

On minimum cost local permutation problems and their application to smart meter data

Nathaniel Charlton¹, Danica Vukadinović Greetham², and Colin Singleton¹

¹ CountingLab Ltd <http://www.countinglab.co.uk/>
{billiejoe,colin}@countinglab.co.uk

² Department of Mathematics and Statistics
University of Reading, UK
d.v.greetham@reading.ac.uk

Abstract. A w -local permutation is a permutation π on $\{1, \dots, n\}$ that displaces each element by a distance of at most w . We define the *minimum cost local permutation* (MCLP) problem, where there is a cost associated with each choice of $\pi(i)$ for $i = 1, \dots, n$ and the goal is to find the w -local permutation of minimum total cost. The MCLP problem generalises the problem of computing the *adjusted error*, a measure of the similarity of two household-level smart meter energy profiles. Existing work has reduced MCLP to the assignment problem, which can be solved in $O(n^3)$ time using the Hungarian algorithm.

We prove a reduction of the MCLP problem to that of computing the shortest path in a directed acyclic graph. This yields an algorithm that, for fixed w , solves MCLP in $O(n)$ time. Analysis of running times for adjusted error computations on a real smart meter data set confirms that the new method is far faster in practice.

Further, we study N -MCLP, a generalisation of MCLP where N permutations are chosen simultaneously to minimise the associated total cost. N -MCLP generalises the problem of computing an appropriate “average” of N smart meter profiles with respect to the adjusted error measure. As in the MCLP case, we prove a reduction to the problem of computing the shortest path in a directed acyclic graph. We apply the resulting algorithm to a smart meter data set, obtaining improved forecasts for household-level energy consumption.

1 Introduction

We introduce and study the *minimum cost local permutation* (MCLP) problem, an optimisation problem regarding local permutations. A permutation π on $\{1, \dots, n\}$ is said to be w -local if it displaces each element by a distance of at most some predefined limit w . In the MCLP problem there is a cost associated with each choice of $\pi(i)$ for $i = 1, \dots, n$ and the goal is to find the w -local permutation of minimum total cost. We further introduce and study an N -way generalisation of the problem, N -MCLP, where instead of a single permutation, N permutations must be chosen simultaneously to minimise a total cost.

These two problems arise as generalisations of two problems we have encountered when dealing with smart meter data detailing the energy consumption over time of individual households:

1. The *adjusted error* [9] has been proposed as a way of measuring how close a household-level energy use forecast is to the subsequent actual use, and is defined in terms of local permutations. How can we compute adjusted errors efficiently?
2. Given N days of historical energy use data for a particular household, how can we combine them into a day-ahead forecast which performs well under the adjusted error measure?

The remainder of this report is structured as follows. Section 2 introduces the MCLP problem, giving a definition and a simple example. Section 3 gives the reduction we use to solve MCLP: we reduce the MCLP problem to the problem of computing the shortest path between two vertices of a directed acyclic graph. We explain the reduction informally through an example, before defining it precisely and proving its correctness. In Section 4 we present the generalised problem N -MCLP, which we solve using a similar graph-based method; again we explain the method informally before proving it correct. In Section 5 we apply our algorithms to the two smart meter-related problems described above. Section 6 examines the computational efficiency of our methods, in theory and by reporting run times in practice, and Section 7 concludes.

A brief description of this work, omitting the technical details but with further discussion of the applications to smart meter data, appears in our workshop paper [2].

2 The minimum cost local permutation (MCLP) problem

The following definitions and example introduce the *minimum cost local permutation* (MCLP) problem.

Definition 1. A function π from the set $\{1, \dots, n\}$ to itself is w -local, for $0 \leq w < n$, if for all i , $|\pi(i) - i| \leq w$. We write $\mathcal{P}(w, n)$ for the set of w -local permutations on $\{1, \dots, n\}$. \square

Definition 2. An instance $\text{MCLP}(n, w, C)$ of the minimum cost local permutation problem comprises:

- integers n and w such that $0 \leq w < n$
- a function $C : \{1, \dots, n\} \times \{1, \dots, n\} \rightarrow \mathbb{R}_{\geq 0}$ assigning costs to the permuted points; $C(i, j)$ is the cost of mapping point i onto point j .

Given a permutation $\pi \in \mathcal{P}(w, n)$ we define the cost of π to be

$$\text{Cost}(\pi) := \sum_{i=1}^n C(i, \pi(i)) \quad (1)$$

A solution to the MCLP problem is a permutation $\pi \in \mathcal{P}(w, n)$ that minimises $\text{Cost}(\pi)$ (in general there may be multiple such permutations). \square

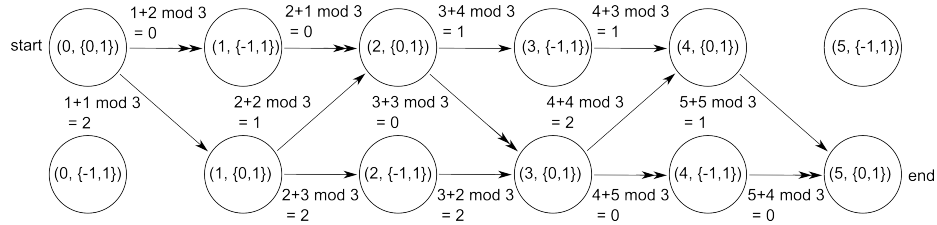


Fig. 1. The directed acyclic graph we use to solve the MCLP problem in Example 1.

Example 1. We give an example instance of MCLP: let $n = 5$, $w = 1$ and define the cost $C(i, j)$ of mapping point i onto point j by

$$C(i, j) := i + j \pmod{3}$$

The cost of the identity permutation I is then

$$\begin{aligned} \text{Cost}(I) &= C(1, 1) + C(2, 2) + C(3, 3) + C(4, 4) + C(5, 5) \\ &= 2 + 1 + 0 + 2 + 1 \\ &= 6 \end{aligned}$$

In this case the permutation $\{1 \mapsto 2, 2 \mapsto 1, 3 \mapsto 3, 4 \mapsto 5, 5 \mapsto 4\}$ is the unique solution, having cost

$$\begin{aligned} &C(1, 2) + C(2, 1) + C(3, 3) + C(4, 5) + C(5, 4) \\ &= 0 + 0 + 0 + 0 + 0 \\ &= 0 \end{aligned}$$

□

3 Reducing MCLP to shortest path in a DAG

In this section we show how to reduce the MCLP problem to that of computing the shortest path in a directed acyclic graph (DAG). We begin with a preliminary definition.

Definition 3. An availability set S is a subset of $\{-w, \dots, w\}$ such that $w \in S$ and $|S| = w + 1$. Let \mathcal{S} be the set of all such availability sets; there are $\binom{2w}{w}$ of these. Define a distinguished availability set $S^+ := \{0, \dots, w\}$. □

Fig. 1 shows the DAG that we will construct from the MCLP instance in Example 1. Using this, we now explain informally the idea of our construction, which we will formalise and prove shortly.

Choosing a path from “start” to “end” corresponds to choosing a w -local permutation π : specifically, choosing the i th edge of such a path (counting from 1) corresponds to choosing the value of $\pi(i)$. Each node is labelled with a pair (i, S) where i gives the number of points of the image of π chosen so far, and

the availability set S records which elements of $\{1, \dots, n\}$ are available when choosing $\pi(i+1)$. Specifically, if $k \in S$ then we are able to choose to map the value $i+1$ onto the value $i+1+k$. The cost of each edge leaving the node (i, S) is the cost associated with mapping $i+1$ to the chosen value.

For example, when we are at the start node $(0, \{0, 1\})$ in Fig. 1 we have chosen 0 points of the image of π so far, and we are about to choose $\pi(1)$. The available choices, corresponding to the two outgoing edges, are 1 and 2. Suppose we take the horizontal edge, which corresponds to choosing 1 from the availability set to put $\pi(1) := 2$. The cost of the edge is $C(1, 2) = 1 + 2 \bmod 3 = 0$, which will be the contribution for $i = 1$ to the summation in (1).

The node $(1, \{-1, 1\})$ that we reach has availability set $\{-1, 1\}$ reflecting that the available options for our next choice, the choice of $\pi(2)$, are 1 and 3; these numbers lie, respectively, at offsets of -1 and 1 from the “current” point 2. The availability set does not contain 0, corresponding to a choice of 2, because we used up 2 already at the previous node.

In fact, however, choosing $\pi(2) := 3$ is not feasible: elements can be displaced by at most $w = 1$, so if we have assigned $\pi(1) := 2$ and then $\pi(2) := 3$, we will not be able to map any of the remaining values 3, 4, 5 onto 1. Thus there is only one edge available from the node $(1, \{-1, 1\})$. Continuing in this way, we gradually choose the whole 1-local permutation π ; when we reach the “end” node we have finished. The path highlighted with double arrowheads in Fig. 1 corresponds to the unique solution given earlier.

Once we have the shortest path, we can recover the required permutation. To find $\pi(i)$, we go to the i th node of the path (counting from 1); this node will have the form $(i-1, S)$ and the next node in the path will have the form (i, S') . Then $\pi(i) = i + m$ where m is the unique number such that $m \in S$ but $m-1 \notin S'$. For example, let us find $\pi(1)$ for the path highlighted with double arrowheads in Fig. 1. Going to the first node we have $S = \{0, 1\}$ and $S' = \{-1, 1\}$. Thus the unique number m described above is 1, so $\pi(1) = 2$.

The following definition shows how to construct the appropriate graph for an arbitrary MCLP problem.

Definition 4. *Given a problem instance $\text{MCLP}(n, w, C)$, we define as follows a DAG $G(n, w, C)$. We take as vertices the elements of $\{0, \dots, n\} \times \mathcal{S}$. We define*

$$\text{decrease}(X) := \{m-1 \mid m \in X\}$$

Next we define a partial function T from pairs of vertices to $\{-w, \dots, w\}$:

$$T((i, S), (i', S')) := \begin{cases} \min(X) & \text{if } i' = i+1 \text{ and} \\ & X = \left\{ t \in S \mid \begin{array}{l} 1 \leq i+1+t \leq n \text{ and} \\ S' = \text{decrease}(S \setminus \{t\}) \cup \{w\} \end{array} \right\} \\ & \text{is nonempty} \\ \perp & \text{otherwise} \end{cases}$$

(Here the symbol \perp means undefined.) We put an edge from vertex $v = (i, S)$ to vertex $v' = (i + 1, S')$ if $T(v, v') \neq \perp$; we set the cost of the edge to $C(i + 1, i + 1 + T(v, v'))$. We define distinguished vertices $v_{\text{start}} := (0, S^+)$ and $v_{\text{end}} := (n, S^+)$. \square

In fact, the set X in the definition of T above is either empty or a singleton set, as the following lemma shows.

Lemma 1. *Let (i, S) and $(i + 1, S')$ be vertices of $G(n, w, C)$, and let $t, t' \in S$ be such that:*

- $1 \leq i + 1 + t \leq n$ and $S' = \text{decrease}(S \setminus \{t\}) \cup \{w\}$
- $1 \leq i + 1 + t' \leq n$ and $S' = \text{decrease}(S \setminus \{t'\}) \cup \{w\}$

Then $t = t'$.

Proof. We have $t' - 1 \notin \text{decrease}(S \setminus \{t'\}) \cup \{w\}$ so $t' - 1 \notin S'$. Suppose for a contradiction that $t \neq t'$. Then $t' - 1 \in \text{decrease}(S \setminus \{t\}) \cup \{w\}$ so $t' - 1 \in S'$. \square

The following definition makes precise the idea explained above of recovering a permutation from the shortest path.

Definition 5. *Let $P = v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_n$ be a path from v_{start} to v_{end} in $G(n, w, C)$. Then we define a corresponding function $\pi_P : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ by $\pi_P(i) := i + T(v_{i-1}, v_i)$. (Note that $T(v_{i-1}, v_i)$ is always defined here, because by assumption there is an edge from v_{i-1} to v_i , and Definition 4 puts edges exactly where T is defined.) \square*

It is easy to see that π_P as just defined is a w -local function. The following theorem formalises the link between a problem instance $\text{MCLP}(n, w, C)$ and the corresponding graph $G(n, w, C)$.

Theorem 1. *Consider a problem instance $\text{MCLP}(n, w, C)$. Let P be a path of minimal length from v_{start} to v_{end} in the associated graph $G(n, w, C)$. Then π_P is a solution to $\text{MCLP}(n, w, C)$. \square*

Once we prove this theorem, we will have an algorithm for solving instances $\text{MCLP}(n, w, C)$ of the MCLP problem: first construct the graph $G(n, w, C)$, then find the shortest path P from v_{start} to v_{end} , and finally recover from P the required permutation π_P .

We conclude this section by proving Theorem 1. The theorem follows from the following two lemmas: one shows how to go from a path with length L to a permutation with total cost L , and the other shows how to go from a permutation with total cost L to a path of length L .

Lemma 2 (from paths to local permutations). *Let P be a path in $G(n, w, C)$ from v_{start} to v_{end} with length L . Then $\pi_P \in \mathcal{P}(w, n)$ and $\text{Cost}(\pi_P) = L$.*

Proof. (Note that although it is clear from Definition 5 that π_P is a w -local function, it is *not* obvious that it is a *permutation*. The first task in our proof is to establish this.) P must have the form

$$v_0 \rightarrow v_1 \rightarrow \cdots \rightarrow v_n$$

where each v_i has the form (i, S_i) and $S_0 = S_n = S^+$. We define a sequence of partial functions $R_0, R_1, \dots, R_n : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ by $R_0(m) := \perp$ and, for $i > 0$, $R_i := \pi_P|_{\{1, \dots, i\}}$. For $i \in \{0, \dots, n\}$ let $\Phi(i)$ be the following statement:

The image $\text{Im}(R_i)$ and the set $\{i + 1 + m \mid m \in S_i\}$ are disjoint, and their union is $\{1, \dots, i + 1 + w\}$.

We will prove by induction that $\Phi(i)$ holds for all $i \in \{0, \dots, n\}$. The base case is easily checked. For the inductive case, suppose that $\Phi(i)$ holds and let us prove $\Phi(i + 1)$. We can write R_{i+1} in terms of R_i as follows:

$$R_{i+1}(m) = \begin{cases} i + 1 + T(v_i, v_{i+1}) & \text{if } m = i + 1 \\ R_i(m) & \text{otherwise} \end{cases}$$

Also by Definition 4 and Lemma 1 we have

$$S_{i+1} = \text{decrease}(S_i \setminus \{T(v_i, v_{i+1})\}) \cup \{w\} \quad (2)$$

We shall now show that

$$\text{Im}(R_{i+1}) \cap \{i + 2 + m \mid m \in S_{i+1}\} = \emptyset$$

So let $x \in \{i + 2 + m \mid m \in S_{i+1}\}$ and we will prove $x \notin \text{Im}(R_{i+1})$, which is equivalent to

$$x \notin \text{Im}(R_i) \cup \{i + 1 + T(v_i, v_{i+1})\}$$

There is some $m' \in S_{i+1}$ such that $x = i + 2 + m'$. From (2) there are two cases.

- (a) $m' = w$: Thus $x = i + 2 + w$. We cannot have $x = i + 1 + T(v_i, v_{i+1})$ because $T(v_i, v_{i+1}) \leq w$. Also we cannot have $x \in \text{Im}(R_i)$ because the induction hypothesis tells us that $\text{Im}(R_i) \subseteq \{1, \dots, i + 1 + w\}$.
- (b) $m' \in \text{decrease}(S_i \setminus \{T(v_i, v_{i+1})\})$: It follows that $m' + 1 \in S_i$. Thus $i + m' \in \{i + m \mid m + 1 \in S_i\}$. Hence $i + m' \in \{i + m - 1 \mid m \in S_i\}$. Therefore $i + 2 + m' \in \{i + m + 1 \mid m \in S_i\}$ i.e. $x \in \{i + m + 1 \mid m \in S_i\}$. The disjointness part of the induction hypothesis now tells us that $x \notin \text{Im}(R_i)$. It remains to show that $x \neq i + 1 + T(v_i, v_{i+1})$, which, since $x = i + 2 + m'$, is equivalent to $i + 2 + m' \neq i + 1 + T(v_i, v_{i+1})$, that is, $m' + 1 \neq T(v_i, v_{i+1})$. But this follows from $m' \in \text{decrease}(S_i \setminus \{T(v_i, v_{i+1})\})$.

Next we need to prove

$$\text{Im}(R_{i+1}) \cup \{i + 2 + m \mid m \in S_{i+1}\} = \{1, \dots, i + w + 2\}$$

Firstly, let x be an element of the RHS and we will show that it is also an element of the LHS. There are two cases:

- (a) $x = i + w + 2$: From (2), $w \in S_{i+1}$. Hence $x \in \{i + 2 + m \mid m \in S_{i+1}\}$.
- (b) $x \in \{1, \dots, i + w + 1\}$: Then by the induction hypothesis we have $x \in \text{Im}(R_i) \cup \{i + 1 + m \mid m \in S_i\}$. If $x \in \text{Im}(R_i)$ then clearly also $x \in \text{Im}(R_{i+1})$ and we are done. Now consider the case when $x \in \{i + 1 + m \mid m \in S_i\}$. There exists $m' \in S_i$ such that $x = i + 1 + m'$. There are two subcases.
- (1) $m' = T(v_i, v_{i+1})$: Then $x = i + 1 + T(v_i, v_{i+1})$ and it is immediate from the definition of R_{i+1} that $x \in \text{Im}(R_{i+1})$.
- (2) $m' \neq T(v_i, v_{i+1})$: It then follows from (2) that $m' - 1 \in S_{i+1}$. Defining $m'' := m' - 1$ we have $m'' \in S_{i+1}$ and $x = i + 2 + m''$. Hence $x \in \{i + 2 + m \mid m \in S_{i+1}\}$.

Secondly, let x be an element of the LHS and we will show that it is also an element of the RHS. There are two cases.

- (a) $x \in \text{Im}(R_{i+1})$: There are two subcases:
- (1) $x \in \text{Im}(R_i)$: By the induction hypothesis we have $x \in \{1, \dots, i + w + 1\}$. Therefore $x \in \{1, \dots, i + w + 2\}$.
- (2) $x = i + 1 + T(v_i, v_{i+1})$: From the induction hypothesis we know that $\{i + 1 + m \mid m \in S_i\} \subseteq \{1, \dots, i + 1 + w\}$; also $T(v_i, v_{i+1}) \in S_i$, and therefore $x \in \{1, \dots, i + w + 2\}$.
- (b) $x \in \{i + 2 + m \mid m \in S_{i+1}\}$: Then there is some $m' \in S_{i+1}$ such that $x = i + 2 + m'$. Equation (2) gives rise to two subcases:
- (1) $m' = w$: Then $x = i + 2 + w$ so clearly $x \in \{1, \dots, i + w + 2\}$.
- (2) $m' \in \text{decrease}(S_i \setminus \{T(v_i, v_{i+1})\})$: It follows that $m' + 1 \in S_i$. Thus $i + m' \in \{i + m \mid m + 1 \in S_i\}$. Hence $i + m' \in \{i + m - 1 \mid m \in S_i\}$. Therefore $i + 2 + m' \in \{i + m + 1 \mid m \in S_i\}$ i.e. $x \in \{i + m + 1 \mid m \in S_i\}$. The induction hypothesis tells us that $\{i + 1 + m \mid m \in S_i\} \subseteq \{1, \dots, i + 1 + w\}$, and thus $x \in \{1, \dots, i + w + 2\}$.

We have thus established that $\Phi(i)$ holds for all $i \in \{0, \dots, n\}$. From $\Phi(n)$ it follows that $\text{Im}(R_n)$ and $\{n + 1 + m \mid m \in S_n\}$ are disjoint with union $\{1, \dots, n + 1 + w\}$. But $S_n = S^+ = \{0, \dots, w\}$, so we have $\text{Im}(P) = \text{Im}(R_n) = \{1, \dots, n\}$ i.e. π_P is onto the set $\{1, \dots, n\}$. It then follows for cardinality reasons that π_P is a bijection i.e. a permutation.

It remains to check that $\text{Cost}(\pi_P) = L$. We have (from (1) and Definition 5)

$$\text{Cost}(\pi_P) = \sum_{i=1}^n C(i, \pi_P(i)) = \sum_{i=1}^n C(i, i + T(v_{i-1}, v_i))$$

From Definition 4 the length of the edge from (i, S_i) to $(i + 1, S_{i+1})$ for $i \in \{0, \dots, n - 1\}$ is $C(i + 1, i + 1 + T(v_i, v_{i+1}))$; hence

$$L = \sum_{i=0}^{n-1} C(i + 1, i + 1 + T(v_i, v_{i+1})) = \text{Cost}(\pi_P)$$

□

Lemma 3 (from local permutations to paths). *Let $\pi \in \mathcal{P}(w, n)$. Then there exists a path P in $G(n, w, C)$ from v_{start} to v_{end} with length $\text{Cost}(\pi)$.*

Proof. For each $i \in \{0, \dots, n\}$ we construct a set $S_i \subseteq \{-w, \dots, w\}$ as follows:

$$S_i := \{m \in \{-w, \dots, w\} \mid i + 1 + m \notin \text{Im}(\pi|_{\{1, \dots, i\}}) \text{ and } i + 1 + m > 0\}$$

It is easy to see that $S_0 = S^+$. We now verify that $S_n = S^+$. S_n simplifies to

$$\{m \in \{-w, \dots, w\} \mid n + 1 + m \notin \{1, \dots, n\}\}$$

and this is equal to S^+ .

Next we will show that, for each $i \in \{0, \dots, n-1\}$, $T((i, S_i), (i+1, S_{i+1}))$ is defined and equal to $\pi(i+1) - i - 1$. By Lemma 1 this amounts to showing that

1. $\pi(i+1) - i - 1 \in S_i$
2. $1 \leq i + 1 + \pi(i+1) - i - 1 \leq n$
3. $S_{i+1} = \text{decrease}(S_i \setminus \{\pi(i+1) - i - 1\}) \cup \{w\}$

For 1.), we first need to show that $\pi(i+1) - i - 1 \in \{-w, \dots, w\}$. This follows from the w -locality of π . Next we need to show that $i + 1 + \pi(i+1) - i - 1 \notin \text{Im}(\pi|_{\{1, \dots, i\}})$, i.e. $\pi(i+1) \notin \text{Im}(\pi|_{\{1, \dots, i\}})$. This follows from the injectivity of π . Finally we must show $i + 1 + \pi(i+1) - i - 1 > 0$ which trivially simplifies to $\pi(i+1) > 0$.

For 2.) we easily simplify our goal to $1 \leq \pi(i+1) \leq n$ which trivially holds.

For 3.) we first prove the \subseteq inclusion. Let $x \in S_{i+1}$. If $x = w$ we are done, so suppose $x \neq w$. Because $x \in S_{i+1}$ we have $x \in \{-w, \dots, w\}$,

$$i + 2 + x \notin \text{Im}(\pi|_{\{1, \dots, i+1\}}) \tag{3}$$

and

$$i + 2 + x > 0 \tag{4}$$

It will suffice to prove that (a). $x + 1 \in S_i$ and (b). $x + 1 \neq \pi(i+1) - i - 1$. For (a). we need to show

$$x + 1 \in \{m \in \{-w, \dots, w\} \mid i + 1 + m \notin \text{Im}(\pi|_{\{1, \dots, i\}}) \text{ and } i + 1 + m > 0\}$$

From $x \in \{-w, \dots, w\}$ and $x \neq w$ it follows that $x + 1 \in \{-w, \dots, w\}$. Next, $i + 1 + (x + 1) \notin \text{Im}(\pi|_{\{1, \dots, i\}})$ follows from (3). Finally, $i + 1 + (x + 1) > 0$ is just (4). (b). is equivalent to $i + 2 + x \neq \pi(i+1)$ which follows from (3).

Secondly we prove the \supseteq inclusion. Let $x \in \text{decrease}(S_i \setminus \{\pi(i+1) - i - 1\}) \cup \{w\}$. We split into two cases; in each we must show $x \in S_{i+1}$.

- (a) $x = w$: To show $x \in S_{i+1}$ it suffices to show $i + 2 + w \notin \text{Im}(\pi|_{\{1, \dots, i+1\}})$ and $i + 2 + w > 0$. The second conjunct holds trivially; for the first, observe that by w -locality of π , $\pi^{-1}(i + 2 + w)$ can be no smaller than $i + 2$.

(b) $x \in \text{decrease}(S_i \setminus \{\pi(i+1) - i - 1\})$: Then $x + 1 \in S_i$ and

$$x + 1 \neq \pi(i+1) - i - 1 \quad (5)$$

The first step to showing $x \in S_{i+1}$ is showing that $x \in \{-w, \dots, w\}$. We know that $-w \leq x + 1 \leq w$ i.e. $-w - 1 \leq x < w$, but we still need to make sure that $x \neq -w - 1$. So suppose for a contradiction that $x = -w - 1$. Then $x + 1 = -w$, so $-w \in S_i$. Therefore

$$i + 1 - w \notin \text{Im}(\pi|_{\{1, \dots, i\}}) \quad (6)$$

and $i + 1 - w > 0$. By w -locality of π we have $\pi^{-1}(i + 1 - w) \leq i + 1$; together with (6) this implies that $\pi(i + 1) = i + 1 - w = i + 2 + x$. But this contradicts (5). Hence we have proved that $x \in \{-w, \dots, w\}$. Now, from $x + 1 \in S_i$ it follows that

$$i + 2 + x \notin \text{Im}(\pi|_{\{1, \dots, i\}}) \quad (7)$$

and $i + 2 + x > 0$.

To show $x \in S_{i+1}$ it suffices to prove that $i + 2 + x \notin \text{Im}(\pi|_{\{1, \dots, i+1\}})$ and $i + 2 + x > 0$. The second conjunct we have shown already; (7) almost implies for the first conjunct, leaving us only to check that $\pi(i + 1) \neq i + 2 + x$. But this follows from (5).

We have thus shown that, for each $i \in \{0, \dots, n - 1\}$, $T((i, S_i), (i + 1, S_{i+1}))$ is defined and equal to $\pi(i + 1) - i - 1$. This means that, according to Definition 4, $P = (0, S_0) \rightarrow (1, S_1) \rightarrow \dots \rightarrow (n, S_n)$ is a path from v_{start} to v_{end} in $G(n, w, C)$.

The length of the i th edge of P (counting from 0) is $C(i + 1, i + 1 + T((i, S_i), (i + 1, S_{i+1})))$ so the length of P is

$$\sum_{i=0}^{n-1} C(i + 1, i + 1 + T((i, S_i), (i + 1, S_{i+1})))$$

which, using $T((i, S_i), (i + 1, S_{i+1})) = \pi(i + 1) - i - 1$, we can rewrite as

$$\sum_{i=0}^{n-1} C(i + 1, \pi(i + 1))$$

and this equals $\text{Cost}(\pi)$ as required. \square

4 Generalisation to N permutations

We now define a generalisation N -MCLP of the MCLP problem, where N local permutations are chosen simultaneously to minimise their joint total cost.

Definition 6. *An instance N -MCLP(n, w, C) of the N -way minimum cost local permutation problem comprises:*

- integers n and w such that $0 \leq w < n$
- a function $C : \{1, \dots, n\} \times \{1, \dots, n\}^N \rightarrow \mathbb{R}_{\geq 0}$ assigning costs to the permuted points; $C(i, j^1, \dots, j^N)$ is the cost of mapping point i onto point j^1 in permutation π^1 , mapping point i onto point j^2 in permutation π^2 , and so on.

Given permutations $\pi^1, \dots, \pi^N \in \mathcal{P}(w, n)$ we define the cost of π^1, \dots, π^N to be

$$\text{Cost}(\pi^1, \dots, \pi^N) := \sum_{i=1}^n C(i, \pi^1(i), \dots, \pi^N(i)) \quad (8)$$

A solution to the N -MCLP problem is a choice of permutations π^1, \dots, π^N that minimises $\text{Cost}(\pi^1, \dots, \pi^N)$. \square

Example 2. We give an example instance of N -MCLP: let $N = 2$, $n = 5$, $w = 1$ and define the cost $C(i, j^1, j^2)$ of mapping point i onto point j^1 in permutation π^1 and mapping point i onto point j^2 in permutation π^2 to be

$$C(i, j^1, j^2) := i + j^1 + j^2 \pmod{4}$$

(Note that the interaction between j^1 and j^2 in the cost function stops us from trivially decomposing the problem into two MCLP problems.) The cost of taking both permutations to be the identity I is then

$$\begin{aligned} \text{Cost}(I, I) &= C(1, 1, 1) + C(2, 2, 2) + C(3, 3, 3) + C(4, 4, 4) + C(5, 5, 5) \\ &= 3 + 2 + 1 + 0 + 3 \\ &= 9 \end{aligned}$$

In this case several solutions achieve the minimum cost of 5, such as:

$$\begin{aligned} \pi^1 &:= \{1 \mapsto 2, 2 \mapsto 1, 3 \mapsto 3, 4 \mapsto 5, 5 \mapsto 4\} \\ \pi^2 &:= \{1 \mapsto 1, 2 \mapsto 2, 3 \mapsto 4, 4 \mapsto 3, 5 \mapsto 5\} \end{aligned} \quad (9)$$

which has cost

$$\begin{aligned} &C(1, 2, 1) + C(2, 1, 2) + C(3, 3, 4) + C(4, 5, 3) + C(5, 4, 5) \\ &= 0 + 1 + 2 + 0 + 2 = 5 \end{aligned}$$

\square

We will solve the N -MCLP problem using a graphical approach similar to the one we used for MCLP. Fig. 2 contains the graph we generate from the N -MCLP problem in Example 2, where $N = 2$.

This graph is similar to the one in Fig. 1, except that now each vertex is labelled with $N = 2$ availability sets rather than one; following an edge from a node (i, S^1, S^2) now corresponds to assigning values $\pi^1(i + 1)$ and $\pi^2(i + 1)$ to two permutations π^1 and π^2 rather than one permutation. Thus as one chooses a path from the “start” node on the left to the “end” node on the right, one simultaneously chooses N permutations. For space reasons, edge costs are not shown in Fig. 2.

We now make this construction precise and prove its correctness.

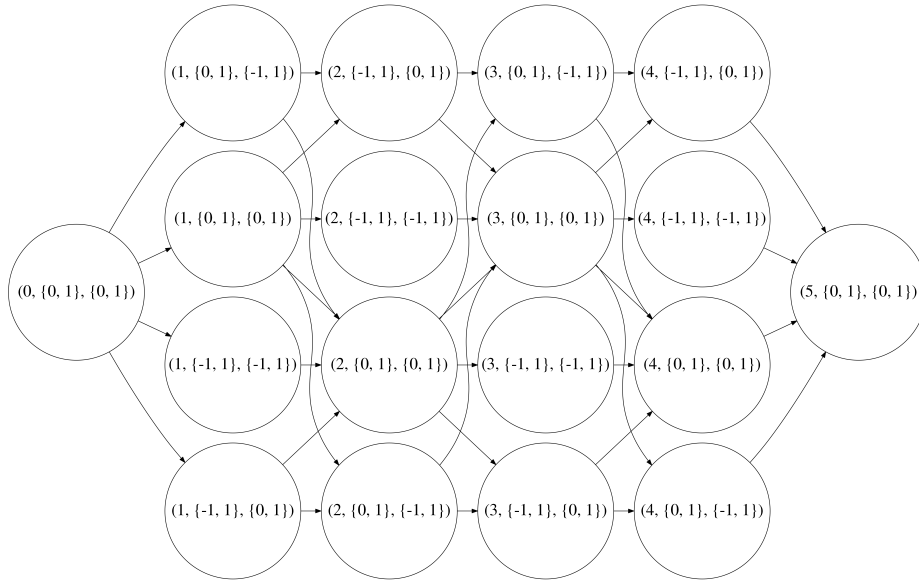


Fig. 2. The directed acyclic graph we use to solve the N -MCLP problem in Example 2. For space reasons, edge costs are not shown.

Definition 7. Given a problem instance N -MCLP(n, w, C), we define as follows a DAG $G_N(n, w, C)$. We take as vertices the elements of $\{0, \dots, n\} \times \mathcal{S}^N$. Let \leq_N be any total order on N -tuples of integers. Next we define a partial function T from pairs of vertices to $\{-w, \dots, w\}$:

$$T((i, S^1, \dots, S^N), (i', S^{1'}, \dots, S^{N'})) := \begin{cases} \min_{\leq_N}(X) & \text{if } i' = i + 1 \text{ and} \\ & X = \left\{ \begin{array}{l} (t^1, \dots, t^N) \mid \text{for all } j \in \{1, \dots, N\}, \\ \in (S^1, \dots, S^N) \mid \begin{array}{l} 1 \leq i+1+t^j \leq n \text{ and} \\ S^{j'} = \text{decrease}(S^j \setminus \{t^j\}) \cup \{w\} \end{array} \end{array} \right\} \\ & \text{is nonempty} \\ \perp & \text{otherwise} \end{cases}$$

where \min_{\leq_N} denotes minimum with respect to the total order \leq_N . Where $T(v, v')$ is defined, we write $T^j(v, v')$ for the j th component of $T(v, v')$. We put an edge from vertex $v = (i, S^1, \dots, S^N)$ to vertex $v' = (i', S^{1'}, \dots, S^{N'})$ if $T(v, v') \neq \perp$; we set the cost of the edge to $C(i+1, i+1+T^1(v, v'), \dots, i+1+T^N(v, v'))$. We define distinguished vertices $v_{\text{start}} := (0, S^+, \dots, S^+)$ and $v_{\text{end}} := (n, S^+, \dots, S^+)$. \square

Definition 8. Let $P = v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_n$ be a path from v_{start} to v_{end} in $G_N(n, w, C)$. Then we define corresponding functions $\pi_P^1, \dots, \pi_P^N : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ by $\pi_P^j(i) := i + T^j(v_{i-1}, v_i)$. \square

Theorem 2. Consider a problem instance $N\text{-MCLP}(n, w, C)$. Let P be a path of minimal length from v_{start} to v_{end} in the associated graph $G_N(n, w, C)$. Then $(\pi_P^1, \dots, \pi_P^N)$ is a solution to $\text{MCLP}(n, w, C)$. \square

We remark that, unlike in the case of MCLP, there is no obvious way of reducing $N\text{-MCLP}$ to the assignment problem; the above algorithm is the only method we know for solving MCLP problems besides exhaustive enumeration.

We conclude this section by proving Theorem 2. The theorem follows from three lemmas, which are the generalisations to N permutations of Lemmas 1, 2 and 3.

Lemma 4. Let (i, S^1, \dots, S^N) and $(i+1, S^{1'}, \dots, S^{N'})$ be vertices of $G_N(n, w, C)$, let $(t^1, \dots, t^N) \in S^1 \times \dots \times S^N$ and let $(t^{1'}, \dots, t^{N'}) \in S^{1'} \times \dots \times S^{N'}$ be such that for all $j \in \{1, \dots, N\}$:

- $1 \leq i+1+t^j \leq n$ and $S^{j'} = \text{decrease}(S^j \setminus \{t^j\}) \cup \{w\}$
- $1 \leq i+1+t^{j'} \leq n$ and $S^j = \text{decrease}(S^{j'} \setminus \{t^{j'}\}) \cup \{w\}$

Then $(t^1, \dots, t^N) = (t^{1'}, \dots, t^{N'})$.³

Proof. We use a simple adaptation to N permutations of the argument used to prove Lemma 1. For a contradiction, suppose $(t^1, \dots, t^N) \neq (t^{1'}, \dots, t^{N'})$. Then for some j , $t^j \neq t^{j'}$. We have $t^{j'} - 1 \notin \text{decrease}(S^j \setminus \{t^j\}) \cup \{w\}$ so $t^{j'} - 1 \notin S^{j'}$. But also $t^{j'} - 1 \in \text{decrease}(S^{j'} \setminus \{t^{j'}\}) \cup \{w\}$ so $t^{j'} - 1 \in S^j$. \square

Lemma 5 (from paths to local permutations). Let P be a path in $G_N(n, w, C)$ from v_{start} to v_{end} with length L . Then $\pi_P^1, \dots, \pi_P^N \in \mathcal{P}(w, n)$ and $\text{Cost}(\pi_P^1, \dots, \pi_P^N) = L$.

Proof. This lemma is proved with a simple adaptation to N permutations of the argument used to prove Lemma 2; the extra superscript index $j = 1, \dots, N$ for the N permutations propagates docilely through that proof. \square

Lemma 6 (from local permutations to paths). Let $\pi^1, \dots, \pi^N \in \mathcal{P}(w, n)$. Then there exists a path P_{π^1, \dots, π^N} in $G_N(n, w, C)$ from v_{start} to v_{end} with length $\text{Cost}(\pi^1, \dots, \pi^N)$.

Proof. The argument that we used to prove Lemma 3 applies *mutatis mutandis*; again the extra superscript index $j = 1, \dots, N$ for the N permutations propagates docilely through that proof. \square

³ Among other things, this result shows that the choice of order \leq_N in Definition 7 is irrelevant; \min_{\leq_N} is used only to provide a well-defined way of selecting an element from a non-empty set.

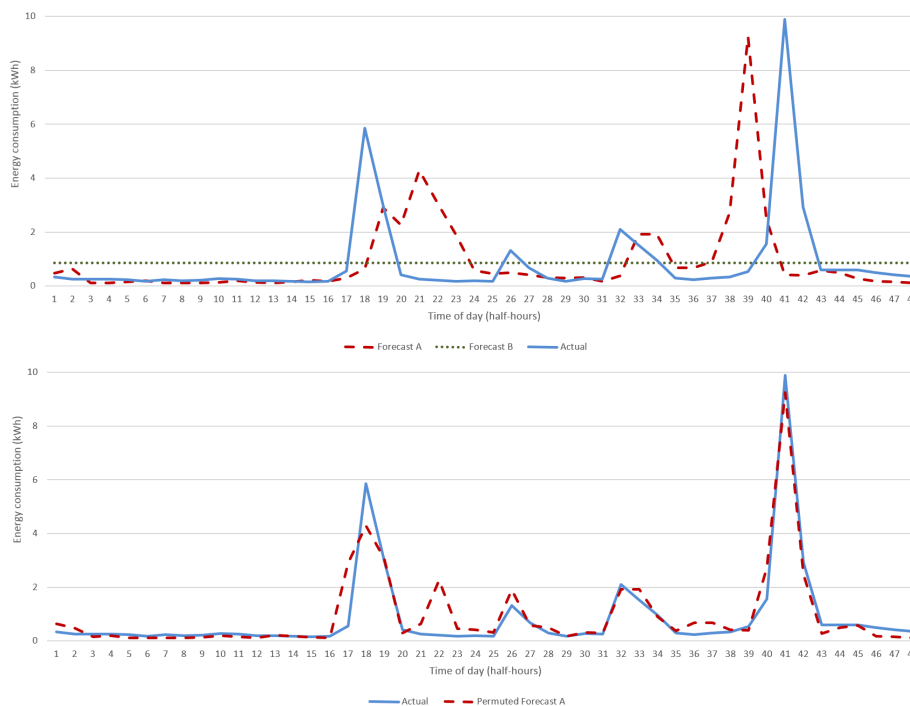


Fig. 3. (top) An informative Forecast A, a flat uninformative Forecast B and the subsequent actual energy consumption (bottom) A locally permuted version of Forecast A matching the actual energy consumption well.

5 Application to smart meter data

In this section we apply our graph-based methods to two problems arising in the field of household-level electricity use forecasting.

By an *energy use profile* we mean a vector \mathbf{x} of n non-negative real numbers (i.e. $\mathbf{x} \in \mathbb{R}_{\geq 0}^n$) where the components represent the (electrical) energy use of a household, sampled at n evenly spaced time points. Typically we work with daily profiles read from smart meters installed in houses, reading every hour or every 30 minutes. Common values of n are thus 24 and 48.

5.1 Computing adjusted errors: evaluating household-level electricity use forecasts

The forecasting of aggregated energy demand, such as at the regional or national level, has been well studied; demand at these levels is quite smooth and a plethora of forecasting techniques have been developed (see for example [8,12]). Such forecasts are typically assessed using RMSE (root mean square error), MAPE (mean absolute percentage error) or variants thereof.

Much less has been done for forecasting at the level of individual households, which appears to be a much harder problem: at the household level one has volatile, non-smooth load functions that are much harder to predict, similar to individual levels of natural gas consumption [1]. In fact, the volatile environment of household-level forecasting requires not only different forecasting techniques, but a different notion of what constitutes a good forecast.

Haben et al. [9] demonstrate that at the household level RMSE does a poor job of distinguishing good (useful) forecasts from poor (useless) forecasts. Fig. 3 (top) shows two forecasts A and B plotted against the subsequent actual energy consumption. As argued in [9], Forecast A is a good, informative forecast: it contains the right number of peaks in consumption, at approximately the right times and with approximately the right magnitudes. Yet under RMSE the flat Forecast B scores better, despite giving us no indication of the expected times and magnitudes of the peaks.

To address this, [9] proposed a new method, the *adjusted error*, of quantifying the similarity between two household-level load curves, typically a forecast and the actual usage. The idea of the adjusted error is to make allowances for small discrepancies in time that may be present between the forecast and actual profiles by allowing a w -local permutation to be applied to the forecast: we then consider a forecast good if, among all the possible w -local permutations of the forecast, there is one that is close to the actual consumption at each time point.

Fig.3 (bottom) shows a locally permuted version of Forecast A matched against the subsequent actual consumption. As they are very close, Forecast A will be given a small adjusted error. Because Forecast B is flat, no permuting of its components can bring it closer to the actual profile. We have taken $w = 3$, so that components of the forecast have been shifted forwards or backwards by no more than an hour and a half.

Formally, [9] defines the adjusted error $E_p^w(\mathbf{x}, \mathbf{y})$ between an actual profile \mathbf{x} and a forecast profile \mathbf{y} by

$$E_p^w(\mathbf{x}, \mathbf{y}) := \min_{\pi \in \mathcal{P}(w, n)} \left(\sum_{i=1}^n |(\pi(\mathbf{y}))_i - x_i|^p \right)^{1/p} \quad (10)$$

where $p \geq 1$. Here we overload notation by allowing permutations π to be applied to vectors of n components; if $\mathbf{x} = (x_1, \dots, x_n)$ is a vector of n reals, then $\pi(\mathbf{x})$ denotes the vector $(x_{\pi^{-1}(1)}, x_{\pi^{-1}(2)}, \dots, x_{\pi^{-1}(n)})$. Note that E_p^w is symmetric: for all \mathbf{x} and \mathbf{y} , $E_p^w(\mathbf{x}, \mathbf{y}) = E_p^w(\mathbf{y}, \mathbf{x})$.

The parameters w and p can be chosen differently for different applications. However if w is set too close to n , adjusted errors are unlikely to yield useful information about profile (dis)similarity: for example if we have daily profiles with $n = 24$, meaning hourly readings, and we set $w = 10$, then a peak in consumption at 8 AM in the first profile could be matched with a peak at 6 PM in the second profile. But even a small value such as $w = 2$ will usually be enough to match events such as cooking dinner, which tend to occur at approximately the same time each day. Thus in practice we are concerned about computing adjusted errors when w is much smaller than n . [9] recommends using $p = 4$ for

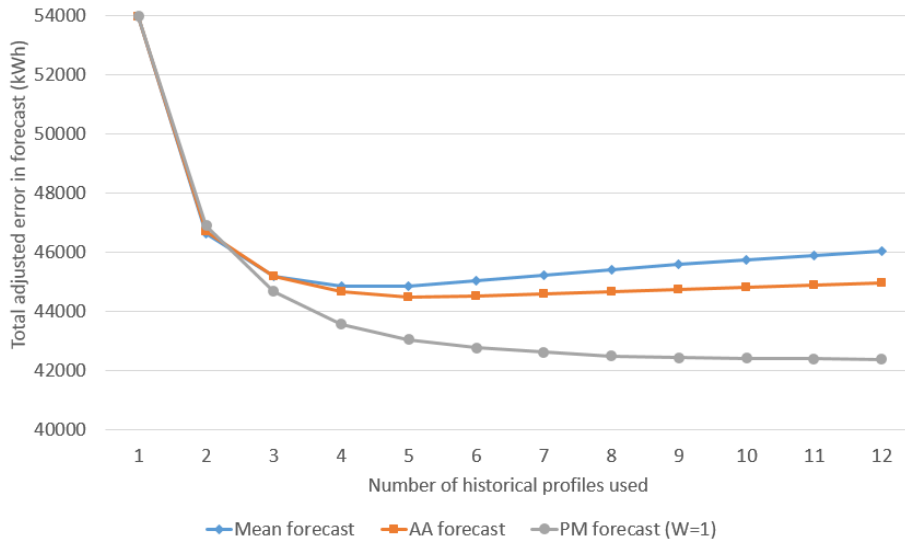


Fig. 4. Comparison of three household-level forecasting methods under the adjusted error measure (with $w = 3$ and $p = 4$). The PM forecast we have introduced performs better than the existing AA forecast [9] and a simple mean forecast.

smart storage control applications, so that a forecast with the right number of peaks but slightly wrong peak amplitudes and timings is preferred over a forecast which predicts exactly the right amplitude and timing for some of the peaks but completely misses another.

We now show how to formulate adjusted error computation as an instance of MCLP.

Remark 1. Given two profiles $\mathbf{x}, \mathbf{y} \in \mathbb{R}_{\geq 0}^n$ and an adjustment limit w , we form an instance of the MCLP problem by defining

$$C(i, j) := |y_i - x_j|^p$$

Then the adjusted error $E_p^w(\mathbf{x}, \mathbf{y})$ is equal to the cost of the solution to the MCLP instance raised to the power $1/p$. Thus we can use our algorithm for MCLP to compute adjusted errors. \square

5.2 Merging profiles

As well as computing adjusted errors quickly, we can use a generalisation of our graph-based method to *create* forecasts that perform well under the adjusted error measure. Our idea is as follows. Suppose we have energy use profiles $\mathbf{x}^1, \dots, \mathbf{x}^N$ for the previous N Tuesdays for a particular household, and we wish to make a forecast of that household’s energy use profile for next Tuesday, using some form of average over the previous N Tuesdays and ignoring for now temperature and seasonal effects.

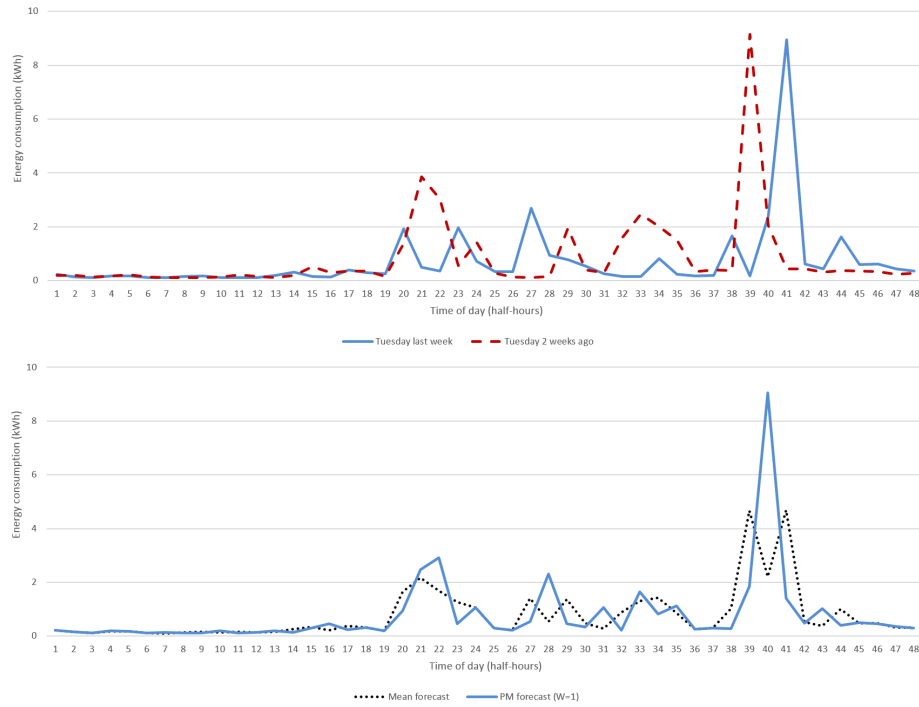


Fig. 5. (top) Actual energy use profiles from two recent Tuesdays (bottom) The mean forecast and PM (permutation merge) forecast for next Tuesday. The PM forecast is much more similar in shape to the recent actual profiles.

We will show shortly that using our algorithm for N -MCLP problems we can compute a profile \mathbf{y} that minimises the following criterion:

$$\sum_{j=1}^N (E_p^W(\mathbf{x}^j, \mathbf{y}))^p \quad (11)$$

for even p . Intuitively an optimal profile, which we denote \mathbf{y}^* , is one that is not too distant (under the adjusted error measure with an adjustment limit W) from any of the N historical profiles; because people often observe the same routine from one week to the next, the profile \mathbf{y}^* works well as a forecast, as we shall shortly demonstrate. We call this forecast the *permutation merge* (PM) forecast.

We emphasise that the adjustment limit W used in the PM forecast need not be the same as the adjustment limit w with which one intends to evaluate the forecasts.

We applied the PM forecast to real data from the Commission for Energy Regulation [4]. We took the 543 control households for which there is complete data for the 22 weeks from 3rd May 2010 to 3rd Oct 2010, and produced forecasts of the last 7 weeks of this period. (In Ireland the effect of temperature on demand

is small over this period.) Fig. 4 shows the results. The horizontal axis shows the number N of historical profiles (vectors) used to produce the forecast. The vertical axis shows the adjusted error in the forecasts in kWh using $w = 3$ and $p = 4$ summed over the 543 households and 7×7 days. Three forecasts are shown for N from 1 up to 12:

1. a simple mean forecast, where at time i we forecast the mean $(x_i^1 + \dots + x_i^N)/N$ of the load at time i in the historical profiles,
2. the *AA forecast* from [9], which is a forecast specifically designed to score well under the adjusted error metric, and
3. the PM forecast using $W = 1$.

Our key finding is that for $N \geq 3$ the PM forecast performs better than the other forecasts, and, unlike the others, continues to improve as more historical profiles are used. An investigation of which values of W and N give the best PM forecasts for the various values of w is future work, as is the incorporation of weather variables.

Fig. 5 shows the PM forecast in action. Fig. 5 (top) shows smart meter profiles for a particular household, from the last two Tuesdays. Fig. 5 (bottom) shows the mean forecast that would be generated from these two historical profiles, and the PM forecast using $W = 1$. Because the evening peaks in the two actual profiles occur one hour apart, the (pointwise) mean forecast does not strongly resemble either of the actuals: it contains two evening peaks, of about half the magnitude. On the other hand, the PM forecast effectively shifts the actual profiles' evening peaks half an hour forward and backward respectively, so that they coincide, *and then* averages them, producing a single evening peak at about the same time, with approximately the same magnitude. By construction, the PM forecast profile is close — in the E_4^1 sense — to both the actual profiles. In this way, the permutation merge method tends to produce better forecasts, as we confirmed in Fig. 4.

We now show how we compute the optimal profile \mathbf{y}^* that minimises the criterion (11). Let $p \geq 2$ be an even number. We form an N -MCLP problem by defining

$$C(k, j^1, \dots, j^N) := \min_{f \in \mathbb{R}} \sum_{i=1}^N (f - x_{j^i}^i)^p \quad (12)$$

This choice of C is well-defined and can be solved numerically; additionally if $p = 2$ or $p = 4$ then a minimum of the function $\sum_{i=1}^N (f - x_{j^i}^i)^p$ can be found symbolically by finding the roots of the derivative. Once we have the solution permutations π^1, \dots, π^N we construct our forecast vector \mathbf{f} by setting its k th component f_k as follows:

$$f_k := \arg \min_{f \in \mathbb{R}} \sum_{i=1}^N (f - x_{\pi^i(k)}^i)^p \quad (13)$$

The values of the components f_k can again be computed numerically, or symbolically in the case that $p = 2$ or $p = 4$. We now prove that the PM forecast does in fact minimise the criterion (11) as claimed.

Theorem 3. Given vectors $\mathbf{x}^1, \dots, \mathbf{x}^N \in \mathbb{R}_{\geq 0}^n$, the vector \mathbf{f} with components defined as in (13) minimises the criterion (11). \square

Before proving this theorem we establish three lemmas.

Lemma 7. Let π^1, \dots, π^N be W -local permutations and let C and \mathbf{f} be as defined in (12) and (13). Then

$$\text{Cost}(\pi^1, \dots, \pi^N) = \sum_{k=1}^n \sum_{i=1}^N (f_k - x_{\pi^i(k)}^i)^p$$

Proof. By definition,

$$\text{Cost}(\pi^1, \dots, \pi^N) = \sum_{k=1}^n C(k, \pi^1(k), \dots, \pi^N(k))$$

Substituting in the definition of C , we have

$$\text{Cost}(\pi^1, \dots, \pi^N) = \sum_{k=1}^n \min_{f \in \mathbb{R}} \sum_{i=1}^N (f - x_{\pi^i(k)}^i)^p$$

But each f_k is chosen exactly to minimise the inner sum, so we are done. \square

Lemma 8. Let $\sigma^1, \dots, \sigma^N$ be W -local permutations and let $y_1, \dots, y_n \in \mathbb{R}_{\geq 0}$. Then

$$\text{Cost}(\sigma^1, \dots, \sigma^N) \leq \sum_{k=1}^n \sum_{i=1}^N (y_k - x_{\sigma^i(k)}^i)^p$$

Proof. By definition,

$$\text{Cost}(\sigma^1, \dots, \sigma^N) = \sum_{k=1}^n C(k, \sigma^1(k), \dots, \sigma^N(k))$$

Substituting in the definition of C , we have

$$\text{Cost}(\sigma^1, \dots, \sigma^N) = \sum_{k=1}^n \min_{f \in \mathbb{R}} \sum_{i=1}^N (f - x_{\sigma^i(k)}^i)^p$$

and the result follows by inspection. \square

Lemma 9. Let π^1, \dots, π^N be W -local permutations and let C and \mathbf{f} be as defined in (12) and (13). Then

$$\sum_{i=1}^N (E_p^W(\mathbf{x}^i, \mathbf{f}))^p \leq \text{Cost}(\pi^1, \dots, \pi^N)$$

Proof. By Lemma 7, it suffices to prove

$$\sum_{i=1}^N (E_p^W(\mathbf{x}^i, \mathbf{f}))^p \leq \sum_{k=1}^n \sum_{i=1}^N (f_k - x_{\pi^i(k)}^i)^p \quad (14)$$

By the defining equation (10) for adjusted error, the left hand side of (14) is equal to

$$\sum_{i=1}^N \min_{\pi \in \mathcal{P}(W, n)} \sum_{k=1}^n (f_{\pi^{-1}(k)} - x_k^i)^p$$

The right hand side of (14) is equal to

$$\sum_{i=1}^N \sum_{k=1}^n (f_k - x_{\pi^i(k)}^i)^p$$

which is in turn equal to

$$\sum_{i=1}^N \sum_{k=1}^n (f_{(\pi^i)^{-1}(k)} - x_k^i)^p$$

Hence it will suffice to show

$$\sum_{i=1}^N \min_{\pi \in \mathcal{P}(W, n)} \sum_{k=1}^n (f_{\pi^{-1}(k)} - x_k^i)^p \leq \sum_{i=1}^N \sum_{k=1}^n (f_{(\pi^i)^{-1}(k)} - x_k^i)^p$$

and this is trivially true. \square

Now we can prove Theorem 3.

Proof (Theorem 3). Let π^1, \dots, π^N be a solution of the N -MCLP problem defined as in (12), and let \mathbf{f} be the forecast vector defined as in (13). Now suppose for a contradiction that there is some vector \mathbf{g} such that

$$\sum_{i=1}^N (E_p^W(\mathbf{x}^i, \mathbf{g}))^p < \sum_{i=1}^N (E_p^W(\mathbf{x}^i, \mathbf{f}))^p \quad (15)$$

By the definition of adjusted error, there exist W -local permutations $\sigma^1, \dots, \sigma^N$ such that

$$\sum_{i=1}^N \sum_{k=1}^n (g_{(\sigma^i)^{-1}(k)} - x_k^i)^p = \sum_{i=1}^N (E_p^W(\mathbf{x}^i, \mathbf{g}))^p$$

Rewriting the left hand side, we have

$$\sum_{i=1}^N \sum_{k=1}^n (g_k - x_{\sigma^i(k)}^i)^p = \sum_{i=1}^N (E_p^W(\mathbf{x}^i, \mathbf{g}))^p \quad (16)$$

Applying Lemma 8 we obtain

$$\text{Cost}(\sigma^1, \dots, \sigma^N) \leq \sum_{k=1}^n \sum_{i=1}^N (g_k - x_{\sigma^i(k)}^i)^p \quad (17)$$

Combining (15), (16) and (17) we get

$$\text{Cost}(\sigma^1, \dots, \sigma^N) < \sum_{i=1}^N (E_p^W(\mathbf{x}^i, \mathbf{f}))^p$$

From Lemma 9 we have

$$\sum_{i=1}^N (E_p^W(\mathbf{x}^i, \mathbf{f}))^p \leq \text{Cost}(\pi^1, \dots, \pi^N)$$

and therefore

$$\text{Cost}(\sigma^1, \dots, \sigma^N) < \text{Cost}(\pi^1, \dots, \pi^N)$$

Now we have a contradiction because π^1, \dots, π^N is a solution to the N -MCLP problem. \square

6 Analysis of running time

We start this section with a theoretical analysis of the time taken by our graph-based algorithm to solve the MCLP and N -MCLP problems, before reporting running times measured in practice.

6.1 Theoretical analysis

When we solve an MCLP problem by reducing to an instance of the problem of finding a shortest path in a DAG, we must do two things:

1. construct the appropriate DAG $G(n, w, C)$ as per Definition 4, and
2. solve the resulting shortest path problem.

Note that the DAG constructed always has a layered structure, as visible in Fig. 1: if we take all the nodes of the form (i, S) to be the i th layer of the graph, then all edges go from a layer j to the next layer $j + 1$.

Suppose we construct $G(n, w, C)$ by starting at layer 0 and generating each successive layer in turn. Each layer contains $\binom{2w}{w}$ nodes as noted in Definition 3. It follows from Definition 4 that each node has out-degree at most $w + 1$. Thus we can construct a layer from the previous layer in time proportional to $(w + 1)\binom{2w}{w}$; there are $n + 1$ layers so we can construct the whole graph in time proportional to $(n + 1)(w + 1)\binom{2w}{w}$.

Thanks to the layered structure of the graphs, a *topological sort* of the nodes is trivial to obtain: first list the nodes of form $(0, S)$, then those of form $(1, S)$,

then those of form $(2, S)$ and so on. Hence we can compute the shortest path from v_{start} to v_{end} in time $O(|E|)$, as detailed in [5, §24.2], where $|E|$ is the number of edges in the graph. But $|E| \leq (n+1)(w+1)\binom{2w}{w}$.

Thus solving an MCLP instance takes time proportional to $(n+1)(w+1)\binom{2w}{w}$. Using the inequality $\binom{2w}{w} \leq 4^w$ we have the following results.⁴

Corollary 1. *Our graph-based method solves the MCLP problem in $O(nw \cdot 4^w)$ time.* \square

Corollary 2. *Our graph-based method computes the adjusted error $E_p^w(\mathbf{x}, \mathbf{y})$ in time $O(nw \cdot 4^w)$.* \square

Existing work [9] has in effect reduced MCLP to the assignment problem, which can be solved in $O(n^3)$ using the Hungarian algorithm [11,13]. When n is large compared to w , as will be the case in our smart meter applications, our running time of $O(nw \cdot 4^w)$, which is linear in n , compares favourably to the $O(n^3)$ running time of the existing method. We explore running times in practice in the next section.

For the N -MCLP problem, each layer of the DAG $G_N(n, w, C)$ contains $\binom{2w}{w}^N$ nodes, and each node has out-degree at most $(w+1)^N$. Thus we can construct a layer from the previous layer in time proportional to $(w+1)^N \binom{2w}{w}^N$; there are $n+1$ layers so we can construct the whole graph in time proportional to $(n+1)(w+1)^N \binom{2w}{w}^N$. We can compute the shortest path from v_{start} to v_{end} in time $O(|E|)$, where $|E| \leq (n+1)(w+1)^N \binom{2w}{w}^N$. Thus solving an N -MCLP instance takes time proportional to $(n+1)(w+1)^N \binom{2w}{w}^N$. Using the inequality $\binom{2w}{w} \leq 4^w$ we have the following result.

Corollary 3. *Our graph-based method solves the N -MCLP problem in time $O(nw^N \cdot 4^{Nw})$.* \square

6.2 Running times in practice

We evaluated our algorithm for MCLP using real data from Ireland’s Commission for Energy Regulation [4]. We took 2,000 household-level electricity use profiles, with half-hourly readings (so $n = 48$) and computed the adjusted error between all pairs of these (resulting in 1,999,000 adjusted error computations). Table 1 compares the CPU time required for this task, using the Hungarian algorithm and using our new algorithm, for values of w from 1 to 6. CPU times reported are measured on a single core of a 2.2Ghz Intel PC, using C++ implementations.

The results show that our new algorithm is very much faster for small values of w . The running time of our algorithm grows more quickly than that of the Hungarian algorithm as w increases, so for sufficiently large w the Hungarian

⁴ Given a set of size $2w$ there are $\binom{2w}{w}$ ways of choosing a subset of size w , whereas there are $2^{2w} = 4^w$ ways of choosing an arbitrary subset. Hence $\binom{2w}{w} \leq 4^w$.

w	Adj. error computation		Clustering using PAM
	Our new algorithm	Hungarian algorithm	
1	1.0	1920.2	< 0.1
2	3.2	3114.0	< 0.1
3	12.2	4048.9	< 0.1
4	91.3	4796.5	< 0.1
5	885.9	5399.8	< 0.1
6	4906.8	5914.3	< 0.1

Table 1. Running times (in seconds) of our algorithm and the Hungarian algorithm computing adjusted errors between all pairs of 2,000 profiles, and of clustering the same 2,000 profiles into 5 clusters using the PAM technique.

algorithm will be faster; for $n = 48$ this happens at $w \geq 7$. Crucially however, as explained in Section 5.1, in practice w is small and our algorithm delivers orders-of-magnitude savings.

Although it has been termed an *error*, $E_p^w(\mathbf{x}, \mathbf{y})$ can be used as a measure of (dis)similarity between any two profiles; one need not be a forecast. We can, for instance, use adjusted error as one measure of profile dissimilarity when clustering profiles. Clustering of smart meter profiles has been proposed as a component in various smart grid management activities such as tariff design [3], targeting of behaviour modification initiatives [6] and improving short term load forecasts [7].

Table 1 also reports the CPU time required for clustering the 2,000 smart meter profiles into 5 clusters using the PAM technique [10]. To use PAM clustering, one defines a distance function between the objects being clustered (here profiles). We took the distance between profiles \mathbf{x} and \mathbf{y} to be a weighted sum of the following ingredients:

- the adjusted error $E_4^w(\mathbf{x}, \mathbf{y})$,
- the (absolute) difference between the largest component of \mathbf{x} and the largest component of \mathbf{y} ,
- the (absolute) difference between the smallest component of \mathbf{x} and the smallest component of \mathbf{y} ,
- the (absolute) difference between the mean component of \mathbf{x} and the mean component of \mathbf{y} .

Hence performing the 1,999,000 adjusted error computations was a necessary precursor to performing the clustering.

The CPU times required for the clustering step are tiny compared to those required for computing the adjusted errors. We include them to illustrate that in smart grid applications, the adjusted error computations can give rise to a substantial part (here the vast majority) of the computational burden, so our faster algorithm will be useful in practice.

7 Conclusions

We studied the *minimum cost local permutation* (MCLP) problem, which generalises the problem of computing the *adjusted error*, a measure of the similarity of two household-level smart meter energy profiles. We proved a reduction of the MCLP problem to that of computing the shortest path in a directed acyclic graph. This yielded an algorithm that, for a fixed adjustment limit w , solves MCLP in $O(n)$; this is better than the existing approach which used the Hungarian algorithm to solve MCLP in $O(n^3)$. We reported running times observed in practice for adjusted error computations on a real smart meter data set, and confirmed that our new method is far faster in practice.

We studied N -MCLP, a generalisation of MCLP where N permutations are chosen simultaneously to minimise the associated total cost. Again we proved a reduction to the problem of computing the shortest path in a directed acyclic graph.

Finally we considered the problem of computing an appropriate “average” of N smart meter profiles with respect to the adjusted error measure. We showed how to use our algorithm for N -MCLP to solve this problem, using the resulting method to produce substantially improved household-level energy consumption forecasts using real smart meter data.

References

1. Marek Brabec, Ondřej Konár, Emil Pelikán, and Marek Malý. A nonlinear mixed effects model for the prediction of natural gas consumption by individual customers. *International Journal of Forecasting*, 24(4):659 – 678, 2008.
2. Nathaniel Charlton, Danica Vukadinovic-Greetham, and Colin Singleton. Graph-based algorithms for comparison and prediction of household-level energy use profiles. In *Proceedings of IWIES (International Workshop on Intelligent Energy Systems)*, Vienna, 2013. To appear.
3. Gianfranco Chicco, Roberto Napoli, Petru Postolache, Mircea Scutariu, and Cornel Toader. Customer characterization options for improving the tariff offer. *Transactions On Power Systems*, 18(1), 2003.
4. Commission for Energy Regulation (CER). Smart metering electricity customer behaviour trials, 2007. Data available from Irish Social Science Data Archive, <http://www.ucd.ie/issda/data/commissionforenergyregulation/>.
5. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition*. The MIT Press and McGraw-Hill Book Company, 2001.
6. Ian Dent, Tony Craig, Uwe Aickelin, and Tom Rodden. Finding the creatures of habit; clustering households based on their flexibility in using electricity. In *Digital Futures 2012, Aberdeen, UK*, 2012.
7. M. Espinoza, C. Joye, R. Belmans, and B. De Moor. Short-term load forecasting, profile identification, and customer segmentation: A methodology based on periodic time series. *IEEE Transactions on Power Systems*, 20(3):1622–1630, 2005.
8. Eugene A. Feinberg and Dora Genethliou. Load forecasting. In Joe H. Chow, Felix F. Wu, and James Momoh, editors, *Applied Mathematics for Restructured*

- Electric Power Systems*, Power Electronics and Power Systems, pages 269–285. Springer US, 2005.
9. Stephen Haben, Jonathan Ward, Danica Vukadinovic-Greetham, Peter Grindrod, and Colin Singleton. A new error measure for forecasts of household-level, high resolution electrical energy consumption. Technical Report MPS-2012-14, University of Reading, 2012.
 10. L. Kaufman and P.J. Rousseeuw. Clustering by means of medoids. In Y. Dodge, editor, *Statistical Data Analysis Based on the L_1 -Norm and Related Methods*, pages 405–416. North-Holland, 1987.
 11. J. Munkres. Algorithms for the assignment and transportation problems. *J. SIAM*, 5:32–38, March 1957.
 12. James W. Taylor and Antoni Espasa. Energy forecasting. *International Journal of Forecasting*, 24(4):561 – 565, 2008.
 13. N. Tomizawa. On some techniques useful for solution of transportation network problems. *Networks*, 1:33–34, 1971.