PHILOSOPHICAL TRANSACTIONS A

Procuring load curtailment from local customers under uncertainty

SCHOLARONE™
Manuscripts

Author for correspondence:

Jure Vogrinc

e-mail: j.vogrinc13@imperial.ac.uk

Procuring load curtailment from local customers under uncertainty

Aleksandar Mijatović[1,2], John Moriarty[3] and Jure Vogrinc[4]

[1]Department of Mathematics, King's College London
[2]The Alan Turing Institute, London
[3]School of Mathematical Sciences, Queen Mary University of London
[4]Department of Mathematics, Imperial College London

Demand Side Response (DSR) provides a flexible approach to managing constrained power network assets. This is valuable if future asset utilisation is uncertain. However there may be uncertainty over the process of procurement of DSR from customers. In this context we combine probabilistic modelling, simulation and optimisation to identify economically optimal procurement policies from heterogeneous customers local to the asset, under chance constraints on the adequacy of the procured DSR. Mathematically this gives rise to a search over permutations, and we provide an illustrative example implementation and case study.

1. Frequently used notation

A_i	Availability payment to i-th customer
$\mathcal{C}_0$	Capacity of the constrained asset
C_i	DSR capacity added by i-th customer
D	Utilisation at time 1 (without DSR)
E_i	Exercise payment to i-th customer at time 1
H	Minimum threshold for DSR procurement
M	Number of utilisation scenarios
N	Number of qualifying DSR customers
p_i	Probability that i-th customer passes the acceptance test
β	Acceptance test cost
γ	Value of lost load
μ_i	Mean of utilisation in i-th scenario
ρ	Permutation (ordering) of N objects
σ_i	Standard deviation of utilisation in i-th scenario

2. Introduction

(a) Demand side response

Demand side response (DSR) may be defined as "actions by consumers to change the amount of electricity they take off the grid at particular times in response to a signal" [20]. As described in [25], DSR offers potential benefits across the entire electricity system, including generation, transmission, distribution and consumption. At the level of generation, DSR may allow a lower margin of installed generation capacity to be maintained relative to the system maximum demand. DSR may also potentially be deployed to increase the level of utilisation in transmission networks, to manage distribution network constraints of various kinds, or to help maintain the balance between generation and load in systems with significant renewable penetration.

According to the UK energy market regulator Ofgem [19] three main methods are used to facilitate DSR:

(i) Tariffs, including time of use pricing; critical peak pricing, where the peak periods and associated prices are communicated to customers a short time before they begin; and real-time pricing,

(ii) Automated devices such as 'smart' appliances which respond either to changing network conditions or to price signals. A related approach is *demand bidding* whereby customers offer demand response via an online platform [21], for example by scheduling their appliances intelligently,

(iii) Contracts with industrial and commercial customers to curtail load. This approach includes load curtailment at pre-agreed times; interruptible contracts, where the utility may shed customer load a limited number of times; or the direct control of loads, whereby utility companies have free access to customer processes.

Method (iii) is our focus in this paper.

The first DSR programmes emerged following the energy crises of the 1970s [27]. Nevertheless, neither the operational nor the planning aspects of DSR can yet be considered mature subjects. A particular challenge has been to value the benefits of flexible solutions such as DSR under uncertainty, and a recent contribution in this direction is [24]. In the latter paper real options analysis is used to value DSR alongside capital network investment on a consistent basis for planning purposes, so that an informed choice may be made between the two. In particular the authors highlight the potentially high value of DSR under uncertain growth in peak load, as it offers to defer capital investment for a number of years and thus potentially avoid the stranding of assets in scenarios of subsequent low peak demand growth.

We focus on the application of DSR by a distribution network operator (DNO) through load curtailment, as described in (iii) above. The goal is to manage the maximum utilisation of a particular cable or transformer, in order to defer or avoid a capital-intensive reinforcement of this asset. The strategic decision to use DSR rather than reinforcement is assumed to have already been made. For clarity we use the following terminology:

- A *qualifying customer* is an industrial or commercial customer who:
  - is served by the constrained asset, and
  - has sufficiently high demand, both to make a significant DSR contribution to and justify the fixed automation costs involved in joining the DSR programme.

- A *contracted customer* is a qualifying customer who passes an *acceptance test*, and is thus assumed to enter the DSR programme. This assessment represents the outcome of a test procedure, whereby both the utility and customer verify the suitability of the customer's load for curtailment within the parameters of the DSR programme.

Page 3 of 19

Submitted to Phil. Trans. R. Soc. A - Issue

3

We therefore do not assume that each qualifying customer will participate in the DSR scheme if invited. Instead an acceptance test is carried out to determine whether both the utility and the customer are willing to proceed (a real-world example of acceptance testing, although in a different DSR scheme, is described in [22]). We assume that each acceptance test carries a fixed cost β for the utility, and so the assessments occur only after the customer is invited to join the programme. Example reasons for failure of the acceptance test include unacceptable disruption caused to the customer's business by load curtailment; technical unsuitability of the customer's devices or control equipment; and loads which may not be available for DSR when needed. Since it is not known *a priori* which qualifying customers will pass, we model the outcome of the acceptance test probabilistically.

This study is timely since DSR programmes are now entering the 'business as usual' practice of both transmission system and distribution network operators. National Grid, the UK transmission system operator, has several current DSR programmes for large customers. These include frequency response, reserve services and the triad system for managing peak load [19]. A triad is defined to be the three half-hour periods in each year at which demand in the transmission system is highest. Demand is financially penalised during the these periods, incentivising customers to reduce consumption. Since the precise timing of demand peaks, and hence the triad, can only be identified at the end of the year, suppliers offer triad prediction services. With an opposite aim National Grid has also recently introduced Demand Turn Up, a trial 'footroom' DSR programme which incentivises large customers to increase demand (or reduce generation) at times when there is an excess of generation, typically overnight and on weekend afternoons. An example using direct load control is the 'managed connections' programme from UK DNO Electricity North West Ltd [14]. This programme offers potentially cheaper and quicker connections for new distributed generation by installing direct control of the generation export, which may be activated during network faults. In this way traditional network reinforcement can be deferred, thus potentially avoiding the associated delay and cost. The company has also recently developed a tool to compare specific investment projects at the grid and primary level in the framework of real options analysis, based on [24]. The latter tool has recently been used to identify the managed connections programme as a preferable alternative to a more capital intensive, specific network reinforcement [15].

In contrast to the larger DSR schemes of transmission system operators, or the online platforms used for demand bidding, in this paper we address local schemes run by DNOs with no more than tens of qualifying customers. It is assumed that the DNO has had initial contact with each qualifying customer and thus knows both their level of directly controllable load and the compensation they require to join the programme. Clearly, insufficient DSR procurement by the DNO would carry the risk of lost load, while excessive or inefficient procurement would result in unnecessarily high costs to be borne by the utility and ultimately by its customers. Uncertainty in future asset utilisation is also a material consideration, since in our model procurement takes place one year in advance. This is in order to take account of the 'lead time' required both for procurement and to commission the necessary communications and automation equipment. Our aim is therefore to address the problem of optimally procuring DSR from qualifying customers, while also taking into account uncertainty in both its procurement and its provision.

To date the literature on DSR operation has been dominated by the study of price responsive demand, with recent examples including its impact on distribution networks [16] and on the scheduling of generation [28]. In contrast there has been relatively little academic study of load curtailment, and this has been focused on aggregating response from many customers. An approach using distributed control is studied in [7], while an algorithm inspired by packet switching in digital communications networks has been explored in [2]. The use of thousands of small loads such as water heaters or air conditioners is studied in [13], and [23] presents a randomised algorithmic approach potentially suitable for controlling millions of appliances. In contrast the setting for the present paper is the direct control of small numbers of relatively large loads, in order to relieve a specific constrained network asset. Since in this context the operational

control problem should be straightforward, we focus instead on the question of optimal customer selection given knowledge of their individual controllable loads and their bids for participation in the DSR programme.

(b) Problem setting and scope

The problem setting, which is identical to that of [24], is as follows. A specific distribution network asset has become constrained by its maximum utilisation and one possible solution is a relatively costly reinforcement of the asset. The DNO considers a feasible and lower-cost alternative to be the direct control of load from selected customers served by the asset. A receiver and control system is installed at each participating customer site, enabling communication from the DNO and automated load control. In return for a reduced electricity bill the DNO is then able to either cycle or shut off an appliance for a certain amount of time, a certain number of times per year, in order to relieve the network asset when its level of utilisation is highest. Since we consider only the utilisation of a specific cable or transformer in the distribution network, we make the approximation that network effects are not modelled.

Our probabilistic modelling of the procurement process, which was outlined above, will have significant consequences for the underlying optimisation problem which is formulated mathematically in Section 2(c) below and solved in Section 4. In the absence of uncertainty over acceptance testing, the problem reduces to one of integer programming, in which each qualifying customer is simply selected or not for the programme. However if each customer's ability to participate is uncertain then any given solution to this integer programming problem may not be implementable in practice, since one or more of the desired customers may not pass the acceptance test. The solution we propose is to create

- an ordered 'wish list', denoted by ρ, which may be interpreted as a ranking of the qualifying customers in decreasing order of their desirability to the DNO, together with
- a *threshold*, denoted by H, which indicates when the desired total amount of DSR has been contracted and thus the procurement process would stop. That is, once this threshold has been met, the process of inviting customers and acceptance testing stops.

Our chosen objective is therefore to identify an optimal order ρ in which qualifying customers are invited to participate. This will necessitate a particular mathematical formulation which we provide in the next section. Finally we note that our modelling of future utilisation at time 1, which uses probability weighted scenarios plus added noise, is the same as that used in [24].

(c) Overview of the modelling and optimisation paradigms

Our study aims to perform an economic optimisation over a particularly large and complex parameter space, with a two-stage model of uncertainty and a chance constraint on the loss of load. In this section we discuss the modelling and optimisation issues that arise in this context and give an overview of our approach for tackling them (for the detailed descriptions see Sections 3 and 4 below).

Let N be the number of qualifying customers and let ρ represent an arbitrary (but fixed) order in which they are to be approached to join the DSR programme. Mathematically ρ is therefore a permutation of the set $\{1, \ldots, N\}$, where the first element of the permutation, written $\rho(1)$, denotes the first customer to be approached, and so on for $\rho(2)$, etc. We write $\mathcal{S}_N$ for the *permutation group*, which consists of all possible such permutations. In contrast the problem of choosing an optimal subset (which, as discussed, is more appropriate in the case of a deterministic procurement process) would correspond to choosing an element of the *power set*. The large size of the considered parameter space can be appreciated by taking $N = 20$ qualifying customers, when the number of possible permutations in $\mathcal{S}_N$ is $20! \approx 2 \times 10^{18}$. In contrast the number of subsets in the power set is $2^{20} \approx 1 \times 10^6$. Identifying the optimal permutation can therefore

Page 5 of 19

Submitted to Phil. Trans. R. Soc. A - Issue

be computationally prohibitive in the context of integer programming techniques, even for a relatively small number of customers N. The problem is further complicated by the necessity to evaluate the performance of each candidate permutation ρ over a detailed model of uncertainty. Put differently, this issue arises because the objective function in the optimisation is given as an expectation that cannot be computed in closed form for any given permutation ρ.

We are hence faced with a *global* optimisation problem over a very large finite set, making an exhaustive search infeasible. A deterministic local search approach would require a predefined notion of a neighbourhood of a permutation ρ in $\mathcal{S}_N$ and would consist of optimising over the neighbourhood of ρ in $\mathcal{S}_N$. This would clearly lead to a local minimum at best. Furthermore, the resulting permutation would depend heavily on the specific choice of the neighbourhood of the starting permutation ρ. A general approach to global optimisation problems over very large finite (and infinite) sets, capable of circumventing these difficulties, is known as *stochastic optimisation*, see e.g. [9]. In our context, the underlying idea behind this class of methods can be described as follows: choose randomly a new permutation ρ' and use a probabilistic criterion, based on the values the objective function at ρ and ρ', to decide whether to move to ρ'. This stochastic search is not exhaustive and is hence feasible, as long as it converges rather quickly to an acceptable solution, sufficiently close to a global minimum.

A paradigm known as *Simulated Annealing* (SA) encapsulates the idea of a non-exhaustive stochastic search, generalising for example stochastic gradient and other stochastic methods. The SA paradigm is crucial in many areas of application [17,18]. Due to the general nature of SA, its definition is broad (see Section 4 below) and requires a development of an optimisation algorithm in each specific application separately. More specifically, our approach to this computational challenge will be two-fold: to simulate outcomes in a Monte Carlo fashion as described in Section 3, and to incorporate this within the SA paradigm for heuristic search, as described in Section 4. In this way we seek to identify an economically approximately optimal permutation $\rho \in \mathcal{S}_N$. However it should be noted that the specific choice of the SA algorithm is for the sake of concreteness in our running example and case study, and in principle other versions of the algorithm, based on a heuristic stochastic search paradigm, could be used.

Given a candidate permutation ρ the two time points in our probabilistic simulation model, namely DSR procurement at *time 0* and then DSR provision at *time 1*, are described in Table 1. The sum of the DNO's simulated costs for time 0 and time 1 is then calculated by adding the

Table 1. The two stages: procurement and provision

Time 0	Procurement is simulated by assessing the qualifying customers in the order specified by ρ, flipping a biased coin each time to determine the outcome of the acceptance test. Procurement stops when either the threshold H has been reached or when all customers have been assessed, whichever is earlier.
Time 1	Provision is simulated *at a single time point in the year following procurement* by simulating the level of utilisation and using the procured DSR in a least cost fashion to relieve the constrained asset. If insufficient DSR has been procured at time 0 this results in involuntary load shedding.

costs of the DSR programme given in Section 3(b) to the value of any load shed involuntarily at time 1 at a flat rate γ per MVA of load unserved. We note an interplay between the parameters γ and H. The role of γ is to make the model of DNO costs more complete by taking account of the value of lost load. In the absence of a constraint on the loss of load probability, γ could potentially therefore cause the economic optimisation to trade off a greater loss of load probability against a lower cost of procurement and provision. However the threshold H provides a guarantee that a certain minimum level of DSR will be procured (if available). Thus unless there is a problem with the availability of directly controllable load, this threshold H places a chance constraint on the loss of load at time 1 by ensuring adequate procurement at time 0. Further, as noted in Table 1, the

threshold H also has the practical purpose of indicating when the procurement process should stop.

(d) Contribution

In this paper we address the procurement and provision of DSR through load curtailment from customers local to a specific constrained network asset, taking account of uncertainty at both stages. In particular we

- provide a probabilistic problem formulation,
- propose a stochastic optimisation procedure in the framework of simulated annealing,
- develop the details of an algorithmic implementation, and
- discuss particular considerations arising from case study results.

3. Simulation model

As described above our simulation model has two parts, namely contracting and then operation. Before describing these parts we first clarify the stochastic model used for utilisation and the contract structure assumed in the DSR programme. This section concludes with a description of the optimisation problem which is based on the simulation model.

(a) Utilisation scenarios

We assume that the DNO's model of the future utilisation at time 1 is expressed using a number M of probability weighted scenarios. These scenarios are typically chosen by expert judgement and may for instance correspond to different economic scenarios. Particularly if the number of scenarios is small, there may remain significant uncertainty over utilisation *within* each scenario. To take account of this, and hence relate the scenarios more closely to the real world, an appropriate amount of random noise may be added in each scenario. Indeed by the use of a continuous distribution for the noise model we in principle allow any surrounding value to be simulated. This has the added benefit of avoiding sensitivity to the use of 'round numbers' or other particular values in the scenario predictions. In our running example and case study we choose the (zero-mean) Normal distribution for this noise. Although this specific choice is merely for concreteness in the paper, it is a well-known symmetric distribution with a single parameter (the standard deviation) and so would be a practical and transparent choice in practice. In this case the overall distribution of future utilisation across all scenarios is a *Gaussian mixture distribution*.

We use the following notation. The considered asset's capacity is denoted by $\mathcal{C}_0$. For $1 \leq i \leq M$, scenario i predicts a level μ_i of utilisation and has probability weight q_i. The utilisation in this scenario is simulated as a normally distributed random variable with mean μ_i and standard deviation σ_i. We denote by D a random variable with the corresponding Gaussian mixture distribution.

(b) DSR programme costs

The cost structure we assume for DSR procurement is as follows. Firstly there is a cost β for each acceptance test, which accounts for the DNO's resources spent in conducting each test. Secondly, provided that the invited customer passes the acceptance test, an *availability* payment accounts for fixed compensation paid, together with any costs arising from the installation of necessary communications and automation equipment. Finally, an additional *exercise* payment is made at time 1 to each customer who is required in the event to provide demand response.

Our notation is as follows. Labelling the potential DSR customers in a list from 1 to N, we attribute to the i-th customer the following data:

- C_i the DSR capacity added by a successful contract with the i-th customer;

Page 7 of 19

Submitted to Phil. Trans. R. Soc. A - Issue

- A_i the availability payment;
- E_i the exercise payment;
- p_i the probability that the i-th customer passes the acceptance test, with $0 < p_i \leq 1$.

These quantities are assumed to be known at time 0. The plausibility of this set-up, as well as alternatives to it, are discussed in Section 6. For concreteness we will take the desired total capacity (that is, the asset's capacity $\mathcal{C}_0$ plus any contracted DSR) as $H = \max_{1 \leq i \leq M}(\mu_i + 3\sigma_i)$. Provided that it is met at time 0 this target ensures that, after DSR, the utilisation at time 1 will not exceed the asset's capacity with probability more than 99.7% under our model.

Running example, Part 1. *We will develop the following example, which is based in part on [24].* *Capacities are measured in MVA and costs in thousands of GBP.*

- *Asset capacity: $\mathcal{C}_0 = 15.45$;*
- *desired total capacity threshold $H = 16.15$;*
- *cost of load unserved per MVA: $\gamma = 5000$;*
- *cost of acceptance test: $\beta = 3$;*
- *number of scenarios: 2;*
- *predicted utilisation means: $\mu_1 = 16.0$, $\mu_2 = 15.8$;*
- *predicted utilisation standard deviations: $\sigma_1 = 0.05$, $\sigma_2 = 0.1$;*
- *scenario weights: $q_1 = 0.4$, $q_2 = 0.6$.*

There are 5 qualifying customers and, for convenience, we specify their characteristics using the following vectors C, A, E and p. The i-th customer, if approached, will pass the acceptance test with probability $p_i = p[i]$, and, if successful, will enter a contract offering $C_i = C[i]$ of DSR capacity in exchange for an availability payment of $A_i = A[i]$ and additional exercise payment of $E_i = E[i]$ if its load is curtailed at time 1. We take

- *$C = [0.6, 0.5, 0.34, 0.3, 0.2]$;*
- *$A = [60, 45, 30, 24, 15]$;*
- *$E = [20, 20, 12, 12, 8]$;*
- *$p = [0.7, 0.6, 0.8, 0.6, 0.7]$.*

(c) Time 0: Procurement

Since future utilisation is uncertain, there is a risk that the DSR contracts entered at time 0 will not provide sufficient DSR capacity to bring utilisation within capacity at time 1. As discussed above, this risk may be constrained by a conservative choice for H. However in order to preserve the generality of the model, and to account for the economic cost of load unserved, we allow this event to be explicitly penalised by associating a cost γ per MVA to the load unserved at time 1. Next we describe the model of the contracts entered at time 0.

The fixed cost β for each acceptance test (which has been introduced above) is incurred irrespective of the test outcome. Independently of the order ρ and irrespective of whether the customer is approached, we associate with each potential customer i a Bernoulli random variable B_i (that is, a biased coin toss) with parameter p_i. The role of B_i is to represent the outcome of the contract negotiation with the i-th customer at time 0, should that negotiation take place: a successful contract being indicated by the value $B_i = 1$ (which has probability p_i), while the value $B_i = 0$ corresponds to no contract being entered with that customer.

The network operator stops approaching customers when either its target amount of DSR capacity has been contracted or, failing this, when all customers have been approached. After k customers have been approached, the capacity contribution from DSR at time 1 is equal to

$$\sum_{i=1}^{k} C_{\rho(i)} B_{\rho(i)}, \tag{3.1}$$

which is random since it depends on the realisation of the indicators $\mathcal{B} = (B_1, B_2, \dots)$ of success in contract negotiation. Suppose that the target is to contract sufficient DSR that a load of H (measured in MVA) can be brought within capacity at time 1. Then from (3.1), the total number of customers who are approached at time 0 will be given by T, the smallest positive integer less than N such that $\mathcal{C}_1^T \geq H$, where

$$\mathcal{C}_1^k = \mathcal{C}_0 + \sum_{i=1}^{k} C_{\rho(i)} B_{\rho(i)},$$

or alternatively if $\mathcal{C}_1^N < H$ then $T = N$. Thus T depends on $\mathcal{B}$ (so that we write $T = T(\mathcal{B})$). Denote by $L_\rho(\mathcal{B})$ the subset of $\{1, 2, \dots, N\}$ containing all the customers who have been approached, tested and successfully contracted (note that $|L_\rho(\mathcal{B})| \leq T(\mathcal{B})$).

(d) Time 1: Provision

Next we describe the operational model, which specifies the manner in which the DSR contracts agreed at time 0 are used at time 1. Given the set of contracted customers $L_\rho(\mathcal{B})$ and given also the realisation of the utilisation level D at time 1, the network operator is faced with the task of selecting the subset $K = K_\rho(\mathcal{B}, D) \subset L_\rho(\mathcal{B})$ of customers from whom demand response is to be requested in return for any agreed exercise payments. The total cost of the DSR scheme to the network operator in this model is thus

$$V_\rho^K(\mathcal{B}, D) = \beta T(\mathcal{B}) + \sum_{i \in L_\rho(\mathcal{B})} A_i + \sum_{i \in K} E_i + \gamma \Big(D - \sum_{i \in K} C_i - \mathcal{C}_0 \Big)^+ ,$$

where the first term on the right hand side is the total cost of acceptance testing for the T customers approached at time 0, the second accounts for the fixed payments under the agreed contracts, the third sums the exercise payments for the contracts exercised at time 1, and the fourth term penalises the load not served (here $(a)^+$ equals a when a is positive and equals 0 otherwise). Thus if K is chosen on a least cost basis, the corresponding total cost is

$$V_\rho(\mathcal{B}, D) = \beta T(\mathcal{B}) + \sum_{i \in L_\rho(\mathcal{B})} A_i + \min_{K \subseteq L_\rho(\mathcal{B})} \left(\sum_{i \in K} E_i + \gamma \Big(D - \sum_{i \in K} C_i - \mathcal{C}_0 \Big)^+ \right). \qquad (3.2)$$

Running example, Part 2. *Take for instance the permutation $\rho = (5, 4, 3, 2, 1)$. Suppose that at time 0 the realisations of the Bernoulli random variables are $\mathcal{B} = [1, 1, 1, 0, 1]$, and that at time 1 the realised load is $D = 15.78$.*

We have $H = 16.15$ and the first customer to be approached is number 5, since $\rho(1) = 5$. This contract is successfully negotiated since $B_5 = 1$. The resulting capacity is $\mathcal{C}_1^1 = 15.65$ since $\mathcal{C}_0 = 15.45$ and $C[5] = 0.2$. The next negotiation is with customer 4 ($\rho(2) = 4$) and is not successful ($B_4 = 0$), so that $\mathcal{C}_1^2 = 15.65$. This is followed by successful contracting with customers 3 and 2 ($B_3 = B_2 = 1$), bringing capacity successively to $\mathcal{C}_1^3 = 15.99$ and $\mathcal{C}_1^4 = 16.49$. At this point the threshold H is exceeded so the negotiation of contracts concludes and customer $\rho(5) = 1$ is not approached. The set of contracted customers is $L_\rho(\mathcal{B}) = \{2, 3, 5\}$ and the cost of acceptance testing plus the fixed contractual payments equals $4 * 3 + (15 + 30 + 45) = 102$.

A search over the possible subsets $K \subseteq L_\rho(\mathcal{B})$ reveals that the economic optimum at time 1 is to request capacity from customer 3 only, thus reaching total capacity 15.79 and exceeding the required level $D = 15.78$. The associated operational cost is then 12, and the total cost equals $V_\rho(\mathcal{B}, D) = 102 + 12 = 114$.

(e) Optimisation problem

We may now state the main optimisation problem as follows:

Page 9 of 19

Submitted to Phil. Trans. R. Soc. A - Issue

Goal: Find ρ, a permutation of $\{1, \ldots, N\}$ that minimises $\mathbb{E}\left[V_\rho(\mathcal{B}, D)\right]$.

Here $\mathbb{E}$ denotes the mathematical expectation over the distributions of $\mathcal{B}$ and D. The goal is therefore to find a permutation ρ giving the lowest average cost for the DSR scheme. This optimisation problem is challenging for the following reasons:

(i) The cost $V_\rho(\mathcal{B}, D)$ depends in a non-linear fashion on the realisation of the random quantities $\mathcal{B}$ and D. Thus its expectation cannot in general be calculated explicitly, and it may also be challenging to approximate deterministically.

(ii) The permutation space (that is, the set of all permutations ρ of N customers) is a large discrete set with no convexity structure. In addition its size grows rapidly with N, so that an exhaustive search over this space becomes computationally intractable even for modest values of N. For example in the case study below with 9 customers there are 362,880 (that is, 9 factorial) permutations, while with 20 customers there are more than 2×10^{18} permutations over which to search.

In contrast the approximation of the expectation $\mathbb{E}\left[V_\rho(\mathcal{B}, D)\right]$ for a given permutation ρ using simulations is relatively computationally inexpensive. We will therefore employ an approximate stochastic optimisation method, as follows. Beginning with an arbitrary permutation ρ we may propose another randomly chosen permutation, approximate its expected cost via simulation, and compare this to the approximate cost of the present permutation ρ. A *random walk* may then be performed over the space of all possible permutations, by moving to the new permutation if its approximate expected cost is lower, or staying at the present permtuation if it is higher. As it progresses, this random walk identifies permutations with progressively lower expected cost. It may be possible to accelerate this progress by proposing the next permutation not entirely randomly but instead in a guided fashion, choosing randomly among "neighbouring permutations", provided we can give a suitable meaning to this concept.

We now discuss the evaluation of the objective function $\mathbb{E}\left[V_\rho(\mathcal{B}, D)\right]$. If there is an analytical formula for this expectation, or an analytical expression which is known to be a close approximation, then this would provide a computationally inexpensive approach. If such an expression is not available, however, the objective function must be approximated. This may be done inexpensively using Monte Carlo simulation. Although this approximation step introduces error, this may be accounted for in the above comparison step. In particular the walk may be permitted to move to a new permutation whose approximate objective function is less favourable. Indeed, when the proposals are selected from among the neighbours of the present permutation, this approach offers the benefit of helping the walk to 'escape' from permutations which are optimal among their neighbours, but not optimal globally.

In the next section we introduce *simulated annealing*, the particular approach to stochastic optimisation considered in this paper.

4. Simulated annealing

For completeness, in this section we provide a general description of the simulated annealing algorithm. Assume that we are given a discrete parameter set Θ and an objective function $f : \Theta \to \mathbb{R}$ to be minimised over Θ, so that the optimisation problem is to find a parameter θ^* such that

$$f(\theta^*) = \min_{\theta \in \Theta} f(\theta). \tag{4.1}$$

Running example, Part 3. *In the present paper the parameter set Θ consists of all orderings of N objects, $\Theta = \mathcal{S}_N$ and the objective function is $f(\rho) = \mathbb{E}[V_\rho(\mathcal{B}, D)]$ (V_ρ is defined as in* (3.2)*).*

Simulated annealing, which was introduced in [6,11,12], is a stochastic algorithm designed to provide approximate solutions to problems of the form (4.1). It produces a time inhomogeneous Markov chain $(\Phi_k)_{k \in \mathbf{N}}$ which is a random walk in Θ, with steps proposed according to a *Markov transition kernel Q*. The walk is designed to converge towards a global minimiser θ^* for the function f.

(a) Algorithm

The SA algorithm is summarised by the following pseudocode, or informal description:

Pseudocode, simulated annealing:

Input: An objective function f, a decreasing sequence $T_1 \geq T_2 \geq T_3 \ldots$ converging to 0, an initial state $\theta_0 \in \Theta$, a Markov transition kernel Q on Θ and a *stopping rule* for the termination of the algorithm.

Output: A random sequence $(\Phi_k)_{k \in \mathbb{N}} \subset \Theta$, designed to converge to an optimal parameter θ^* solving (4.1).

(I) Set $\Phi_0 = \theta_0$;

(II) Given Φ_k simulate a proposal $\Psi_{k+1} \in \Theta$ according to the distribution $Q(\Phi_k, d\theta)$;

(III) Evaluate the acceptance probability

$$\alpha(\Phi_k, \Psi_{k+1}, T_k) := \min\left(1, \frac{\pi_k(\Psi_{k+1})}{\pi_k(\Phi_k)}\right) = \min\left(1, \exp\left(\frac{f(\Phi_k) - f(\Psi_{k+1}))}{T_k}\right)\right). \quad (4.2)$$

Throw an independent coin with probability $\alpha(\Phi_k, \Psi_{k+1}, T_k)$ of heads. In case of heads accept the proposal and set $\Phi_{k+1} = \Psi_{k+1}$. Otherwise reject the proposal and set $\Phi_{k+1} = \Phi_k$;

(IV) If the stopping rule is not reached, increase k to $k+1$, decrease T_k to T_{k+1}, go back to step (II) and repeat.

It is appropriate to use pseudocode at this point since, as mentioned in Section 2(c), SA is a general stochastic optimisation framework rather than a formal algorithm. This is because the stopping rule, cooling schedule (that is, the decreasing sequence $T_1 \geq T_2 \geq T_3 \ldots$) and the Markov kernel Q should ideally be tailored to each particular problem. In Section 4(b) below and in the case study of Section 5 we highlight particular considerations for the procurement problem under study.

SA is in fact an adaptation of the well-known Metropolis-Hastings algorithm (see [8] and references therein). That is, its goal is to obtain a sequence of random samples from a probability distribution from which direct sampling is computationally challenging. The particular idea in SA is that the mode(s) of these distributions are the minimiser(s) of the objective function f, and hence the resulting samples will tend to be located in parts of the parameter space which are approximately optimal.

Note from (4.2) that a proposed move to a smaller f value is always accepted. A proposed move to a larger f value, in contrast, is accepted with a probability which depends on T_k (and which converges to zero as $T_k \downarrow 0$). The constants T_k are chosen to decrease to 0 at an appropriate speed. In fact T_k may be called the *temperature at time k*, and the name 'simulated annealing' arises from an analogy between the algorithm and the controlled cooling used in the annealing process in metallurgy.

There are two main questions of interest regarding simulated annealing:

(i) Does the Markov chain $(\Phi_k)_{k \in \mathbb{N}}$ converge to an optimal parameter θ^*?

(ii) How quickly does this convergence take place?

The first question of convergence can be answered theoretically in certain specific cases. The papers [5,10] present sufficient conditions under which SA converges on a finite parameter space.

11

There are also results about convergence in the case of a continuous parameter space (see [3]). In general, however, establishing convergence is a challenging mathematical problem. Nevertheless simulated annealing has been applied successfully to a wide range of optimisation problems, many where no theoretical guarantee is available, and has thus proved itself to be of undisputed practical value (see for instance [26] and references therein).

Even in settings with provable convergence, though, very little has been established about the second question of the speed of convergence. Indeed the number of iterations needed to guarantee a certain level of near-optimality is usually prohibitively large for real applications (see e.g. [1]). Hence in practice SA is typically terminated long prior to any level of theoretical near-optimality through the use of a heuristic *stopping rule* with the aim of obtaining a good working solution.

(b) Discussion

Some clarifications are in order regarding the above pseudocode. Step (II) could be called *proposing a neighbour*: the transition kernel Q is simply a mathematical way to encode a rule that governs how, given the current location $\Phi_k = \theta$ of the random walk, the proposed next step $\Psi_{k+1} \in \Theta$ is generated. Although almost any such rule may be used it is advantageous if the proposal is computationally inexpensive to evaluate (there are additional considerations including *exploration* and *exploitation* of the parameter space, which are discussed below). If the parameter set Θ was continuous, for instance $\mathbb{R}^d$, then a centred uniform or normal random variable could for example be added to the current location to generate a proposal. In our present context with a discrete parameter set, if Θ can be regarded as the set of vertices of a graph (as is the case with the permutations ρ treated in this paper) then a common proposal is to choose another vertex, among the vertices that are connected to Φ_k by an edge, uniformly at random.

Running example, Part 4. *Given the current permutation $\rho = \Phi_k$, we give three methods for proposing the next step $\rho' = \Phi_{k+1}$ of a random walk over permutations. We will work with the second one in this example and with a mixture of all three in the Case Study.*

 (i) **Random permutation.** *Pick ρ' uniformly at random among all permutations.*
 (ii) **Random transposition.** *Pick two numbers i, j from $\{1, 2, \ldots, N\}$ independently and uniformly at random and then exchange $\rho(i)$ and $\rho(j)$ to obtain ρ'.*
 (iii) **Random neighbouring transposition.** *Pick a number i from $\{1, 2, \ldots, N-1\}$ uniformly at random and then exchange $\rho(i)$ and $\rho(i+1)$ to obtain ρ'. For instance, if $\rho = (5, 4, 3, 2, 1)$ and $i = 2$ then $\rho' = (5, 3, 4, 2, 1)$.*

The second and third methods use different concepts of a 'neighbouring' permutation. Here the representation of Θ as a graph, as illustrated in Figure 1, is helpful. The neighbouring transposition corresponds to moving along a single edge in this graph and so provides the most local move. Thus if the current position of the random walk is not a local optimum, the third method is typically effective in proposing an improvement when contrasted with the first method, which may easily propose an absurd permutation with significantly higher cost. Conversely the third method is incapable of improving upon a local optimum. In this case the first method, which ignores the structure of the parameter space, and also the second method which offers a compromise between the first and third, offer the opportunity to propose a better permutation and hence for the walk to escape from a local minimum.

A random mixture of all three methods thus provides a balance between exploration *of the whole parameter space and* exploitation *of the current position of the random walk, this balance depending on the relative frequencies with which the three methods are applied.*

Step (III) of the SA algorithm is the *accept-reject* step, in which the proposed next location Ψ_{k+1} of the random walk is considered. The interpretation of (4.2) is that if Ψ_{k+1} improves upon Φ_k then the walk moves there, otherwise the proposed next location Ψ_{k+1} is accepted according to the toss of a biased coin. The sequence $(T_k)_{k \in \mathbb{N}}$ of temperatures is called the *cooling schedule*. It governs

Submitted to Phil. Trans. R. Soc. A - Issue

Page 12 of 19

12

these biased coin probabilities by influencing the balance between exploration and exploitation as the random walk progresses on the parameter set Θ. It is common to use a sequence $(T_k)_{k \in \mathbb{N}}$ which decreases to zero, so that exploration gives way progressively to exploitation. A cooling schedule often used in the theoretical literature that we will also use here is of the form $T_k = \frac{h}{\log k}$ for some constant h (see [5]). The choice of cooling schedule is an interesting topic which is beyond the scope of this paper and the interested reader is referred to, for example, section 9 of [1]. We note that a systematic approach to choosing the constant h in the cooling schedule $T_k = \frac{h}{\log k}$ is described in [4] and references therein.

The starting value for the random walk, specified in step (I), is typically chosen either heuristically or randomly.

Running example, Part 5. *Initially we sort the potential customers by their unit cost of capacity, as measured by the ratio $(A_i + E_i)/C_i$. Given the parameters specified in Part 1, the random walk thus begins at the initial permutation $\Phi_0 = (5, 4, 3, 2, 1)$. We specify the cooling schedule $T_k = \frac{1}{20 \log(k)}$.*

As discussed above, the objective function $f = \mathbb{E}[V_\rho(\mathcal{B}, D)]$ is approximated by sampling. At each iteration of step (III) $n_{MC} = 500$ independent Monte Carlo simulations of $\mathcal{B}$ and D (denoted by $\mathcal{B}^n$ and D^n, $1 \leq n \leq n_{MC}$) are generated. The empirical distribution of this sample is then used to approximate the objective function as follows. We compute $V_{\Phi_k}(\mathcal{B}^n, D^n)$, $V_{\Psi_{k+1}}(\mathcal{B}^n, D^n)$ and take $\frac{1}{n_{MC}} \sum_{n=1}^{n_{MC}} V_{\Phi_k}(\mathcal{B}^n, D^n)$ and $\frac{1}{n_{MC}} \sum_{n=1}^{n_{MC}} V_{\Psi_{k+1}}(\mathcal{B}^n, D^n)$ instead of $f(\Psi_{k+1})$ and $f(\Phi_k)$ respectively. This formulation of the accept-reject step follows the treatment in, for example, [17,18].

Step (IV) terminates the algorithm using the stopping rules, which is typically derived from heuristics about the optimisation problem at hand. A common choice is to fix numbers n_R and n_S and to stop either when the last n_R proposals have all been rejected, or when n_S steps have been taken, whichever is earlier.

Running example, Part 6. *We apply the following heuristic for the stopping rule. Suppose that DSR is readily available (that is, suppose that if all potential DSR customers were contracted then the total capacity would be large compared to H, and the contracting probabilities p_i are not small). Then variations towards the end of the permutation ρ would tend to have little effect on the objective function, since in this setting the corresponding customers would be relatively unlikely to be approached at time 0. Indeed this heuristic can frequently be exact in all 500 Monte Carlo samples so that*

$$\sum_{n=1}^{500} |V_{\Phi_k}(\mathcal{B}^n, D^n) - V_{\Psi_{k+1}}(\mathcal{B}^n, D^n)| = 0.$$

Since the proposal is accepted in such cases, it may take an unacceptably long time to reject n_R consecutive proposals for any moderately large n_R. Hence we modify the stopping rule by excluding such instances, as follows. We take $n_R = 50$ and stop when the last n_R proposals, satisfying

$$\sum_{n=1}^{500} \left(V_{\Phi_k}(\mathcal{B}^n, D^n) - V_{\Psi_{k+1}}(\mathcal{B}^n, D^n) \right) \neq 0,$$

have all been rejected (or when the total number of steps reaches $n_S = 2000$). The choice $n_R = 50$ is based on the following heuristic concerning the set of 'neighbouring' parameters ρ. There are 10 possible transpositions available in proposal method (ii) above. If there is a transposition that would lower the cost, we wish the probability that this transposition goes untried before stopping to be limited to 0.5%. With $n_R = 50$ this probability equals the chance of not choosing one particular transposition (from these 10) in 50 consecutive attempts, that is $\left(\frac{10-1}{10} \right)^{50} = 0.9^{50} = 0.0051 \approx 0.5\%$.

The SA algorithm thus has the following implementation:

(I) Set $\Phi_0 = (5, 4, 3, 2, 1)$, $n = 0$ and $k = 0$;
(II) Given Φ_k pick distinct numbers i, j uniformly at random from $\{1, 2, 3, 4, 5\}$ and denote by Ψ_{k+1} the permutation equal to Φ_k with $\Phi_k(i)$ and $\Phi_k(j)$ interchanged;

Page 13 of 19

Submitted to Phil. Trans. R. Soc. A - Issue

(IIIa) For each $n \in \{1, 2 \ldots, 500\}$ simulate independent realisations of the random variables D^n and coin tosses $\mathcal{B}^n = (B_1^n, B_2^n, \ldots, B_5^n)$;

(IIIb) For each $n \in \{1, 2 \ldots, 500\}$ determinine $L_{\Phi_k}(\mathcal{B}^n)$ and $L_{\Psi_{k+1}}(\mathcal{B}^n)$ as in Subsection 3(c) and evaluate $V_{\Phi_k}(\mathcal{B}^n, D^n)$ and $V_{\Psi_{k+1}}(\mathcal{B}^n, D^n)$ as in (3.2) by cycling through all subsets of $L_{\Phi_k}(\mathcal{B}^n)$ and $L_{\Psi_{k+1}}(\mathcal{B}^n)$ respectively;

(IIIc) Set

$$\alpha(\Phi_k, \Psi_{k+1}) := \min\left(1, \exp\left(\frac{20\log(k)}{500} \sum_{n=1}^{500} V_{\Phi_k}(\mathcal{B}^n, D^n) - V_{\Psi_{k+1}}(\mathcal{B}^n, D^n)\right)\right). \tag{4.3}$$

Throw an independent coin with probability $\alpha(\Phi_k, \Psi_{k+1})$ of heads. In case of heads accept the proposal, set $\Phi_{k+1} = \Psi_{k+1}$ and set $n = 0$. Otherwise reject the proposal, set $\Phi_{k+1} = \Phi_k$ and increase n by one;

(IV) If $n < n_R = 50$ and $k < n_S = 2000$, increase k to $k + 1$, decrease T_k to T_{k+1}, go back to step (II) and repeat. Otherwise stop.

In a single run of the above SA algorithm this stopping rule was achieved after 135 steps. The final permutation was $(2, 5, 1, 4, 3)$, with approximate cost 131.74. A second independent run took 458 steps and returned the same permutation with an approximate cost of 127.90.

5. Case study application

In the absence of general theoretical guidelines regarding the choice of Markov transition kernels for proposals or the choice of stopping rules, the implementation details of the SA algorithm should typically be tuned in order to achieve reasonable results with computational efficiency. A particular feature of the optimisation problem under study is the nonlinear structure of the parameter space, since the set Θ of permutations instead has the more general structure of a graph. Our aim in this section is therefore to present a case study implementation.

(a) Utilisation scenarios and DSR procurement

As in the above example we take $\mathcal{C}_0 = 15.45$, $\gamma = 5000$ and $\beta = 3$. We now take 3 scenarios for utilisation and 9 potential DSR customers, which are as follows:

Utilisation scenarios:

- means: $\mu_1 = 16.0$, $\mu_2 = 15.6$, $\mu_3 = 15.5$;
- standard deviations: $\sigma_1 = 0.1$, $\sigma_2 = 0.1$, $\sigma_3 = 0.1$;
- probabilities of occurrence: $q_1 = 0.2$, $q_2 = 0.6$, $q_3 = 0.2$.

This results in the threshold $H = \max_{1 \leq i \leq 3}(\mu_i + 3\sigma_i) = 16.3$.

Potential DSR customers:

- $C = [0.5, 0.45, 0.4, 0.3, 0.275, 0.25, 0.25, 0.2, 0.2]$;
- $A = [40, 35, 32, 23, 22, 24, 21, 20, 18]$;
- $E = [15, 15, 13, 12, 12, 6, 8, 8, 7]$;
- $p = [0.7, 0.6, 0.8, 0.5, 0.9, 0.5, 0.8, 0.8, 0.7]$.

(b) SA algorithm implementation

- **Initialisation:** Heuristically a reasonable initial location for the random walk is the permutation which puts customers in increasing order according to their cost per unit capacity $(A + E)/C$. That is, we take $\Phi_0 = (1, 2, 3, 7, 4, 6, 5, 9, 8)$.
- **Random walk:** recalling the above example, we use the following mixture of methods to propose the next permutation in the random walk:
  - with 5% probability choose a random permutation;
  - with 15% probability choose a random transposition;

14

– with 80% probability choose a random neighbouring transposition.

Independently of the cooling schedule, the above frequencies themselves emphasise exploitation rather than exploration by weighting more highly the more local proposal methods, as discussed in part 4 of the above example. The accept-reject step is again evaluated using $n_{MC} = 500$ independent Monte Carlo simulations of utilisation D and contract success $\mathcal{B}$; the cooling schedule $T_k = \frac{1}{50 \log k}$ is used; and the stopping rule is as described above with $n_R = 50$ and $n_S = 1000$. This choice of n_R is again based heuristically on ensuring a very low probability that an improvement via a local move is missed prior to stopping. In contrast to the running example, however, here we interpret 'local move' as a neighbouring transposition, since the latter is now both the most frequent and most local type of proposal.

(c) Results

Since simulated annealing is a stochastic algorithm, over multiple independent runs several different approximately optimal permutations are returned. As our parameter space Θ of permutations has a nonlinear structure, which may instead be represented by a graph, the question of how to represent the set of results is of interest.

The case study was repeated independently 100 times (each initialised with a different seed for random number generation). Figure 1 shows the set of most commonly returned permutations in these 100 runs, together with their graph structure as permutations. Mathematically they form a cyclic graph which is generated by two (non-adjacent) transpositions, namely the transposition $(4, 6)$ which exchanges 4 and 6, and the transposition $(7, 9)$.

$$(3, 2, 1, 9, 7, 5, 8, 4, 6) \xleftrightarrow{(4,6)} (3, 2, 1, 9, 7, 5, 8, 6, 4)$$
$$\updownarrow (7,9) \qquad\qquad\qquad \updownarrow (7,9)$$
$$(3, 2, 1, 7, 9, 5, 8, 4, 6) \xleftrightarrow{(4,6)} (3, 2, 1, 7, 9, 5, 8, 6, 4)$$

Figure 1. The subgraph of most commonly returned approximately optimal permutations. These permutations are related by two non-adjacent transpositions, namely $(4, 6)$ and $(7, 9)$.

These four permutations were returned in 82 out of 100 runs (7 returns of the top left, 19 of the bottom left, 27 of the top right and 29 of the bottom right). They have the same initial three choices of customer, namely (3,2,1). In fact, the returned permutation started with $(3, 2, 1)$ in 98 out of 100 runs.

By the heuristics discussed in the running example, part 6, variations at the beginning of the permutation are likely to have the greatest effect on the objective function, and hence on the accept-reject step of the SA algorithm. The random walk should therefore provide a clear indication of the optimal initial order in which customers should be approached, and this heuristic is clearly borne out in Figure 1. Conversely we may expect ambiguity over the later stages of this order. The same argument, however, indicates that this ambiguity should not be problematic from the point of view of economic optimisation. This is because the ambiguity arises when the difference in the objective function is relatively small across the set of alternative solutions. Figure 1 illustrates that in our case study there is ambiguity over the relative ordering of two pairs of customers: namely, whether customers 4 and 6 should be interchanged in the optimal solution, and whether customers 7 and 9 should be interchanged. This suggests that the objective functions of the four permutations in Figure 1 may be almost indistinguishable.

To investigate this point further we performed a more extensive Monte Carlo simulation of D and $\mathcal{B}$. An experiment with 10^5 Monte Carlo simulations confirmed the closeness of the objective functions of the four permutations in Figure 1: indeed it yielded identical average costs

Page 15 of 19

Submitted to Phil. Trans. R. Soc. A - Issue

of 99.084×10^3 GBP for the two permutations on the left of Figure 1, and also identical costs of 99.079×10^3 GBP for the two permutations on its right. While these values were expected to be close, these pairs of *identical* costs are striking. They may be explained by considering the capacities specified in part (a) for customers 7 and 9. At time 0, if at least two of the customers $1, 2$ and 3 (who are always approached first in the subgraph shown in Figure 1) are contracted then the target H is reached before customers 7 and 9 are approached and so they do not influence the cost. Alternatively if at most one of customers $1, 2$ and 3 is contracted at time 0 then *both* customers 7 and 9 are needed in order to reach the target H, so that the order in which they are approached does not have an effect on cost (irrespective also of the contract success indicators B_7 and B_9). Thus in Figure 1, transposing 7 and 9 has no effect whatsoever on cost. Interestingly the latter discussion shows that part of the ambiguity illustrated by Figure 1 is inherent in the optimisation problem of this case study and is not the result of the iterative stochastic approach of the SA algorithm. In particular any alternative exact and deterministic approach would also be unable to distinguish between the top and bottom rows of Figure 1.

The majority of runs (71 out of 100) of this case study were stopped before the maximum run time length of $n_S = 1000$ was achieved. The average number of steps needed to satisfy the stopping rule was around 712 and the shortest run took 88 steps. Using an Intel® Core™ i7-6700 CPU @ 3.40GHz processor an average computation time required for a run was $295.3s$ with a standard deviation of $132.3s$ (with the fastest and the slowest run lasting $22.6s$ and $494.0s$, respectively). However, all of the 500 Monte Carlo simulations of utilisation D^n and contract success $\mathcal{B}^n$ that are required each time an expected cost of a permutation needs to be evaluated can be run in parallel. Thus, the algorithm can be accelerated by nearly 500 times if sufficient computing resources are available.

In order to illustrate behaviour when termination does not occur in the first 1000 steps, Figure 2 plots the evolution of cost over a randomly selected run which reached $n_S = 1000$ steps. The vertical scattering in adjacent costs is due to the stochastic nature of the algorithm, as the random variables D and $\mathcal{B}$ are independently resampled at each step rather than being held constant. To estimate the evolution of the *average* cost we also show a linear regression of these points for each 100 steps. It can be seen that in this run the average cost decays over the first 200 steps, remaining almost constant afterwards at a cost of approximately 100. This example thus provides some evidence that taking $n_S = 1000$ steps is more than enough for SA to achieve near-optimal solutions.

Figure 2 also illustrates the economic value of this optimisation procedure when compared with the simple heuristic solution of inviting qualifying customers to join the programme in increasing order of their total cost per unit capacity, $(A + E)/C$. We recall that the latter heuristic was used to define the starting permutation $(1, 2, 3, 7, 4, 6, 5, 9, 8)$. Indeed this initial permutation has an average cost (over 10^5 Monte Carlo simulations) of 106.01×10^3 GBP, which compares to an average cost of approximately 99×10^3 GBP for the best permutations obtained by SA, as discussed above.

6. Discussion and conclusion

(a) Implementation issues

We recall that during DSR provision at time 1, the discrete exhaustive search (3.2) is undertaken to identify an optimal subset K. Therefore when the number of contracted DSR customers is larger (for example 50, cf. Section 2(c)), this search also becomes computationally prohibitive. In this case an additional layer of stochastic optimisation would have to be employed to find a near optimal subset $K \subseteq L_\rho(\mathcal{B})$. Note that the simple alternative of requesting capacity in the same order as the customers were contracted is clearly suboptimal for this operational problem since the availability payments, which are influential at time 0, are sunk economic costs at time 1 and hence do not play a role in the minimisation problem (3.2).
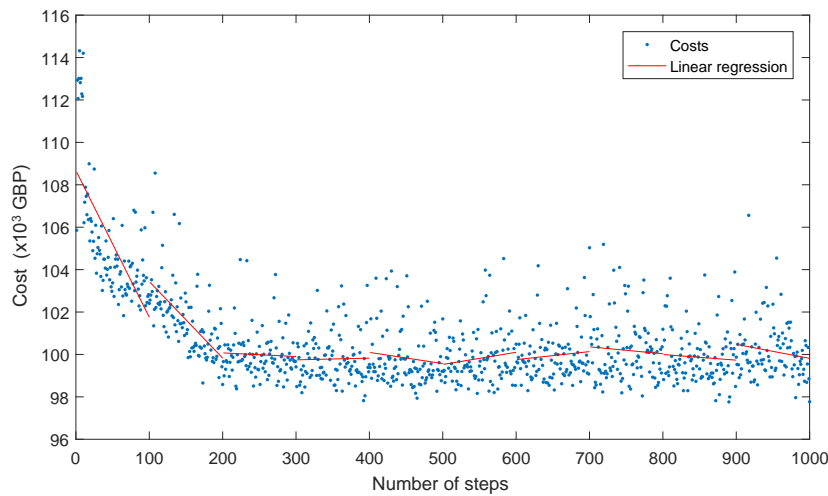
Figure 2. Evolution of estimated cost in a typical run of $n_S = 1000$ steps and linear regression lines indicating average growth of cost over every hundred steps.

The model assumes that the customers' respective probabilities p_i of passing the acceptance test are known. We acknowledge that in practice basing these numbers on expert opinion or on contract success history might not be either feasible or satisfactory. Interestingly, though, sensitivity analysis (data not shown) indicates that precision in the values of p_i is unnecessary. That is, the near optimal permutations in which the algorithm terminates seem to be robust with respect to changes in the values of p_i, provided that the majority of them are significantly different from 1, say at most 0.95. This can be understood in light of the discussion in Section 2(b). If most of the values p_i are very high then the uncertainty in the procurement process becomes insignificant, and our problem of choosing an optimal permutation thus starts to collapse into the integer programming problem of choosing an optimal subset. Conversely if the probabilities p_i are significantly lower than 1 then the optimisation must take account of randomness in the procurement process, and the resulting solution can then be understood as a 'wish list' which ranks the qualifying customers in decreasing order of their desirability to the DNO. In contrast the average cost objective function is of course somewhat sensitive to the values of p_i, since when these probabilities are lower a greater number of costly acceptance tests tends to be required before the procurement threshold H is reached.

(b) Extension to multiple periods

As we have noted, DSR contracts may in practice cover multiple years and we therefore comment on potential multiple period extensions to the above framework. There are many possible directions for such generalisations, and developing them case by case with industrial feedback could be an interesting direction for future research. For instance different customers' participation in the DSR programme could be planned to start in different years, or an option could be added for each customer to withdraw from a contract after a prescribed time period by paying a penalty fee. In the mathematical formulation of Section 2(c) the representation ρ of candidate solutions would then need to be extended to an appropriate analogue, along with the simulation models for procurement and provision and the associated cost model. At the level

Page 17 of 19

Submitted to Phil. Trans. R. Soc. A - Issue

of utilisation modelling (Section 3(a)), each utilisation scenario would then become a pathway over the considered future time points rather than a point forecast. Although the computational complexity of the heuristic search then increases accordingly, only relatively small numbers of customers need to be considered in the setting of this paper. Further because of their flexibility and relatively low capital costs, DSR programmes can be planned for a relatively small number of years. Therefore it is not anticipated that typical extensions to multiple time periods should cause SA to become computationally intractable, rather stochastic optimisation algorithms like SA should remain an appropriate method for heuristic search.

Further in the case of multiple time periods there is the opportunity to apply the above model in a 'receding horizon' fashion, as uncertainty gives way to additional information (see for example [24]). That is, the optimisation may be repeated as each successive time period passes, to verify whether any refinements should be made to the optimal policy in the light of the new information.

Authors' Contributions. JV designed and carried out the numerical experiments and drafted the manuscript. JM and AM conceived of the study. All authors edited and approved the manuscript.

Competing Interests. The authors declare that they have no competing interests.

References

1. Emile H. L. Aarts, Jan H. M. Korst, and Peter J. M. van Laarhoven.
   Simulated annealing.
   In *Local search in combinatorial optimization*, Wiley-Intersci. Ser. Discrete Math. Optim., pages 91–120. Wiley, Chichester, 1997.
2. Mahnoosh Alizadeh, Anna Scaglione, and Robert J Thomas.
   From packet to power switching: Digital direct load scheduling.
   IEEE Journal on Selected Areas in Communications, 30(6):1027–1036, 2012.
3. Christophe Andrieu, Laird A. Breyer, and Arnaud Doucet.
   Convergence of simulated annealing using Foster-Lyapunov criteria.
   J. Appl. Probab., 38(4):975–994, 2001.
4. Walid Ben-Ameur.
   Computing the initial temperature of simulated annealing.
   Comput. Optim. Appl., 29(3):369–385, 2004.
5. Dimitris Bertsimas and John Tsitsiklis.
   Simulated annealing.
   In *Probability and algorithms*, pages 17–29. Nat. Acad. Press, Washington, DC, 1992.
6. Vladimír Černý.
   Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm.
   Journal of optimization theory and applications, 45(1):41–51, 1985.
7. Chen Chen, Jianhui Wang, and Shalinee Kishore.
   A distributed direct load control approach for large-scale residential demand response.
   IEEE Transactions on Power Systems, 29(5):2219–2228, 2014.
8. Siddhartha Chib and Edward Greenberg.
   Understanding the metropolis-hastings algorithm.
   The american statistician, 49(4):327–335, 1995.
9. Dimitris Fouskakis and David Draper.
   Stochastic optimization: a review.
   International Statistical Review, 70(3):315–349, 2002.
10. Bruce Hajek.
    Cooling schedules for optimal annealing.
    Math. Oper. Res., 13(2):311–329, 1988.

17

18

11. S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi.
Optimization by simulated annealing.
Science, 220(4598):671–680, 1983.

12. Scott Kirkpatrick.
Optimization by simulated annealing: quantitative studies.
J. Statist. Phys., 34(5-6):975–986, 1984.

13. Junji Kondoh, Ning Lu, and Donald J Hammerstrom.
An evaluation of the water heater load potential for providing regulation service.
In *Power and Energy Society General Meeting, 2011 IEEE*, pages 1–8. IEEE, 2011.

14. Electricity North West Ltd.
Managed connections.
http://www.enwl.co.uk/our-services/connection-services/generation/
managed-connections.
[Online; accessed 7th February 2017].

15. Electricity North West Ltd.
Real options decision support model.
http://www.enwl.co.uk/realoptions.
[Online; accessed 7th February 2017].

16. Killian McKenna and Andrew Keane.
Residential load modeling of price-based demand response for network impact studies.
IEEE Transactions on Smart Grid, 7(5):2285–2294, 2016.

17. Peter Müller.
Simulation-based optimal design.
In *Bayesian statistics, 6 (Alcoceber, 1998)*, pages 459–474. Oxford Univ. Press, New York, 1999.

18. Peter Müller, Bruno Sansó, and Maria De Iorio.
Optimal Bayesian design by inhomogeneous Markov chain simulation.
J. Amer. Statist. Assoc., 99(467):788–798, 2004.

19. UK Office of Gas and Electricity Markets.
Demand side response: A discussion paper.
https://www.ofgem.gov.uk/ofgem-publications/57026/dsr-150710.pdf,
2010.
[Online; accessed 1st March 2017].

20. UK Office of Gas and Electricity Markets.
Making the electricity system more flexible and delivering the benefits for consumers, 2015.

21. Peter Palensky and Dietmar Dietrich.
Demand side management: Demand response, intelligent energy systems, and smart loads.
IEEE transactions on industrial informatics, 7(3):381–388, 2011.

22. National Grid plc.
Technical guidance and testing procedure for static and dynamic demand response and
battery storage providers of frequency balancing services.
www2.nationalgrid.com/WorkArea/DownloadAsset.aspx?id=40571, 2015.
[Online; accessed 1st March 2017].

23. Vinayak V Ranade and Jacob Beal.
Distributed control for small customer energy demand management.
In *Self-adaptive and self-organizing systems (SASO), 2010 4th IEEE international conference on*,
pages 11–20. IEEE, 2010.

24. Jonathan A. Schachter, Pierluigi Mancarella, John Moriarty, and Rita Shaw.
Flexible investment under uncertainty in smart distribution networks with demand side
response: Assessment framework and practical implementation.
Energy Policy, 97:439 – 449, 2016.

25. Goran Strbac.
Demand side management: Benefits and challenges.
Energy policy, 36(12):4419–4426, 2008.

26. Cher Ming Tan, editor.
Simulated Annealing.
InTech, 2008.

27. Jacopo Torriti.
Peak energy demand and demand side response.

Page 19 of 19

Submitted to Phil. Trans. R. Soc. A - Issue

Routledge, 2015.

28. Vamsi Krishna Tumuluru and Danny HK Tsang.
A two-stage approach for network constrained unit commitment problem with demand response.
IEEE Transactions on Smart Grid, 2016.

19