 3$ assets with tank capacity $c_a = 20$ and initial tank level $s_a = 10$ for all $a = 1, \ldots, A$. The location of each asset at each time period is illustrated in Table 3. Fuel consumption of each asset at each time period is illustrated in Table 4; the penalty cost per litre of fuel short is $p = 100$. Finally, the bowser capacity is $c_b = 300$ and the initial bowser level is $s_b = 10$. An IBM ILOG OPL implementation of this example is provided in our Electronic Addendum EA1.[2]

						j					
		1	2	3	4	5	6	7	8	9	10
	1		96			107					
	2	121									
	3		92		103	103			92		77
	4		90								91
	5						102			126	
i	6			72		139		89			
	7						80		83		
	8		119							90	91
	9	83									
	10					79					

Table 2: Distance matrix representing the distances $d_{i,j}$ for the numerical example; an empty cell denotes a forbidden transit

	Period									
	1	2	3	4	5	6	7	8	9	10
Asset 1	5	10	1	1	4	4	2	6	6	6
Asset 2	6	1	9	3	2	3	7	7	10	5
Asset 3	10	5	7	5	10	2	10	10	6	3

Table 3: Location (i.e. node index in the site graph) of each asset at each time period.

	Period									
	1	2	3	4	5	6	7	8	9	10
Asset 1	4	4	2	1	3	1	4	4	3	3
Asset 2	4	2	3	4	3	1	4	2	4	4
Asset 3	2	4	1	2	2	4	1	1	2	2

Table 4: Fuel consumption (in liters) of each asset at each time period.

We solve the MILP model discussed by using IBM ILOG CPLEX Optimization Studio Version 12.6; the optimal solution, which yields a cost of 494, is found in 1.08 seconds on a 2.2GHz Intel Core i7 Macbook Air fitted with 8Gb of RAM. In this specific instance, this figure simply represents the total distance covered by the bowser, since there are no fuel

[2] Our Electronic Addendum is available on GitHub `https://github.com/gwr3n/dbrp`.

shortages. The optimal bowser routing plan is displayed in Fig. 11 and 12; the optimal refuelling plan is shown in Fig. 13.

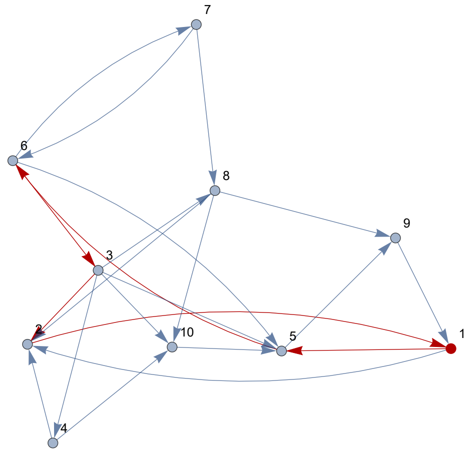


Figure 11: Optimal bowser routing plan.

t	Transition
1	1→ 1
2	1→ 1
3	1→ 5
4	5→ 6
5	6→ 3
6	3→ 2
7	2→ 1
8	1→ 1
9	1→ 1

Figure 12: Optimal bowser routing plan.

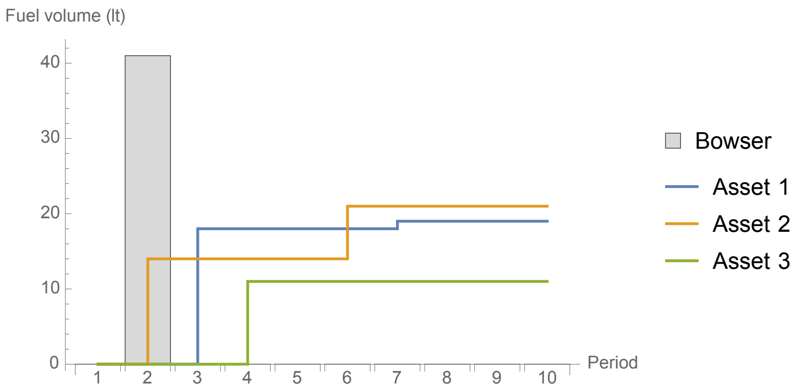


Figure 13: Bowser and asset refueling plan.

4 The Stochastic Bowser Routing Problem

In the DBRP asset *location* and *fuel consumption* throughout the planning horizon are known with certainty. In this section we investigate stochastic variants of the problem; we shall name this problem the Stochastic Bowser Routing Problem (SBRP).

4.1 A Stochastic Dynamic Programming formulation

We now introduce a dynamic programming formulation that can seamlessly capture deterministic as well as stochastic variants of the problem; in other words, this approach makes it possible to relax the assumption of certainty for asset *location* or *fuel consumption*. This flexibility comes at a price; in fact, the curse of dimensionality makes this approach far less scalable than the mixed integer linear programming approach previously discussed. However, at least for small instances, this formulation can be effectively employed to derive optimal policies and investigate the cost of uncertainty.

Introduced by Bellman in the late Fifties [11] Dynamic Programming is a powerful modelling and solution framework for decision making under uncertainty. To keep our discussion focused, we restrict the discussion to discrete stochastic dynamic programs. To formulate a problem as a stochastic dynamic program the decision maker must define the *planning horizon* length, the *state space*, the set of *feasible actions* for each state of the state space, the *transition probability* from one state-action pair to the set of admissible future states and the *immediate value* function associated with every given state-action pair. At the core of stochastic dynamic programs we find a "functional equation" — also known as Bellman's equation. In its most general form, a functional equation $v_t(s)$ can be expressed as

$$v_t(s) = \min_{k \in A_s} \left\{ c_s^k + \sum_{s'} p_{s,s'}^k v_{t+1}(s') \right\}, \tag{17}$$

where A_s denote the set of all feasible actions in state s; c_s^k denotes the immediate cost incurred for the state-action pair $\langle s, k \rangle$; and $p_{s,s'}^k$ is the transition probability from state s to state s' when action k is selected. Eq. 17 is clearly independent of the problem at hand. Note that, without loss of generality, the problem is here expressed in cost minimization form. Once all aforementioned problem elements have been defined, one can apply a backward or a forward recursion algorithm to tabulate the functional equation — a process known in the literature as "memoization" [46] — and determine an optimal policy.

We next formulate the SBRP as a stochastic dynamic program.

Planning horizon: T periods.

States: a state is encoded as a 5-tuple $\langle t, b_{\text{tank}}, b_{\text{loc}}, m_{\text{tank}}, m_{\text{loc}} \rangle$; where t is the period associated with the state, b_{tank} is the bowser tank level, b_{loc} is the bowser position in the network, m_{tank} is an array of A asset tank levels, m_{loc} is an array of A asset locations.

Actions: an action is a 4-tuple $\langle s, b_{\text{ref}}, b'_{\text{loc}}, m_{\text{ref}} \rangle$; where s is the state associated with the action, b_{ref} is the bowser refuelling quantity for the current period, b'_{loc} is the bowser location in the next period, m_{ref} is an array of A asset refuelling quantities.

Transition probabilities: the transition probabilities depend on what problem parameters are modelled as random variables. We have developed two variants of the problem: one in which asset movements from period t to period $t+1$ are captured by means of a probability mass function over the nodes of the construction site graph; and another in which an asset fuel consumption in period t is expressed as a generic probability distribution over a discrete support. In both cases, transition probabilities $p_{s,s'}^k$ are immediately obtained from the given distributions.

Immediate value function: the immediate value function is given by the distance travelled by the bowser over the arc $(b_{\text{loc}}, \bar{b}_{\text{loc}})$ selected by action k, plus the expected penalty cost paid at the end of the period for each unit of fuel short.

Once all problem elements have been defined, the decision maker may tackle the problem numerically by applying established complete approaches such as forward or backward recursion algorithms, or approximate dynamic programming algorithms [54]. We implemented the model discussed using an open source general purpose library called `jsdp`.[3]. Our implementation of the DBRP is available in package `jsdp.app.routing`; this package comprises a deterministic formulation, as well as two stochastic variants of the problem in which *asset location* and *asset fuel consumption*, respectively, are modelled as random variables. Unfortunately, this approach can only tackle small instances; the interested reader may refer to [16] for applications of stochastic dynamic programming to inventory routing instances of realistic size.

In what follows we will present an effective Mixed-Integer Linear Programming heuristic for the case in which asset fuel consumption is random. We leave the investigation of effective heuristics for the random asset location case as future work.

4.2 A Mixed-Integer Linear Programming heuristic

We consider an SBRP in which asset a's fuel consumption in period t is a random variable f_t^a with known probability distribution over nonnegative support, e.g. Poisson; our aim is to model and solve heuristically this problem.

The SBRP under random fuel consumption is a complex multi-stage stochastic optimisation problem; as mentioned in the previous section, a stochastic dynamic programming approach can only solve small instances. To develop our heuristic, we proceed as follows: we approximate the original multi-stage problem as a two-stage stochastic optimisation problem with complete recourse [17]; and we then employ a "receding horizon" approach [10] to apply this approximation in the context of the original multi-stage setting.

Formulating the SBRP as a two-stage stochastic optimisation problem effectively means determining, at the beginning of the planning horizon, a refuelling plan comprising the optimal bowser route as well as asset replenishment quantities for all future period of the planning horizon — note that these decisions are fixed once and for all at the beginning of the planning horizon and they do not depend on random variable realisations. Once the plan is determined, we observe random fuel consumption for all assets and periods and determine the values of recourse variables, which represent the amount of fuel spare/short for each asset at each time period. The optimal plan is the one that minimises travel costs as well as the expected total penalty cost incurred as a consequences of fuel shortages.

To model this problem we introduce the following recourse decision variables:

- $[I_t^a]^-$, the expected fuel shortage for asset a at time t, where $[I_0^a]^- = \max(-s_a, 0)$;
- $[I_t^a]^+$, the expected fuel inventory for asset a at time t, where $[I_0^a]^+ = \max(s_a, 0)$;
- $[E_t^a]$, the expected fuel quantity exceeding tank capacity for asset a at time t.

The certainty equivalent MILP formulation of the problem is presented in Fig. 14. The objective function (18) now minimizes the sum of the total bowser routing cost and the expected total penalty cost incurred for fuel shortages.

Constraints (2), (3), (4), (5), (6), (7), (8), (9), (12) are those originally discussed for the DBRP and do not change. However, asset refuelling and inventory conservation

[3] http://gwr3n.github.io/jsdp/

$$\min \sum_{t=2}^{T} \sum_{i,j|\delta_{i,j}=1} T_{t-1}^{i,j} d_{i,j} + p \sum_{t=1}^{T} \sum_{a=1}^{A} [I_t^a]^- \tag{18}$$

Subject to

$$(2), (3), (4), (5), (6), (7), (8), (9), (12)$$

$$[I_t^a]^- = \mathcal{L}_{1,\ldots,t}^a(s_a + \sum_{k=1}^{t} Q_k^a + \sum_{k=1}^{t-1} [I_k^a]^- - \sum_{k=1}^{t} [E_k^a]) \quad t = 1, \ldots, T;\ a = 1, \ldots, A \tag{19}$$

$$[I_t^a]^+ = \widehat{\mathcal{L}}_{1,\ldots,t}^a(s_a + \sum_{k=1}^{t} Q_k^a + \sum_{k=1}^{t-1} [I_k^a]^- - \sum_{k=1}^{t} [E_k^a]) \quad t = 1, \ldots, T;\ a = 1, \ldots, A \tag{20}$$

$$[E_t^a] = \max([I_{t-1}^a]^+ + Q_t^a - c_a, 0) \quad t = 1, \ldots, T;\ a = 1, \ldots, A \tag{21}$$

$$\begin{array}{ll} T_t^{i,j},\ V_t^i \in \{0, 1\} & t = 1, \ldots, T;\ i, j|\delta_{i,j} = 1 \\ B_t \geq 0 & t = 1, \ldots, T \\ Q_t^a,\ [I_t^a]^+,\ [I_t^a]^-,\ [E_t^a] \geq 0 & t = 1, \ldots, T;\ a = 1, \ldots, A \\ 0 \leq S_t^a \leq f_t^a & t = 1, \ldots, T;\ a = 1, \ldots, A \end{array} \tag{22}$$

Figure 14: An MILP model for the Bowser Routing Problem under stochastic fuel consumption

constraints (10) and (11) must be adapted in order to take into account the fact that f_t^a is now a random variable.

Given a fuel quantity Q, to compute the expected fuel shortage $[I_t^a]^-$ for asset a at time t as well as the expected fuel inventory $[I_t^a]^+$ for asset a at time t, we leverage the first order loss function $\mathcal{L}_{1,\ldots,t}^a(Q)$ and the complementary first order loss function $\widehat{\mathcal{L}}_{1,\ldots,t}^a(Q)$, respectively. These are defined as follows

$$\widehat{\mathcal{L}}_{1,\ldots,t}^a(Q) = \sum_{k=0}^{Q} (Q - k) g_{1,\ldots,t}(k) \tag{23}$$

$$\mathcal{L}_{1,\ldots,t}^a(Q) = \sum_{k=Q}^{\infty} (k - Q) g_{1,\ldots,t}(k) \tag{24}$$

where $g_{1,\ldots,t}$ is the probability mass function of $f_1^a + \ldots + f_t^a$; these expressions are easily generalised to the case in which random variables are continuous [56].

Our system operates under a "lost sales" setting; by construction, $\mathcal{L}_{1,\ldots,t}^a(Q)$ represents the sum of expected shortages observed in periods $1, \ldots, t$ if the available fuel is Q.

Lemma 4. *Let $\mathcal{L}_t^a(Q)$ represent expected shortages observed in period t, then*

$$\mathcal{L}_t^a(Q) = \mathcal{L}_{1,\ldots,t}^a(Q) - \mathcal{L}_{1,\ldots,t-1}^a(Q);$$

similarly,

$$\widehat{\mathcal{L}}_t^a(Q) = \widehat{\mathcal{L}}_{1,\ldots,t}^a(Q) + \mathcal{L}_{1,\ldots,t-1}^a(Q).$$

Proof. Follows from linearity of expectation. □

Lemma 5. *We can approximate $\mathcal{L}^a_t(Q)$ and $\widehat{\mathcal{L}}^a_t(Q)$ as follows*

$$\mathcal{L}^a_t(Q) \approx \mathcal{L}^a_{1,\dots,t}(Q + \mathcal{L}^a_{1,\dots,t-1}(Q));$$

$$\widehat{\mathcal{L}}^a_t(Q) \approx \widehat{\mathcal{L}}^a_{1,\dots,t}(Q + \mathcal{L}^a_{1,\dots,t-1}(Q)).$$

Proof. In a "lost sales" setting, shortages are reset at the end of every period. The intuition behind our approximation is to increase the available fuel Q by an amount $\mathcal{L}^a_{1,\dots,t-1}(Q)$ equal to the sum of expected shortages observed in previous periods. □

Lemma 6. *The expected fuel quantity exceeding tank capacity for asset a at time t is*

$$[E^a_t] = \max([I^a_{t-1}]^+ + Q^a_t - c_a, 0).$$

Proof. Under a "lost sales" setting shortages are reset at the end of every period, hence the expected initial fuel level at period t is $[I^a_{t-1}]^+$; Q^a_t and c_a are constant. □

Constraints (19) and (20) are obtained by applying Lemma 5 and by reducing the available fuel by the sum of expected fuel quantities exceeding tank capacity for asset a at times $1, \dots, t$. Constraints (21) follow directly from Lemma 6.

Constraints (19) and (20) can be easily implemented via piecewise linearization techniques presented in [55, 56]. More specifically, let $\omega = f^a_1 + \dots + f^a_t$ and $g_\omega(\cdot)$ denote the probability density function of ω. Consider a partition of the support Ω of ω into R disjoint compact subregions $\Omega_1, \dots, \Omega_R$. The complementary first order loss function is piecewise linearised as

$$\widehat{\mathcal{L}}^a_{1,\dots,t}(Q) = \sum_{r=1}^{N} p_r \max(Q - \mathrm{E}[\omega|\Omega_r], 0), \tag{25}$$

where $p_r = \int_{\Omega_r} g_\omega(t)\mathrm{d}t$ and $\mathrm{E}[\omega|\Omega_r] = \frac{1}{p_r}\int_{\Omega_r} t g_\omega(t)\mathrm{d}t$ for all $r = 1, \dots, R$ [56, Lemma 10]. A linear expression for the first order loss function is derived by applying the linear transformation $\mathcal{L}^a_{1,\dots,t}(Q) = \widehat{\mathcal{L}}^a_{1,\dots,t}(Q) - (Q - \mathrm{E}[\omega])$ [56, Lemma 3].

The $\max(a, b)$ operator, which appears in constraints (21) and (25) can be implemented using the IBM ILOG OPL `maxl` command; alternatively, one may rely on the `piecewise` command in IBM ILOG OPL [39].

Note that our approximation is similar to the one adopted in [32] to model expected waste in perishable inventory control.

4.3 Numerical example

We consider the same numerical example presented in Section 3.3. However, fuel consumption in each period is now random. More specifically, fuel consumptions in different periods are independently distributed random variables that follow a Poisson distribution with mean values presented in Table 4. We solve this instance by using the MILP approach presented in the previous section; we employ a piecewise linearisation comprising 8 segments.

By comparing Fig. 12 and 16, it is easy to see that the optimal routing plan did not change with respect to the deterministic problem; its cost is therefore still 494. However, by contrasting Fig. 13 and Fig. 17 it is apparent that the optimal refuelling plan has changed. Moreover, since now fuel consumption is stochastic, we do expect to observe shortages. An overview on expected shortages is shown in Table 5; the expected total amount short is 1.68. Since the penalty cost is 100, the expected total cost predicted by the MILP model — comprising bowser routing cost and expected fuel shortage cost is $494+100\cdot(1.68)\approxeq 662$. The expected total cost obtained via Monte Carlo simulation (500 replications) is 655 — 95.0% confidence interval for mean (student): $(633, 676)$. Ideally, one may want to compare this cost against the expected total cost of the optimal plan obtained via stochastic dynamic programming. However, this instance is already too large to be solved to optimality. In Section 5.2 we will carry out a comprehensive numerical study on smaller instances to investigate the quality of our approximation.

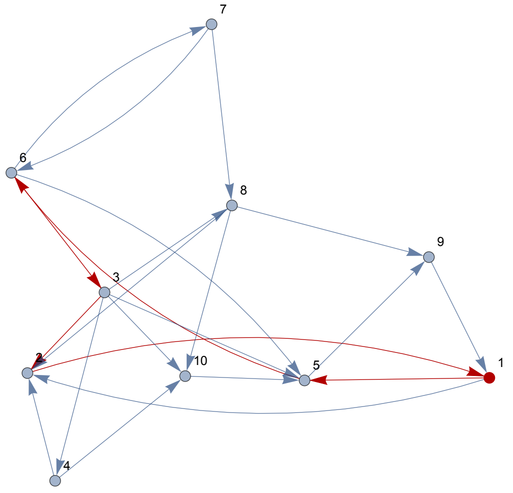


Figure 15: Optimal bowser routing plan.

t	Transition
1	1→ 1
2	1→ 1
3	1→ 5
4	5→ 6
5	6→ 3
6	3→ 2
7	2→ 1
8	1→ 1
9	1→ 1

Figure 16: Optimal bowser routing plan.

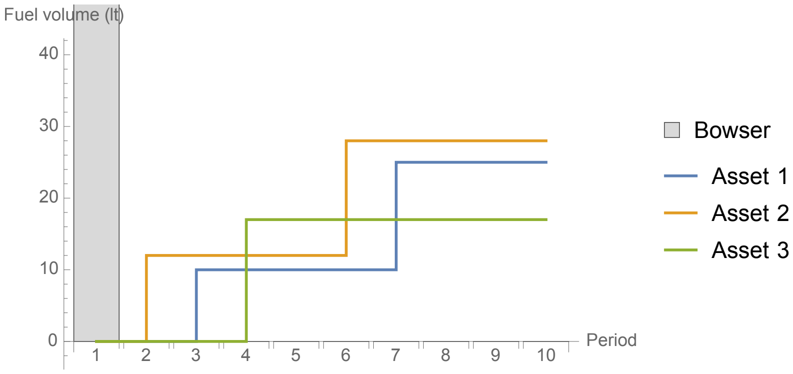


Figure 17: Bowser and asset refueling plan.

An IBM ILOG OPL implementation of this example is provided in our Electronic

Period	1	2	3	4	5	6	7	8	9	10
Asset 1	0.027	0.340	0.032	0.038	0.039	0.037	0.049	0.051	0.058	0.066
Asset 2	0.027	0.035	0.032	0.040	0.043	0.040	0.050	0.048	0.055	0.066
Asset 3	0.022	0.034	0.153	0.032	0.038	0.041	0.043	0.045	0.045	0.053

Table 5: Expected fuel shortages; the expected total amount short is 1.6831.

Addendum EA2.

5 Computational Experience

In this section we present our computational study for the deterministic (Section 5.1) and the stochastic (Section 5.2) settings. All experiments involving MILP models were performed on a MacBook Air 2.2 GHz Intel Core i7 8 GB of RAM. The optimisation environment used was IBM ILOG CPLEX Optimization Studio Version 12.6 with default settings and a time limit of 10 minutes (600 sec). Experiments involving stochastic dynamic programming were performed on an Intel(R) Xeon(R) @ 3.5GHz with 16Gb of RAM; the library used was `jsdp`.[4]

5.1 Dynamic Bowser Routing Problem

In this section we contrast the computational efficiency of the MILP model presented in Section 3 to tackle the DBRP. We consider two variants of the model: with and without valid inequalities presented in Section 3.2. In what follows, we first introduce our test bed and then present our results.

5.1.1 Test bed

Fuel bowsers come in different sizes and forms, from towable tanks whose capacity ranges from 500lt to 2000lt, to tanker trucks that may reach up to 15000lt. However, due to space constraints, a large tanker truck is unlikely to be deployed within a building site. Generally, these trucks are used to carry fuel to a site cistern, and then smaller towable fuel tanks or tankers are deployed on site. For this reason we consider three levels of bowser capacity: 500lt, 1000lt, and 2000lt. Asset tank capacity vary in size depending on the type of the asset. We will consider the following assets: the JCB 540-170 telehandler, which fits a 125lt tank; the JCB 531-70 telehandler, which fits a 146lt tank; the JCB JS130, a 13 tons excavator, which fits a 235lt tank; the JCB 86C-1 mini excavator, which fits a 112lt tank. Asset initial fuel levels are uniformly distributed between 0 and 20% of an asset tank capacity. Fuel consumption in each period of the planning horizon is randomly generated for each asset by following the distributions in Table 1, which we obtained from our analysis of JCB LiveLink$^{\text{TM}}$ asset consumption data.

We consider a test bed comprising a total of 108 instances generated as follows: the planning horizon covers $T = 50$ periods. The bowser initial tank capacity takes values $c_b \in \{500, 1000, 2000\}$. The topologies considered in our test bed are shown in Fig. 18;

[4] The code employed in our experiments is available on `http://gwr3n.github.io/jsdp/`.

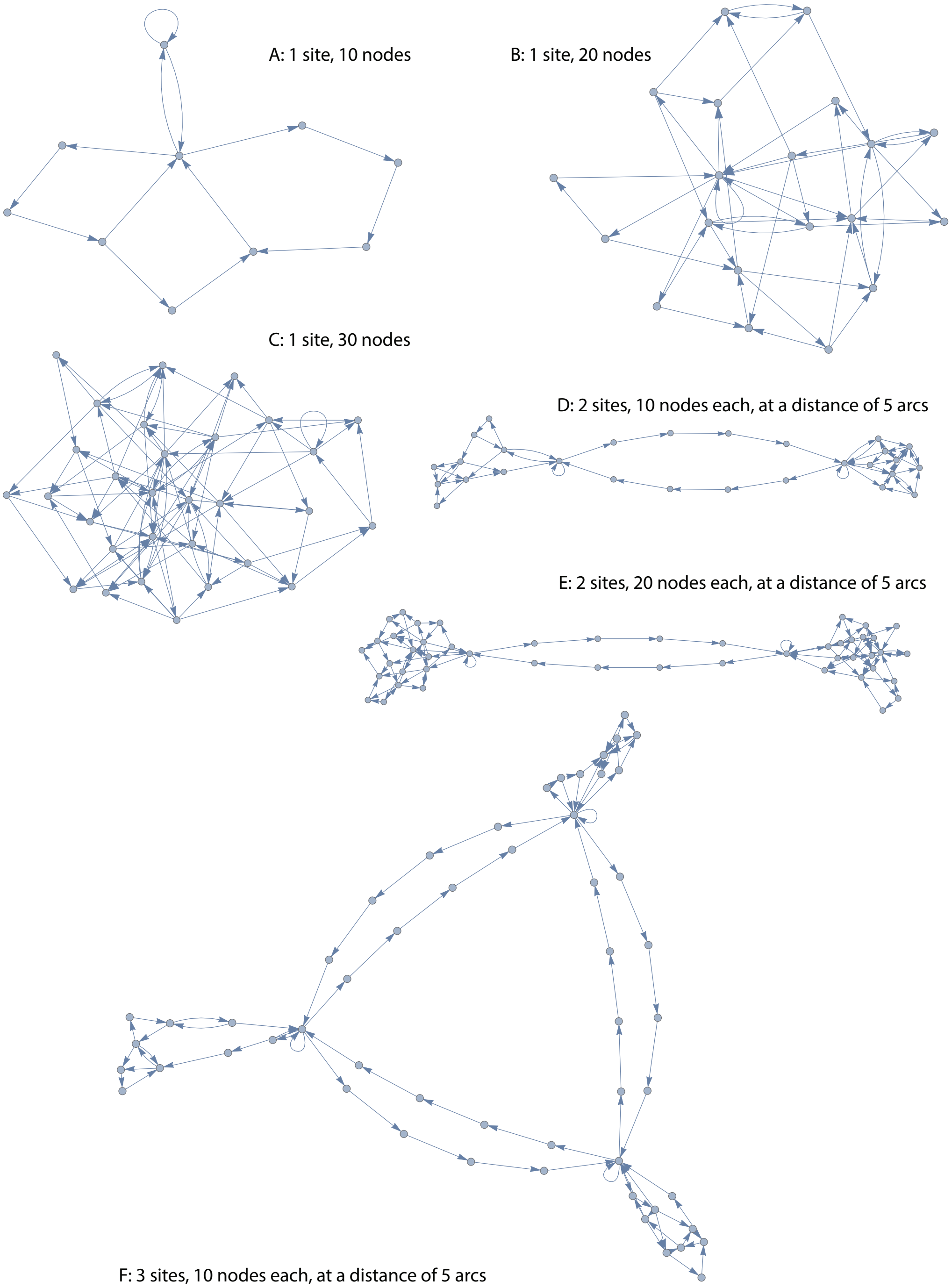

A: 1 site, 10 nodes
B: 1 site, 20 nodes
C: 1 site, 30 nodes
D: 2 sites, 10 nodes each, at a distance of 5 arcs
E: 2 sites, 20 nodes each, at a distance of 5 arcs
F: 3 sites, 10 nodes each, at a distance of 5 arcs

Figure 18: Site topologies considered in our DBRP numerical study.

these include networks including 1, 2, and 3 sites. Site networks have been generated as Bernoulli graphs constructed starting with a complete directed graph with $n \in \{10, 20, 30\}$ vertices and selecting each edge independently through a Bernoulli trial with probability $p = 0.1$; this process is repeated until a connected graph is obtained. Since, as discussed in Section 3, nodes represent adjacent accessible locations in building sites, these graphs are relatively sparse. Arc lengths are randomly generated from a normal distribution with mean 100 meters and standard deviation 20 meters. On each site we deploy a number of assets that ranges in the set $\{5, 10, 15\}$ — assets on each site are randomly picked from asset types listed above. Fuel shortage penalty cost take values $p \in \{100, 500\}$. The experimental design is full factorial.

To ensure replicability, a complete set of IBM ILOG data files for the test bed is provided in our Electronic Addendum EA3.

5.1.2 Results

In what follows, we shall refer to the MILP model presented in Section 3 as "MP;" the MILP model augmented with valid inequalities will be referred to as "MPVI."

Average key performance indicators recorded by CPLEX are presented in Table 6. There are a total of 108 instances in our test bed, MP solved 84 of them to optimality — i.e. 24 instances timed out at 600 sec), MPVI solved 90 of them to optimality — i.e. 18 instances timed out at 600 sec. The average solution time (including timeouts) is 217 sec for MP and 122 sec for MPVI; the average time difference between MP and MPVI is charted in Fig. 19. The average number of nodes explored is 4898 for MP and 319 for MPVI; the average node difference between MP and MPVI is charted in Fig. 20. The average number of simplex iterations[5] is almost 800k for MP and about 150k for MPVI; the average simplex iterations difference between MP and MPVI is charted in Fig. 21. For those instances that either MP or MPVI could not solve to optimality, the average optimality gap at timeout is 45.3% for MP and 33.4% for MPVI; the average optimality gap difference between MP and MPVI is charted in Fig. 22. This analysis demonstrates the effectiveness of valid inequalities introduced in Section 3; further insights on the performance on each individual set of inequalities are presented in Appendix II.

5.2 Stochastic Bowser Routing Problem

In this section we investigate the effectiveness and scalability of the MILP heuristic presented in Section 4.2 for the SBRP. We investigate effectiveness by comparing the simulated cost of the "true" optimal refueling plan obtained via stochastic dynamic programming against the simulated cost of the plan obtained via our MILP heuristic. Computing an optimal refueling plan is computationally demanding, therefore our experiments will be conducted on a test bed (Section 5.2.1) comprising smaller instances than those presented in Section 5.1.1. Conversely, scalability will be investigated on this latter test bed.

[5] This is the total number of simplex method pivoting operations carried out by CPLEX [39].

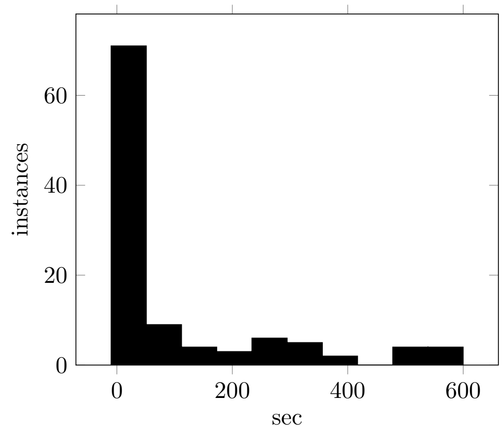


Figure 19: Time difference (sec) between MP and MPVI

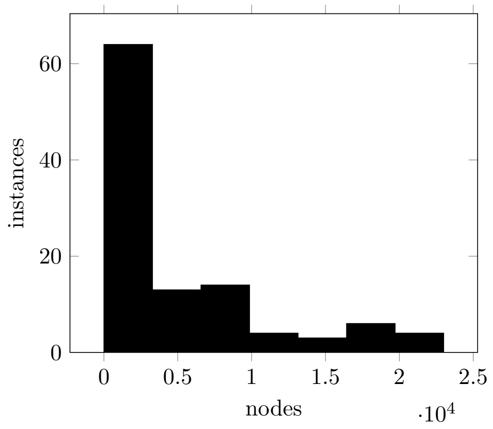


Figure 20: Difference in # explored nodes between MP and MPVI

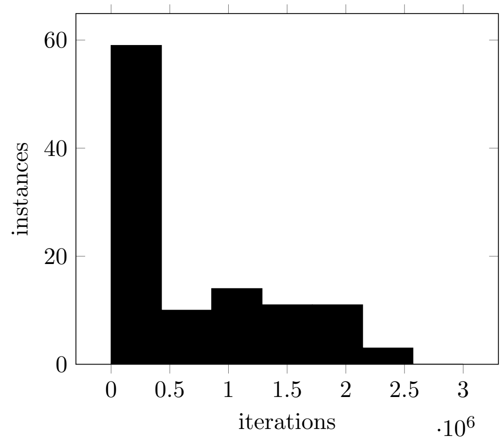


Figure 21: Difference in # simplex iterations between MP and MPVI

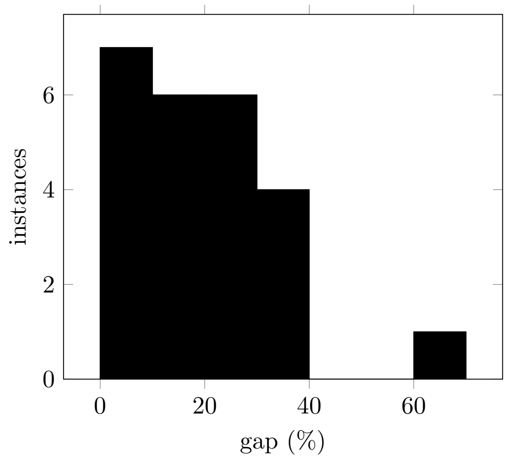


Figure 22: Difference between MP and MPVI optimality gap (%)

5.2.1 Test bed

We consider a problem over a planning horizon of $T = 5$ periods. The bowser capacity is set to 20 and the bowser initial tank level is 0. We consider 3 assets each of which features a tank with capacity 6; there are three initial tank level (ITL) configurations: $s^1 = \{0, 0, 0\}$, $s^2 = \{3, 0, 5\}$, and $s^3 = \{5, 5, 5\}$, where s_a^k is the initial tank level of asset a in the k-th configuration. Fuel consumption of asset k in period t follows a Poisson distribution with mean λ_t^k which has been truncated by fixing a maximum consumption of 7.[6] We consider three possible consumption patterns (CP): in the first pattern, $\lambda_t^k = 3$ for all assets and periods; in the second pattern, $\lambda_t^1 = 2$, $\lambda_t^2 = 1$, $\lambda_t^3 = 3$, for all $t = 1, \dots, T$; in the third pattern, $\lambda^1 = \{1, 2, 3, 4, 5\}$, $\lambda^2 = \{5, 4, 3, 2, 1\}$, $\lambda^3 = \{3, 3, 1, 1, 2\}$, where λ_t^k is the t-th entry of λ^k. We consider two possible values for the fuel stockout penalty cost $p = \{50, 100\}$. There are six possible site topologies as shown in Fig. 23 randomly generated in line with what we previously discussed; arc lengths are randomly generated from a normal distribution with mean 100 and standard deviation 20. We utilise 5 segments in our

[6] Probability masses have been normalised to ensure total probability over the truncated support is 1.

	time (sec)		timeouts		nodes		simplex iterations	
	MP	MPVI	MP	MPVI	MP	MPVI	MP	MPVI
Topology								
A	0.32	1.78	0	0	0	0	167	156
B	1.97	4.16	0	0	113	0	5697	627
C	232	22.0	4	0	2208	46	503413	23675
D	173	43.2	0	0	11484	484	1034887	76533
E	386	237	8	6	4247	547	1076786	270875
F	509	422	12	12	11334	837	2165064	547446
Assets								
5	157	26.2	2	0	5696	287	595398	51087
10	135	114	6	6	2759	268	575213	165201
15	358	224	16	12	6238	402	1222397	243367
Penalty (p)								
50	209	122	12	9	4782	309	767015	144513
100	224	121	12	9	5014	329	828323	161924
Overall								
	217	122	24	18	4897	319	797669	153218

Table 6: Pivot table comparing average key performance indicators for MP and MPVI

piecewise linearisation. Our test bed comprises a total of 108 instances; the experimental design is full factorial.

5.2.2 Results

We first investigate the **effectiveness** of the MILP heuristic presented in Section 4.2 for the SBRP. We do so by analysing two key performance indicators: the *linearisation gap* produced by our model, that is the difference between the expected total cost predicted by the MILP model and the expected total cost estimated by Monte Carlo simulation; and the *optimality gap* between the simulated cost of the refuelling plan suggested by our model and the cost of an optimal plan obtained via stochastic dynamic programming.[7]

In addition to investigating the cost of implementing a "here-and-now" plan (HN), which is a plan that fixes the bowser route and all asset replenishment quantities at the beginning of the planning horizon, we also investigate the performance of our heuristic in a "receding horizon" (RH) setting [10]. Under this setting, at any period $t = 1\dots,T$ we solve our MILP model over planning horizon $t,\dots,T$, but we only implement time t decisions — i.e. refuelling of assets and bowser movement that occur at time t. After implementing these decisions, we observe fuel consumption for period t and in period $t+1$ we solve again our MILP model over planning horizon $t+1,\dots,T$ by taking into account the impact of previously observed realisations. This process continues until we have reached the end of the planning horizon. Since RH is computationally cumbersome, we limit Monte Carlo simulation to 500 replications with common random numbers [18].

Table 7 report results over the test bed here investigated; the average *linearization gap* for our MILP model is 7.71%, the average *optimality gap* is 7.59% (HN) and 5.18%

[7] The stochastic dynamic programming code employed in our experiments is available on `http://gwr3n.github.io/jsdp/`.

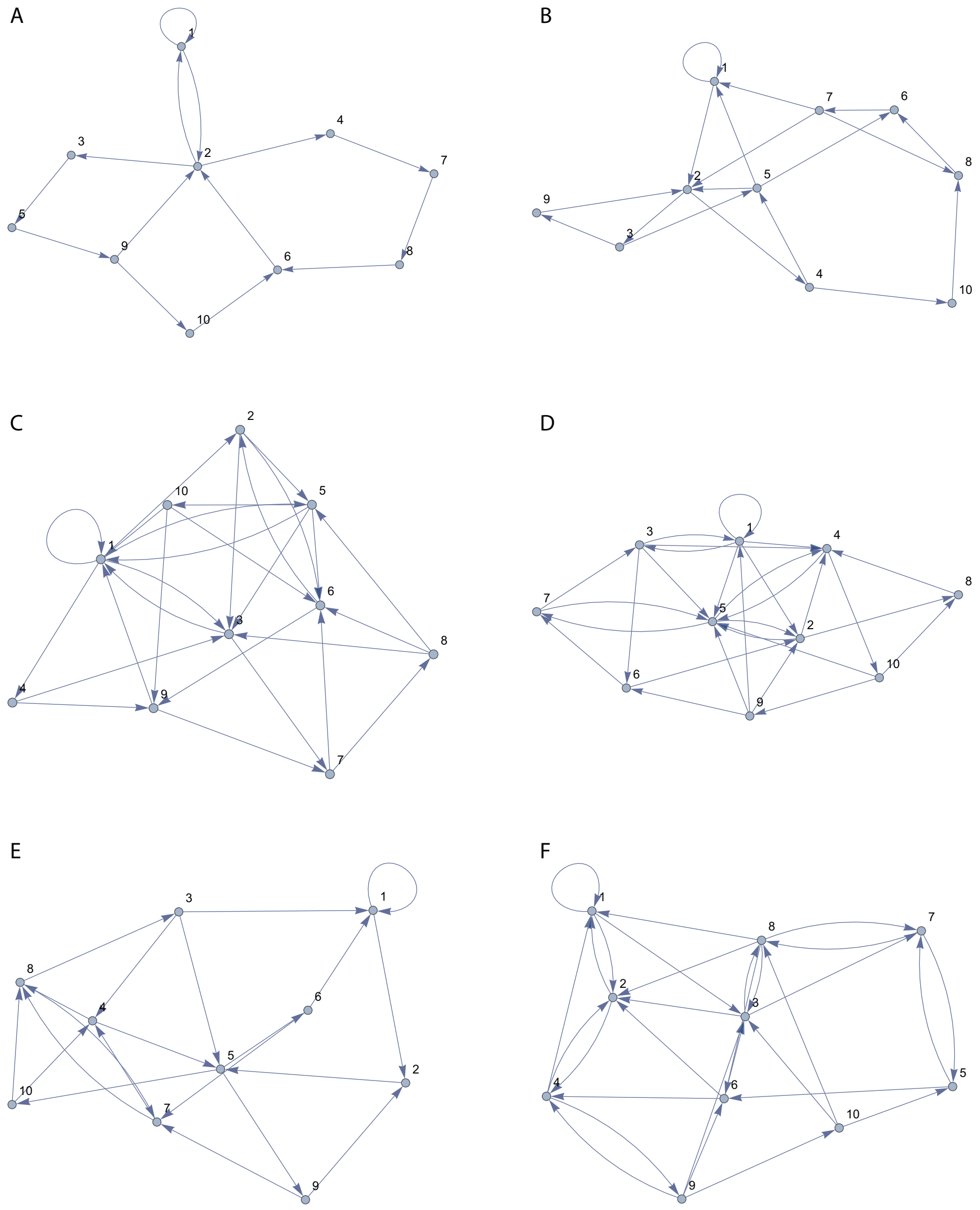


Figure 23: Site topologies considered in our SBRP numerical study.

(RH). Average solution time for the stochastic dynamic programming (SDP) approach is 2640 sec, while it is 0.13 sec for our MILP heuristic. These results show that our MILP approximation is effective and that its performance is enhanced under a RH strategy.

	Gap			Time (sec)	
	linearization	HN	RH	SDP	MILP
Topology					
A	5.14	5.80	5.54	950	0.16
B	9.20	13.1	5.51	2068	0.14
C	6.86	6.69	4.94	3850	0.13
D	7.00	5.04	4.49	2661	0.08
E	8.92	7.53	5.30	3742	0.17
F	9.16	7.33	5.29	2569	0.08
Initial tank level					
ITL1	6.55	4.53	4.02	1898	0.11
ITL2	7.33	7.47	5.39	2647	0.12
ITL3	9.25	10.78	6.13	3375	0.15
Consumption pattern					
CP1	5.87	6.01	4.04	2639	0.10
CP2	8.88	5.96	2.36	2653	0.17
CP3	8.39	10.82	9.14	2629	0.11
Penalty					
100	6.87	7.34	5.06	2645	0.12
500	8.56	7.85	5.30	2635	0.13
Overall					
	7.71	7.59	5.18	2640	0.13

Table 7: Optimality and linearization gap assessment for our SBRP MILP heuristic

We next investigate **scalability**. The MILP heuristic presented in Section 4.2 for the SBRP is not as scalable as the MILP model presented in Section 3 for the DBRP. We have run experiments on a test bed generated in line with the discussion in Section 5.1.1; however, we have now considered a shorter planning horizon comprising $T = 10$ periods — given a time bucket size of 20-30 minutes, this roughly corresponds to a three- to five-hour plan. As previously discussed, on each site we deploy a number of assets that ranges in the set $\{5, 10, 15\}$; these are randomly picked from asset types previously listed. For each asset type, fuel consumption in each period of the planning horizon follows the distribution in Table 1. Once more, to ensure replicability; a complete set of IBM ILOG data files for the test bed is provided in our Electronic Addendum EA4. Results of this study are presented in Table 8. All instances could be solved within the given time limit of 600 seconds. Average solution time was 8.21 second. This study was limited to default CPLEX settings; hence improvements may be achieved by using more efficient hardware and solver setups. Future research may investigate dedicated branch-and-cut strategies similar to those discussed in [60], which have been shown to boost scalability of MILP model based on piecewise linearisation strategies like the one here adopted.

	time (sec)	nodes	simplex iterations
Topology			
A	2.20	4286	9530
B	2.83	5465	12131
C	1.44	1416	5932
D	8.53	49396	80924
E	1.93	2447	8698
F	32.3	76087	132668
Assets			
5	2.40	3755	9938
10	16.8	39472	69368
15	5.41	26321	45636
Penalty (p)			
50	14.9	44482	76571
100	1.52	1884	6724
Overall			
	8.21	23183	41647

Table 8: Pivot table with average key performance indicators for our SBRP MILP heuristic

6 Conclusion

In this work we focussed on efficient refuelling of assets across construction sites. We introduced the Bowser Routing Problem and discussed its deterministic (DBRP) and stochastic (SBRP) variants. For each of these variants, we developed mathematical programming models that, by leveraging data supplied by different assets, schedule refuelling operations by balancing the cost of dispatching a bowser truck with that of incurring fuel shortages. To enhance computational performances, we discussed valid inequalities. We carried out a comprehensive set of experiments on a testbed designed around data derived from our experience at Crossrail sites. In a deterministic setting, this study shows that our mathematical programming models can tackle instances of realistic size in reasonable time — generally ranging from seconds to minutes; valid inequalities on average halve computational time and dramatically reduce the number of explored nodes and simplex iterations required to reach an optimal solution. In the stochastic case, our mathematical programming heuristic is effective and produces tight linearisation (approx. 7%) as well as optimality (approx. 5%) gaps. Computationally, when tested on our original test bed comprising larger instances, our heuristic can solve all instances in a few seconds on average and in the worst case within ten minutes. An interesting direction for future research is to investigate more effective reformulations for this latter MILP heuristic.

Acknowledgements

This work has been developed in the context of project ReCCEL, which was supported by Innovate UK grant number 55202 - 418137.

References

[1] Daniel Adelman. A Price-Directed approach to stochastic Inventory/Routing. *Operations Research*, 52(4):499–514, July 2004.

[2] Yossiri Adulyasak, Jean-François Cordeau, and Raf Jans. Formulations and Branch-and-Cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing*, 26(1):103–120, February 2014.

[3] AEMP. AEMP Telematics Data Standard V1.2. Standard, Association of Equipment Management Professionals, Glenwood Springs, CO 81602, 2010.

[4] Henrik Andersson, Arild Hoff, Marielle Christiansen, Geir Hasle, and Arne Løkketangen. Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research*, 37(9):1515–1536, 2010.

[5] Claudia Archetti, Luca Bertazzi, Gilbert Laporte, and Maria G. Speranza. A Branch-and-Cut algorithm for a Vendor-Managed Inventory-Routing problem. *Transportation Science*, 41(3):382–391, August 2007.

[6] Claudia Archetti, Luca Bertazzi, Alain Hertz, and M. Grazia Speranza. A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, 24(1): 101–116, February 2012.

[7] Okan Arslan and Oya E. Karaşan. A benders decomposition approach for the charging station location problem with plug-in hybrid electric vehicles. *Transportation Research Part B: Methodological*, 93:670–695, November 2016.

[8] J. W. Barnes, V. D. Wiley, J. T. Moore, and D. M. Ryer. Solving the aerial fleet refueling problem using group theoretic tabu search. *Mathematical and Computer Modelling*, 39(6-8):617–640, March 2004.

[9] Walter J. Bell, Louis M. Dalberto, Marshall L. Fisher, Arnold J. Greenfield, R. Jaikumar, Pradeep Kedia, Robert G. Mack, and Paul J. Prutzman. Improving the distribution of industrial gases with an On-Line computerized routing and scheduling optimizer. *Interfaces*, 13(6):4–23, December 1983.

[10] J. Bellingham, A. Richards, and J. P. How. Receding horizon control of autonomous aerial vehicles. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, pages 3741–3746 vol.5. IEEE, 2002.

[11] Richard Bellman. *Dynamic Programming.* Princeton University Press, Princeton, NJ, USA, 1 edition, 1957.

[12] Gaetan Belvaux and Laurence A. Wolsey. bc — prod: A specialized Branch-and-Cut system for Lot-Sizing problems. *Management Science*, 46(5):724–738, May 2000.

[13] Mike Benchimol, Pascal Benchimol, Benoît Chappert, Arnaud de la Taille, Fabien Laroche, Frédéric Meunier, and Ludovic Robinet. Balancing the stations of a self service “bike hire” system. *RAIRO - Operations Research*, 45(1):37–61, May 2011.

[14] Luca Bertazzi and Maria Grazia Speranza. Inventory routing problems: an introduction. *EURO Journal on Transportation and Logistics*, 1(4):307–326, 2012.

[15] Luca Bertazzi, Martin Savelsbergh, and Maria Grazia Speranza. Inventory routing. In Bruce Golden, S. Raghavan, and Edward Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *Operations Research/-Computer Science Interfaces*, pages 49–72. Springer US, 2008.

[16] Luca Bertazzi, Adamo Bosco, and Demetrio Laganà. Managing stochastic demand in an inventory routing problem with transportation procurement. *Omega*, 56:112–121, October 2015.

[17] John R. Birge and François Louveaux. *Introduction to Stochastic Programming.* Springer New York, New York, NY, 2011.

[18] Zdravko Botev and Ad Ridder. Variance reduction. In N. Balakrishnan, Theodore Colton, Brian Everitt, Walter Piegorsch, Fabrizio Ruggeri, and Jozef L. Teugels, editors, *Wiley StatsRef: Statistics Reference Online*, pages 1–6. John Wiley & Sons, Ltd, Chichester, UK, April 2014.

[19] Leandro C. Coelho and Gilbert Laporte. The exact solution of several classes of inventory-routing problems. *Computers & Operations Research*, 40(2):558–565, February 2013.

[20] Leandro C. Coelho, Jean-François Cordeau, and Gilbert Laporte. The inventory-routing problem with transshipment. *Computers & Operations Research*, 39(11): 2537–2548, November 2012.

[21] Leandro C. Coelho, Jean-François Cordeau, and Gilbert Laporte. Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies*, 24:270–287, October 2012.

[22] Jean-François Cordeau, Gilbert Laporte, Martin W. P. Savelsbergh, and Daniele Vigo. *Chapter 6 Vehicle Routing*, volume 14, pages 367–428. Elsevier, 2007.

[23] Gérard Cornuéjols. Valid inequalities for mixed integer linear programs. *Mathematical Programming*, 112(1):3–44, 2008.

[24] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, October 1959.

[25] Luca Di Gaspero, Andrea Rendl, and Tommaso Urli. Balancing bike sharing systems with constraint programming. *Constraints*, 21(2):318–348, 2016.

[26] A. Drexl and A. Kimms. Lot sizing and scheduling — survey and extensions. *European Journal of Operational Research*, 99(2):221–235, June 1997.

[27] Awi Federgruen and Paul Zipkin. A combined vehicle routing and inventory allocation problem. *Operations Research*, 32(5):1019–1037, 1984.

[28] LeRoy E. Foster. *Telemetry Systems.* John Wiley & Sons, Inc., first edition, 1965.

[29] Michel Gendreau, Gilbert Laporte, and René Séguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88(1):3–12, January 1996.

[30] Luc van Hamme Gilbert Laporte, Franois V. Louveaux. An integer l-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, 50(3):415–423, 2002.

[31] Bruce Golden, S. Raghavan, and Edward Wasil, editors. *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *Operations Research/Computer Science Interfaces.* Springer US, Boston, MA, 2008.

[32] Alejandro Gutierrez-Alcoba, Roberto Rossi, Belen Martin-Barragan, and Eligius M. T. Hendrix. A simple heuristic for perishable item inventory control under non-stationary stochastic demand. *International Journal of Production Research*, 55(7): 1885–1897, April 2017.

[33] Ford W. Harris. How many parts to make at once. *Operations Research*, 38(6):947–950, December 1990. Reprinted from Factory, The Magazine of Management, Volume 10, Number 2, February 1913, pp. 135–136, 152.

[34] HMG. Construction 2025. Report URN/BIS/13/955, Her Majesty's Government, 10 Downing Street, London, July 2013.

[35] HMT. Investing in Britain's Future. Report PU1524, Her Majesty's Treasury, 1 Horse Guards Road, Westminster, London, June 2013.

[36] HMT. Infrastructure Carbon Review. Report PU1593, Her Majesty's Treasury, 1 Horse Guards Road, Westminster, London, November 2013.

[37] Lars M. Hvattum and Arne Løkketangen. Using scenario trees and progressive hedging for stochastic inventory routing problems. *Journal of Heuristics*, 15(6):527–557, December 2008.

[38] Lars M. Hvattum, Arne Løkketangen, and Gilbert Laporte. Scenario Tree-Based heuristics for stochastic Inventory-Routing problems. *INFORMS Journal on Computing*, 21(2):268–285, May 2009.

[39] IBM. IBM ILOG CPLEX Optimization Studio OPL Language User's Manual. Technical report, International Business Machines, 2014.

[40] Patrick Jaillet. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Operations Research*, 36(6):929–936, 1988.

[41] Sezgin Kaplan and Ghaith Rabadi. Exact and heuristic algorithms for the aerial refueling parallel machine scheduling problem with due date-to-deadline window and ready times. *Computers & Industrial Engineering*, 62(1):276–285, February 2012.

[42] Anton J. Kleywegt, Vijay S. Nori, and Martin W. P. Savelsbergh. The stochastic inventory routing problem with direct deliveries. *Transportation Science*, 36(1):94–118, February 2002.

[43] Anton J. Kleywegt, Vijay S. Nori, and Martin W. P. Savelsbergh. Dynamic programming approximations for a stochastic inventory routing problem. *Transportation Science*, 38(1):42–70, February 2004.

[44] Michael Kuby and Seow Lim. The flow-refueling location problem for alternative-fuel vehicles. *Socio-Economic Planning Sciences*, 39(2):125–145, June 2005.

[45] Philippe Lacomme, Christian Prins, and Wahiba Ramdane-Cherif. Competitive memetic algorithms for arc routing problems. *Annals of Operations Research*, 131 (1-4):159–185, 2004.

[46] Donald Michie. “memo” functions and machine learning. *Nature*, 218(5136):19–22, April 1968.

[47] S. A. MirHassani and R. Ebrazi. A flexible reformulation of the refueling station location problem. *Transportation Science*, 47(4):617–628, November 2013.

[48] NASA. Telemetry: Summary of concept and rationale. *NASA STI/Recon Technical Report N*, 89, December 1987.

[49] S. Nora and A. Minc. *L'informatisation de la société: rapport à M. le Président de la République*. Points politique. Seuil, 1978.

[50] Andrew W. Palmer, Andrew J. Hill, and Steven J. Scheding. Stochastic collection and replenishment (SCAR): Objective functions. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3324–3331. IEEE, November 2013.

[51] Andrew W. Palmer, Andrew J. Hill, and Steven J. Scheding. Stochastic collection and replenishment (SCAR) optimisation for persistent autonomy. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2943–2949. IEEE, September 2014.

[52] Andrew W. Palmer, Andrew J. Hill, and Steven J. Scheding. Methods for stochastic collection and replenishment (SCAR) optimisation for persistent autonomy. *Robotics and Autonomous Systems*, 87:51–65, January 2017.

[53] Yves Pochet and Laurence A. Wolsey. *Production Planning by Mixed Integer Programming*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006.

[54] Warren B. Powell. What you should know about approximate dynamic programming. *Naval Research Logistics (NRL)*, 56(3):239–249, 2009.

[55] Roberto Rossi and Eligius M. T. Hendrix. Computing linearisation parameters of arbitrarily distributed first order loss functions. In *Proceedings of MAGO'14, XII Global Optimization Workshop (GOW)*. University of Malaga, 2012.

[56] Roberto Rossi, S. Armagan Tarim, Steven Prestwich, and Brahim Hnich. Piecewise linear lower and upper bounds for the standard normal first order loss function. *Applied Mathematics and Computation*, 231:489–502, March 2014.

[57] Rahul Saxena and Anand Srinivasan. *Business Analytics*, volume 186. Springer New York, New York, NY, 2013.

[58] Oğuz Solyalı, Jean-François Cordeau, and Gilbert Laporte. Robust inventory routing under demand uncertainty. *Transportation Science*, 46(3):327–340, August 2012.

[59] Horst Tempelmeier. Stochastic lot sizing problems. In J. MacGregor Smith and Barış Tan, editors, *Handbook of Stochastic Models and Analysis of Manufacturing System Operations*, volume 192 of *International Series in Operations Research & Management Science*, pages 313–344. Springer New York, 2013.

[60] Huseyin Tunc, Onur A. Kilic, S. Armagan Tarim, and Roberto Rossi. An extended Mixed-Integer programming formulation and dynamic cut generation approach for the stochastic Lot-Sizing problem. *INFORMS Journal on Computing*, 30(3):492–506, August 2018.

[61] Harvey M. Wagner and Thomson M. Whitin. Dynamic version of the economic lot size model. *Management Science*, 5(1):89–96, October 1958.

[62] Yugang Yu, Chengbin Chu, Haoxun Chen, and Feng Chu. Large scale stochastic inventory routing problems with split delivery and service level constraints. *Annals of Operations Research*, 197(1):135–158, July 2012.

Appendix I

Symbols used in the manuscript are listed in Table 9.

Parameters	
A	number of assets;
$\langle V, E \rangle$	directed graph with nodes V and edges E;
T	number of time periods;
N	number of nodes in the site network (i.e. $N = \|V\|$);
$d_{i,j}$	distance between node i and node j in the site network, if $i = j$, $d_{ij} = 0$;
$\delta_{i,j}$	a binary parameter that is set to one iff it is possible to travel from node i to node j in one time period;
$l^a_{t,i}$	a binary parameter that is set to one iff asset a is at node i during time period $t \in T$; i.e iff $l^a_t = i$;
c_a	tank capacity of asset a;
s_a	initial tank level of asset a;
f^a_t	fuel consumption of asset a in time period $t \in T$
c_b	bowser tank capacity;
s_b	initial bowser tank level;
p	penalty cost per litre of fuel short;

Decision variables	
V^i_t	a binary variable that is set to one iff, at time t, the bowser is at node i;
$T^{i,j}_t$	an auxiliary binary variable that is set to one iff the bowser transits from node i to node j by the end of period t;
Q^a_t	the quantity of fuel delivered to asset a at time t;
S^a_t	the fuel shortage for asset a at time t;
B_t	the quantity of fuel transferred from the site cistern to the bowser at time t;
$[I^a_t]^-$	the expected fuel shortage for asset a at time t, where $[I^a_0]^- = \max(-s_a, 0)$;
$[I^a_t]^+$	the expected fuel inventory for asset a at time t, where $[I^a_0]^+ = \max(s_a, 0)$;
$[E^a_t]$	the expected fuel quantity exceeding tank capacity for asset a at time t.

Functions	
$\widehat{\mathcal{L}}^a_{1,\ldots,t}(\cdot)$	complementary first order loss function for consumption over period $1, \ldots, t$;
$\mathcal{L}^a_{1,\ldots,t}(\cdot)$	first order loss function for consumption over period $1, \ldots, t$.

Table 9: Summary table of symbols used in this work.

Appendix II

In this appendix we present a more detailed overview on the effectiveness of the valid inequalities discussed in Section 3.2. From Tables 10 and 11 it is clear that the model that combines all valid inequalities (MPVI) dominates, in terms of explored nodes and simplex iterations performed, models embedding individual valid inequalities (Eq. 14, Eq. 15, Eq. 16) separately, as well as the model without valid inequalities (MP). This means that all valid inequalities are effective.

			Nodes		
	MP	Eq. 14	Eq. 15	Eq. 16	MPVI
Topology					
A	0	0	0	0	0
B	113	3	53	2	0
C	2208	43	845	45	46
D	11484	404	5056	514	484
E	4247	852	2630	486	547
F	11334	4751	6651	1032	837
Assets					
5	5696	278	2565	307	287
10	2759	1450	1690	301	268
15	6238	1299	3363	432	402
Penalty (p)					
50	4782	985	2406	341	309
100	5014	1033	2673	352	329
Overall					
	4898	1009	2539	347	319

Table 10: Pivot table comparing average nodes explored for individual valid inequalities

			Simplex Iterations		
	MP	Eq. 14	Eq. 15	Eq. 16	MPVI
Topology					
A	167	156	182	171	156
B	5697	435	3089	809	627
C	503413	27304	238844	28782	23675
D	1034887	48067	587795	79790	76533
E	1076786	381125	728284	257024	270875
F	2165064	1584704	1521861	563083	547446
Assets					
5	595398	34483	347684	52806	51087
10	575213	416857	397354	159315	165201
15	1222397	569556	794990	252708	243367
Penalty (p)					
50	767015	305086	492632	146168	144513
100	828323	375511	534054	163718	161924
Overall					
	797669	340299	513343	154943	153218

Table 11: Pivot table comparing average simplex iterations for individual valid inequalities