Information Sciences 430–431 (2018) 179–199

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

User driven multi-criteria source selection

Edward Abel*, John Keane, Norman W. Paton, Alvaro A.A. Fernandes, Martin Koehler, Nikolaos Konstantinou, Julio Cesar Cortes Rios, Nurzety A. Azuan, Suzanne M. Embury

School of Computer Science, University of Manchester, Manchester M13 9PL, UK

ARTICLE INFO

Article history:
Received 26 April 2017
Revised 3 November 2017
Accepted 14 November 2017
Available online 11 December 2017

Keywords:
Source selection
Multi-criteria decision analysis
Multi-objective optimization
Information retrieval
Data science
Data wrangling

ABSTRACT

Source selection is the problem of identifying a subset of available data sources that best meet a user's needs. In this paper we propose a user-driven approach to source selection that seeks to identify sources that are most fit for purpose. The approach employs a decision support methodology to take account of a user's context, to allow end users to tune their preferences by specifying the relative importance between different criteria, looking to find a trade-off solution aligned with his/her preferences. The approach is extensible to incorporate diverse criteria, not drawn from a fixed set, and solutions can use a subset of the data from each selected source, rather than require that sources are used in their entirety or not at all.

The paper describes and motivates the approach, presenting a methodology for modelling a user's context, and its collection of optimisation algorithms for exploring the space of solutions, and compares and evaluates the resulting algorithms using multiple real world data sets. The experiments show how source selection results are produced that are attuned to each user's preferences, both with respect to overall weighted utility and through faithful representation of a user's preferences within a result, while scaling to potentially thousands of sources.

1. Introduction

There are increasingly many data sources; as well as individual organisations producing numerous data sources for their own internal purposes (e.g. [1]), further impetus comes from policy initiatives in open data, and technological developments such as web data extraction [2]. As a result, data scientists are often only interested in a subset of available sources. With potentially very large numbers of possible sources to choose data from it is clearly important that well-founded decisions are made as to which sources are selected. Moreover, the number and size of sources will only increase in the future as will the importance of such decisions. The most important criteria for informing source selection, and the relative importance between them, are likely to be specific to the user and to the use that is to be made of the data. With an ever increasing number of data sources there is a clear necessity to be able to capture a user's needs so as to be able to select data that will be most valuable for them, otherwise we run the risk of selecting data that is unsuitable and unusable for the user's

* Corresponding author.
E-mail address: edward.abel@manchester.ac.uk (E. Abel).

https://doi.org/10.1016/j.ins.2017.11.019

needs. In this paper we approach the problem of source selection through explicit consideration of the contextual needs of a user undertaking source selection tasks. In this way, different users with different priorities will seek and obtain different bespoke results, potentially in terms of different sets of criteria, reflecting what is most valuable and most fit-for-purpose for them.

Motivating example: Consider an example in the real estate domain. Assume an application that can retrieve data regarding housing from a plethora of sources. When using such an application, two users may have different requirements, reflected in their user contexts, which will result in different results being the best fit for purpose for each.

For example, a criterion of the quality of data relating to postcodes may vary from source to source, such that some sources have complete accurate postcodes whilst others only have rough area information, if any at all. The importance of the quality of location and postcode data will be different for different users. For example, for a user seeking results to compare to additional land registry data, exact full postcodes will be most suitable. Conversely, for another user seeking results to determine rough distances from the nearest motorway, partial postcode information is likely to suffice. Therefore, for these two users, different source selection results will constitute results that are most fit for purpose based upon their contextual needs. Moreover, within such an application the user context considerations will invariably be multi-faceted, with not just a single attribute of the data, such as location data quality, being of importance. For example, the quantity of data pertaining to when homes are available will again differ from source to source, and the importance of it will differ in importance from one user to another. Through tackling source selection with regards to multiple criteria the importance between the criteria to the user can be modelled.

Therefore, it is clear from this example that what represents suitable and valuable data will be different from one user to another, and significantly, it is only through modelling a user's contextual needs that source selection can seek to retrieve data that represents value to them. Through consideration of source selection as a multi-criteria problem that looks to capture the contextual needs of a user between such criteria, we aim to retrieve data from sources that are a better fit to a user's needs.

In this paper we define an approach to source selection that, firstly models a user's context, by eliciting their preferences relating to a set of criteria. Next, the user preferences are then used within an optimisation framework to search for the most suitable sources for the user, which represents a solution that is most fit for purpose to them. In this way the multi-criteria problem can be cast as a multi-dimensional optimization problem where criteria essentially act as dimensions to the optimization.

Our approach has the following features:

- *User context-driven and multidimensional* – The approach employs a decision support methodology to enable users to formalise their preferences by specifying the relative importance of different criteria. Users specify the relative priority to them between pairs of criteria, an approach that is widely used in decision support and for which tool support is available [3]. Through explicit modelling of user preferences, our approach considers results whose merit is explained not just with respect to overall utility, but also because they more closely match a user's preferences. This is in contrast to other work on source selection which does not explicitly consider a user's preferences [4]. Sources are then selected from the multi-criteria objective space to find a trade-off solution aligned with a user's preferences; in contrast to other results on source selection which have tended to map a multi-criteria space onto a single criterion for optimization [4], or to explore the multi-criteria space only via sampling without consideration of user preferences [5]. In our approach we preserve the multi-dimensional nature of the multi-criteria problem, utilising the user context to guide (and focus) the search. We introduce and contrast a variety of optimization strategies for addressing the multi-dimensional optimization problem. To the best of our knowledge there is no directly comparable approach that tackles source section as we do – as a multi-criteria problem that explicitly models user preferences between the criteria to be utilised within searching for a solution; hence, we evaluate between our optimization strategies and to various baseline and comparison strategies.
- *Group functionality* – The approach is able to handle scenarios where multiple users seek a single combined source selection result. For such scenarios each user is able to give his/her preferences regarding the criteria then, through the aggregation of their views, the approach seeks a single source selection solution that best meets all the users' needs. Moreover, the approach is able to handle differing weights of importance between the users, regarding the importance of their views upon the result.
- *Flexible and extensible* – Diverse criteria can be used to capture the user's data requirements; the only requirement in the current formulation is that criteria can be defined as linear functions.[1] Moreover, the criteria used are not drawn from a fixed set; the approach can use (generic and application-specific) quality metrics, and can accommodate performance metrics (such as availability) or cost metrics (relating to time or payment).
- *Fine-grained* – A solution can be fine-grained through the use of a subset of the data from each selected source, rather than requiring that sources are used in their entirety or not at all. The focus on complete sources in existing methods

[1] Defining criteria via linear functions trades-off model complexity and performance, as well as usability and interpretability. It is the case that there are criteria which might be more satisfactory modelled through non-linear functions however, multiple real-world experiment examples later show how we are able to satisfactorily model many criteria via linear functions. Moreover, such functions facilitate finding optimal solutions rapidly, whereas for non-linear functions neither optional solutions nor swift performance are guaranteed.

[5], reduces the potential for the search result to closely match a user's requirements, especially in the presence of large sources.

In addressing source selection, we make the following contributions:

(1) We describe a multi-criteria based approach to source selection that is *user context-driven, flexible and extensible* and *fine-grained* that can also facilitate finding a single result for a group of users.
(2) We introduce and contrast a variety of strategies to addressing the multi-dimensional optimization problem in (1).
(3) We empirically evaluate the strategies in (2) on source selection problems involving multiple real world data sets, exploring both the properties of the search results and their scalability, by defining and exploring multiple evaluation metrics.

The rest of the paper is structured as follows related work is presented in Section 2. Section 3 discusses technical background; Section 4 defines and describes our source selection approach; experimental results and evaluation are presented in Section 5; and Section 6 concludes.

2. Related work

We tackle source selection as a multi-criteria problem, looking to explicitly model a user's preferences between criteria, to direct the source selection process towards a solution that most meets user needs. Furthermore, our approach is flexible and extensible, regarding the possible criteria that can be utilised, whilst also being fine-grained, as it may choose to use all or part of each source.

Work on source selection has mostly focused upon a single criterion aspect of the data. In [4], a subset of sources is selected to be utilised in data integration, with respect to the single criterion of marginal gain – looking to balance the quality of data with the cost of integration. Sources are selected in their entirety and the focus is on data fusion (integration of data representing the same real-world object), looking to select a subset of sources which yields highest profit.

Similarly, the approach in [6] focuses on a single aspect, that of coverage, through analysis of coverage between sources. The approach looks to successively select sources, each in its entirety, such that at each selection the gain in residual coverage is highest. Considering as input precise coverage information for only a few subsets of sources, the approach looks to estimate complete coverage information between sources. Coverage between sources has also been considered in [7]. Here sources are organised into hierarchies to seek to estimate coverage with respect to the set of classes for which each source is a part. Within these approaches the focus is only on coverage without considering the multi-criteria nature of source selection and user preferences between criteria.

Previous work that considers multiple criteria does not look to model user preferences between the criteria. In [8] the problem of dynamic sources whose content changes over time is addressed, to select a sub-set of sources with the highest perceived gain. To define gain, a fixed set of quality criteria are considered: coverage (the amount of the total population of objects present in sources), accuracy and freshness (the level of up-to-date data in sources), to evaluate and differentiate between sources. Multiple criteria of the data are considered, without consideration of user preferences between the criteria and how such preferences could affect a result.

The demo system SourceSight [5,9] does consider source selection as a multi-criteria problem and is the most related to our work. A vision of a data source management system is outlined along with a fixed set of data quality criteria, from which a user can select a subset to utilise. The relative importance of the criteria is not elicited from the user or utilised to help guide the selection process. Instead, the system seeks to provide a user with multiple trade-off solutions, between which they can compare. As finding a comprehensive set of trade-off solutions in a multi-objective space may not be straightforward, the approach seeks to find a set of trade-off solutions through sampling different weighting configurations between the criteria via a weighted linear combination of the set of metrics. Without prior consideration of a user's preferences between the criteria, effort may be spent calculating trade-off solutions which are of little user relevance. Furthermore, even sampling of weights may not result in similar even spacing between solutions within the objective space, resulting in an uneven spread of sample solutions presented to a user.

Other work on source selection has significantly different emphases. For example, proposals have been made to use queries over source metadata to identify candidate sources [10], to use social profile information for source selection within distributed information retrieval (IR) environments [11], to use sampling to locate the most relevant sources within uncooperative distributed IR environments [12], and to use fuzzy decision support methods to identify a single source by aggregating a group of decision makers' preferences [13].

3. Technical background

The approach in this paper is inspired by Multi-Criteria Decision Analysis (MCDA) for modelling source selection as a multi-criteria problem and eliciting user's preferences between the criteria. Generally, an MCDA problem seeks to determine the suitability of alternative outcomes to a goal evaluated with respect to a set of criteria. A criterion is a factor against which an alternative can be evaluated. Within source selection the decision as to which data from which sources to use can be evaluated with respect to multiple criteria relating to properties of data quality, such as those introduced in our

Table 1
1–9 numerical scale mapping from verbal scale.

Verbal preference strength	Numerical
Equal importance	1
Weak or slight importance	2
Moderate Importance	3
Moderate plus	4
Strong Importance	5
Strong plus	6
Very strong importance	7
Very very strong	8
Extreme importance	9

motivating example. The relative importance of each criterion to a given user will typically be different. Criteria weights, numerical values denoting importance, can be used to define the significance of each criterion to a user.

Pairwise Comparison (PC) is employed in multiple MCDA methodologies (e.g. [14,15]) for systematic and transparent elicitation of preferences over a set of criteria. PC enables consideration of a pair of criteria at a time, to allow a user to define their preference, and strength of preference, between the pair. Allowing a user to consider only a pair of elements at a time induces a separation of concerns that helps to achieve an accurate reflection of user preferences [16].

Given a pair of criteria, the user is asked which is most important and, through a verbal scale of descriptive importance, by how much. Using a verbal scale is intuitively appealing and user friendly [17]. Various numerical scales may be used to represent the strength of preference; the most widely used being the Saaty 1–9 scale [18]. Table 1 gives an example of definitions of a verbal scale, the 1–9 numerical scale, and the mapping between them.

The set of PCs, one for each pair of criteria (along with self-comparison values and the reciprocal values) can be collated into a two-dimensional Pairwise Comparison Matrix (PCM), as shown in (1) for a set of n elements, where a_{ij} represents a PC between elements i and j.

$$PCM = \begin{pmatrix} 1 & a_{12} & \cdots & a_{1n} \\ 1/a_{12} & 1 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \\ 1/a_{1n} & 1/a_{2n} & & 1 \end{pmatrix} \tag{1}$$

Moreover, a PCM and its properties can be visualised as a Directed Acyclic Graph (DAG), as in Fig. 2. In such, each criterion is represented as a graph node and each judgement is depicted by a directed arc from the preferred criterion to the other. Equally preferred elements can be shown via an undirected arc. Each arc is labelled with the judgement preference strength value. To aid graph clarity self-comparisons and reciprocal judgements are usually omitted.

From a completed PCM of the type (1) of (n x n) elements a *preference vector* - $w = [w_1, w_2 \ldots w_n]^T$ - that assigns weights to criteria, can be derived via use of a prioritization method (where w_i represents the weighting of the element i for $i = 1$–n). Many prioritization methods exist for this task (see [19] for a comprehensive discussion). In our approach we utilise the Geometric Mean (GM) prioritization method to derive a preference vector, as it can be swiftly calculated and is widely used in the MCDA community [20]. The GM method calculates a preference vector via the product of each row of the PCM raised to the inverse power of n (2). These weights are then normalized to sum to 1.[2]

$$w_i' = \prod_{j=1}^{n} a_{ij}^{1/n} \tag{2}$$

for $i = 1, 2 \ldots n$

$$w_i = \frac{w_i'}{\sum_{i=1}^{|n|} w_i'} \tag{3}$$

for $i = 1, 2 \ldots n$

4. Our approach

We tackle the problem of source selection taking inspiration from MCDA and supplier selection optimization. Various work has applied MCDA to supplier selection problems, see [21,22]. In (multiple source) supplier selection, with multiple suppliers each supplying multiple products, the problem is to identify which suppliers to select and the quantity of each product to order from each. For such problems, there are various criteria by which suppliers can be evaluated, such as

[2] Hence, due to the set of criteria weights summing to 1, each weight is fractional in nature.

Table 2
Source selection notation.

Notation	Definition
Decision variables	
Q	Integer vector of length S of the numbers of tuples to choose from each source
q_i	Integer quantity of tuples chosen from source i
Parameters	
S	Number of sources
B	Total number of tuples to select across all sources.
$Size_i$	The number of tuples in source i.
C	Number of criteria
c_{ji}	Measurement of quality of criterion j for source i
Multi-Objective Parameters	
W	Vector of weights of size C representing the user preferences regarding the criteria
w_i	Weight of criterion i
Z_i	Value of objective function for criterion i
Z_i^*	Ideal solution of the objective function Z_i
Z_i^{**}	Negative ideal solution of the objective function Z_i
D_i	Deviation of criteria i from its ideal solution as ratio between its ideal and negative ideal solution
MD	Maximum deviation value in the set of C deviation values

product prices and delivery times. Whilst not an exact analog, source selection can be seen as similar as it seeks to determine which sources to choose and how much data to choose from each. MCDA has been applied to such supplier selection problems to model multiple criteria and their relative importance, see [23,24]. Within our approach, the selection process is guided by first modelling the user context regarding the relative importance, to the user, of the set of problem criteria. This elicited information, along with information relating to the sources regarding the set of criteria of the problem, is then utilised within an optimization model to determine a source selection result.

4.1. Problem definition

We define the problem of source selection as: given a set of possible sources S to retrieve data from and a total, defined by a user, of the amount of data to retrieve B, we select a subset of sources along with the quantities of data to select from each selected source. Further, we conjecture that various criteria against which sources can be evaluated will have varying degrees of importance to a user and that the preferences between these criteria are part of the problem. Given a set of C of criteria, then a set of weights $W = [w_1, w_2, \ldots, w_3]$ can model the user's preferences between the set of criteria. A source selection result is an integer vector of length $|S|$ representing the quantity values of tuples to choose from each source (where 0 denotes a source that is not selected).

In order to evaluate and compare solutions we define metrics for evaluating a source selection solution. These characterize (a) the solution's overall utility and (b) the degree to which the solution matches the user's stated preferences. The notation of the problem and our approach is shown in Table 2.

4.2. Evaluation metrics

4.2.1. Overall Weighted Utility (WU)

Given a source selection solution, for each criterion its average weighted utility can be determined, with respect to the range of possible values the criterion can take. In this way normalisation is performed with respect to the range of possible values a criterion can take, therefore allowing aggregation of different criteria values. The normalised value, the Selection Utility (SU), of the quantity q_i from source i for criterion j is calculated via:

$$SU_{ji} = \frac{q_i(c_{ji} - Z_j^{**})}{Z_j^* - Z_j^{**}} \tag{4}$$

where Z_j^* is the ideal solution for criterion j, Z_j^{**} the negative ideal solution for criterion j and c_{ji} the quality measure of criterion j for source i. Each ideal solution represents the possible optimal solution for a single criterion (given the amount of data to retrieve). Similarly, each negative ideal solution represents the worst possible solution for a single criterion.

Then the Criterion average Weighted Utility (CWU) for criterion j is calculated as follows:

$$CWU_j = \frac{W_j(\sum_{i=1}^{S} SU_{ji})}{B} \tag{5}$$

where W_j is the weight of criterion j and B is the total number of tuples that is selected.

The Overall Weighted Utility (WU) for a solution is then calculated via:

$$WU = \frac{\sum_{i=1}^{C} CWU_i}{C} \tag{6}$$

Table 3
Illustrative example data.

	Equal Weights			W=[2/3, 1/3]			W=[1/3, 2/3]		
	C1	C2	Range	C1	C2	Range	C1	C2	Range
Solution 1	0.7	0.35	0.35	0.93	0.23	0.6	0.47	0.47	0.0
Solution 2	0.4	0.4	0.0	0.53	0.27	0.26	0.27	0.53	0.26
Solution 3	0.38	0.75	0.37	0.5	0.5	0.0	0.25	1	0.75

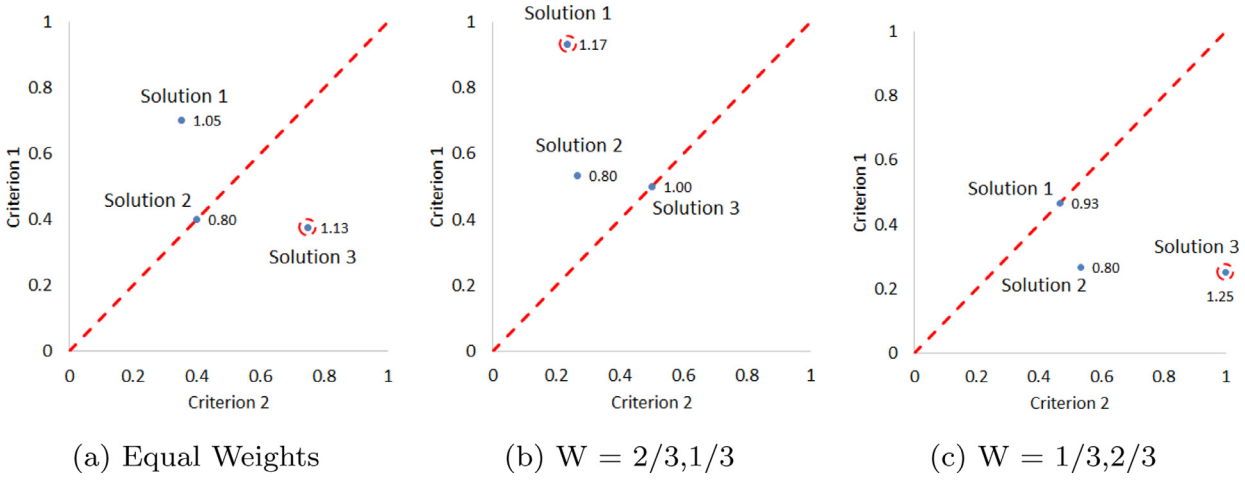

Fig. 1. Illustrative example solutions' *WU* and *Range* comparisons.

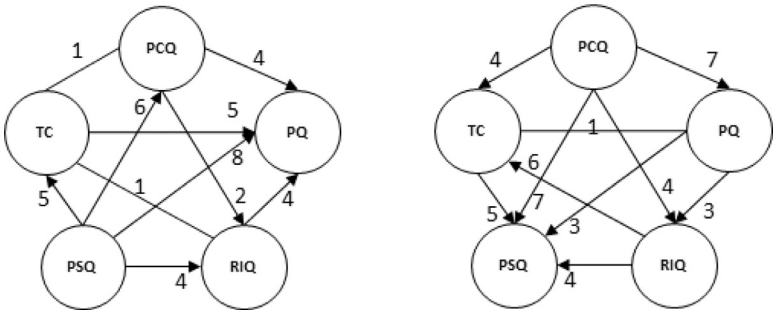

Fig. 2. Left: User1 judgements as a DAG, Right: User2 judgements as a DAG.

We can compare the *WU* of solutions. For example, data for 3 illustrative solutions for a 2 criterion problem are shown in Table 3. Considering equal weights first, the values of the 2 criteria for 3 solutions are shown. Here, for example, the 0.7 value of criterion 1 for solution 1 denotes that the CWU_1 value of the solution is almost 3 quarters towards the ideal solution for the criterion in relation to the range of values between the criterion's ideal and negative ideal solutions. We can plot these three solutions in the objective space as shown in Fig. 1(a). The *WU* of each solution is given next to each solution and the solution with the highest, solution 3, is denoted via a dotted circle. Furthermore, Table 3 details the Weighted utility values for the 3 solutions under two different user weights for the two criteria, the plots of which are shown in Fig. 1(b) and (c) respectively. From this example, the impact of changing weights upon the solutions' *WU* can be observed and the solution with the greatest overall utility can be identified.

4.2.2. Faithful preference mapping - Range

Given a source selection solution, we can also look to discern how faithfully the relative importance between the criteria to the user has been maintained within the solution. In a faithful mapping of user preferences, the Weighted compromise for each criterion, with respect to the compromise from its ideal value, should be as proportionally close as possible to the user's weights.

We can determine, for each criterion, the amount of weighted compromise, regarding that criterion's range of possible values, that a solution entails. The compromise, as a weighted measure of Deviation (D), for each criterion can be calculated. The deviation for a criterion therefore represents a measure of the amount a solution is from the ideal solution for that criterion, as a ratio of its distance from the ideal solution within the range between the criterion's ideal and negative ideal

Table 4
Property dataset criteria definitions.

Criterion	Definition
1.TC	$= (\sum_{i=1}^{S}$ Non Null Attributes$)$/No. Of Attributes
2.PCQ	$= \begin{cases} \text{Full Valid Postcode} = 1 \\ \text{Valid Outcode} = 0.5 \\ \text{Valid UK City} = 0.2 \\ \text{other/null} = 0 \end{cases}$
3.PQ	$= \begin{cases} \text{Valid Numeric Price} = 1 \\ \text{POA (Price On Application)} = 0.4 \\ \text{other/null} = 0 \end{cases}$
4. RIQ	$= \begin{cases} \text{Number of bedrooms} + 0.5 \\ \text{Number of bathrooms} + 0.3 \\ \text{Number of receptionrooms} + 0.2 \end{cases}$
5. PSQ	$= \begin{cases} \text{Specified Available Date} = 1 \\ \text{Available but unspecified when} = 0.5 \\ \text{other/null} = 0 \end{cases}$

solution. The D for criterion j can be determined via:

$$D_j = \frac{w_j\left(Z_j^* - \frac{\sum_{i=1}^{|S|} q_i \cdot c_{ij}}{B}\right)}{(Z_j^* - Z_j^{**})} \tag{7}$$

where Z_j^* is the ideal solution for criterion j, Z_j^{**} the negative ideal solution for criterion j, W_j the weight of criterion j and c_{ji} the quality measure of criterion j for source i.

The faithfulness of a solution to the user's preferences can then be determined as the *Range* between the largest weighted deviation value from the set of calculated weighted deviation values and the smallest via:

$$Range = Max\{D_1, D_2, \ldots, D_c\} - Min\{D_1, D_2, \ldots, D_c\} \tag{8}$$

Therefore, a solution's range is a weight adjusted measure of the amount of divergence within the set of criteria deviations of a solution. Comparisons between solutions regarding their ability to faithfully map user preferences into the solution can be done through comparing their Range values. Range will be a value between 0 and 1; where the lower the value the faithfully proportionally represented a user's preferences are in a solution.

Returning to the illustrative source selection solutions presented in Table 3, the *Range* values for the 3 solutions, shown in Table 3, can be plotted in the objective space such that the vector from the origin dividing the objective space in two represents the plane for which the weighted range between criteria would be equal. In Fig. 1(a)–(c) this vector allows the most faithful representation of the weights solution to be determined; through comparing how close to this vector each solution is. From these plots, we can observe the changing ranges for the solutions under the different weight configurations and how the solution with the lowest range can be identified.

The plots in Fig. 1 highlight the trade-off between seeking the highest utility solution and the most faithful preference maintaining solution. Clearly overall utility and range are desirable properties of a solution to a source selection problem, and considering both should lead to more satisfactory solutions.

4.3. Preference elicitation

As we saw in the motivating example, a source selection problem inherently involves multi-criteria, and when expressed in this way representable as a MCDA problem. Given a problem with C criteria, we use PC to elicit judgements from the user, about each pair of criteria, i.e. which of the pair they see as most important and by how much, in terms of the 1–9 scale. From the set of PC, one for each pair of criteria, a preference vector ranking of the weights of the criteria can be derived. To illustrate the preference election stage we return to our motivating example. We define a set of five representative criteria against which a set of real estate sources can be evaluated. The set of criteria is inspired by quality views [25] which advocate the use of quality measures that may be problem bespoke. The criteria are: (1) Tuple (row) Completeness (TC), (2) Post Code Quality (PCQ), (3) Price Quality (PQ), (4) Room Information Quality (RIQ) and (5) Property Status Quality (PSQ). These are formalized in Table 4.

Assume two users, User1 and User2. For this set of criteria we firstly elicit a PC from User1 of his/her preferences regarding each pair. For example, taking tuple completeness and postcode quality, User1 may decide that these criteria are of equal importance, so the PC judgement between the pair is 1. Next, taking the importance between tuple completeness and price quality User1 may decide that tuple completeness has strong importance over price quality, so the PC judgement between the pair is 4. After eliciting PCs in a similar manner for the rest of the criteria pairs we can show the set of

Table 5
Example user preference vectors.

	TC	PCQ	PQ	RIQ	PSQ
User1	0.14	0.15	0.04	0.12	0.55
User2	0.10	0.54	0.15	0.16	0.04

judgements as a DAG (see Fig. 2:Left). From the set of PCs a preference vector ranking can be calculated for User1 as shown in Table 5. From this we see that, for User1, the quality of data relating to a property's availability status is the most important criterion. We can similarly elicit preferences from User2 between each pair of criteria (show as DAG in Fig. 2:Right) and calculate a preference vector, also shown in Table 5. From Table 5 we observe that for User2 postcode quality is most important, with property status quality being of little importance.

A user's preferences can then be utilised within the optimization stage to search for a source selection result that best meets user needs.

4.4. Optimization strategies

For a set of sources, given a set of criteria, typically some sources contain high quality data according to some criteria but lower quality according to other criteria. Therefore, any source selection result involves a trade-off.

In such a multi-criteria scenario, cast as a multi-dimensional optimization problem, a range of possible trade-off solutions exist. For each solution, any improvement in one objective will result in a decrease in one or more of the other objectives. Discovering all possible trade-off solutions is expensive [5], whilst also resulting in a potentially large set of solutions to explore [26]. Moreover, due to a user's preferences between the criteria, seeking to find a mapping of the whole trade-off front may result in time spent searching within regions of the objective space of little importance to the user. Therefore, we aim to efficiently find a single trade-off solution by considering user preferences.

Next, various optimization strategies of our approach are defined. Here the strategies are explained and their properties discussed. The experiments that compare the various strategies and their properties are presented in Section 4. The notation of the metrics and approaches is shown in Table 2.

4.4.1. Maximizing weighted utility - MinSum optimization model

MinSum considers a user's preferences in determining a trade-off solution that has high overall weighted utility (without explicit consideration of the *Range* for the solution). Such an optimization model is implemented here using Multi-Objective Linear Programming (MOLP). First, the model finds ideal and negative ideal solutions for each criterion, then uses this information, along with a user's preference vector of criteria weights, to find a trade-off solution.

The model finds the ideal solution Z^* for each criterion separately via single objective optimisation; each represents an optimal solution for a single criterion (given the amount of data to retrieve and the model constraints outlined below); similarly, the negative ideal solution Z^{**} for each criterion is found in the same way. Each represents the worst possible solution for a criterion.

To find these ideal and negative ideal values, each criterion is optimised with respect to a single objective function (Z) defined, for criterion j as:

$$Z_j = \frac{\left(\sum_{i=1}^{|S|}(q_i \cdot c_{ji})\right)}{B} \tag{9}$$

where c_{ji} represents the quality value of criteria j for source i.

Such an objective function can be efficiently solved optimally in both the minimisation and maximisation cases. Such a solution represents the set of B tuples for which the average quality for a single criterion is optimal (or negatively optimal when minimised).

A set of C such objective functions, one for each criterion, is solved both as maximisation and as minimization objective functions constrained by:

1. *Source size constraints* – the quantity of tuples chosen from source i, q_i, cannot be greater than its total size

$$q_i <= Size_i \tag{10}$$

2. *Total data constraint* – the total quantity of tuples chosen must equal the amount requested by the user

$$\sum_{i=1}^{|S|} q_i = B \tag{11}$$

3. *Decision variable non-negativity constraints* – each quantity value has a non-negativity restriction

$$q_i >= 0 \tag{12}$$

4. *Overlap* – When data across sources overlaps then our approach has a strategy to tackle overlap during optimization, whether information regarding the overlap is available or not, as discussed in Section 4.5.

From this a set of ideal values and a set of negative ideal values are calculated for each criterion, that define the upper and lower boundaries of the ranges of values the criterion could take, given the sources and the required amount of tuples. In the MinSun model this information, along with user preference weights, is used to find a solution that minimises the sum of the set of criteria deviations. Each criterion deviation is a measure of the compromise for a solution in relation to each criterion's ideal value, weight adjusted to reflect a user's preferences. The MinSum model objective function optimises:

$$min \sum_{j=1}^{|C|} D_j \tag{13}$$

where D_j represents the weighted deviation value for criterion j, see (14). The model is solved subject to the set of constraints (10)–(12) along with an additional set of constraints to determine the deviation for each criterion. The additional constraint for criterion j is calculated via:

$$\frac{\sum_{i=1}^{|S|} q_i \cdot c_{ij}}{B} + \frac{D_j \cdot (Z_j^* - Z_j^{**})}{w_j} = Z_j^* \tag{14}$$

where w_j is the user's weight of criteria j.

4.4.2. Faithful preference mapping - Minimum Range optimization

Alternatively, optimization specifically seeking a faithful mapping of user preferences could be utilised. Such a model – the MinRange optimization model – seeks to determine a trade-off solution with the smallest Range without consideration of the overall weighted utility of the solution. The MinRange model, similar to the MinSum model, first finds the sets of ideal and negative ideal solutions for each criterion via single objective optimization. This information along with user weights is then utilised within the MinRange optimisation. The MinRange model objective function is:

$$min[Max\{D_1, D_2, \ldots, D_c\} - Min\{D_1, D_2, \ldots, D_c\}] \tag{15}$$

where D_i represents the weighted deviation value for criterion i, see (14).

The MinRange model is solved subject to the set of constraints as defined in (10)–(12) along with an additional set of C constraints defined to determine the deviation for each criterion, as defined in (14).

4.4.3. Utility and faithful preference mapping - MinMax optimization

The MinSum and MinRange models seek a solution with a focus on either high overall utility or faithful mapping of user preferences respectively. As previously discussed, both overall utility and Range are desirable properties of a solution to a source selection problem, and consideration of both should lead to more satisfactory solutions. Therefore, an alternative optimization strategy – the MinMax optimization model – seeks to minimise the maximum weighted deviation of the set of criteria, to look for a solution that is both of high utility and faithful to the user's preferences. The MinMax model first finds the sets of ideal and negative ideal solutions for each criterion via single objective optimization. This information along with user weights is then utilised within the MinMax optimisation. The MinMax model objective function seeks to minimise the Maximum Deviation value MD.

The MinMax model is solved subject to the set of constraints defined in (10)-(12), along with an additional set of C constraints, one for each criterion, to determine the maximum deviation. The additional constraint for criterion j is:

$$\frac{(\sum_{i=1}^{|S|} q_i \cdot c_{ij})}{B} + \frac{MD(Z_j^* - Z_j^{**})}{w_j} >= Z_j^* \tag{16}$$

where w_j is the user's weight for criteria j.

4.4.4. Optimization strategies summary

Within our approach a suite of optimisation models (MinMax, MinSum and MinRange), are available to enable flexibility in various scenarios. Tackling the problem via these optimization approaches allows our approach to (a) select a subset of data from each source, which sometimes may yield a more appropriate set of tuples, (b) consider the possible ideal achievable values of each criterion in a solution, and consequently be able to establish, e.g., that a seemingly low value for a criterion is actually a high value in relation to the best possible value for that criterion, and (c) find solutions that lie at non-turning points within the objective space, which any weighted sum approach cannot.[3]

[3] A turning point is a point upon the trade-off front of the objective space at which the gradient for the objective function changes within the multi-dimensional objective space.

4.5. Overlap

An important consideration is potential overlap between data within sources. Our approach can consider overlap between sources whether information regarding some or all of overlap between sources is present or not.

When any information regarding overlap across sources is available then additional constraints regarding overlap can be added to the optimization strategies to ensure a selected solution is valid with respect to overlap information. When there is overlap between sources then the total number of tuples chosen from the set of sources cannot exceed the aggregated total number of distinct tuples. For example, given a level of overlap between sources i and j defined as ol_{ij}, then the maximum total quantity from the two sources is constrained, and added to our approach's optimization models:

$$q_i + q_j <= Size_i + Size_j - ol_{ij} \tag{17}$$

Alternatively, without information relating to overlap, our approach can find a solution that is valid with respect to overlap via iterative optimization. Given no information regarding overlap, an initial optimization solution can be found and its validity regarding its overlap determined through calculation of the number of distinct tuples it contains.[4] Then, any discrepancy between the number of distinct tuples selected and the value of B represents the amount of *deficit* due to overlap in the initial solution. After such initial optimization, the sizes of the sources from which tuples were selected can be altered to reflect the new state, the number of requested tuples set to the deficit and the optimization process re-run. In this way the approach converges to a deficit of 0 and determines a solution that is valid with respect to overlap, without any overlap information supplied. Pseudo-code of the 'overlap' algorithm is given in Algorithm 1. See Section 5.4 for experimentation regarding the overlap strategy's convergence and scalability.

Algorithm 1 Algorithm for overlap.

```
while number of requested tuples not reached do
    Run optimization process
    Select tuples from suggested sources
    Evaluate deficit
    if (Deficit >0) then
        Update sizes of sources and set requested amount to deficit
    end if
end while
```

4.6. Group source selection considerations

We envisage that in most cases source selection will be a single user task however, there may be occasions when multiple users seek to collectively find a single source selection solution. For example, given a scenario in which multiple users seek to retrieve data for a set of possible sources for which there is significant cost associated with access and retrieval of data from each source. Here, it may be more beneficial for the users' views to be aggregated to calculate a single source selection result that seeks to best meet the views of all the users. Combining multiple users' views, to avail of collective expertise and experience, is predicated on the view that a group can make superior decisions compared with a single user [27] whilst additionally, bias that may be present with just a single DM could be eliminated [20] (or at least diluted). Therefore, our approach considers the aggregation of preferences, of the relative importance of different criteria, of multiple users. Then we seek to find a source selection solution that is most fit for purpose for the *user group*. Various strategies can be taken to aggregate multiple users' views. In our approach aggregation is performed by eliciting a set of PC judgements from each user from which, a single aggregated set of pairwise judgements is calculated, from which a single weights vector can then be calculated.

There are other group decision making strategies and considerations. An approach presented in [28] defines a model to reach a group consensus regarding a set of alternatives and a set of dynamic criterion. Moreover, an approach in [29] has addressed the issue of group members expressing their views dishonestly in order to more likely obtain a consensus on their own interests. An approach in [30] tackles group aggregation through revealing the compromise needed between the user views for multiple trade-off solutions including the global compromise solution. Approaches in [31,32] have studied reaching a consensus in large scale Group Decision making and consideration of the degree of trust of the opinions of different group members when looking to reach a consensus have been explored in [33]. We feel, however, that for our approach aggregation of a set of PC from each user into a single weights vector is appropriate and sufficient as our approach only elicits preferences regarding the set of criteria, from which a weights vector is derived and then utilised within the optimisation stage. Therefore, we do not have a set of finite alternative outcomes for which users express any preferences.

[4] Our focus here is to find a source selection solution using source level information relating to the set of criteria. To select and realise a solution, so as to evaluate overlapping, tuples are selected randomly from the sources. More sophisticated strategies could be employed, however, we seek to realise solutions to allow us to explore that our overlap algorithm can swiftly find a valid solution.

Table 6
Example group preference vector.

	TC	PCQ	PQ	RIQ	PSQ
Combined aggregation	0.18	0.24	0.13	0.19	0.25

Moreover, the philosophy of our approach is to model a user's context so as to provide the user with bespoke results that are most fit for their purpose. For small, more harmonious groups of users, it may be the case that the aggregation of their views can still return a single solution that is able to satisfactorily meet all the member's needs and be fit for purpose for everyone. However, we feel that consideration of applying our approach, of user-driven source selection, for large group and/or adversary group scenarios would weaken its fundamental philosophy. Therefore, we feel that the suitability of our approach, to deliver a bespoke source selection result to a user, one that is fit for purpose for each user's needs, would be impaired within such scenarios.

For a group scenario our approach takes a set of PC of each user's judgements regarding their preferences between criteria. The problem then becomes to aggregate these individual sets of judgements, and how to incorporate the weights of importance of each user's views within the aggregation. This should be by either Aggregation of Individual Judgements or Aggregation of Individual Priorities [17]. Our approach utilises Aggregation of Individual Judgements via The Weighted Geometric Mean Method (WGMM) [34], as it is prominent in the MCDA community and considered able to calculate a representative compromise of a set of user's views [35]. Given a group of D users, and a vector of weights of user importance UW of length D, a set of judgements is elicited from each user. From this an aggregated PCM is calculated where each aggregated judgement Aj is calculated via:

$$Aj_i = \sqrt[1/D]{\prod_{i=1}^{|D|}(j_i^{Uw_i})} \tag{18}$$

subject to $\sum_{i=1}^{|D|} Uw_i = 1$
where Uw_i is the weight of importance of the ith User.

From this single set of aggregate judgements a preference vector can be calculated. Therefore, when multiple users' preferences are being considered the preference elicitation stage performs aggregation of the multiple users' judgements into a single preference vector. For example, if our two users from earlier in Section 4.3 now seek a single source selection result, and given example user weights of importance of $UW = [2/3, 1/3]$, then the aggregation of the two sets of judgements can be performed using the WGMM. From this aggregated set of judgements a preference vector ranking can be calculated as shown in Table 6. From this we see that the two most important criteria within the aggregated vector are, the quality of data relating to a property's availability status and the postcode quality, which were the most important criteria for each user when calculated separately earlier. Such a preference vector is then utilised as in the single user setting within the optimization model of our approach. Therefore, within the Experiments section any of the defined weights used in each experiment could conceivability be weights derived as a result of group aggregation.

5. Experimentation

We now present experimental evaluation of our approach investigating (1) the properties and effectiveness of the optimization strategies presented, (2) comparative effectiveness of the strategies for different numbers of sources, amounts of requested tuples, systematic patterns in user weight distribution, and overlapping data, and (3) the scalability of the optimization models in terms of the time needed to return a solution.

5.1. Comparison approaches

To the best of our knowledge there is no directly comparable approach that tackles source section as we do, as a multi-criteria problem that explicitly models user preferences between the criteria to be utilised within the search for a solution. Therefore, we evaluate our optimization strategies with various baseline and comparison strategies to solve the source selection problem.

5.1.1. Random

The baseline strategy is a **Random** selection of the required number of tuples from sources. Here there is no consideration of the quality of different sources with respect to the criteria and hence alignment of the user preferences to the data is not considered.

5.1.2. Ranking sources

By considering sources with respect to the set of criteria, a ranking of the quality of the sources can be determined. **Ranking sources** considers the source criteria without consideration of user preferences. For each source, a Source Score

(*SS*) is determined via:

$$SS_i = \sum_{j=1}^{|C|} c_{ji} \tag{19}$$

where c_{ji} is the value of criterion j from source i.

From this a ranking of the sources is determined. Then, the top source contributes all its tuples, the second source contributes all its tuples, and so on, until the required number of tuples is reached.[5]

5.1.3. Weighted ranking sources

User preferences could be incorporated within a ranking of sources. Such a **Weighted Ranking Sources** considers source qualities and user preferences. For each source, a Weighted Source Score (WSS) is determined via:

$$WSS_i = \sum_{j=1}^{|C|} w_i \cdot c_{ji} \tag{20}$$

where c_{ji} is the value of criterion j from source i.

A ranking of sources is then determined and tuples selected as in Ranking sources. Ranking sources, weighted or unweighed, is unable to select only parts of data from sources and does not consider the possible ideal achievable values of each criterion in a solution, consequently it is unable to establish, for example, that a seemingly low value for a criterion is actually a high value in relation to the best possible value for that criterion. Additionally, as it only considers overall utility of each source, it has no consideration of the faithfulness to the user preferences of the solutions it derives.

5.2. Experimentation setup

For the experimentation we utilise multiple real world data sets and synthetic data; the real world data sets are from the real estate and food domains.

5.2.1. Property dataset

Our first real world dataset consists of web-scraped data from the real-estate domain, obtained via the DIADEM automated wrapper generation and extraction system [2]. The dataset contains a total of 36,818 tuples, each relating to a single real-estate property, from 146 UK real-estate sites. For each property the wrapper aims to obtain details of its location, its room composition and other particulars. The quality of such data will vary from source to source, due to the amount of information that is extractable as well as the success of the extraction process. The quality of the extracted data from each source can be assessed with respect to various criteria. Using our set of representative criteria from Section 3.2, the qualities of interest are computed as described via the representative functions outlined in Table 4. From this we obtain a measure of each source's average quality, for each criterion. Such average quality values could be estimated from source sampling.

5.2.2. Food dataset

Our second real world dataset comes from the food domain.[6] This data set contains nutritional information about world food products in an open database format. The information has been gradually collected from vendors websites by unpaid contributors. The dataset consists of 117 sources (where each source represents a contributor to the food facts database). The quality of such data will vary from source to source, due to variability of each contributor. The quality of extracted data from each source can be assessed with respect to various criteria. The set of representative criteria considered were: (1) correctness, (2) relevance, (3) usefulness, (4) consistency, (5) conciseness and (6) interpretability. Each can be calculated for each source via:

$$C_i = \frac{TP_i}{(TP_i + FP_i)} \tag{21}$$

where TP_i is number of True positives in relation to criterion i and *FP* is the number of false positives in relation to criterion (calculated via crowd souring feedback). From this we obtain a measure of each source's average quality, for each criterion.

5.3. General comparison experimentation

We first explore the propositions set out in Section 3, namely, the properties of the proposed optimization models (and comparison approaches) and their effectiveness at finding solutions of high utility and/or low range.

Experiment 1 - *WU* and *Range*

[5] In general, the last source to be selected will only contribute a subset of its tuples, which could be selected in various ways. In this paper, in the absence of additional information, the necessary amount of tuples is taken at random to complete the total, i.e., every tuple is assumed to be as good as the average over all tuples in the source.

[6] http://world.openfoodfacts.org/data.

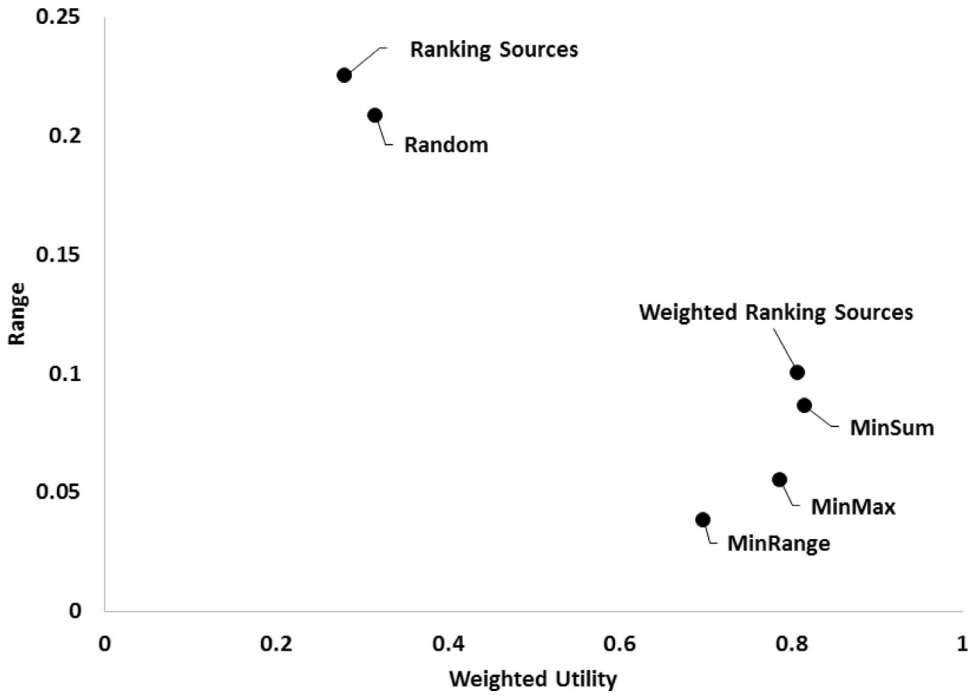

Fig. 3. Experiment 1 Property dataset.

Table 7
Single user from Experiment 1 evaluation metrics.

red	WU	Range
Weighted ranking sources	0.7638	0.1423
MinSum	0.7774	0.1193
MinRange	0.7370	0.0649
MinMax	0.7612	0.0766

This first experiment explore whether the propositions made about the characteristics of the optimization models hold and how these characteristics might impact upon the solutions they find. In this experiment we first use all 146 sources of the property data with the five criteria defined in Table 4. With the number of requested tuples set as 4000, 1000 runs were performed with randomly created user weights representing each of the 1000 users. The *WU* and *Range* for each solution found by each of the six approaches was determined. From this the average *WU* and *Range* Values for each approach can be calculated and plotted together as shown in Fig. 3. This plot illustrates the trade-off between high Weighted Utility and low Range with a trade-off front mapped out between MinSum, MinRange and MinMax. MinSum is able to find solutions with on average the highest *WU*, but at the expense of *Range*; MinRange is able to find solutions with on average the lowest *Range* but at the expense of *WU*; MinMax is able to find on average solutions with high *WU* and low *Range* that sit along the trade-off front between MinSum and MinRange. Furthermore, we observe MinSum on average finds solutions that dominate Weighted Ranking Sources solutions with respect to both *WU* and *Range* (for all 1000 solutions MinSum finds solutions that are equal or better than Weighted Ranking Sources solutions). The plot also shows the performance of approaches that do not consider user preferences, Random and Ranking Sources, which both perform poorly compared to other approaches. Performing the same experimentation using the food data set the averages of *WU* and *Range* Values for each approach over 1000 runs are shown in Fig. 4. From this we observe similar properties between the approaches.

An important consideration is whether the differences observed between the strategies translate into significant solution variation and thus into concrete benefits to a user. To investigate this we take a single user from Experiment 1, using the property data, and observe the solution more closely. The values of *WU* and *Range* for MinSum, MinRange, MinMax and Weighted Ranking Sources for the single solution are shown in Table 7. The selected sources and quantities of tuples in the solutions can be compared, as shown in Fig. 5. Here we observe how small differences in the approaches' *WU* and *MD* values have significant impact upon the actual sources and quantities that are chosen.

Experiment 2 - Sensitivity to changes in weights

In this experiment we explore how faithfully the proposed strategies are able to capture a user's preferences in the solutions they return by observing their response to subtle changes in the weights. For this experiment we use our property dataset, this time considering just two criteria, viz., PCQ and RIQ (from Table 4), with B = 400. A systematic evaluation

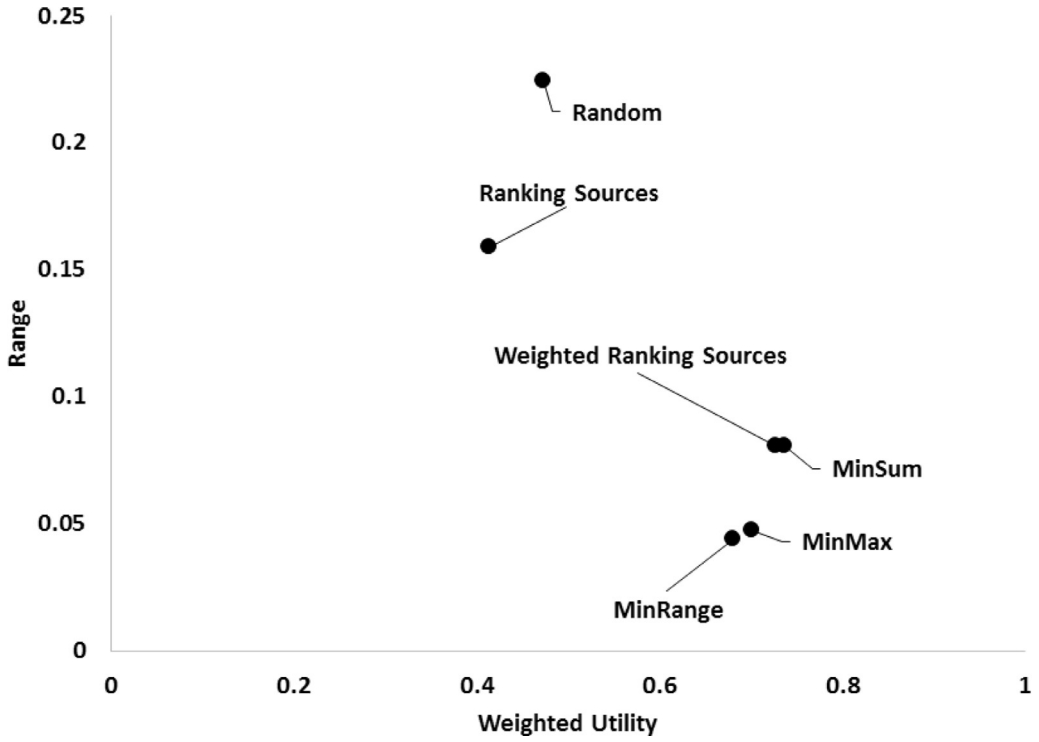

Fig. 4. Experiment 1 Food dataset.

of the impacts resulting from changes in the criteria weights was performed beginning with PCQ:0.01 and RIQ:0.99, then incrementing PCQ and decrementing RIQ, in steps of 0.01, finishing with PCQ:0.99 and RIQ:0.01. The results for Weighted Ranking Sources, MinSum, MinRange and MinMax are shown in Fig. 6. From Fig. 6(a) we observe that Weighted Ranking Sources has found only four unique solutions, which highlights that it is insensitive to subtle changes in user preferences due to its focus on overall utility. From Fig. 6(b) we observe that MinSum has found five unique solutions highlighting that it is also relatively insensitive to subtle changes in user preferences due to its focus on overall utility. From Fig. 6(c) we observe that, from the 99 experiment runs, MinRange has found 99 unique trade-off solutions within the objective space; highlighting MinRange's sensitivity to subtle changes in user preference weights, however, due to its focus on *Range* without consideration of utility it is unable to always find solutions on the trade-off front. In contrast, in Fig. 6(d) we observe that, from the 99 experiment runs, MinMax has found 99 unique trade-off solutions within the objective space; this highlights MinMax's sensitivity to subtle changes in user preference weights, to find solutions that are a faithful representation of user preferences, as well as its ability to seek high utility solutions that lie on the trade-off front between the criteria.

5.4. Studying robustness to variation

In this section Experiments 3–5 explore how much the properties of the selection strategies are preserved through systematic varying of the parameters of the number of sources, the amount of data retrieved, and the set of user weights respectively. Experiment 6 explores how well our approach is able to deal with varying levels of overlap within the data sources.

Experiment 3 - number of sources.

First, the approaches were compared with increasing numbers of synthetically created sources, from 100 to 10000. For each source random criteria values for each of 5 criteria were created and at every step the number of requested tuples was 5% of the overall total number of tuples. A set of user weights was randomly created and utilised over all the steps. The results for the 6 approaches, with respect to their *WU* values, are shown in Fig. 7(a). From this we observe that MinSum always performs equal to or better than Weighted Ranking Sources and performs better than the other approaches. Further, we observe that MinMax finds solutions whose utility is slightly better than MinRange finding solutions that are between MinSum and MinRange. Moreover, we observe how the approaches that do not consider user preferences perform poorly, with (unweighted) Ranking Sources performing similarly to Random. Regarding *Range*, the results for the approaches are shown in Fig. 7(b). From this we observe that MinRange is able to find solutions that most faithfully preserve user preferences. Moreover, we observe that MinMax finds solutions between MinRange and MinSum and is able to find solutions close to the MinRange solutions found. Moreover, we observe again how the approaches that do not consider user prefer-

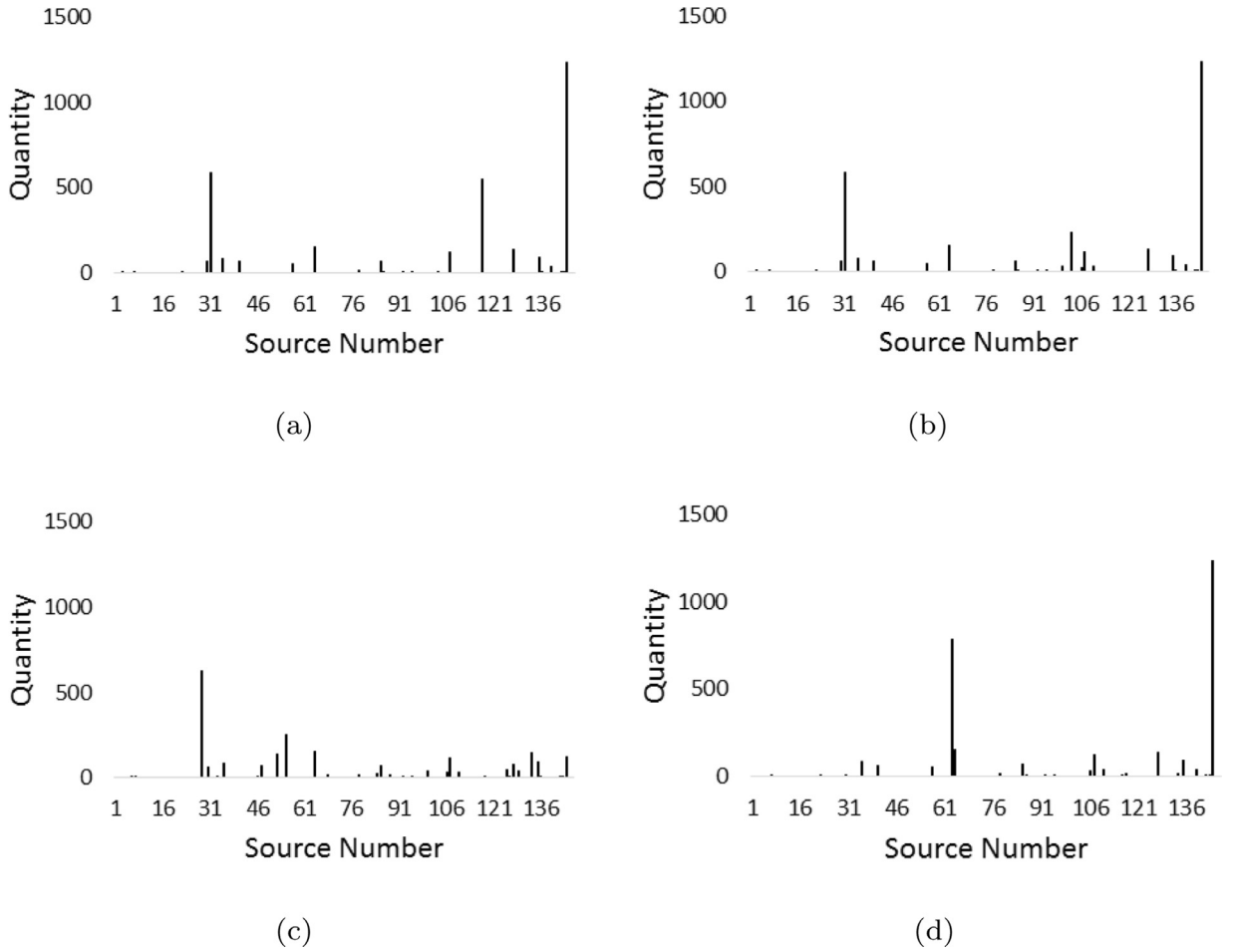

Fig. 5. Single user from Experiment 1 (a) Weighted ranking sources (b) MinSum (c) MinRange (d) MinMax.

ences perform poorly. These two plots show that the relative properties of *WU* and *Range* for the strategies are preserved as the number of sources increases.

Experiment 4 - amount of data requested

Next, the approaches were compared over increasing amounts of data requested. Using the food dataset,[7] runs were performed, with the percentage of the overall total amount of tuples requested increasing from 1% to 99%. For each run, the set of 6 criteria was utilised and a set of user weights was randomly created and utilised over all steps. The results are shown with respect to their *WU* values in Fig. 8(a). In this plot, we observe, how MinSum performs as well as or better than the other approaches whilst MinMax performs better than MinRange, finding solutions with *WU* between MinRange and MinSum. We further observe that most of the approaches improve as the amount of data selected increases, as each criterion's ideal weighted utility can be more satisfactorily met as more and more sources make up the result. Regarding *Range*, the results for this experiment are shown in Fig. 8(b). Here we observe again that MinRange performs better than other approaches and that MinMax is able to find solutions with lower *Range* than MinSum and Ranking Sources and finds solutions close to the MinRange solutions.

Experiment 5 - user preference weights

Next, the approaches were compared through systematic alteration of user weights. The property dataset was used,[8] with the set of five criteria and the amount of data requested set to 5000. Starting from equal weights, each criterion was taken in turn and its weight gradually increased at the expense of the other four. The *WU* results (with one plot per criterion) are shown in Fig. 9. Here we observe that MinSum again always performs equal to or better than the other approaches regarding *WU* across the range of weights values, and that MinMax finds solutions that are between MinRange and MinSum, here closer to the MinSum solutions. Interestingly we see that, as each criterion's weight increases, the differences between

[7] Running the experiment with the property dataset, similar characteristics were observed.

[8] Running the experiment using the Food dataset, similar characteristics were observed.

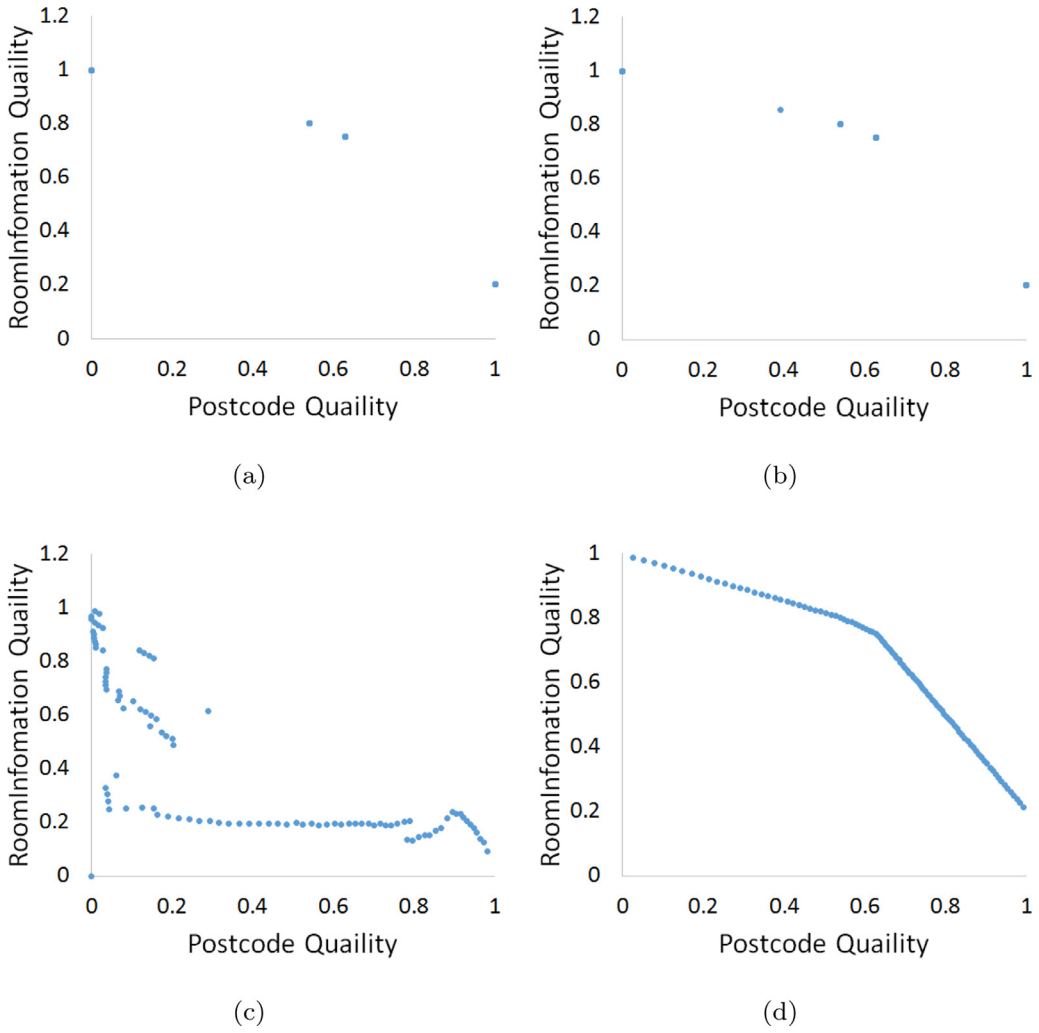

Fig. 6. Weights Sensitivity (a) RankingSources, (b) MinSum, (c) MinRange, (d) MinMax.

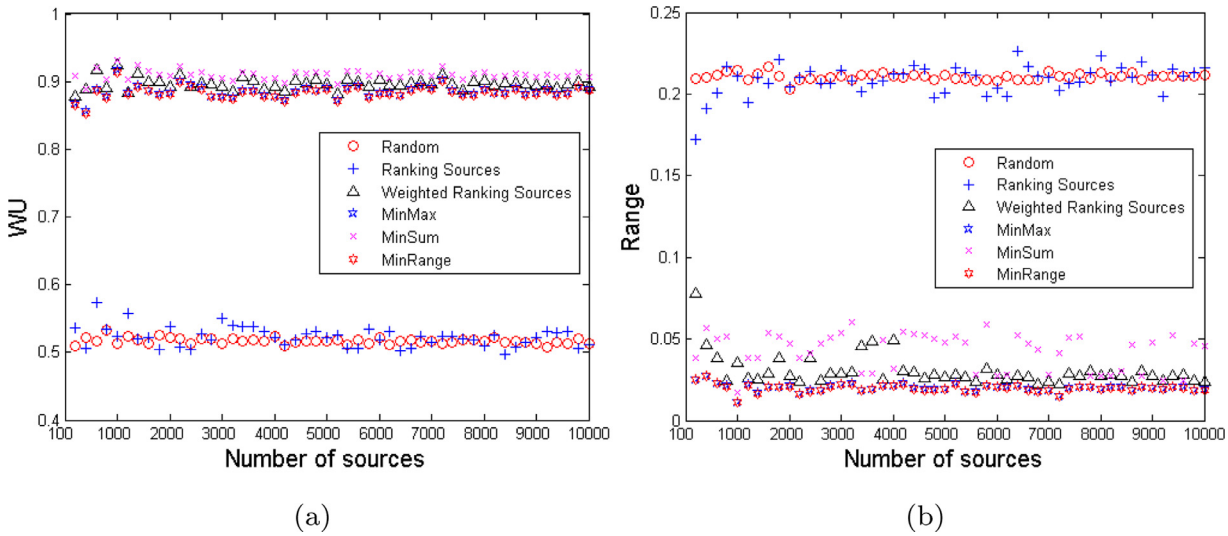

Fig. 7. Experiment 3 (a) *WU*, (b) *Range*.

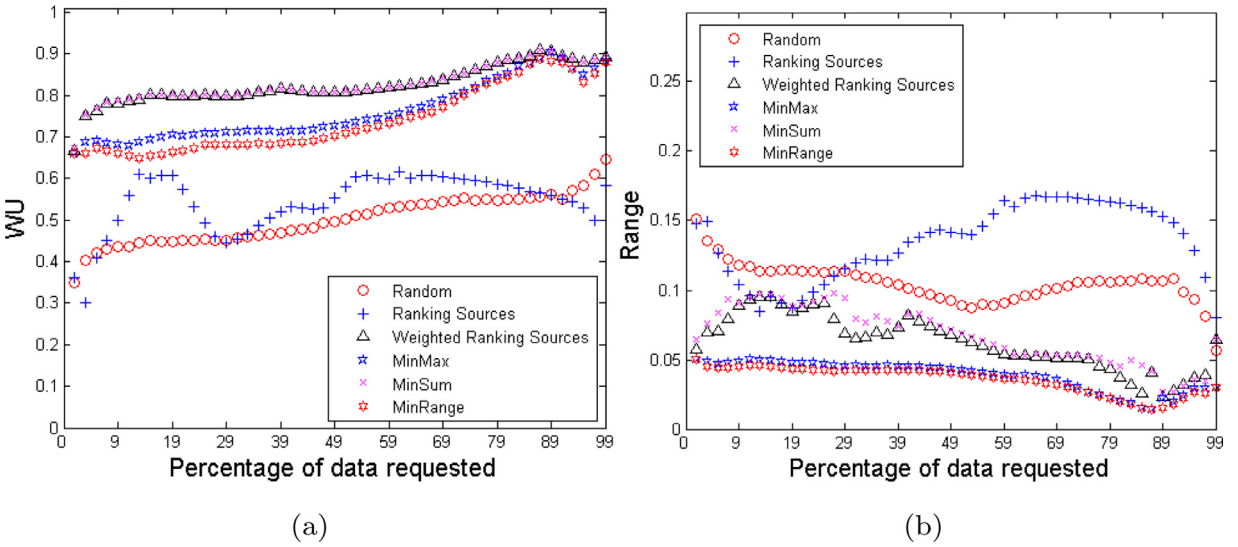

Fig. 8. Experiment 4 (a) *WU*, (b) *Range*.

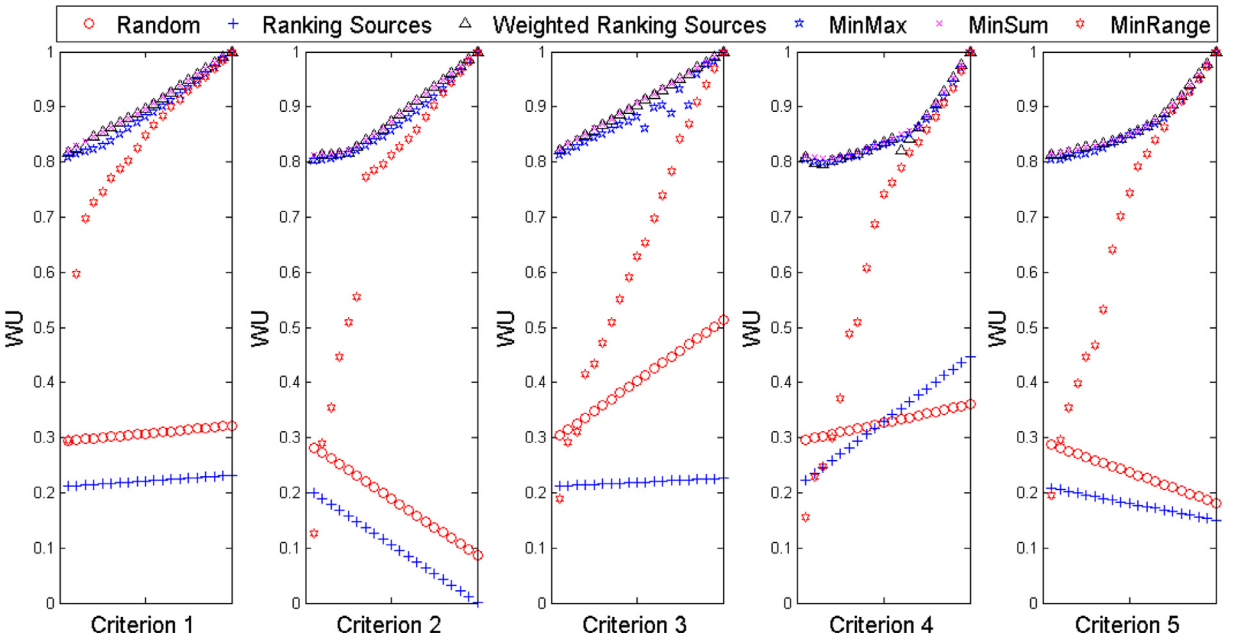

Fig. 9. Experiment 5 *WU*.

strategies become less pronounced, as a single criterion of the problem becomes more dominant. When there is close to equal weights between the criteria we notice more pronounced differences between the MinSum and MinRange solutions, highlighting that in these scenarios the trade-off between seeking Highest Utility and Minimising Range is higher. The results for *Range* are shown in Fig. 10. We observe that MinRange has the best performance and that MinMax performs better than MinSum and Ranking Sources, finding solutions between MinRange and MinSum. We further see that again, as each criterion's weight increases, the differences between the strategies become less pronounced. Interestingly we further see that, for most criterion, MinRange is able to find solutions closest to 0 Range when the weights are closer to equal or when one criterion has a very high weight.

Experiment 6 - overlapping sources

Finally, we experiment with increasing amounts of overlap within sources. For this experiment, synthetic data was created; for each source random criteria values were created and its tuples assigned an ID from an overall list of the number of distinct tuples within the whole dataset. Then, by reducing the number of distinct tuples within the whole dataset, the density of overlap can be increased. 200 sources were created each of size 100, creating a 20,000 tuple dataset, with the

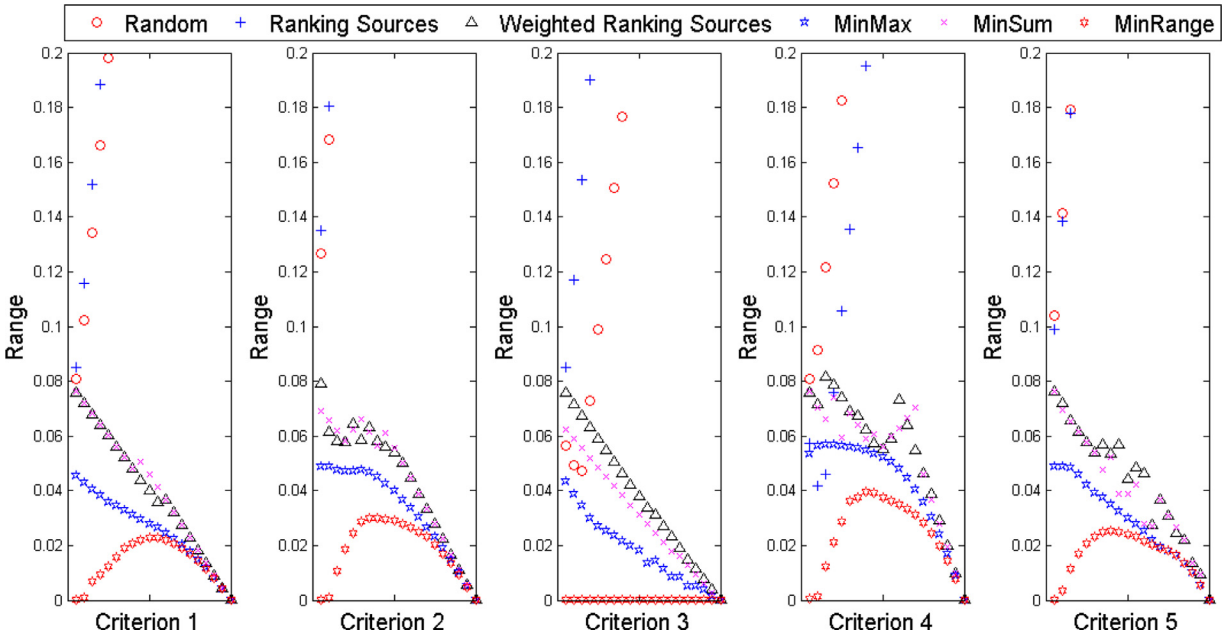

Fig. 10. Experiment 5 *Range*.

Table 8
Overlap experiment steps.

Step no.	No. of distinct tuples	Percentage of overlap	No. of iterations	Time (secs)
1	10,000	50%	3	0.03
2	5000	75%	4	0.275
3	4000	80%	4	0.088
4	3000	85%	5	0.125
5	2000	90%	5	0.262
6	1000	95%	57	0.499

number of requested tuples set as 1000. 10 runs for each overlap step were performed and averaged using the geometric mean. Initially, the number of distinct tuples within the dataset is defined as 10,000, denoting that it will have on average 2 occurrences of each tuple hence, overlap of about 50% (that is 50% of the tuples are repeats). Next, the number of distinct tuples was reduced (and thus the percentage of overlap increased) in the steps shown in Table 8.

Using MinSum,[9] the results of the number of iterations required to reduce the deficit to 0 (that is to find 1000 distinct tuples) for all the steps is shown in Fig. 11(a) and listed in Table 8. We see that, for up to 90% overlap between the sources, our approach is able to find a solution with 0 deficit in 5 or less iterations. Further, even for the final step - where there are only 1000 distinct tuples in the whole dataset and 1000 tuples are sought - our approach is able to converge, taking 57 iterations to find a solution with 0 deficit.

Accessing scalability

We now present scalability results for the optimization models considered here.[10] We recorded the elapsed time taken by MinSum, MinRange and MinMax for the experiments in Section 5.4. For all the optimization experimentation performed here a time-out of five seconds was employed to terminate an optimization and utilise the solution found at the time-out. Due to the nature of the search space of the objective functions for MinMax and MinRange, sometimes a near optimal solution may be swiftly found, followed by refinement that gives minuscule improvements, sometimes for several minutes. Therefore, by utilising a time-out a near optimal solution is found within a reasonable time-frame.

Elapsed time for MinSum, MinRange and MinMax during Experiment 4 - with increasing amounts of data requested - is shown in Fig. 12(a). We observe an operational time of milli-seconds for the MinSum model, even when requesting large percentages of data. MinRange and MinMax have longer elapsed times than MinSum and occasionally times-out occurs[11]. Similar observations were made for the elapsed times incurred in Experiment 5. Elapsed time for MinSum, MinRange and MinMax during Experiment 3 - with increasing numbers of sources - is shown in Fig. 12(b). We observe that MinSum scales

[9] Here the MinSum model is utilised, due to its efficiently (See below).
[10] All experiments were performed on a 64bit Windows 7 Machine with i7-3770 CPU @ 3.4 GHz and 12GB of RAM

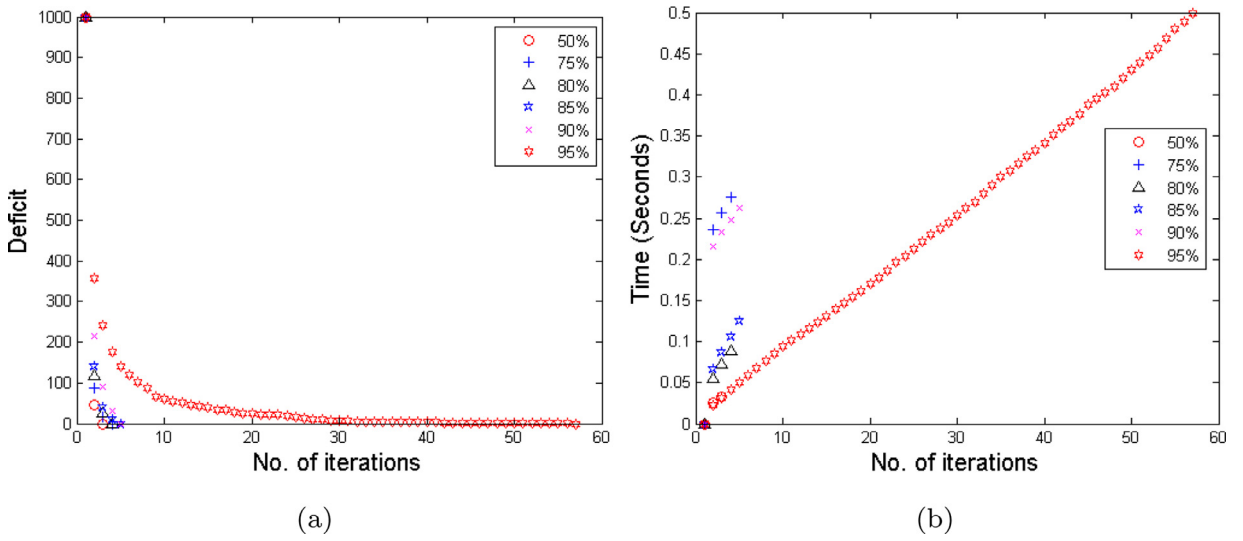

Fig. 11. Experimentation 6 (a) Deficit, (b) Elapsed time.

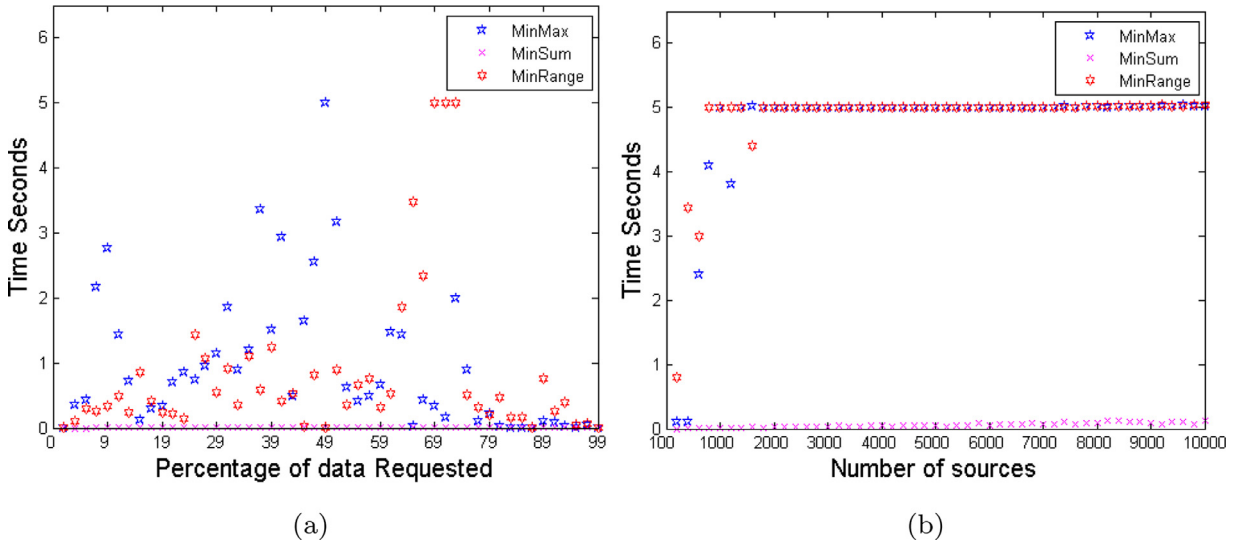

Fig. 12. Elapsed time (a) Experiment 4, (b) Experiment 3.

well, even for 10,000 sources it is able to find solutions in a fraction of a second, and that MinRange and MinMax frequently time-out.[11]

Elapsed time for each step from Experiment 6 is shown in Fig. 11(b) (and the total elapsed times are listed in Table 8[12]). Here we observe that for up to 90% overlap between sources our approach finds a solution with 0 deficit in less than 1/3 of a second. Further, even for the pathological case which takes 57 iterations, our approach finds a solution with 0 deficit in less than 1/2 a second.

In summary, our approach's optimization-based strategies are capable of finding optimal (or near optimal when time-out is utilised) solutions efficiently, that scale up to thousands of sources; further, MinSum is able to find valid solutions even with high density of data overlap.

[11] However, as was shown in the Experiments performance analysis, it has little impact upon the resulting characteristics of the solutions as in these cases near optimal solutions are found

[12] Here time relates to the total optimization running time without consideration of actual data selection procedures and their potential time concerns. However, here we are concerned with exploring the scalability of our approach regarding its optimization times for heavily overlapping sources.

6. Conclusions

In this paper we have proposed a user-driven approach that tackles source selection as a multi-criteria problem, seeking to identify sources, and data therein, that are most fit for purpose. The approach takes account of user context, allowing users to tune their preferences by specifying the relative importance of different criteria. Through explicit modelling of user preferences, our approach finds solutions that exhibit high overall weighted utility with high *WU* values, and/or are a faithful representation of a user's preferences with low *Range* values. The approach can incorporate criteria that are diverse and not drawn from a fixed set. Furthermore, a solution can use a subset of the data from each selected source, rather than require that sources are used in their entirety or not at all.

We have presented a collection of optimisation strategies for exploring the space of solutions, and compared and evaluated them against different baselines, using multiple real world datasets. Experimentation has shown that solutions can be efficiently computed at scale that are attuned to each user's preferences, both with respect to overall weighted utility and through faithful representation of a user's preferences within a result. Furthermore, experimentation has shown the approach's ability to deal with overlap between sources.

We have shown that the MinSum model is able to finds solutions with high overall weighted utility and always performs equal to or better than the weighted sum approaches, such as Weighted Ranking Sources. Further, we have shown that the MinRange model is able to find solutions that are a faithful representation of a user's preferences. Moreover, we have shown that the MinMax strategy considers both of these valuable properties of the MinSum and MinRange strategies and is able to find solutions with good overall weighted utility whilst also being a faithful representation of a user's preferences.

Future work will utilise user context modelling within other stages of a data wrangling pipeline, as well as investigating the modelling of non-linear criteria functions and analysis of their optimisation, complexity and performance.

Acknowledgements

This work has been carried out within the VADA Programme Grant of the UK Engineering and Physical Sciences Research Council, grant number EP/M025268/1, whose support we are pleased to acknowledge.

References

[1] A. Halevy, F. Korn, N.F. Noy, C. Olston, N. Polyzotis, S. Roy, S.E. Whang, Goods: organizing Google's datasets, in: SIGMOD, ACM Press, New York, New York, USA, 2016, pp. 795–806, doi:10.1145/2882903.2903730.
[2] T. Furche, G. Gottlob, G. Grasso, X. Guo, G. Orsi, C. Schallhart, C. Wang, DIADEM: thousands of websites to a single database, PVLDB 7 (14) (2014).
[3] S. Siraj, L. Mikhailov, J. Keane, PriEsT : atool to estimate priorities from inconsistent judgments, in: Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on, 2013, pp. 44–49.
[4] X.L. Dong, B. Saha, D. Srivastava, Less is more, Proc. VLDB Endow. 6 (2) (2012) 37–48, doi:10.14778/2535568.2448938.
[5] T. Rekatsinas, A. Deshpande, X.L. Dong, L. Getoor, D. Srivastava, SourceSight: enabling effective source selection, in: ACM SIGMOD, 2016, pp. 2157–2160.
[6] M. Salloum, X.L. Dong, D. Srivastava, V.J. Tsotras, Online ordering of overlapping data sources, in: VLDB 2014, 2014, pp. 133–144.
[7] S. Kambhampati, U. Nambiar, Effectively mining and using coverage and overlap statistics for data integration, IEEE TKDE 17 (5) (2005) 638–651, doi:10.1109/TKDE.2005.76.
[8] T. Rekatsinas, X.L. Dong, D. Srivastava, Characterizing and selecting fresh data sources, in: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data - SIGMOD '14, ACM Press, New York, New York, USA, 2014, pp. 919–930, doi:10.1145/2588555.2610504.
[9] T. Rekatsinas, L. Getoor, Finding quality in quantity : the challenge of discovering valuable sources for integration, CIDR (2015).
[10] G.A. Mihaila, L. Raschid, M.-e. Vidal, Using quality of data metadata for source selection and ranking, in: Proceedings of the Third International Workshop on the Web and Databases, WebDB 2000, 2000, pp. 93–98.
[11] Z. Saoud, S. Kechid, Integrating social profile to improve the source selection and the result merging process in distributed information retrieval, Inf. Sci. 336 (2016) 115–128, doi:10.1016/j.ins.2015.12.012.
[12] G. Paltoglou, M. Salampasis, M. Satratzemi, Collection-integral source selection for uncooperative distributed information retrieval environments, Inf. Sci. 180 (14) (2010) 2763–2776, doi:10.1016/j.ins.2010.03.020.
[13] J. Tian, D. Yu, B. Yu, S. Ma, A fuzzy TOPSIS model via chi-square test for information source selection, Knowl. Based Syst. 37 (2013) 515–527, doi:10.1016/j.knosys.2012.09.010.
[14] T.L. Saaty, The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation, Mcgraw-Hill, 1980.
[15] T.L. Saaty, Decision Making with Dependence and Feedback: The Analytic Network Process : the Organization and Prioritization of Complexity, Rws Publications, 2001.
[16] J. van Til, C. Groothuis-Oudshoorn, M. Lieferink, J. Dolan, M. Goetghebeur, Does technique matter; a pilot study exploring weighting techniques for a multi-criteria decision support framework., Cost Ef.f Resour. Alloc. 12 (1) (2014) 22, doi:10.1186/1478-7547-12-22.
[17] A. Ishizaka, D. Balkenborg, T. Kaplan, Influence of aggregation and measurement scale on ranking a compromise alternative in AHP, J. Oper. Res. Soc. (2010) 1–11.
[18] T.L. Saaty, A scaling method for priorities in hierarchical structures, J. Math. Psychol. 15 (3) (1977) 234–281, doi:10.1016/0022-2496(77)90033-5.
[19] E. Choo, W. Wedley, A common framework for deriving preference values from pairwise comparison matrices, Comput. Oper. Res. 31 (6) (2004) 893–908, doi:10.1016/S0305-0548(03)00042-X.
[20] A. Ishizaka, P. Nemery, Multicriteria Decision Analysis: Methods and Software, Wiley, 2013.
[21] A. Ishizaka, Comparison of fuzzy logic, AHP, FAHP and hybrid fuzzy AHP for new supplier selection and its performance analysis, Int. J. Integr. Supply Manage. 9 (1/2) (2014) 1, doi:10.1504/IJISM.2014.064353.
[22] A. Ishizaka, C. Pearman, P. Nemery, AHPSort: an AHP-based method for sorting problems, Int. J. Prod. Res. 50 (17) (2012) 4767–4784, doi:10.1080/00207543.2012.657966.
[23] S.-C. Ting, D.I. Cho, An integrated approach for supplier selection and purchasing decisions, Supply Chain Manage. Int. J. 13 (2) (2008) 116–127, doi:10.1108/13598540810860958.
[24] S.-p. Wan, G.-l. Xu, J.-y. Dong, Supplier selection using ANP and ELECTRE II in interval 2-tuple linguistic environment, Inf. Sci. 385 (2017) 19–38, doi:10.1016/j.ins.2016.12.032.
[25] P. Missier, S. Embury, M. Greenwood, A. Preece, B. Jin, Quality views: capturing and exploiting the user perspective on data quality, PVLDB (2006) 977–988.

[26] E. Zio, R. Bazzo, A comparison of methods for selecting preferred solutions in multiobjective decision making, in: C. Kahraman (Ed.), Computational Intelligence Systems in Industrial Engineering, Atlantis Press, 2012, pp. 23–43.
[27] S. French, J. Maule, N. Papamichail, Decision Behaviour, Analysis and Support, Cambridge University Press, 2009.
[28] Y. Dong, H. Zhang, E. Herrera-Viedma, Consensus reaching model in the complex and dynamic MAGDM problem, Knowl. Based Syst. 106 (2016) 206–219, doi:10.1016/j.knosys.2016.05.046.
[29] Y. Dong, Y. Liu, H. Liang, F. Chiclana, E. Herrera-Viedma, Strategic weight manipulation in multiple attribute decision making, Omega (2017), doi:10.1016/j.omega.2017.02.008.
[30] E. Abel, L. Mikhailov, J. Keane, Group aggregation of pairwise comparisons using multi-objective optimization, Inf. Sci. 322 (2015) 257–275, doi:10.1016/j.ins.2015.05.027.
[31] Á. Labella, Y. Liu, R. Rodríguez, L. Martínez, Analyzing the performance of classical consensus models in large scale group decision making: a comparative study, Appl. Soft Comput. (2017), doi:10.1016/j.asoc.2017.05.045.
[32] H. Zhang, Y. Dong, E. Herrera-Viedma, Consensus building for the heterogeneous large-scale GDM with the individual concerns and satisfactions, IEEE Trans. Fuzzy Syst. (2017) 1–13, doi:10.1109/TFUZZ.2017.2697403.
[33] Y. Liu, C. Liang, F. Chiclana, J. Wu, A trust induced recommendation mechanism for reaching consensus in group decision making, Knowl. Based Syst. 119 (2017) 221–231, doi:10.1016/j.knosys.2016.12.014.
[34] T. Saaty, Group decision making and the AHP, in: B. Golden, E. Wasil, P. Harker (Eds.), The Analytic Hierarchy Process. Applications and Studies, Springer Berlin Heidelberg, 1989, pp. 59–67, doi:10.1007/978-3-642-50244-6_4.
[35] R. Van Den Honert, F. Lootsma, Group preference aggregation in the multiplicative AHP. The model of the group decision process and Pareto optimality, Eur. J. Oper. Res. 96 (2) (1997) 363–370, doi:10.1016/0377-2217(95)00345-2.